

Solving the Electric Vehicle Routing Problem Using a Hybrid Adaptive Large Neighborhood Search Method

Erdelić, Tomislav

Doctoral thesis / Disertacija

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:138947>

Rights / Prava: [In copyright / Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-27**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences -
Institutional Repository](#)





University of Zagreb

FACULTY OF TRANSPORT AND TRAFFIC SCIENCES

Tomislav Erdelić

**SOLVING THE ELECTRIC VEHICLE ROUTING
PROBLEM USING A HYBRID ADAPTIVE LARGE
NEIGHBORHOOD SEARCH METHOD**

DOCTORAL THESIS

Zagreb, 2021



University of Zagreb

FACULTY OF TRANSPORT AND TRAFFIC SCIENCES

Tomislav Erdelić

**SOLVING THE ELECTRIC VEHICLE ROUTING
PROBLEM USING A HYBRID ADAPTIVE LARGE
NEIGHBORHOOD SEARCH METHOD**

DOCTORAL THESIS

Supervisor: Professor Tonči Carić, Ph.D.

Zagreb, 2021



Sveučilište u Zagrebu
FAKULTET PROMETNIH ZNANOSTI

Tomislav Erdelić

**RJEŠAVANJE PROBLEMA USMJERAVANJA
ELEKTRIČNIH VOZILA METODOM HIBRIDNOGA
ADAPTIVNOGA PRETRAŽIVANJA VELIKOGA
SUSJEDSTVA**

DOKTORSKI RAD

Mentor: Prof. dr. sc. Tonči Carić

Zagreb, 2021.

Doctoral thesis has been made at the University of Zagreb, Faculty of Transport and Traffic Sciences, Department of Intelligent Transport Systems, Chair of Applied Computing.

Mentor: Professor Tonči Carić, Ph.D.

Doctoral thesis has: 308 pages

Doctoral thesis committee:

1. Associate Professor Edouard Ivanjko, Ph.D. (committee chair)
Faculty of Transport and Traffic Sciences, University of Zagreb
2. Professor Tonči Carić, Ph.D.
Faculty of Transport and Traffic Sciences, University of Zagreb
3. Professor Domagoj Jakobović, Ph.D.
Faculty of Computing and Electric Engineering, University of Zagreb
4. Assistant Professor Luka Novačko, Ph.D. (replacement)
Faculty of Transport and Traffic Sciences, University of Zagreb

Date:

4 October, 2021

About the mentor:

Tonči Carić received his B.Sc. and the M.Sc. degree in Computer Science from the Faculty of Electrical Engineering and Computing, the University of Zagreb, in 1993 and 2000, respectively. He received his Ph.D. degree in 2004 from the Faculty of Transport and Traffic Engineering, University of Zagreb. He is currently working at the Department of Intelligent Transportation Systems, Faculty of Transport and Traffic Sciences, the University of Zagreb as Professor and Head of Chair for Applied Computing. He participated in R&D projects funded by the European Commission, national research projects, and professional projects in cooperation with industry. His main research interests are combinatorial optimization, moving object databases, and intelligent transportation systems. He is a member of INFORMS and IEEE.

O mentoru:

Tonči Carić diplomirao je 1993. i magistrirao 2000. na Fakultetu elektrotehnike i računarstva, Sveučilišta u Zagrebu. Obranio je doktorsku disertaciju 2004. na Fakultetu prometnih znanosti, Sveučilišta u Zagrebu. Trenutno radi na Zavodu za Inteligentne transportne sustava Fakulteta prometnih znanosti kao profesor i voditelj je Katedre za primijenjeno računalstvo. Sudjelovao je kao istraživač na znanstveno razvojnim projektima financiranih od Europske komisije, nacionalnim znanstvenim projektima i stručnim projektima na projektima u gospodarstvu. Njegovi glavni istraživački interesi vezani su za kombinatorne optimizacije, baze podataka pokretnih objekata i inteligentne transportne sustave. Član je udruženja INFORMS i IEEE.

*Dedicated to my family for their love and
support.*

Acknowledgements

I would like to take this opportunity to express my gratitude and to thank all the people who helped me throughout my life, and without which, I would not be able to reach this goal and complete my doctoral thesis.

First of all, I want to thank my wife Martina, who supported me since I enrolled my Ph.D. study. Thank you for being there for me, especially in hard times, and for putting up with all the frustration that has been present in our lives due to the Ph.D. Thank you for pushing me to never give up, even when I would rather quit and leave everything behind. Thank you for showing me the important things in life: you and Jan brighten my life.

Second, I would like to thank my parents, Marija and Josip, who supported me from an early age to follow my dreams. They did everything they could to back me up and encouraged me to enroll both master and Ph.D. studies. Thank you for all the love and care you gave me through life, even when I was not at my best. Thank you for passing on the important life values and showing me to never give up.

I would like to thank my older brothers, Dario and Mario, and sisters-in-laws, Martina and Vlatka, for their love and support. Sometimes, it may seem that I forgot, but thank you for always watching over me from an early age and taking me with you whenever you could. I will forever cherish those moments. To my brother Mario, thank you for taking care of me and showing me what a hard-work means, as you are the most hardworking person I know. Especially, I want to thank my older brother Dario, who was, and still is, my inspiration in life. Sometimes, you were like a father to me, and I always wanted to walk the same path as you did. I am really grateful for all your advices.

I would like to thank my mentor, professor Tonči Carić, who supported and mentored me through my Ph.D. study. Thank you for allowing me to chose the research topic of Ph.D. by myself. I want to thank you for all the time and effort that you invested in me, as well as all the help and understanding that you provided.

I want to thank all the colleagues from the Department of Intelligent Transport Systems, who were always supportive and helped me out whenever needed. It was an enjoyment to work and collaborate with you during these last several years. Especially, I want to thank Marko Matulin for all advices regarding the publication of my research papers. I also want to thank all other professors who motivated me to pursue a career in teaching and research.

Finally, I also want to thank all my friends who always supported me during these last several years. Marko Đ. thank you for being such a good friend and for the numerous suggestions regarding the research papers, Ph.D. thesis, teaching activities, programming, etc. I hope to be as half as good a researcher as you are. Thank you "escape kuleri" Ines, Ivan, and Tomislav, for your support and always listening whenever I needed an out-of-scope opinion.

Abstract

In order to perform a high-quality and on-time delivery in logistic systems, it is necessary to efficiently manage a delivery fleet. Nowadays, due to the new policies and regulations related to greenhouse gas emission in the transport sector, logistic companies are paying higher penalties for each emission gram of CO₂/km. With Electric Vehicle (EV) market penetration, many companies have started to evaluate the integration of EVs in their fleet, as EVs do not have local greenhouse gas emission, produce minimal noise, and are independent of the fluctuating oil price. Well-researched Vehicle Routing Problem (VRP) is extended to the Electric Vehicle Routing Problem (EVRP), which takes into account specific characteristics of EVs. EVRP aims to determine a set of least-cost electric vehicle delivery routes from a depot to a set of geographically scattered customers, subject to side constraints. As VRP is an NP-hard problem, the EVRP is also an NP-hard problem, which incurs the use of heuristic and metaheuristic procedures to solve the problem. Over the years, various heuristic procedures were applied to solve the VRP problem. In the last several years, these procedures were modified for the application on the EVRP problem. In the literature, for each problem variant of the EVRP, i.e., time windows, partial recharging, full recharge, different charging stations, etc., a specifically designed metaheuristic procedure is proposed.

The main objective of this thesis is to develop a Hybrid Adaptive Large Neighborhood Search (HALNS) method for solving different variants of the EVRP problem. The proposed method includes a local search for improving the solution and exact procedure for optimal Charging Station (CS) placement. In the first part of the thesis, the proposed hybrid method was implemented and compared to the non-hybrid method used for solving the EVRP problem. Also, the advantages of the metaheuristic methods were highlighted in comparison to the exact method for solving the problem defined as a mixed integer linear program. The developed hybrid method was applied to solve different EVRP variants. In the end of the first part of the thesis, the results were analyzed and compared to the so-far best-known solutions. In the second part, the new time-dependent EVRP problem with time windows and charging time dependent on the state-of-charge was presented. The problem considers temporal changes in the traffic network while routing EVs, which are usually caused by congestion. In the last part, the adapted delivery problem of a Croatian company was modeled as EVRP problem and solved by the HALNS method. Instead of using conventional vehicles, the fleet of EVs with equal vehicle characteristics (load and battery capacity) was considered.

Keywords: electric vehicles, electric vehicle routing problem, hybrid metaheuristic, heuristics, exact procedures, dynamic programming, time-dependent routing, logistic

Prošireni sažetak

Rješavanje problema usmjeravanja električnih vozila metodom hibridnoga adaptivnoga pretraživanja velikoga susjedstva

Uvod

Problem usmjeravanja vozila (eng. *Vehicle Routing Problem, VRP*) je NP-težak optimizacijski problem koji nastoji odrediti skup dostavnih ruta od skladišta do geografski raspršenih korisnika uz određena ograničenja. S dolaskom električnih vozila (eng. *Electric Vehicle, EV*) na tržište, mnoge logističke tvrtke evaluirale su uvođenje električnih vozila u svoju flotu vozila kako bi smanjili emisiju stakleničkih plinova, a time i naknade koje moraju plaćati za svaki ispušteni gram CO₂/km. EV vozila imaju nekoliko prednosti u odnosu na vozila s motorima na unutarnje izgaranje (eng. *Internal Combustion Engine Vehicle, ICEV*): nemaju lokalnu emisiju stakleničkih plinova, stvaraju minimalnu buku, mogu se napajati električnom energijom dobivenom iz obnovljivih izvora energije, te su neovisni o promjenjivoj cijeni nafte i politički nestabilnim zemljama koji proizvode naftu. Postoje dvije osnovne izvedbe EV-a: baterijska električna vozila (eng. *Battery Electric Vehicles, BEV*), koja se napajaju isključivo iz baterija ugrađenih u vozilo, i plug-in hibridna električna vozila (eng. *Plug-in Hybrid Electric Vehicle, PHEV*) koja se napajaju integracijom baterija i drugih izvora energije, npr. motora s unutarnjim izgaranjem. U ovoj disertaciji se razmatraju BEV vozila, koja se suočavaju sa dva problema: ograničeni domet vožnje i potrebna infrastruktura za punjenje. Zbog ograničenog kapaciteta baterije, domet koji mogu postići dostavna BEV vozila s baterijom napunjenom do kraja znatno je manji od dometa ICEV vozila. Zbog toga, BEV vozila moraju češće posjetiti stanicu za punjenje kako bi obnovila energiju, što se može postići zamjenom baterija ili punjenjem u stanici za punjenje.

Uvođenje BEV vozila u logističke procese dostave dovelo je do definiranja problema usmjeravanja električnih vozila (eng. *Electric Vehicle Routing Problem, EVRP*). Inačice EVRP problema mogu uzeti u obzir neka dobro poznata ograničenja VRP problema: teretni kapacitet vozila, vremenski prozori korisnika, radno vrijeme i dr. Dodatno, EVRP problemi moraju uzeti u obzir i ograničeni domet BEV vozila, koji se odražava u potrebi za punjenjem vozila u stanici za punjenje. Stanica za punjenje može biti smještena na zasebnoj lokaciji ili na nekoj od lokacija korisnika. Općenito u VRP-u primarni je cilj minimizacija broja korištenih vozila, a sekundarni cilj minimizacija udaljenosti, vremena, emisije i sl., jer se veće uštede mogu postići s manjim brojem vozila (trošak vozila, plaća vozača i dr.). U posljednjih nekoliko godina s pojavom električnih vozila na tržištu, istraživači su se usmjerili na rješavanje različitih inačica EVRP problema koje najčešće uključuju: heterogenu flotu BEV i ICEV vozila, parcijalno punjenje,

različite vrste stanica za punjenje, nelinearnu karakteristika punjenja, zamjenu baterija, istovremeno usmjeravanje vozila i postavljanje stanica za punjenje, i dr. Iako je već istraženo mnogo inačica problema, pojedine bitne izmjene problema još nisu razmatrane: vremenski ovisno putovanje na prometnoj mreži, različite vrste stanica za punjenje s punjenjem do kraja, mogućnosti ograničavanja na samo jedno punjenje u ruti vozila i sl.

Budući da je VRP NP-težak problem, egzaktne metode mogu riješiti probleme koji se sastoje od relativno malog broja korisnika, primjerice, od 50 do 100 korisnika za VRP problem s vremenskim prozorima. Zbog navedenog je kroz godine razvijen veliki broj heurističkih i metaheurističkih metoda za rješavanje problema, koje su većinom, uz izmjene primijenjene na EVRP inačice. Zbog navedenih izmjena koje se odnose na potrebu za punjenjem u stanici za punjenje, EVRP problem teže je riješiti u odnosu na izvorni VRP problem. Prema tome, za rješavanje pojedinih inačica EVRP problema većinom su predložene posebno dizajnirane metaheuristike i heuristike, koje osim što se ističu u svojoj raznolikosti, često imaju i zahtjevnu implementaciju. U području istraživanja nedostaje primjena općenite heuristike koja će uspješno riješiti različite inačice EVRP problema.

Motivacija

Glavna motivacija disertacije jest ostvariti nove znanstvene doprinose u području usmjeravanja električnih vozila koji do sada u postojećoj literaturi nisu bili dovoljno istraženi. Područja koja su razmatrana u ovoj disertaciji su: razvoj hibridne metaheuristike sposobne za rješavanje različitih postojećih inačica problema usmjeravanja električnih vozila i razvoj novog problema vremenski ovisnog usmjeravanja električnih vozila.

Problem usmjeravanja vozila jedan je od osnovnih problema koji je doveo do značajnog razvoja optimizacijskih metoda. Zbog NP-težine problema, egzaktne metode sposobne su riješiti problem za relativno mali broj korisnika te istraživači najčešće primjenjuju heurističke postupke zasnovane na iskustvu, koje su u kratkom vremenu sposobne pronaći zadovoljavajuće rješenje. No heuristički postupci zbog uskog područja pretraživanja, često završavaju u lokalnom optimumu te se za bijeg iz lokalnog optimuma i pretraživanje većeg prostora rješenja najčešće primjenjuju metaheuristički postupci. U posljednjem desetljeću, s obzirom na veliki broj inačica osnovnog VRP problema, istraživači su usmjerili svoje napore na razvijanje općenite metaheuristike/heuristike koja će dovoljno dobro riješiti različite inačice problema. Istovremeno, s dolaskom električnih vozila, pojavio se problem usmjeravanja električnih vozila, što je utjecalo na eksponencijalni rast istraživanja vezanih uz usmjeravanje električnih vozila te pojavu različitih inačica EVRP problema. Na osnovu pregleda literature, može se uvidjeti da je za rješavanje svake od inačica predložena specijalna metaheuristika ili heuristika. Navedeno značajno utječe na mogućnosti implementacije na različite inačice problema, te se javlja potreba za razvojem općenite metaheuristike koja će efikasno riješiti različite inačice problema.

Za razliku od VRP problema, u EVRP problemu, raspored stanica za punjenje značajno utječe na kvalitetu rješenja. Većina istraživanja predlažu pojedine specijalne heurističke postupke koji će odrediti raspored stanica punjenje u ruti vozila i iznos punjenja u stanici za punjenje. S obzirom na složenost određivanja navedenog rasporeda, u ovoj disertaciji će se koristiti egzaktna metoda za određivanje rasporeda stanica za punjenje u ruti i iznos punjenja, što zajedno s korištenom metaheuristikom čini hibridni postupak. Osim navedenog, u pregledu literature, pojedini specijalno razvijeni postupci rješavanja često imaju veliku složenost i daju suboptimalna rješenja, čime se ostavlja prostor za pronalazak jednostavnijih postupaka koji bi mogli polučiti vrlo dobre rezultate.

Većina istraživača u EVRP području razmatralo je osnovne inačice EVRP problema, poput parcijalnog punjenja, punjenja do kraja, različitih stanica za punjenje i heterogene flote vozila. Navedene inačice problema, razmatraju statično stanje prometne mreže, odnosno vremenski neovisno usmjeravanje vozila. U stvarnom svijetu dobro je poznato da se vrijeme putovanja na prometnoj mreži mijenja ovisno o trenutku polaska. Primjerice u gradskim područjima mogu se odrediti vršni sati u kojima se događaju prometne gužve, koje značajno povećavaju vrijeme putovanja. U literaturi se mogu pronaći radovi vezani uz osnovni VRP problem koji se bave vremenski ovisnim usmjeravanjem, ali vremenski ovisno usmjeravanje električnih vozila još nije razmatrano. Uzimanje u obzir vremenski ovisnog putovanja mrežom može značajno promijeniti raspored korisnika i stanica za punjenje u ruti vozila, s obzirom da se primjerice može više isplatiti punjenje vozila u stanici za punjenje za vrijeme vršnih sati. Također, iako su već dobro istražene osnove inačice EVRP problema, postoji prostor za razmatranjem dodatnih karakteristika postojećih problema poput: višestrukog ili jednog punjenja u stanici za punjenje u ruti, punjenje do kraja u problemima s različitim stanicama za punjenje i sl.

U EVRP problemu razmatra se hijerarhijska funkcija optimizacije koja prvotno uključuje minimizaciju broja vozila te potom minimizaciju određene specijalne funkcije koja je u većini istraživanja ukupna prijeđena udaljenost vozila ili ukupni troškovi punjenja. Prema tome javlja se potreba za razmatranjem različitih kriterija sekundarne minimizacije, poput vremena putovanja, ukupnog vremena putovanja, ukupne potrošnje energije i sl.

Pregled disertacije

Disertacija je podijeljena na sedam poglavlja, pri čemu prva tri poglavlja daju uvod i motivaciju za izradu metode hibridnoga adaptivnog pretraživanja velikoga susjedstva (eng. *Hybrid Adaptive Large Neighborhood Search, HALNS*) i novog vremenski ovisnog problema usmjeravanja električnih vozila, dok iduća dva poglavlja opisuju dobivene rezultate i ostvarene znanstvene doprinose. U šestom poglavlju opisana je primjena metode na adaptirani problem dostave iz realnog svijeta. U posljednjem poglavlju dan je zaključak o provedenom istraživanju te smjernice za buduća istraživanja.

Prvo poglavlje daje uvod u disertaciju, poglavito uvod u područje zelene logistike koja razmatra probleme usmjeravanja električnih vozila. U uvodnom poglavlju su naglašeni prednosti i nedostaci korištenja električnih vozila prilikom njihovog usmjeravanja, te osnovne inačice električnih vozila. Također, naglašene su prednosti korištenja metaheurističkih postupaka za rješavanje VRP problema. U poglavlju je dan i pregled izvornih znanstvenih doprinosa koji su ostvareni u sklopu disertacije. Na kraju poglavlja dan je pregled disertacije.

U drugom poglavlju dan je detaljan uvod u područje usmjeravanja vozila, s naglaskom na električna vozila. Opisana je kombinatorna eksplozija mogućih rješenja problema, te je dan pregled karakteristika i primjena električnih vozila u dostavnim procesima. U poglavlju je definiran osnovni EVRP problem - problem usmjeravanja električnih vozila s vremenskim prozorima i punjenjem do kraja, zajedno s pregledom testnih instanci i matematičkom formulacijom problema. Potom je dan detaljan pregled osnovnih inačica problema, od kojih su neki razmatrani u disertaciji, i kratak pregled preostalih inačica problema.

Treće poglavlje opisuje postupke rješavanja VRP i EVRP problema, s naglaskom na metaheurističke postupke. U poglavlju su opisane najčešće korištene heuristike za konstrukciju početnog rješenja, kao i najčešće primijenjeni operatori lokalne pretrage za poboljšavanje rješenja problema. Nadalje, dan je pregled metaheurističkih postupaka primijenjenih za rješavanje problema. Na kraju poglavlja opisana je važnost razmatranja funkcije cilja s penalima za prekoračenja ograničenja problema i važnost evaluacije osnovnih promjena u konfiguraciji rješenja problema.

U četvrtom poglavlju dan je detaljan opis razvijene HALNS metode. Poglavlje započinje s opisom generalnog okvira metode i potom opisom pojedinih dijelova metode. Prvo je opisana metoda korištena za konstruiranje početnog rješenja EVRP problema. Potom su opisane funkcije i varijable korištene za evaluaciju različitih inačica EVRP problema koji se razmatraju: (i) EVRP s vremenskim prozorima i parcijalnim punjenjem, (ii) EVRP s vremenskim prozorima i punjenjem do kraja, (iii) EVRP s vremenskim prozorima, punjenjem do kraja i različitim punjačima u stanici za punjenje, i (iv) EVRP s vremenskim prozorima, parcijalnim punjenjem i različitim punjačima u stanici za punjenje. Nakon toga, opisani su korišteni operatori za izbacivanje i ubacivanje korisnika, stanica za punjenje i cijelih ruta vozila, te su dodatno opisani i preostali operatori koji su razmatrani prilikom razvijanja metode. Nakon toga su navedeni korišteni operatori lokalne pretrage, te je prikazan postupak odabira operatora i njihovog redoslijeda izvođenja. Nakon lokalne pretrage, opisan je postupak optimalnog postavljanja stanica za punjenje u ruti vozila koristeći egzaktne metode. Potom su opisani korišteni postupci za ubrzanje izvođenja metode te je provedena optimizacija parametara metode. Na kraju četvrtog poglavlja prikazani su rezultati primjene metode na malim i velikim testnim problemima te su uspoređeni s: (i) egzaktnom metodom, (ii) implementiranom metodom adaptivnog pretraživanja velikog susjedstva (eng. *Adaptive Large Neighborhood Search, ALNS*) bez postupaka poboljšavanja

rješenja prezentirana u radu [1], (iii) najbolje objavljenim rezultatima na nekoliko inačica EVRP problema, i (iv) postojećim problemima s drugačijom sekundarnom funkcijom cilja.

Peto poglavlje započinje s uvodom u vremenski ovisan problem usmjeravanja vozila i daje motivaciju za razvoj novog problema vremenski ovisnog usmjeravanja električnih vozila. Potom su opisane osnovne karakteristike vremenski ovisnog problema usmjeravanja vozila koje uključuju način izračuna vremena putovanja i osiguranja svojstva ne-pretjecanja. U petom poglavlju je dana matematička formulacija problema kao mješovitog programa cjelobrojnog programiranja. Potom je u petom poglavlju, testirana razvijena HALNS metoda na srodnim problemima vremenski ovisnog usmjeravanja vozila s ICEV vozilima. Na kraju petog poglavlja dani su rezultati primjene metode na instancama novo-razvijenog problema.

Šesto poglavlje prikazuje primjenu razvijene metode na prilagođenom dostavnom problemu iz stvarnoga svijeta. Konkretan dostavni problem je prilagođen na način da su uključene stanice za punjenje u problem te je razmatrana dostava električnim vozilima. U rezultatima je razmatrano rješavanje različitih inačica problema usmjeravanja vozila s različitim sekundarnim funkcijama cilja.

U sedmom poglavlju dan je zaključak disertacije te su istaknuti izvorni znanstveni doprinosi koji su ostvareni kroz disertaciju. Nadalje, dan je pregled tema za buduće istraživanje u području usmjeravanja električnih vozila, kao što su primjena razvijene metode na probleme s heterogenom flotom vozila koja uključuje i ICEV i EV vozila, ili izrada središnjeg sustava za prikupljanje najbolje objavljenih rezultata u EVRP području.

Zaključak

Glavni cilj istraživanja ove disertacije je: (i) razviti hibridnu metodu adaptivnoga pretraživanja velikog susjedstva za rješavanje različitih inačica postojećih EVRP problema i (ii) predložiti novi model vremenski ovisnog usmjeravanja električnih vozila te na njemu primijeniti razvijenu metodu. U sklopu disertacije ostvarena su sljedeća tri izvorna znanstvena doprinosa:

- razvoj hibridne ALNS metode za rješavanje postojećih inačica problema usmjeravanja električnih vozila;
- razvoj modela vremenski ovisnog usmjeravanja električnih vozila s vremenskim prozorima i pripadnog mješovitog cjelobrojnog programa;
- rješavanje razvijenog modela problema vremenski ovisnog usmjeravanja električnih vozila s vremenskim prozorima prilagođenom hibridnom ALNS metodom.

Kroz postignute rezultate može se zaključiti kako je predložena HALNS metoda sposobna učinkovito riješiti različite postojeće inačice EVRP problema, prema različitim kriterijima minimizacije. Na skoro svim testiranim testnim problemima, metoda je pronašla nekoliko, do sada neobjavljenih, najbolje ostvarenih rezultata, čime se najbolje ukazuje na učinkovito rješavanje EVRP problema. Istovremeno predložena metoda ima vrijeme izvršavanja kraće od većine

drugih metoda primijenjenih za rješavanje EVRP problema. Razvijeni novi problem vremenski ovisnog EVRP-a nastoji uzeti u obzir vremenski promjenjivo stanje prometne mreže, i kao rezultat izbjegavati kretanje vozila po zagušenim prometnicama, te ako postoji koristiti vremenski brži put. Na taj način može se uštedjeti na vremenu putovanja, na uštrb povećanja prijeđene udaljenosti. Za rješavanje problema predložena HALNS metoda modificirana je s dijelovima za izračun vremenski ovisnog vremena putovanja. Rezultati na srodnim problemima ukazuju da se predložena metoda može primijeniti za rješavanje vremenski ovisnog VRP problema. Ostvareni rezultati disertacije pokazuju veliki potencijal za provođenje daljnjeg istraživanja u području usmjeravanja električnih vozila.

Ključne riječi: električna vozila, problem usmjeravanja električnih vozila, hibridna metaheuristika, heuristika, egzaktne metode, dinamičko programiranje, vremenski ovisno usmjerenje, logistika

Contents

1. Introduction	1
1.1. Research motivations	3
1.2. Major contributions of the thesis	4
1.3. Outline of the thesis	5
2. Electric vehicle routing problem	7
2.1. Battery electric vehicles in routing problems	10
2.2. Electric vehicle routing problem with time windows and full recharge	11
2.3. Electric traveling salesman problem with time windows	16
2.4. Electric vehicle routing problem with time windows and partial recharging	17
2.5. Electric vehicle routing problem with different charging stations	19
2.6. Time-dependent vehicle routing problem	22
2.7. Hybrid and mixed fleet electric vehicle routing problem	26
2.8. Electric vehicle routing problem with nonlinear charging functions	27
2.9. Electric location routing problem and battery swap stations	27
2.10. Other variants of the electric vehicle routing problem	28
3. Problem solving methods	31
3.1. Initial solution	33
3.1.1. Sweep algorithm	33
3.1.2. Clark and Wright savings algorithm	34
3.1.3. Nearest neighbor heuristic	35
3.2. Local search	36
3.2.1. Intra operators	36
3.2.2. Inter operators	40
3.3. Metaheuristics	42
3.3.1. Population based metaheuristics	42
3.3.2. Simulated annealing	45
3.3.3. Tabu search	46

3.3.4.	Variable neighborhood search	46
3.3.5.	Iterated local search	47
3.3.6.	Adaptive large neighborhood search	48
3.4.	Feasibility	51
3.5.	Move evaluation	52
4.	Hybrid adaptive large neighborhood search method	57
4.1.	Framework	58
4.2.	Pseudo-greedy time-oriented nearest neighbor heuristic	60
4.3.	Penalty functions	64
4.3.1.	EVRPTW-PR	69
4.3.2.	EVRPTW-FR	78
4.3.3.	EVRPTWDCS-FR	80
4.3.4.	EVRPTWDCS-PR	83
4.4.	Destroy and repair operators	90
4.4.1.	Worst removal	91
4.4.2.	Related removal	93
4.4.3.	Shaw removal	94
4.4.4.	CS vicinity	96
4.4.5.	Sequential insertion operator	98
4.4.6.	Other tested operators	100
4.4.7.	ALNS example	105
4.5.	Route removal operators	106
4.5.1.	Random and greedy route removal	107
4.5.2.	ALNS based route removal operator	108
4.5.3.	Repeated ALNS based route removal	110
4.5.4.	Ejection pool	110
4.5.5.	Evaluation	113
4.6.	Local search	114
4.6.1.	Order and selection	114
4.6.2.	Generate feasible solution with LS	116
4.7.	Optimal CS placement	117
4.7.1.	EVRPTW-PR	117
4.7.2.	EVRPTW-FR and EVRPTWDCS-FR	124
4.7.3.	EVRPTWDCS-PR	124
4.8.	Speed-up techniques and implementation	127
4.8.1.	Infeasible arc removal	127
4.8.2.	Nearest neighbors in local search	128

4.8.3.	Nearest CSs	130
4.8.4.	LS inter-intra relation	130
4.8.5.	Minimum number of vehicles	131
4.8.6.	Implementation	131
4.9.	Parameter tuning	132
4.10.	Results	134
4.10.1.	Small instances	134
4.10.2.	Comparison between HALNS and KESK-ALNS	139
4.10.3.	EVRPTW-PR	141
4.10.4.	EVRPTW-FR	145
4.10.5.	EVRPTWDCS-FR	148
4.10.6.	EVRPTWDCS-PR	153
5.	Time-dependent electric vehicle routing problem with time windows	155
5.1.	FIFO property and travel time computation	157
5.2.	MIP formulations	160
5.2.1.	TD-EVRPTW-FR	160
5.2.2.	TD-EVRPTWDCS-FR	163
5.2.3.	TD-EVRPTW-PR	164
5.3.	HALNS modification	164
5.4.	Testing	165
5.4.1.	VRPTW	167
5.4.2.	TD-VRPTW	169
5.5.	TD-EVRPTW-FR and TD-EVRPTWDCS-FR	170
6.	Adapted real-world delivery problem	173
6.1.	Digital map	174
6.2.	VRPTW	176
6.3.	Speed profiles	178
6.4.	TD-VRPTW	181
6.5.	EVRPTW	185
6.5.1.	Adaptation	185
6.5.2.	Results	192
6.5.3.	Minimization of energy consumption	194
6.6.	Note on the computation of time-dependent matrices	196
7.	Conclusion	199
7.1.	Achieved contributions and main conclusions	200

7.1.1.	Hybrid ALNS method development for solving different existing variants of the electric vehicle routing problem	201
7.1.2.	Development of the mixed integer program based model for time-dependent electric vehicle routing problem with time windows	201
7.1.3.	Solving the developed model of time-dependent electric vehicle routing problem with time windows using adapted hybrid ALNS method	202
7.2.	Future research	202
Bibliography		205
List of Figures		227
List of Tables		231
List of Algorithms		235
Nomenclature		237
Appendix		241
A.	List of all thesis contributions	241
B.	TD-VRPTW travel time	243
C.	TD-VRPTW total time	255
D.	TD-EVRPTW-FR travel time	259
E.	TD-EVRPTW-FR total time	271
F.	TD-EVRPTWDCS-FR recharging cost	275
Biography		279
Životopis		281

Introduction

The Vehicle Routing Problem (VRP) is an NP-hard optimization problem that aims to determine a set of least-cost delivery routes from a depot to a set of geographically scattered customers, subject to side constraints [2]. The problem is a generalization of the well-known Traveling Salesman Problem (TSP), which aims to design one least-cost route to visit all customers subject to side constraints. The problem has application in several real-life optimization problems, which has led to the definition of many problem variants over the years [3, 4].

In the past decade, the European Union (EU) has announced many new actions and regulations related to Greenhouse Gas (GHG) emissions in the transport sector [5]. External factors, such as policies and incentives, and the rise of social and ecological awareness have prompted green initiatives in many companies. Conventional Internal Combustion Engine Vehicles (ICEVs), which depend on limited fossil fuels, severely pollute the environment, especially in congested urban areas. The EU tends to decrease GHG emissions by 40% by 2030 [6]. This led to the definition of the green logistics research field [7], which deals with the sustainability of delivery processes by taking into account environmental and social factors. With the Electric Vehicle (EV) market penetration, many logistic companies evaluated the use of EVs in their vehicle fleet in order to decrease GHG emissions, and therefore to reduce the charges for every emission gram of CO₂/km. EVs have several advantages compared to ICEVs: (i) do not have local GHG emissions; (ii) produce minimal noise; (iii) can be powered from renewable energy sources; and (iv) are independent of the fluctuating oil price [8, 9]. There are two basic configurations of EVs: the Battery Electric Vehicle (BEV), which is exclusively powered from batteries mounted inside the vehicle; and the Hybrid Electric Vehicle (HEV), which can be powered from batteries inside the vehicle or by other energy sources, most commonly internal combustion engine. The Plug-in HEV (PHEV) can be recharged by connecting the plug to the electric power source. In this thesis, only BEVs for logistic purposes are considered, where two main problems come to the fore: limited driving range and the need for additional recharging

infrastructure.

Due to the limited battery capacity, the range that delivery BEVs can achieve with a fully charged battery is 160-240 km [10], which is much lower than the 480-650 km range of ICEVs [11]. To achieve a similar driving range as ICEVs, BEVs have to visit Charging Stations (CSs) more frequently. Today, there is still a lack of CSs in the road network infrastructure. Their locations and energy demand should be planned in future infrastructural plans. For an empty BEV to become operable again, battery energy has to be renewed at CS. This can be performed in two ways: (i) by swapping empty batteries with fully charged ones at a Battery Swapping Station (BSS); or (ii) by charging at CS [12, 13]. The former process can be performed in a time comparable to the refueling time of ICEVs. In the latter process, BEVs recharge their batteries at CSs by plugging into the electric power source. The recharge time depends on the State of Charge (SoC) level when entering CS, the desired SoC level when leaving CS, and the CS charging function characteristic.

With BEVs penetration in logistic distribution processes, a problem of routing a fleet of BEVs has emerged: the Electric Vehicle Routing Problem (EVRP). The EVRP aims to design least-cost BEV routes in order to serve a set of customers by taking into account often used constraints: vehicle load capacity, customer time windows, working hours, etc. [4, 14]. Additionally, BEVs have a limited driving range, which directly corresponds to more frequent recharging events at CSs. A CS can be built at separate location as a public CS or mounted at customer location as a private CS. The time needed to travel to a CS as well as recharging time are important aspects of fleet routing, especially if customer time windows are taken into account. In the last few years, several variants of EVRP emerged that most often include: heterogeneous fleet of BEVs and ICEVs, partial recharging, different CSs, nonlinear charging characteristic etc. Although several variants have already been proposed, some problem variants have not been addressed: time-dependent travel time, different CSs with full recharge option, single or multiple recharge per route, etc.

Due to the NP-hardness of the problem, exact procedures are only capable of optimally solving small-sized problems: up to 360 customers for Capacitated Vehicle Routing Problem (CVRP) [15], and 50-100 customers for Vehicle Routing Problem with Time Windows (VRPTW) [16]. Over the years, a vast number of heuristic, metaheuristic, and hybrid procedures were proposed for solving different VRP problems. Most of them are with some modifications applied to solve EVRP problems. Due to the additional modifications of charging requests, the EVRP problem is harder to solve than the similar VRP problem. Therefore, to solve different EVRP variants, researchers usually proposed specially designed metaheuristic and heuristic procedures, which not only differ in their versatility but are also often hard to implement. Therefore, the research field lacks the application of a general heuristic, which will successfully solve several different EVRP variants.

1.1 Research motivations

The main motivation of this dissertation is to give new scientific contributions in the field of electric vehicle routing, which have not been sufficiently researched in the existing literature so far. The research fields considered in this dissertation are: (i) the development of a hybrid metaheuristic capable of solving different existing variants of the EVRP problem, and (ii) the development of a new problem variant - the time-dependent EVRP.

The VRP is one of the basic optimization problems that has led to the significant development of optimization methods [4]. Due to the NP-hardness of the problem, exact methods can solve problems with a relatively small number of customers. Thus, researchers usually apply heuristic procedures based on their experience, that are able to find an acceptable solution on larger problems in reasonable computation time. The heuristic procedures, due to the narrow search space, often end up in local optima. In such cases, metaheuristic procedures are applied to escape the local optima and to search a larger solution space [3]. Given a large number of VRP variants, in the last decade, researchers have focused on developing a general heuristic that will efficiently solve different variants of the problem [17]. At the same time, with electric vehicle market penetration, the EVRP problem emerged [18], which incurred the exponential growth of research papers dealing with electric vehicle routing [1, 19, 20, 21, 22]. Based on the literature review, it can be seen that effective special metaheuristic or heuristic procedures have been proposed to address each of the EVRP variant [21]. This significantly affects the implementation possibilities for different variants of the problem and opens a gap for the development of a more general metaheuristic that will efficiently solve different EVRP variants.

In the EVRP problem, the schedule of CSs in route significantly affects the solution quality. Most of the researchers suggest special heuristic procedures for determining the schedule of CSs, as well as the recharging amount [1, 23]. Such heuristic procedures do not guarantee optimality and can produce solutions far from the optimal one. To increase the solution quality, in this thesis, the exact procedure was used to determine the optimal CS schedule. As this procedure is time-consuming, it is important to balance its calls within the metaheuristic procedure.

Most of the researchers in the EVRP field considered basic variants of the EVRP problem, such as partial charging, full charging, different CSs, and a heterogeneous vehicle fleet [1, 19, 20, 21, 22]. These variants of the EVRP problem consider static traffic network, i.e., time-independent routing. In the real world, it is a well known fact, that the travel time on the traffic network changes depending on the departure time [24, 25]. For example, in urban areas, during rush hours, when traffic jams usually occur, travel time increases significantly. In the literature, there are papers that dealt with time-dependent routing in VRP [25, 26, 27], but the time-dependent routing of electric vehicles has not yet been addressed. Taking into account

time-dependent travel time can significantly change the schedule of customers and CSs in the vehicle route. For example, it can be cost-effective to charge vehicles in CSs during peak hours and not to waste time traveling in such traffic conditions. Although, the basic variants of the EVRP problem have already been well-researched, some additional characteristics of the existing problems have still not been addressed: multiple or single charging in route, full charging in problems with different CSs, the limited capacity of CSs, waiting times at CSs, public or private CS property, energy network load, depot charge scheduling, etc. [28].

In EVRP, a hierarchical optimization function is considered, which first minimizes the number of vehicles and then minimizes total routing costs, most often consisting of the total traveled distance [1, 18, 29]. Therefore, there is a need to consider different criteria for secondary minimization such as total travel time, total time, total energy consumption, recharging costs, etc.

1.2 Major contributions of the thesis

The first objective of the thesis is to develop the HALNS method for solving different existing EVRP variants. The aim is to develop an effective method in terms of both solution quality and execution time that will be able to solve different existing EVRP variants with consideration of different secondary minimization objectives. The most important parts of the method that have not been addressed in such way are: (i) penalty functions, variables and concatenation operators, (ii) exact procedure to solve the CS placement problem, and (iii) vehicle route removal algorithms. The main goal in this part is to validate and compare the results of the proposed method to the other methods in the literature. For that purpose, the benchmark instances and Best-Known Solutions (BKSs) are used [1, 18].

The second objective of the thesis is to formulate a new time-dependent electric vehicle routing problem with time windows (TD-EVRPTW) as a mathematical mixed integer program. This is an important goal, as such a problem has not yet been addressed in the literature. The mixed integer formulation is commonly used in the VRP research field when a new problem is introduced. The linear mixed integer programs can be solved by various available solvers based on the exact procedures to find an optimal solution. As the formulated problem is not linear, the linearization of the problem will be provided to obtain a mixed integer linear program.

The third objective of the thesis is to apply the developed HALNS method to solve a newly defined problem. To apply the method on the newly-defined problem, several modifications of the proposed method are performed, mostly related to the computation of travel times and to the assurance of a non-passing property. To test whether HALNS is able to solve such a problem efficiently, the proposed method is used to solve similar time-dependent VRP problems. The test results are used to adjust HALNS parameters. The results are compared to the case without the time-dependent travel time to indicate how much the consideration of time-dependent travel

times affects the total travel time and total traveled distance.

Based on the previous description, the major contributions of this thesis can be summarized through the following three points:

1. hybrid ALNS method development for solving different existing variants of the electric vehicle routing problem;
2. development of the mixed integer program based model for time-dependent electric vehicle routing problem with time windows;
3. solving the developed model of time-dependent electric vehicle routing problem with time windows using adapted hybrid ALNS method.

1.3 Outline of the thesis

The dissertation is divided into seven chapters, with the first three chapters providing an introduction and motivation for the development of the HALNS method and a new time-dependent EVRP problem. The next two chapters describe the obtained results and scientific contributions. The sixth chapter presents the application case of the method on the adapted problem of real-world delivery. The last chapter provides a conclusion on the conducted research and guidelines for future research.

Chapter 1 provides an introduction to the dissertation, especially an introduction to the field of green logistics, which considers the problem of routing a fleet of electric vehicles. The first introduction chapter highlights the advantages and disadvantages of using electric vehicles in a delivery fleet and gives an overview of basic EV variants. Additionally, the advantages of using a metaheuristic approach to solve the VRP problem are highlighted. The first introductory chapter also provides an overview of the original scientific contributions made as a part of this thesis.

Chapter 2 provides a detailed introduction to the VRP field, with an emphasis on EVs. The combinatorial explosion of the VRP problem is described, and an overview of the characteristics and applications of EVs in delivery processes is given. The second chapter defines the basic variant of the EVRP problem - the EVRP with time windows and full recharge, along with an overview of the test instances and a mathematical formulation of the problem. Then, a detailed overview of the basic variants of the problem is given, together with a brief overview of the remaining problem variants which are not considered within this thesis.

Chapter 3 describes the procedures applied to solve VRP and EVRP problems, with an emphasis on metaheuristic procedures. The third chapter describes the most commonly used heuristic procedures for the construction of the initial solution, as well as the most commonly used local search operators to improve the solution. Further on, an overview of the metaheuristic procedures applied to solve the EVRP problem is given. At the end of the third chapter, the

importance of considering the objective function with penalties for the constraints violations and the technique for solution evaluation are discussed.

In chapter 4, a detailed description of the developed HALNS method is given. The fourth chapter begins with a description of the general framework of the HALNS method. Afterward, the descriptions of the individual parts of the HALNS method are presented. First, the method used to construct an initial solution of the EVRP problem with time windows and full recharge is described. Then, the functions and variables used to evaluate the different variants of the EVRP problem are described: (i) EVRP with time windows and partial recharge, (ii) EVRP with time windows and full recharge, (iii) EVRP with time windows, full recharge and different charger types in CSs, and (iv) EVRP with time windows, partial recharge and different charger types in CSs. Afterwards, the fourth chapter describes the operators used to remove and insert customers, CSs, entire vehicle routes, and additionally describes the remaining operators that were considered during the development phase. Next, local search operators are selected based on their performance, together with the determination of their execution order. After the local search, the exact procedure for optimal CS placement in a fixed vehicle route is described. Prior to the results, the used speed-up techniques and tuned HALNS parameters are outlined. At the end of the fourth chapter, the results of the HALNS method on small and large test problems are presented and compared with: (i) exact method, (ii) implemented ALNS method without problem improvement procedures presented by Keskin et al. [1], (iii) best published solutions on several problem variants, and (iv) existing problems with different secondary objective functions.

Chapter 5 begins with an introduction to the time-dependent VRP problem and provides motivation for the development of a new time-dependent EVRP problem. Then, the basic characteristics of time-dependent routing are described, which include the method for travel time computation and insurance of the no-passing property. The chapter presents a mathematical formulation of the problem as a mixed integer program. The developed HALNS method is tested on related time-dependent VRP problems. At the end of the fifth chapter, the results of the HALNS method on a newly developed problem instances are presented.

Chapter 6 presents the application of the developed HALNS method on the adapted real-world delivery problem. The delivery problem is modified in such a way that CSs and BEVs are included. The results cover solving different versions of the VRP and EVRP problems with different secondary objective functions.

In the final chapter 7, the conclusion of the thesis is given, and the original scientific contributions that were achieved throughout the thesis are highlighted. Furthermore, an overview of topics for future research in the field of EVRP is presented, such as applying the developed HALNS method on heterogeneous vehicle fleet problems involving both ICEVs and EVs or developing a central system for the collection of the best published results in the EVRP field.

Electric vehicle routing problem

The electric vehicle routing problem is a special case of a more general VRP problem where EVs perform the delivery of goods to customers. VRP is one of the most important problems in logistic distribution. The VRP problem was first defined as a truck dispatching problem [30]. The basic VRP problem consists of a depot, geographically scattered customers, and a set of delivery vehicles positioned at the depot. Each customer has to be visited only once, and in the end, all vehicles have to return to the depot. Meanwhile, the goal is to minimize overall routing costs, which includes minimization of vehicle number and specific vehicle routing costs. The real-world application has led to many different variants of the basic VRP problem. Here the most important ones are listed:

- Capacitated Vehicle Routing Problem (CVRP) [4] - each customer has a demand - a quantity representing the amount of goods that need to be delivered to the respective customer, and each vehicle has a limited load (cargo) capacity, usually expressed in volume or mass unit,
- Vehicle Routing Problem With Time Windows (VRPTW) [31] - each customer has to be visited within its time window (early and late time of the service start time),
- Multi-Depot Vehicle Routing Problem (MDVRP) [4] - multiple depots where the start and end depot do not have to be the same,
- Vehicle Routing Problem with Pickup and Delivery (VRPPD) [32] - each customer has a request for pickup and delivery (drop-off), and each request must be handled by the same vehicle,
- Heterogenous or Mix Fleet Vehicle Routing Problem (MFVRP) [33] - vehicles have different load capacities,
- Time-Dependent Vehicle Routing Problem (TD-VRP) [25] - travel time between customers depends on the departure time.

In the VRP, often the primary objective is to minimize the total number of vehicles used

and then to minimize the total traveled distance or some other objective function [34]. The total vehicle number is a primary objective as generally greater savings can be achieved with fewer vehicles (vehicle fixed costs, labor cost, maintenance cost). Such an objective is contradictory as with fewer vehicles, total traveled distance increases, and vice versa. By taking into account BEVs high purchase price, such a hierarchical objective seems even more reasonable in BEV routing applications [18, 35].

In EVRP problem, the objective functions can be complex with simultaneous minimization of: vehicle number, total traveled distance [1], total travel times [36], total routing cost and planning horizon [9, 20, 29, 35], GHG emission [37, 38], energy consumption [39, 40, 41], recharging cost [22], etc. Total routing costs of BEVs usually consist of acquisition cost, circulation tax, maintenance, costs related to the energy consumption (electric energy price), cost of battery pack renewal after its lifetime, labor costs, etc. Instead of a single-objective function, some researchers use a multi-objective function, i.e., fuel consumption and total driving time [42], fuel consumption and route cost [43], battery swapping and charge scheduling [44], etc.

The VRP problem can be modeled on a complete directed or undirected graph, where customers are modeled as graph vertices, and paths between customers are modeled as graph arcs. Most of the VRP variants consider undirected graphs, meaning that the path between customers is the same in both directions. To better model a real traffic network, the directed graph can be used in cases when the path between customers changes depending on the direction or in the cases of one-way roads and forbidden turns.

The important aspect of all VRP problems is the combinatorial explosion, which happens due to the NP-hardness of the VRP problem. For example, for a basic TSP problem without capacity or time-window constraints, humans are only capable of sub-optimally solving problems with a small number of customers, up to 15, depending on the spatial distribution of the customers. Figure 2.1a presents an example of an instance with 15 customers, where the purple square is the depot and the black circles are the customers. The Euclidean distance measure is used. Based on the intuition, humans would usually connect customers near each other to form the route, and in most cases, end up in a sub-optimal solution as in Figure 2.1b, with the total traveled distance of 105.9. Also, for two presented solutions such as those in figures 2.1b and

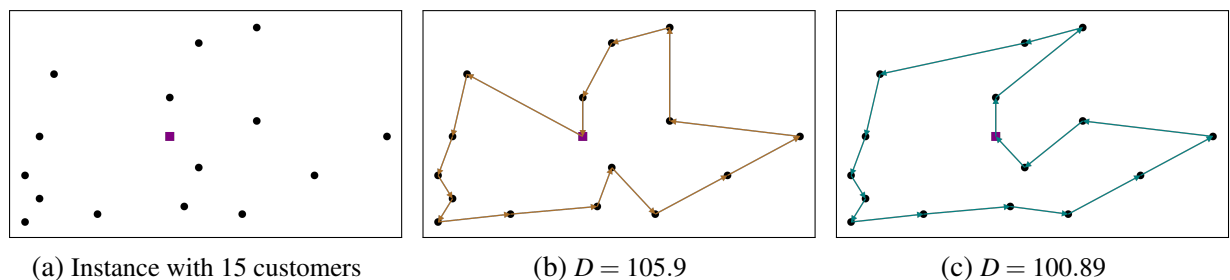


Figure 2.1: Examples of TSP instance with 15 customers

2.1c, for humans without the measure value, it is visually hard to determine which solution is better. The decisions get much harder for a larger number of customers as in Figure 2.2 or if additional constraints are included. The TSP can be considered as a special case of VRP, which is harder to solve than TSP.

To solve larger instances, the advantages of computer resources can be used, which are able to perform a lot more operations than humans in the same unit of time. But still, even with the use of computers, this basic TSP problem is hard to solve. For example, let there be 50 customers without a depot, in the asymmetric TSP problem. The goal is to go through all possible combinations to find the optimal solution - the so-called brute force approach. For 50 customers there are $50! = 3.04 \cdot 10^{64}$ permutations of customers schedule. If assumed, that the check of one combination can be performed in the smallest ever possible measured time called Planck Time $t_p = 5.39 \cdot 10^{-44}$ (best modern computers perform $2.36 \cdot 10^{12}$ instructions per s $\rightarrow 4.24 \cdot 10^{-13}$ s per instruction), then the time to check all combinations would be $1.64 \cdot 10^{21}$ s. To put into the real context, the estimated age of the universe is $13.7 \cdot 10^9$ years $= 4.32 \cdot 10^{17}$ s. This shows that solving a TSP problem with a brute force algorithm is impossible, and some other techniques need to be used.

In the rest of this chapter, first, the characteristic of BEVs are described. Then in the following sections, different EVRP variants are described, with the notation of the variables used in mathematical problem formulations presented in Table 2.1.

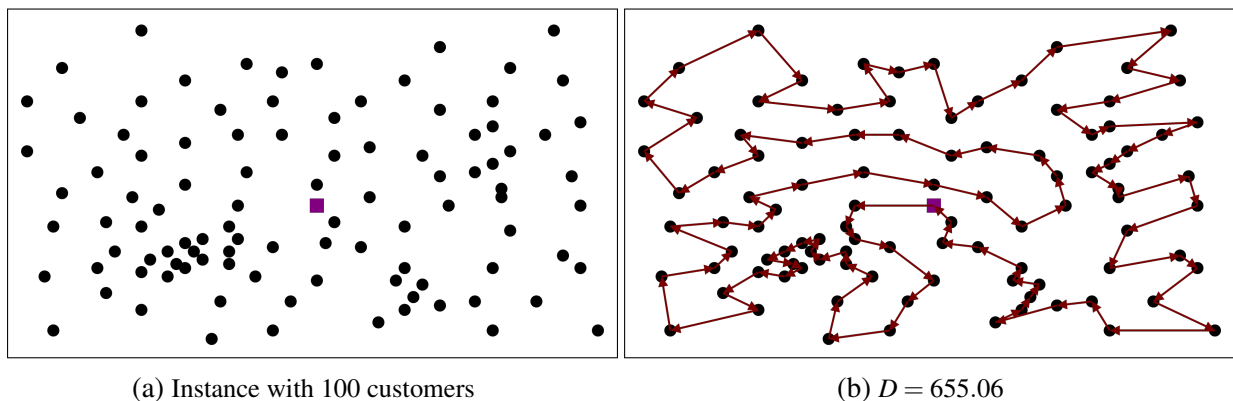


Figure 2.2: Examples of TSP instance with 100 customers

2.1 Battery electric vehicles in routing problems

The major problem that BEVs in delivery processes face is the limited driving range. Grunditz et al. [45] analyzed over 40 globally available BEVs, which can be categorized into small, medium-large, high-performing, and sport cars. Most of the BEV models utilize lithium-based batteries, especially lithium-ion [46], with battery capacity and distance varying between 12-90 kWh and 85-528 km, respectively. An average medium-sized personal BEV has a battery capacity of 30 kWh, which is enough to travel 250 km. In the delivery processes, mostly light vans and freight BEVs are used, which have a shorter driving range (160-240 km) compared to the driving range of ICEVs (480-650 km) [11, 47, 48]. The main reason for the shorter driving range is that the battery has a lower specific energy (130 Wh/kg) than the fossil oil (1233 Wh/kg). Batteries mounted in BEVs are mostly the main cause of high acquisition costs and technical limitations, as the battery degrades over time, resulting in decreased capacity. There are also other factors that influence such battery degradation: overcharging, overdischarging, high and low temperatures, high SoC during storage, large depth of discharge, etc. Usually, the battery should be replaced after five to ten years or after 1000 to 2000 cycles with large SoC variations [49].

BEVs are more likely to be used on short distances or in urban areas where they are more effective than ICEVs due to the low driving speed, low noise production, frequent stops, and financial incentives. In cases when the average route length is short, such as the average FedEx route length in the USA, which is 68 km [10], BEVs can be applied directly, and recharging can be performed on the return to the depot. BEVs are already being applied in such occasions in companies like: DHL, UPS, FedEx, and Coca-Cola [50, 51], use BEVs mostly for the last-mile delivery as distances are shorter and vehicle loads are lower. Many companies are performing case studies of integrating BEVs in their delivery fleet, but are still cautious, because the use of BEVs in time-precise deliveries causes hard completion of an on-time delivery, which then significantly increases the overall routing costs [52]. Some authors also reported that BEVs are not competitive if the solution to the same problem results in a higher number of BEVs than the number of ICEVs [47]. There are several key elements for BEVs to be competitive: daily traveled distance close to the maximum BEV driving range, low speeds and congestions, frequent customer stops, the decrease of a BEV's purchase cost by tax incentives or technology development, and long planning horizon [8]. On the other hand, several researchers conducted case studies of using freight BEVs for delivery and point out that there are no operational limitations when using BEVs compared to ICEVs, as the used number of vehicles and total traveled distance are competitive, and at the same time, the overall costs are lower with almost 25% less CO₂ emission [35]. To fully assess the integration of BEVs in logistic processes, there are three key elements: solution to the problem in terms of the used number of CSs and

BEVs, future CO₂ emission policies, and future technology development (battery capacity and charging infrastructure) [9].

2.2 Electric vehicle routing problem with time windows and full recharge

The Electric Vehicle Routing Problem with Time Windows and Full Recharge (EVRPTW-FR) proposed by Schneider et al. [18] is the first researched problem that considers routing of a BEV fleet, possible visits to CSs, and charging time dependent on the SoC level. The problem considers a set of homogeneous BEVs with equal load and battery capacities and a set of customers with demands and time windows. The constraints of the problem are:

- (i) each customer is visited exactly once, by only one vehicle,
- (ii) each vehicle starts and ends in a depot,
- (iii) the sum of demand of customers in a vehicle does not exceed vehicle load capacity,
- (iv) the SoC level never drops below 0,
- (v) full charging at CS does not exceed vehicle battery capacity,
- (vi) each customer is visited within its time window,
- (vii) the total vehicle time does not exceed depot working hours.

The problem can be formulated as a Mixed Integer Linear Program (MILP) on the complete directed graph G where customers are modeled as graph vertices, and paths between customers are modeled as graph arcs. The variables used to model the EVRPTW-FR problem are presented in Table 2.1.

Let $V = \{1, \dots, N\}$ be a set of geographically scattered customers that need to be served, and let F be a set of CSs. In order to allow multiple visits to same CS, a virtual set of CSs F' is defined, where β represents the maximal number of virtual CSs (visits) to a single CS. Vertices 0 and $N + 1$ denote the depot instances, and every route begins at vertex 0, and ends at vertex $N + 1$ ($V_{0,N+1} = V \cup \{0\} \cup \{N + 1\}$). Graph G is defined as $G = (V_{0,N+1} \cup F', A)$, where A is the set of arcs $A = \{(i, j) | i, j \in V_{0,N+1} \cup F', i \neq j\}$. The binary variable $x_{ij} \in \{0, 1\}$ (equation 2.1) is equal to 1 if arc (i, j) is traversed in the solution, and 0 otherwise. The arc value d_{ij} represents the arc distance, t_{ij} represents the time needed to traverse the arc, and e_{ij} the energy consumption on the arc. The distance matrix representing the shortest path between each vertex is computed in advance in the preprocessing step. The hierarchical objective function consists, first of vehicle number minimization (equation 2.2), and then the total traveled distance minimization (equation 2.3). Each BEV has a load capacity C and battery capacity Q . Recharge time is computed as linear function value of recharged capacity with recharge rate g . The energy consumption on the arc is computed as linear function value of arc distance, $e_{ij} = rd_{ij}$, where r is the energy consumption rate. Each vertex i (customer, station or depot - further on referred

Table 2.1: Notation used in EVRP problems

Notation	Description
V	Set of customers
F	Set of CSs
F'	Set of CSs with dummy vertices
β	Number of dummy vertices
$0, N + 1, (DD, AD)$	Depot beginning and ending instances
x_{ij}	Binary variable for arc (i, j)
d_{ij}	Arc distance
t_{ij}	Arc travel time
$t_{ij}(\delta_i)$	Arc travel time function depending on the departure time δ_i at user i
e_{ij}	Arc energy consumption
C	Vehicle load capacity
Q	Vehicle battery capacity
r	Energy consumption rate
s_i	Service time at user i
e_i	Early time window at user i
l_i	Late time window at user i
q_i	Load demand of user i
τ_i	Earliest begin time at user i
u_i	Remaining vehicle load capacity at user i
y_i	Remaining battery capacity at user i
Y_i	Remaining battery capacity at the departure from CS i
g, g^m	Energy recharge rate of charger type m
c^m	Charging cost of charger type m
a_i, b_i	Binary variables for determining charger type at CS i
θ_i^m	Amount of energy recharged at CS i using charger type m
c_t	Route duration cost
c_d	Route distance cost
K	Maximum number of vehicles

as user) has a service time s_i , load demand q_i and time window $[e_i, l_i]$, while CSs and depots have the time window $[e_0, l_0]$ i.e. working hours. Beside the x_{ij} decision variable, three more decision variables for vertices $i \in V_{0,N+1} \cup F'$, are used: τ_i - begin time, u_i - remaining load capacity, and y_i - remaining battery capacity.

Equation 2.4 ensures the arc connectivity of customers, meaning that each customer can have only one exit arc to a customer or CS. Equation 2.5 ensures the arc connectivity of CSs, meaning that each virtual CS can have at maximum one exit arc to either a customer or another CS. Thus, there is also a possibility of no exit arc if a CS is not visited. Equation 2.6 for each user ensures that the number of entry arcs is the same as the number of exit arcs, which together with 2.4 and 2.5 ensure that each customer is visited exactly once, and that each virtual CS if visited, has only one entry and exit arc. Equation 2.7 ensures travel time feasibility of arcs between customer i and user j , which can be a customer or CS. If arc (i, j) is traversed, then

the begin time at user j has to be equal to or greater than the sum of begin time at customer i , travel time between i and j , and service time of customer i . Otherwise, if arc (i, j) is not traversed, then the difference between begin times at users i and j has to be lower than the depot working hours. This is done in order to reduce the search space. Equation 2.8 ensures travel time feasibility of arcs between CS i and user j . If arc (i, j) is traversed, then the begin time at vertex j has to be equal to or greater than the sum of begin time at CS i , travel time between users i and j , and recharging time at CS i . Otherwise, the search space is reduced in a similar way as in equation 2.7. Equation 2.9 ensures the travel time feasibility of a user, meaning that the begin time has to be within the user's time window. Equations 2.10, 2.12 and 2.13 ensure arcs load and battery capacity, in similar way as equations 2.7 and 2.8. Equation 2.11 ensures that the leaving depot instance has a remaining load capacity equal to the vehicle load capacity C .

$$x_{ij} \in \{0, 1\}, \forall i \in V_0 \cup F', j \in V_{N+1} \cup F', i \neq j \quad (2.1)$$

$$\min \sum_{j \in V \cup F'} x_{0j} \quad (2.2)$$

$$\min \sum_{i \in V_0 \cup F'} \sum_{j \in V_{N+1} \cup F', i \neq j} d_{ij} x_{ij} \quad (2.3)$$

$$\sum_{j \in V_{N+1} \cup F', i \neq j} x_{ij} = 1, i \in V \quad (2.4)$$

$$\sum_{j \in V_{N+1} \cup F', i \neq j} x_{ij} \leq 1, i \in F' \quad (2.5)$$

$$\sum_{i \in V_{N+1} \cup F', i \neq j} x_{ji} - \sum_{i \in V_0 \cup F', i \neq j} x_{ij} = 0, j \in V \cup F' \quad (2.6)$$

$$\tau_i + (t_{ij} + s_i)x_{ij} - l_0 \cdot (1 - x_{ij}) \leq \tau_j, \forall i \in V_0, \forall j \in V_{N+1} \cup F', i \neq j \quad (2.7)$$

$$\tau_i + g(Q - y_i) + x_{ij}t_{ij} - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j, \forall i \in F', \forall j \in V_{N+1} \cup F', i \neq j \quad (2.8)$$

$$e_j \leq \tau_j \leq l_j, \forall j \in V_{0, N+1} \cup F' \quad (2.9)$$

$$0 \leq u_j \leq u_i - x_{ij}(q_i + C) + C, \forall i \in V_0 \cup F', \forall j \in V_{N+1} \cup F', i \neq j \quad (2.10)$$

$$u_0 = C \quad (2.11)$$

$$0 \leq y_j \leq y_i - (e_{ij} + Q)x_{ij} + Q, \forall j \in V_{N+1} \cup F', \forall i \in V, i \neq j \quad (2.12)$$

$$0 \leq y_j \leq Q - e_{ij}x_{ij}, \forall j \in V_{N+1} \cup F', \forall i \in 0 \cup F', i \neq j \quad (2.13)$$

Beside the distance minimization for the secondary objective, the other objectives can be considered:

- Travel time:

$$\min \sum_{i \in V_0 \cup F'} \sum_{j \in V_{N+1} \cup F', i \neq j} t_{ij}x_{ij}, \quad (2.14)$$

- Energy consumption:

$$\min \sum_{i \in V_0 \cup F'} \sum_{j \in V_{N+1} \cup F', i \neq j} e_{ij}x_{ij}, \quad (2.15)$$

- Total time (requires several instances of end depot (AD) to track begin time):

$$\min \sum_{i \in V \cup F'} \sum_{j \in AD} x_{ij}(\tau_j - e_0). \quad (2.16)$$

For problems with load capacity, the theoretical lower bound on the number of vehicles can be determined by equation 2.17, as the ceiling integer value of the sum of customers demand q_i divided by the vehicle load capacity C .

$$n_v^{min} = \left\lceil \frac{\sum_{i \in V} q_i}{C} \right\rceil \quad (2.17)$$

Schneider et al. [18] were the first one to propose the test instances for the Electric Vehicle Routing With Time Windows (EVRPTW), which are widely used in the research community for testing various procedures applied to solve the problem [1, 20, 23, 29, 53, 54, 55]. The authors created two sets of benchmark instances: (i) 56 large instances, each with 100 customers and 21 CSs, and (ii) 36 small instances, with 5, 10 and 15 customers per instance. All instances are created based on the well-known Solomon benchmark instances for VRPTW problem [31]. Small instances are usually used to compare exact and metaheuristic procedures applied to solve the problem [1, 18, 20]. The instances are divided into three groups, depending on the geographical distribution of the customers: clustered customer distribution (C) (example in Figure 2.3a), random customer distribution (R) (example in Figure 2.3b), and a mixture of both (RC) (example in Figure 2.3c). Additionally, the instances are also divided into two groups based on the scheduling horizon: short scheduling horizon with narrow time windows (1), and long scheduling horizon with wide time windows (2). As a result, there are in total six instance types.

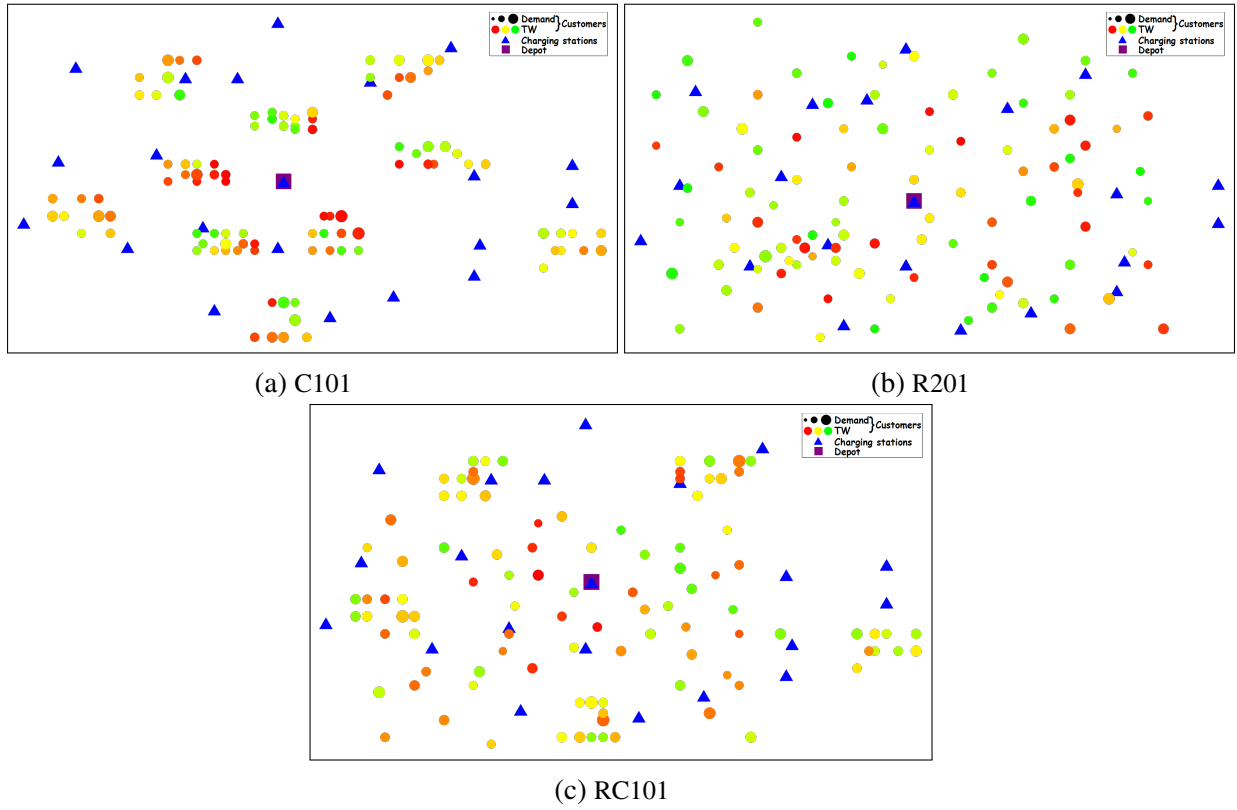


Figure 2.3: Examples of EVRPTW instances

In all of the instances, the Euclidean distance is considered for distance computation, given by equation 2.18. The instances were created in such way, that each customer is directly reachable from the depot regarding time windows, while regarding the energy feasibility, at maximum two CS are needed to visit the customer. The locations of customers in instances are the same as in Solomon instances [31], but the time window values were relaxed in EVRPTW instances, as some instances were infeasible in the original Solomon instances if BEVs were included, due to the charging time at CSs.

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.18)$$

Three examples of different instances are presented in Figure 2.3. For example, in instance RC101, RC represents the geographical distribution, the first value afterward represents the scheduling horizon of type 1, and the last two numbers represent the instance number. In this thesis, in all figures related to the EVRP, the customers are represented as filled circles which have two attributes: (i) size, which represents the demand of a customer - the greater the demand is, the larger the circle is, and (ii) color from red to green, with red representing customers that close sooner (need to be visited sooner) and green representing customers that close later, and thus can be visited later in the route. The depot is represented with the purple rectangle, while the CSs are represented with blue triangles.

Table 2.2: EVRPTW instances - group values

	C1	C2	R1	R2	RC1	RC2
Q	79.69	[117.66, 118.31]	[62.14, 67.15]	[181.23, 267.18]	79.69	[159.68, 273.13]
C	200	700	200	1000	200	1000
r	1	1	1	1	1	1
g	3.39	[2.28, 2.29]	[0.45, 0.48]	[0.11, 0.17]	0.38	[0.11, 0.19]

Vehicle battery capacity Q , vehicle load capacity C , energy consumption rate r , and recharge rate g per instance types are presented in Table 2.2. It can be seen that instance types with wider time windows have larger vehicle load and battery capacities, which results in a lower number of vehicles per instance.

Figure 2.4 presents BKS for the EVRPTW-FR problem on the instance C101. In total, there are 12 vehicles, each represented with a different color. The total traveled distance is 1053.83, while the total time is 12904.61.

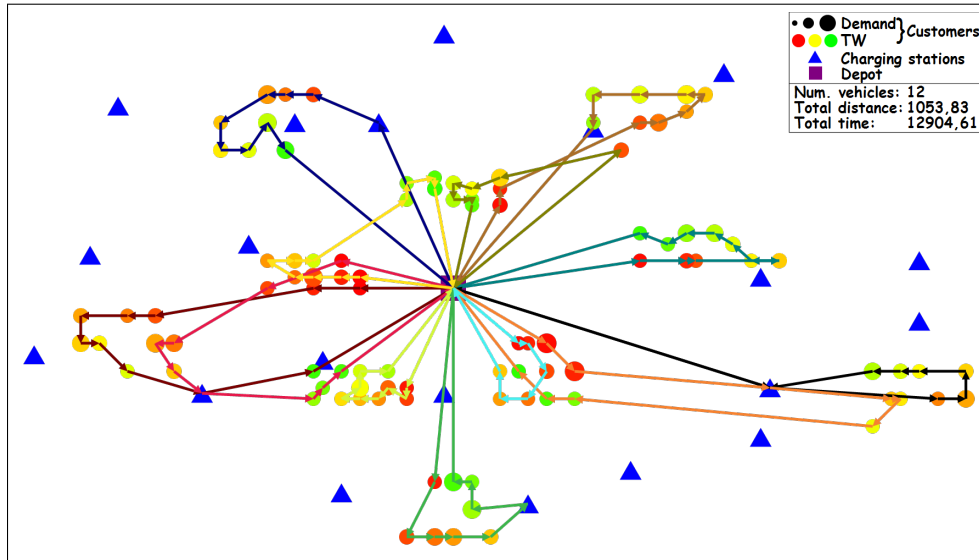


Figure 2.4: C101: EVRPTW-FR, BKS

2.3 Electric traveling salesman problem with time windows

As VRP is a generalization of the TSP, the EVRP is closely related to the Electric Traveling Salesman Problem (ETSP), in which a set of customers has to be served by only one BEV [56]. There exist only a few variants of the ETSP problem: (i) hybrid TSP, in which hybrid vehicles with four operating modes are used [57], and (ii) EV touring problem in which BSSs are used and the total travel time is minimized [48]. Here, the Electric Traveling Salesman Problem with Time Windows and Full Recharge (ETSPTW-FR) is observed in which customers have no demand, and only the battery capacity of the vehicle is considered. The example of the

ETSPTW-FR optimal solution for instance C202-15 is presented in Figure 2.5. The general ETSP can be modeled as a compact MILP program, in the same way as the EVRPTW-FR in section 2.2. Here only the differences are highlighted. Equation 2.2 is omitted, as only one vehicle is used, as well as equations 2.10 and 2.11 for tracking the remaining vehicle load capacity.

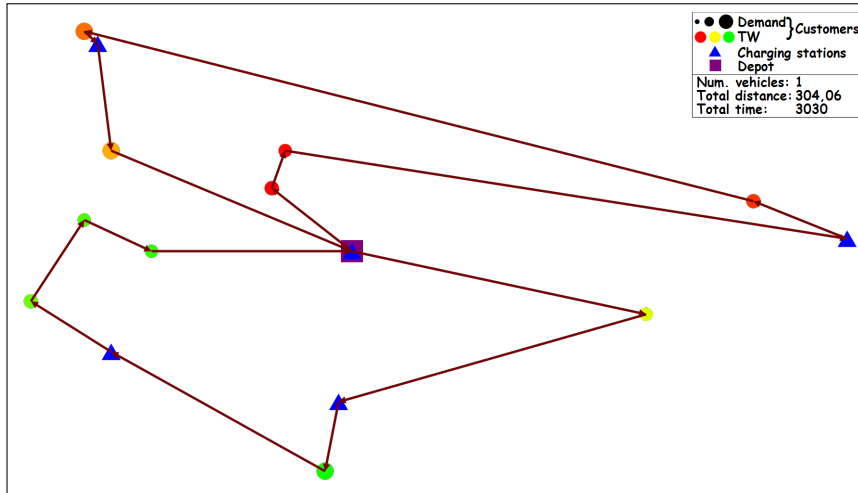


Figure 2.5: C202-15: ETSPTW-FR, optimal

2.4 Electric vehicle routing problem with time windows and partial recharging

The full recharge strategy can be time-consuming because, depending on the SoC level, available charging technology, and battery capacity, the vehicle can charge from five minutes to eight hours [13]. In real-world applications, the battery should be charged enough to complete the whole route or to surpass a fear that the vehicle range will not be enough to perform designated tasks, the so-called range anxiety [58]. The benefits of a partial recharging strategy are particularly visible on customers with narrow time windows, where efficient charge scheduling can enable the feasibility of the route, which would be infeasible with a full recharge strategy. Additionally, partial recharge strategy generally reduces routing costs by roughly 3.80% compared to the full recharge strategy [55]. On the economic side, significant savings can be achieved by partial recharging as a minimal amount of energy could be recharged during the day, when the electricity cost and energy network load are higher, and the rest of the energy could be replenished during the night [23, 59]. In some cases, it is natural to maintain the energy reserve. This can be done by SoC limitation, i.e., in [20, 95]% range [59, 60, 61]. Having an energy reserve seems even more important if energy consumption and range anxiety are taken into account because up to 30% of the consumed energy can be spent on BEV's auxiliary devices [62]. Lim-

iting SoC value also helps to preserve the battery, as battery capacity decreases by overcharging and over-discharging [49]. This all led to the definition of Electric Vehicle Routing Problem with Time Windows and Partial Recharging (EVRPTW-PR) [1, 23, 36, 63].

The EVRPTW-PR can be modeled as a compact MILP program, the same as EVRPTW-FR in section 2.2. The used variables are presented in Table 2.1. Here, only the differences between the models are highlighted. First of all, a new decision variable Y_i is added, which represents the remaining battery capacity on the departure from CS i , given by equation 2.19. The remaining battery capacity after recharging Y_i has to be, somewhere in between the remaining battery capacity without charging y_i and full recharge capacity Q . The equations that changed are the equations for CSs exit arc travel feasibility and battery feasibility ensurance (2.8 and 2.13), and are replaced with equations 2.20 and 2.21. The only change is that instead of charging to the full capacity Q , the vehicle is charged up to the decision variable Y_i .

$$y_i \leq Y_i \leq Q, \forall i \in F'_0, \quad (2.19)$$

$$\tau_i + g(Y_i - y_i) + x_{ij}t_{ij} - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j, \forall i \in F'_0, \forall j \in V_{N+1} \cup F', i \neq j \quad (2.20)$$

$$0 \leq y_j \leq Y_i - (e_{ij} + Q)x_{ij} + Q, \forall i \in F'_0, \forall j \in V_{N+1} \cup F', i \neq j \quad (2.21)$$

Figure 2.6 presents BKS for EVRPTW-PR problem on the instance C101. It can be seen that the total traveled distance decreased compared to the full recharge strategy, while the number of vehicles remained the same.

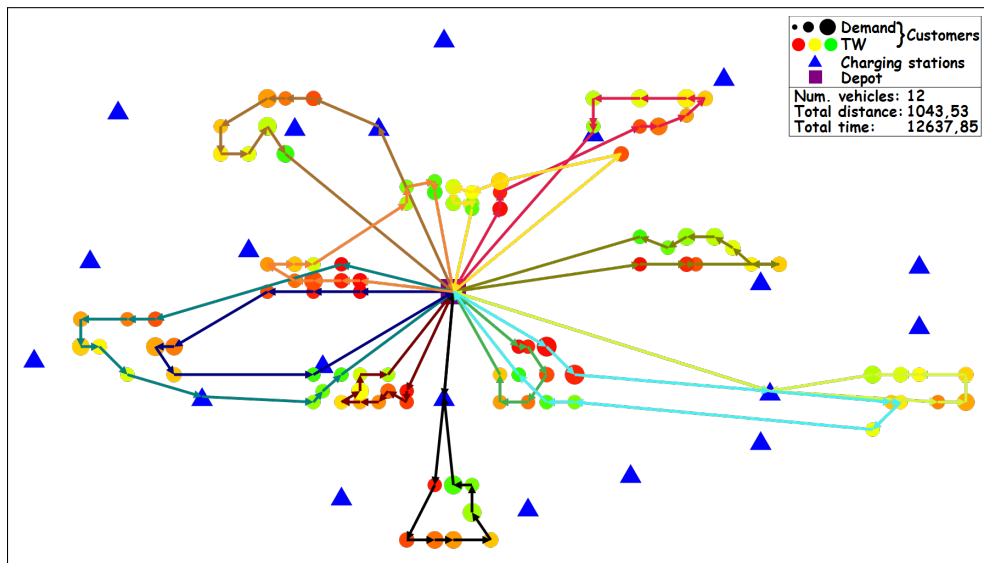


Figure 2.6: C101: EVRPTW-PR, BKS

2.5 Electric vehicle routing problem with different charging stations

Today, multiple charging technologies are present: (i) slow - 3 kW (6-8 h); (ii) fast - 7-43 kW (1-2 h); and (iii) rapid - 50-250 kW (5-30 min) [13, 64]. To better control the charging time when routing BEVs, the selection of possible charging technology could also be optimized. This could make some customers who have narrow time windows more accessible by fast charging at previous CSs, or if the time windows are wide, an economically better approach could be slow charging. Such problem could be extended by taking into account CS working hours, time-dependent charging costs, number of available chargers, compatibility of chargers with BEVs, power grid load, charger power, etc. [23, 59, 61, 65, 66, 67, 68, 69].

The use of only rapid charging option can reduce the fleet size and decrease the total energy consumption, but it can increase the overall routing costs as the rapid charging option is the most expensive one [70]. The lowest overall routing cost can be obtained if joint technologies are used [23]. Keskin et al. [22] formulated the Electric Vehicle Routing Problem with Time Windows and Fast Charging (EVRPTW-FC), in which different charger types in CSs and different charging costs are considered. The authors considered partial recharging and formulated the problem as an MILP model. To ease the understanding of the observed problems, in this thesis, this problem will be named Electric Vehicle Routing Problem with Time Windows, Different Charging Stations and Partial Recharging (EVRPTWDCS-PR), where the term "different charging stations" refers to different charger types used in CSs. The natural extension of the EVRPTW-PR problem to EVRPTWDCS-PR problem would be to replicate each CS for each different CS technology. As pointed by Keskin et al. [22] such model is inferior in both execution time and solution quality compared to the model in which each CSs can have different charging technology, and binary variables are used for the decision of the technology used.

The EVRPTWDCS-PR can be modeled as an MILP program, similar as the MILP program for EVRPTW-PR problem presented in section 2.4. Variables used for the formulation of the problem are presented in Table 2.1. To keep track of BEV's energy consumption, the authors created several copies of depot instances: DD set for leaving depot instances at the beginning of the route, and AD set for arrival depot instances at the end of the route. The customer and CS sets are changed to include those instances, i.e. $V_{DD,AD} = V \cup \{DD\} \cup \{AD\}$. Graph G is defined as $G = (V_{DD,AD} \cup F', A)$, where A is the set of arcs $A = \{(i, j) | i \in V_{DD} \cup F', j \in V_{AD} \cup F', i \neq j\}$. Additionally, binary decision variables a_i and b_i (equation 2.43) are added to each CS $i \in F'$ to determine which charging technology is used to charge the vehicle. Only one charging technology can be used per visited CS. If rapid charging technology is used then $a_i = 1$ and $b_i = 0$, if fast charging technology is used then $a_i = 0$ and $b_i = 1$, and if slow charging technology is used then $a_i = 0$ and $b_i = 0$. The battery recharging rate and unit energy

2. Electric vehicle routing problem

cost depend on the charger type $m \in M$ and are referred to as g^m and c^m , respectively. Set $M = \{1, 2, 3\}$ contains three charging technologies, where 1 represents rapid charging technology, 2 fast charging technology, and 3 slow charging technology. The decision variable θ_i^m is added to determine the amount of energy recharged at CS i using charger type m . The minimization of the vehicle number is given by equation 2.22, while the minimization of the total recharging costs is given by equation 2.23. The recharging costs consist of the recharging costs produced in CSs (first part) and the recharging costs corresponding to the amount of the energy spent from the starting full battery charged at depot with the slow charger (second part). Due to the new depot instances equations 2.27-2.29 are added to allow at maximum one exit arc for DD set and one entry arc for AD set. The sum of charging amount per technology must be equal to the difference of vehicle battery capacity after charging and vehicle battery capacity before charging (equation 2.39). This charging amount is then used in equation 2.31 as a replacement for charging up to a certain threshold. Equations 2.37, 2.38, and 2.40-2.42 limit the remaining battery capacity before and after charging, and charging amount to maximum battery capacity Q . The rest of the equations are similar to the MILPs for EVRPTW-FR and EVRPTW-PR, with the only difference that DD and AD sets are used.

$$\min \sum_{i \in DD} \sum_{j \in V \cup F'} x_{ij} \quad (2.22)$$

$$\min \sum_{i \in F'} \sum_{m \in M} c^m \theta_i^m + c^3 \left(Q \sum_{i \in DD} \sum_{j \in V \cup F'} x_{ij} - \sum_{i \in AD} y_i \right) \quad (2.23)$$

$$\sum_{j \in V_{AD} \cup F', i \neq j} x_{ij} = 1, i \in V \quad (2.24)$$

$$\sum_{j \in V_{AD}, i \neq j} x_{ij} \leq 1, i \in F' \quad (2.25)$$

$$\sum_{i \in V_{AD} \cup F', i \neq j} x_{ji} - \sum_{i \in V_{DD} \cup F', i \neq j} x_{ij} = 0, j \in V \cup F' \quad (2.26)$$

$$\sum_{j \in V \cup F', i \neq j} x_{ij} \leq 1, i \in DD \quad (2.27)$$

$$\sum_{i \in V \cup F', i \neq j} x_{ij} \leq 1, j \in AD \quad (2.28)$$

$$\sum_{i \in DD} \sum_{j \in V \cup F'} x_{ij} = \sum_{i \in AD} \sum_{j \in V \cup F'} x_{ji} \quad (2.29)$$

$$\tau_i + (t_{ij} + s_i)x_{ij} - l_0 \cdot (1 - x_{ij}) \leq \tau_j, \forall i \in V_{DD}, \forall j \in V_{AD} \cup F', i \neq j \quad (2.30)$$

$$\tau_i + \sum_{m \in M} g^m \theta_i^m + x_{ij} t_{ij} - (l_0 + g^3 Q)(1 - x_{ij}) \leq \tau_j, \forall i \in F', \forall j \in V_{AD}, i \neq j \quad (2.31)$$

$$e_j \leq \tau_j \leq l_j, \forall j \in V_{DD,AD} \cup F' \quad (2.32)$$

$$0 \leq u_j \leq u_i - x_{ij}(q_i + C) + C, \forall i \in V_{DD} \cup F', \forall j \in V_{AD} \cup F', i \neq j \quad (2.33)$$

$$0 \leq u_i \leq C, \forall i \in DD \quad (2.34)$$

$$0 \leq y_j \leq y_i - (e_{ij} + Q)x_{ij} + Q, \forall j \in V_{AD} \cup F', \forall i \in V, i \neq j \quad (2.35)$$

$$0 \leq y_j \leq Y_i - e_{ij}x_{ij} + Q(1 - x_{ij}), \forall j \in V_{AD}, \forall i \in DD \cup F', i \neq j \quad (2.36)$$

$$0 \leq y_j \leq Y_i \leq Q, \forall i \in DD \cup F' \quad (2.37)$$

$$Y_i = Q, \forall i \in DD \quad (2.38)$$

$$Y_i - y_i = \sum_{m \in M} \theta_i^m, \forall i \in F' \quad (2.39)$$

$$0 \leq \theta_i^1 \leq Qa_i, \forall i \in F' \quad (2.40)$$

$$0 \leq \theta_i^2 \leq Qb_i, \forall i \in F' \quad (2.41)$$

$$0 \leq \theta_i^3 \leq Q(1 - a_i - b_i), \forall i \in F' \quad (2.42)$$

$$a_i, b_i \in \{0, 1\}, \forall i \in F' \quad (2.43)$$

$$x_{ij} \in \{0, 1\}, \forall i \in V_{DD} \cup F', \forall j \in V_{AD} \cup F', i \neq j \quad (2.44)$$

The BKS for EVRPTWDCS-PR problem on instance R101 is presented in Figure 2.7. The total charging cost is 1686.77, with the utilization of 10 fast charges and 22 slow chargers.

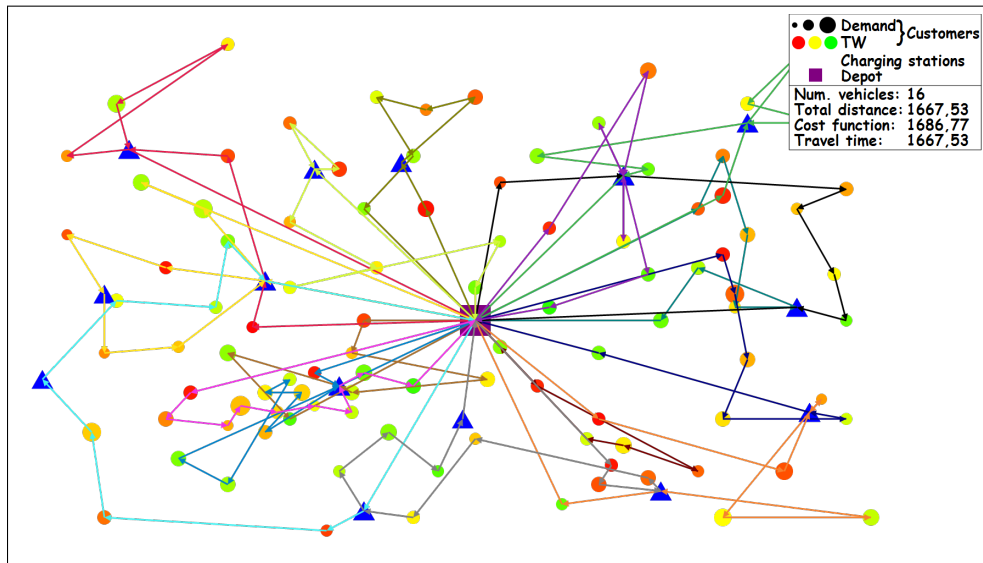


Figure 2.7: R101: EVRPTWDCS-PR, BKS

2.6 Time-dependent vehicle routing problem

Most of the researchers in VRP and EVRP field consider static traffic conditions on the road network. But, the traffic conditions change recurrently, depending on the time of the day, day of the week, and season; or non-recurrently when a traffic incident occurs, such as an accident [71, 72, 73]. Neglecting congestion and time-varying speeds in logistic operations can lead to an increased value of travel time and vehicle operating costs consisting of increased wages, higher prices, and penalties for late deliveries. Kellner et al. [74] reported that regular traffic congestion increases total logistic transportation costs by 3% and travel time by 5.5%. Integrating congestion in logistic operations led to the definition of the Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) [25, 75, 76].

The TD-VRPTW problem can be formulated as a Mixed Integer Program (MIP) [77], in a similar way as MILP for the EVRPTW-FR presented in section 2.2. The MIP formulation instead of the MILP formulation is used as the problem considers travel times that are nonlinear. The variables used in the formulation are presented in Table 2.1. Here, the MIP formulation of Figliozzi [25] is presented, although a stronger MIP formulation can be achieved. The graph is defined without the set of CSs, $G = (V_{0,N+1}, A)$, where A is the set of arcs $A = \{(i, j) | i, j \in V_{0,N+1}, i \neq j\}$. The set of available vehicles located at depot is denoted as K . Instead of a constant arc travel time value, each arc has a travel time $t_{ij}(\delta_i)$ expressed as a function of the departure time δ_i from customer i . The cost per unit of route duration is denoted as c_t ,

while the cost per unit distance traveled is denoted as c_d . The x_{ij}^k (equation 2.45) is a binary decision variable that indicates whether vehicle k travels between customers i and j . The τ_i^k variable indicates the begin time at customer i , which is served by vehicle k . The primary objective function of the TD-VRPTW is the minimization of vehicle number (2.46), and the secondary objective is the minimization of total time costs and total traveled distance costs (2.47). Equation 2.48 ensures that vehicle load capacity is not violated. Equations 2.49-2.53 ensure arc connectivity, while equations 2.54 and 2.55 ensure that each customer is visited within its time window. Equation 2.56 ensures arc travel time feasibility, where arc travel time depends on the departure time.

$$x_{ij}^k \in \{0, 1\}, \forall i \in V_0, \forall j \in V_{N+1}, i \neq j, \forall k \in K \quad (2.45)$$

$$\min \sum_{k \in K} \sum_{j \in V} x_{0j}^k \quad (2.46)$$

$$\min c_d \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_{N+1}, i \neq j} d_{ij} x_{ij}^k + c_t \sum_{k \in K} \sum_{j \in V} (\tau_{N+1}^k - \tau_0^k) x_{0j}^k \quad (2.47)$$

$$\sum_{i \in V} q_i \sum_{j \in V_{0,N+1}, i \neq j} x_{ij}^k \leq C, \forall k \in K \quad (2.48)$$

$$\sum_{k \in K} \sum_{j \in V_{0,N+1}, i \neq j} x_{ij}^k = 1, \forall i \in V \quad (2.49)$$

$$\sum_{j \in V_{0,N+1}, i \neq j} x_{ji}^k - \sum_{j \in V_{0,N+1}, i \neq j} x_{ij}^k = 0, \forall i \in V, \forall k \in K \quad (2.50)$$

$$x_{i0}^k = 0, x_{N+1i} = 0, \forall i \in V_{0,N+1}, \forall k \in K \quad (2.51)$$

$$\sum_{j \in V_{0,N+1}} x_{0j}^k = 1, \forall k \in K \quad (2.52)$$

$$\sum_{j \in V_{0,N+1}} x_{j,N+1}^k = 1, \forall k \in K \quad (2.53)$$

$$e_i \sum_{j \in V_{0,N+1}, i \neq j} x_{ij}^k \leq \tau_i^k, \forall i \in V_{0,N+1}, \forall k \in K \quad (2.54)$$

$$l_i \sum_{j \in V_{0,N+1}, i \neq j} x_{ij}^k \geq \tau_i^k, \forall i \in V_{0,N+1}, \forall k \in K \quad (2.55)$$

$$x_{ij}^k(\tau_i^k + s_i + t_{ij}(\tau_i^k + s_i)) \leq \tau_j^k, \forall i \in V_0, \forall j \in V_{N+1}, i \neq j, \forall k \in K \quad (2.56)$$

As presented in [25, 26], sometimes, the better secondary objective is to minimize total traveled distance 2.57, total traveled time 2.58 or total time 2.59.

$$\min \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_{N+1}, i \neq j} d_{ij} x_{ij}^k \quad (2.57)$$

$$\min \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_{N+1}, i \neq j} t_{ij} x_{ij}^k \quad (2.58)$$

$$\min \sum_{k \in K} \sum_{j \in V} (\tau_{N+1}^k - \tau_0^k) x_{0j}^k \quad (2.59)$$

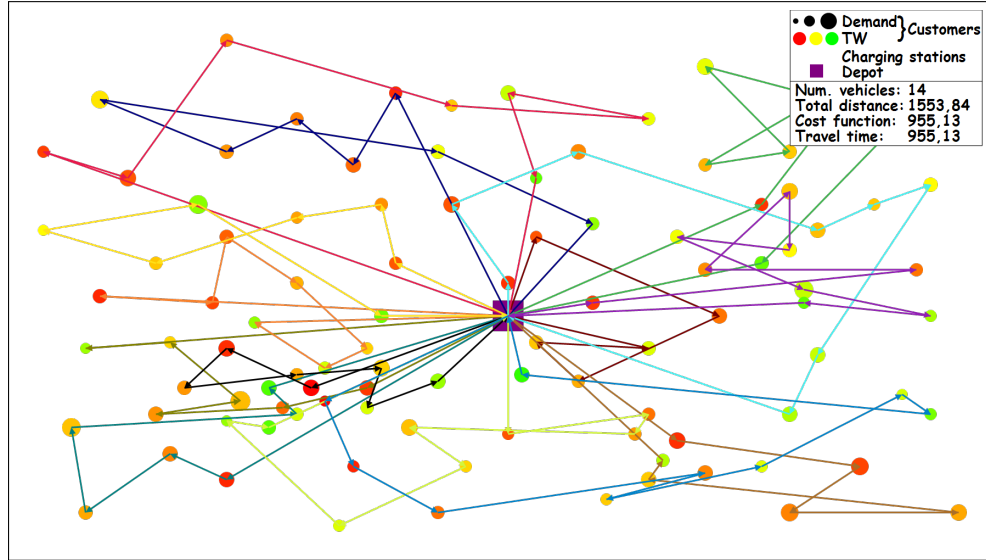
Figliozzi [25] proposed test instances for the TD-VRPTW problem. The TD-VRPTW test instances are based on the Solomon instances for the VRPTW [31], but include different travel speeds per discretized time intervals. As Solomon instances have narrow time windows, a small decrease in travel speed could result in infeasible solution. Thus, higher travel speeds were introduced, as speed values in the original Solomon problems were set to one. The ratio in travel speeds is set in $[1, 2.5]$ range. The depot working time $[e_{an0}, l_0]$ is discretized into five time buckets of equal duration. The time buckets and all 12 sets of travel speeds are presented in Table 2.3. The travel speed sets are divided into four groups:

- (A) - higher travel speeds are found in the rush hour periods,
- (B) - higher travel speeds are found at the extremes of the working day,
- (C) - higher travel speeds are found at the beginning of the working day,
- (D) - higher travel speeds are found at the end of the working day.

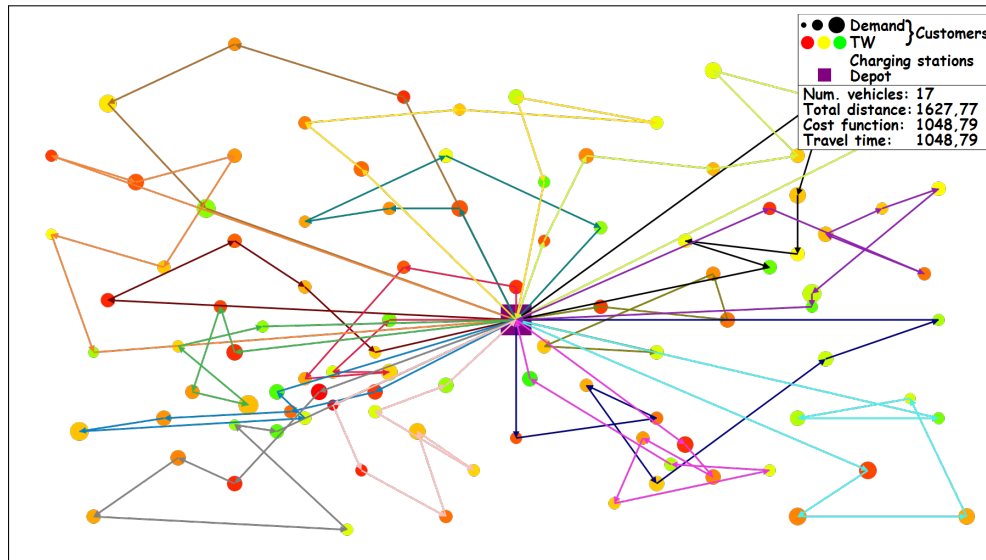
Table 2.3: Figliozzi speeds in time buckets

Set	Time buckets				
	$[0, 0.2l_0)$	$[0.2l_0, 0.4l_0)$	$[0.4l_0, 0.6l_0)$	$[0.6l_0, 0.8l_0)$	$[0.8l_0, l_0]$
A1	1.00	1.60	1.05	1.60	1.00
A2	1.00	2.00	1.50	2.00	1.00
A3	1.00	2.50	1.75	2.50	1.00
B1	1.60	1.00	1.05	1.00	1.60
B2	2.00	1.00	1.50	1.00	2.00
B3	2.50	1.00	1.75	1.00	2.50
C1	1.60	1.60	1.05	1.00	1.00
C2	2.00	2.00	1.50	1.00	1.00
C3	2.50	2.50	1.75	1.00	1.00
D1	1.00	1.00	1.05	1.60	1.60
D2	1.00	1.00	1.50	2.00	2.00
D3	1.00	1.00	1.75	2.50	2.50

The example of BKSs for TD-VRPTW on the instance R101 for two different sets A3 and B3 are presented in Figure 2.8. As set A3 has higher travel speeds during the middle of the day, total travel time is lower compared to the total travel time of set B3. The same goes for the total time value and total traveled distance. Additionally, it can be seen that lower speeds significantly increase the total vehicle number in the solution.



(a) A3



(b) B3

Figure 2.8: R101: TD-VRPTW, BKSs

Only one research considered time-dependent travel time in EVRP field and named the problem Electric Vehicle Routing Problem with Charging Time and Variable Travel Time (EVRP-CTVTT) [78]. The problem considers fixed recharging time to charge the vehicle to maximum, which is a significant simplification of the model. The proposed research lacks mathematical formulation of the problem, as well as an efficient method to solve the problem.

2.7 Hybrid and mixed fleet electric vehicle routing problem

As a compensation for the limited driving range of BEVs, HEVs, which have both an internal combustion engine and an electric engine, have been developed. Two main types of HEVs are present on the market: (i) the series hybrid, in which only the electric motor drives the vehicle, and the internal combustion engine is used to recharge the batteries, and (ii) the parallel hybrid, which uses both internal combustion engine and electric engine to drive the vehicle, where the electric engine is more efficient in stop-and-go activities, and the internal combustion engine is more efficient at high speeds. The HEVs have an option to decide during the route to either run on the electric energy or the fossil oil. This enables the service of customers far from the depot with no need for a refuel. As HEVs have two engines, their load is heavier than BEVs and ICEVs, and therefore they have a higher energy consumption rate. However, the time spent on the delivery is shorter, which makes it easier to achieve time-precise deliveries and to reduce recharging costs at the expense of higher traveling costs due to the fossil oil consumption.

The main problem can be defined as Hybrid Vehicle Routing Problem (HVRP) [79]. The objective of the proposed problem is to minimize the routing costs on the internal combustion engine while satisfying the demand and time window constraints. Usually, the following four working modes can be utilized when routing HEVs: combustion-only, electric-only, charging mode, and the boost mode when the combined internal combustion engine and electric engine are used. Several variants of the HVRP problem depending on the HEVs vehicle types and objective function have been proposed [54, 57, 79, 80, 81].

In today's vehicle fleets, mostly conventional ICEVs are present. Transition to an exclusively electric fleet is a very challenging economic task. Therefore, most companies are gradually integrating BEVs into their existing ICEV fleet. This led to the definition of Electric Vehicle Routing Problem with Time Windows and Mixed Fleet (EVRPTWMF) [20, 29, 54, 59, 82]. The problem considers different vehicle types (BEVs, ICEVs, and PHEVs), with each vehicle type having either equal or different load and battery capacities. A comprehensive case study regarding different vehicle types was conducted by Lebeau et al. [82], where the authors defined seven groups of vehicle types that could be used for the delivery, from small vans and quadricycles, through diesel- and electric-only groups, to a group of all vehicle types. Results showed the following aspects: (i) the fleet with different vehicle types reduced the total routing costs the most; (ii) in the large van group, ICEVs outperformed BEVs; and (iii) HEVs showed a great application in deliveries solely made by trucks. Solving the EVRPTW instances with different vehicle types indicated that BEVs are preferred for clustered instances, ICEVs for randomly distributed instances, and PHEVs for randomly clustered instances [54].

2.8 Electric vehicle routing problem with nonlinear charging functions

In most of the EVRP related literature, either linear or constant charging time is considered. Most of the BEVs have lithium-ion batteries installed, which are often charged in constant-current constant-voltage (CC-CV) phases: first by constant current until approx. 80% of the SoC value and then by a constant voltage. In the CC phase, SoC increases linearly, and in the CV phase, the current drops exponentially, and SoC increases nonlinearly, which prolongs charging time [46, 49]. This led to the definition of the Electric Vehicle Routing Problem with NonLinear charging functions (EVRP-NL) [83]. The nonlinear charging time in EVRP-NL is either linearized per segments or estimated by data-driven approaches [67, 69, 83, 84, 85]. Montoya et al. [83] point out that neglecting the nonlinear charging process can lead to infeasible or overly expensive solutions: 12% of the routes in good EVRP-NL solutions recharged the battery in the nonlinear part, after 80% of the SoC value.

2.9 Electric location routing problem and battery swap stations

Due to the currently low BEV market share, the number of CSs installed in the road infrastructure is also relatively low. Therefore, great potential lies in the simultaneous decision-making of CS locations and BEV routes. Classic Location Routing Problem (LRP) consists of determining the locations of the depots and vehicle routes supplying customers from these depots [86]. A modification of the LRP that deals with CSs is formulated as Electric Location Routing Problem (ELRP) [9, 35, 53, 87]. Several conducted case studies indicated the viability of combining CS sitting and BEV routing for specific cases when a delivery range is not far from the depot [9, 35]. It is also important to note that if CS can be located at a customer location, the service time can be used for recharging. The comparison of ELRP and EVRP solutions on EVRPTW instances [87], indicated that ELRP produced an equal or better solution on all instances, as charging while serving and no need to visit separately located CS, reduce the overall routing costs. If routing is not considered, then the problem considers only the determination of CS locations [88].

Instead of charging at CS, at specially designed BSS, an empty or nearly empty battery can be replaced with a fully charged one [12]. The main advantages of such procedure are the fast swapping time and reduced costs from charging the batteries during the lower energy network load. A whole replacement procedure could last less than ten minutes, which is competitive to the refueling time of ICEVs, and much faster than one of the fastest BEV charging technologies.

The drawbacks of such procedure are the non-standardized batteries and their complex installation in BEVs. In EVRP with BSSs the locations of BSS are usually determined in advance [89], but they have can also be determined simultaneously with BEV routing decisions [90, 91].

2.10 Other variants of the electric vehicle routing problem

Beside the previously described EVRP variants, which are in the main focus of the research field of this thesis, there are also many other variants that try to model real-world delivery constraints.

The Green Vehicle Routing Problem (GVRP) [38, 43, 92, 93] focuses on the reduction of routing pollution on the environment by the use of alternative fuel vehicles powered by bio-diesel, ethanol, hydrogen, methanol, natural gas or electricity. Vehicles refuel at separately located stations, with fixed refueling time. The main idea is to promote the use of sustainable energy sources and minimize overall vehicle emissions. As BEVs have no local CO₂ emission, the EVRP is closely related to the minimization of GHG emission, where a problem-specific GVRP variant called the Pollution Routing Problem (PRP) emerged [37, 94]. The main objective in PRP problem is the optimization of vehicle speeds and minimization of GHG emissions.

In Electric Two-Echelon Vehicle Routing Problem (E2EVRP) [95, 96], goods are transported in two echelons. In the first echelon, goods are transported by conventional freight vehicles from the depot to the satellite facilities. In the second echelon, goods are transported from the satellite facilities to the customers by light BEVs. Thus, two vehicle types are observed in the problem: ICEVs with higher load capacity located at the depot and BEVs with lower load capacity located at the satellite facilities. BEVs are used for last-mile delivery due to the lower pollution, noise reduction, and smaller size.

Many companies that use BEVs prefer charging the vehicles in their own facilities to charge the vehicles between the delivery routes or during the specific periods in day. In such occasions, there is usually a limited number of chargers at the depot, typically fewer than the fleet size, and charging schedule has to be determined. Such problem is called the Electric Freight Vehicle Charge Scheduling Problem (EFV-CSP) [39, 67]. It includes multiple charging technologies, realistic charging process (piecewise linearization of nonlinear charging function), time-dependent charging costs, grid power restriction, battery degradation costs (cyclic and calendar aging), and facility-related demand charges representing the maximal demand registered over the billing period. A comprehensive case study was performed by Pelletier et al. [67] with the following conclusions: (i) model tries to keep the SoC lower when battery degradation costs are included; (ii) in summer, vehicles are rarely charged in peak hours, which results in more vehicles charging simultaneously or the use of fast chargers that retrieve more power from the grid in non-peak hours but incur higher facility-related demand charges; (iii) to avoid cycling in high SoC values, it is preferable to split the long routes into smaller ones; (iv) fast chargers

are heavily used in a high BEV utilization scenarios; (v) grid power restriction increases overall energy costs, especially in summer months, and leads to infeasible solutions - limiting the number of vehicles simultaneously charging; and (vi) total costs are always lower with larger batteries, as smaller batteries have frequent large discharge cycles.

In a one-way electric car-sharing problem [97], the BEVs are shared between multiple users. At the beginning of the service, BEVs are relocated to the user entry point. The relocation is performed by a worker who comes with a bicycle to the pick-up point, puts the bicycle into the BEV's trunk, and drives the BEV to the user entry point. A Dial-a-Ride Problem with EVs and BSSs (DARP-EV) [98] considers routing of BEVs for customers with special needs and disabilities. The BEVs have special characteristics such as handicapped person's seat, stretcher, wheelchair or accompanying person's seat. In Multi-Depot Electric Vehicle Location Routing Problem with Time Windows (MDEVLRPTW) [99], customers can be served from multiple depots, where each depot has its own vehicles. The Robust Electric Location Routing Problem (RELRP) [100] considers uncertain customer patterns related to the customers' spatial distribution, demands, and service time windows.

A natural extension of the EVRPTW problem is to consider waiting times at the CSs [61, 68], with three key aspects regarding the queuing at CS: (i) the vehicle arrival distribution, (ii) the service time distribution, and (iii) service strategy, i.e., First-In-First-Out (FIFO). The availability of CSs in terms of public or private property is another aspect that can be considered within the EVRPTW problem [68].

Problem solving methods

Since VRP is a well-researched problem, many methods for solving the problem have been proposed. The general classification of optimization methods used to solve the VRP problem is presented in Figure 3.1. Due to the NP-hardness of the problem and a large number of customers in real-world problems, most of the methods used in real-world applications are heuristics, metaheuristics, and hybrid combinations. Many of the VRP solving methods in the available literature are with modifications applicable on the EVRP problem. Exact procedures are able to find optimal solutions on instances with a lower number of customers: up to 360 customers for CVRP problem and up to 100 customers for VRPTW problem [15, 16]. They can be split into enumerative and calculus procedures. Some of the most used exact algorithms in the EVRP research field are: branch & price [29], branch & cut [101], branch & bound [69], branch & cut & price [55], and dynamic programming [9, 35, 53, 54]. Additionally, many researchers formulate the problem as MILP program and solve the small instances with commercially available softwares, such as [MATLAB](#), [IBM CPLEX](#), [GUROBI](#), etc. [18, 22, 102]. Lately, in EVRP problems, exact procedures are applied to determine the optimal CS placement and charging schedule [22, 53].

Heuristic methods seek to solve the problem based on the specific knowledge of the problem. They are much faster than the exact methods but usually produce suboptimal solutions or solutions close to a satisfactory solution. Heuristic methods can be split into constructive and improvement heuristics. Constructive heuristics are often used to generate an initial solution by serial or parallel route construction. Solutions are constructed in a greedy way, which often produces solutions to the VRP that are 10-15% far from the optimal solution [3]. In EVRP, constructive heuristics are modified and adapted to BEV characteristics and feasibility checks. Improvement heuristics, most often referred as Local Search (LS) procedures, explore the neighborhood of the current solution, searching for a better solution. The neighborhood is explored by applying perturbation moves based on the composite neighborhood operators. The

local search stops when no improving solution can be found in the neighborhood of the current solution, which is then called the local optima.

Many researchers employ metaheuristics to continue the exploration after the local optima occurrence. Metaheuristics can be defined as heuristics guiding other heuristics. They can be divided into neighborhood-oriented metaheuristics and population based metaheuristics. Neighborhood-oriented heuristics iteratively explore the neighborhood of the current solution, while the population metaheuristics use natural selection to evolve a population and select the fittest individual as the best solution. In the next section, the review of the most used heuristic and metaheuristic procedures for solving the EVRP problem is presented. For a recent survey on the various methods used to solve the EVRP problem, the reader is referred to [19, 21, 103].

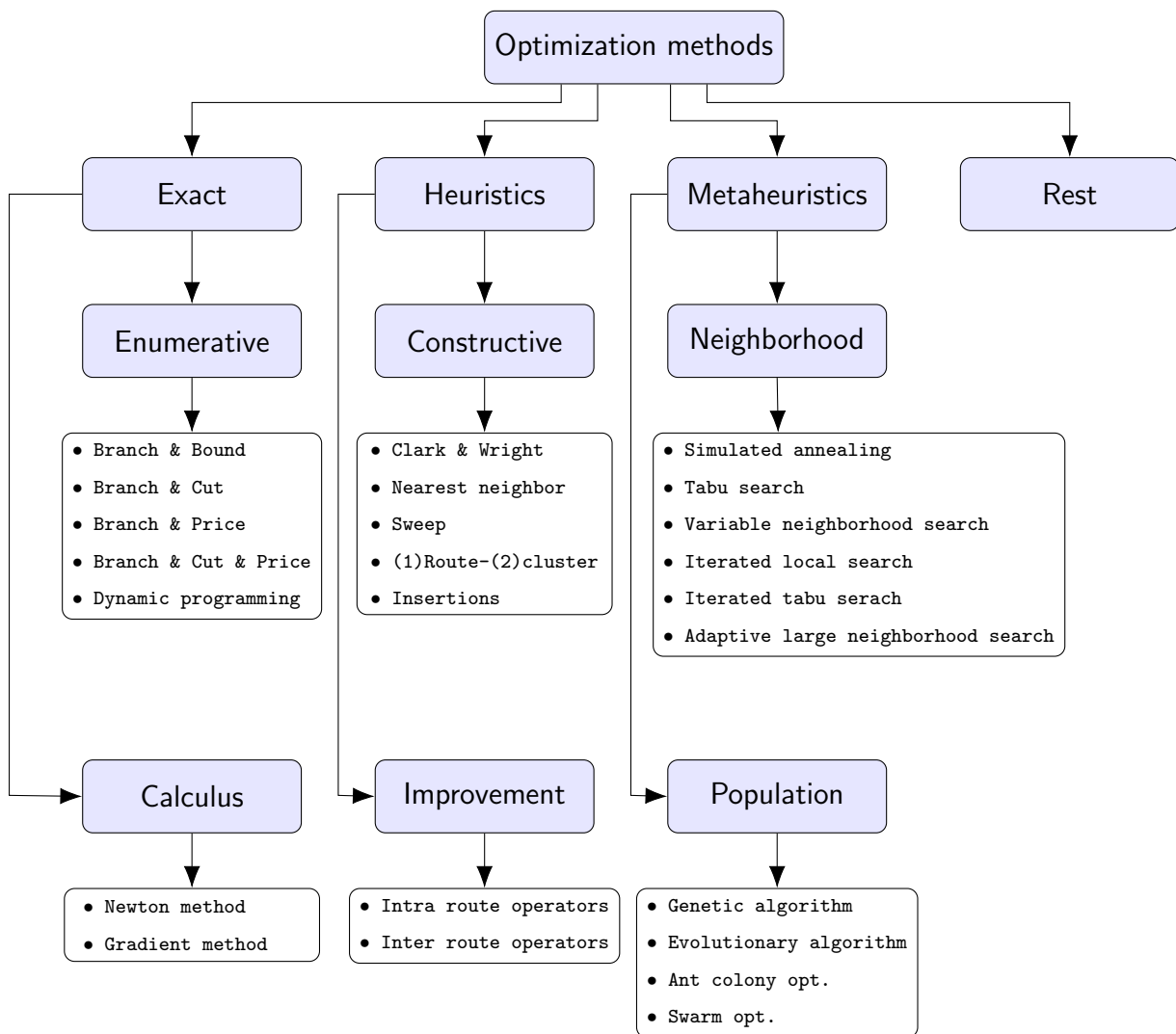


Figure 3.1: Optimization methods

3.1 Initial solution

Constructive heuristics build a solution step by step, meanwhile considering overall routing costs. Vehicle routes are usually constructed in a serial or parallel way [3]. The serial strategy considers constructing one route from start to the end and then going to another route, while in parallel strategy, multiple routes are built simultaneously. Here, several most-applied VRP methods for the initial route creation, adapted on the EVRP problem, are presented.

3.1.1 Sweep algorithm

The sweep algorithm [104] inserts customers in the active route in a circular manner, resulting in an efficient space division. Customers are sorted based on the value of the polar angle between the depot and the randomly chosen point. Customers are then iteratively added to the active route until a constraint is violated. If violation occurs, then a new route is opened, and the procedure is repeated until all customers are served. The basic idea of how the sweep algorithm works is best visible in Figure 3.2b for the CVRP problem, where customers are added in routes in a circular manner. When considering time windows, user u can be inserted between users i and j only if $e_i \leq e_u \leq e_j$. This does not ensure the feasibility of the solution but leads to somewhat better solutions. If some customers could not be inserted in either route (usually far from the depot), then these customers will be added to the last vehicle route. The sweep algorithm applied in the EVRP field usually assumes that there is no possibility to visit CS within route [18, 90, 96], as shown in Figure 3.2a, which means that the created solution is usually infeasible. The CSs are inserted later in the improvement part. As it can be seen in Figure 3.2 the number of vehicles and total traveled distance are significantly higher in the EVRPTW-FR problem than in the CVRP problem for the same customer configuration. Also, in EVRPTW-FR solution, one large route is present, containing all infeasible customers.

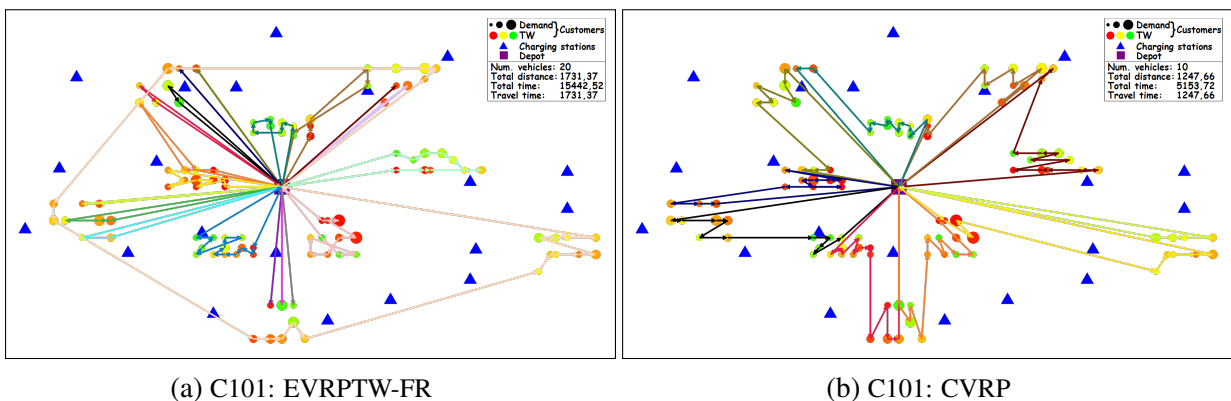


Figure 3.2: Examples of Sweep algorithm

3.1.2 Clark and Wright savings algorithm

The Clark and Wright Savings algorithm (CWS) [105] is one of the oldest algorithms applied on CVRP problems. The algorithm starts with the creation of back-and-forth routes for each customer. Then, the savings of each possible route merging is computed as long as the capacity of the vehicle is not violated as $s_{ij} = d_{i0} + d_{0j} - d_{ij}$. To merge the routes, the last arc in the first route and the first arc in the second route have to be deleted, and a new arc between the last customer in the first route and the first customer in the second route has to be inserted, as shown in Figure 3.3. The merging with the highest saving value is performed. The algorithm ends when no more savings can be achieved by route merging. The example of solutions for the CVRP and EVRPTW-FR problems on instance C101 are presented in Figure 3.4. In each back-and-forth vehicle routes, CSs are inserted to make the route energy feasible [37, 82, 91, 92], but often the solution is still infeasible.

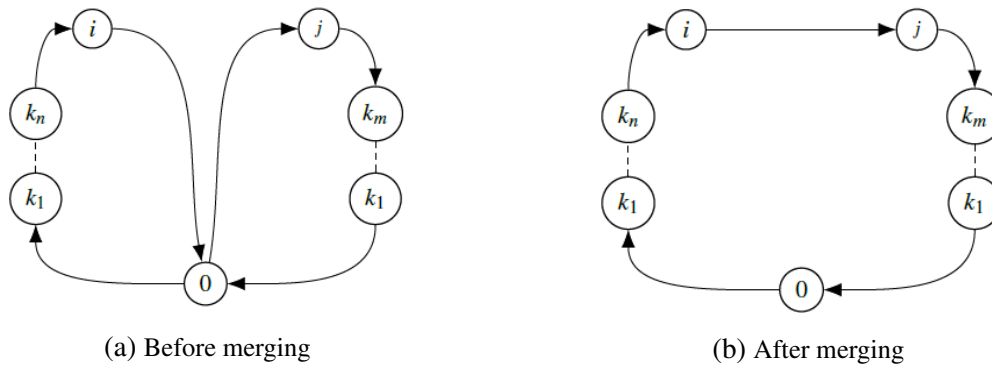


Figure 3.3: Clark & Wright savings algorithm

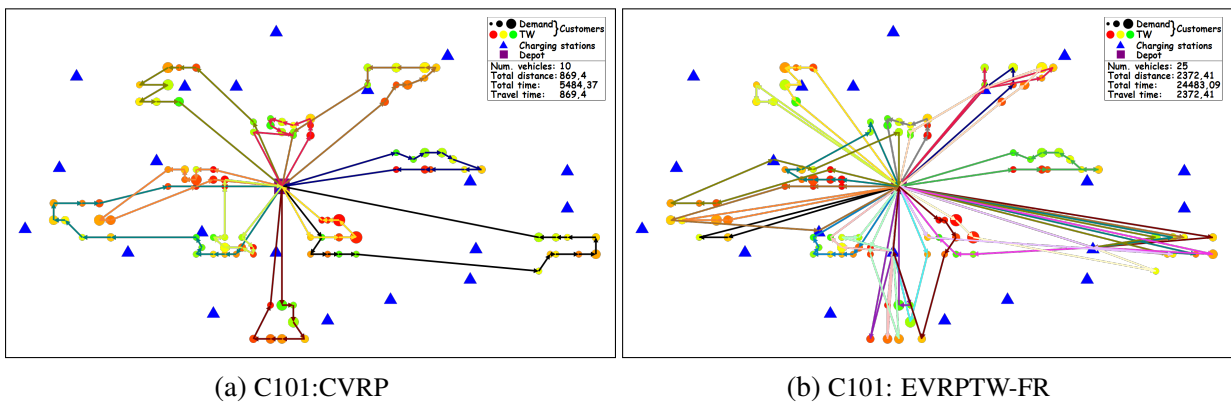


Figure 3.4: Examples of Clark & Wright savings algorithm

3.1.3 Nearest neighbor heuristic

The Nearest Neighbor Heuristic (NNH) is one of the most used greedy algorithms for the construction of an initial solution. The heuristic starts from the depot, and in each iteration, the feasible customer with the least cost increase from the previously inserted customer is added to the route. The route is terminated when any constraint is violated, and then a new route is opened. In NNH for EVRPTW-FR problem [23], the next feasible customer has to satisfy customer time-window, vehicle load capacity, and vehicle battery capacity, and additionally it has to have enough energy to return from customer to the depot on time. If energy is violated at the customer or at the depot, then if possible, the nearest CS between users is inserted immediately before to make the route energy feasible. Otherwise, a new route is opened, and the procedure is repeated until all customers are served. The example of an initial solution created with NNH is presented in Figure 3.5. As it can be seen, NNH algorithm produces a solution with a large number of vehicles. Compared to the Sweep and CWS algorithm, the solution with the NNH algorithm is often feasible, but still not in all occasions.

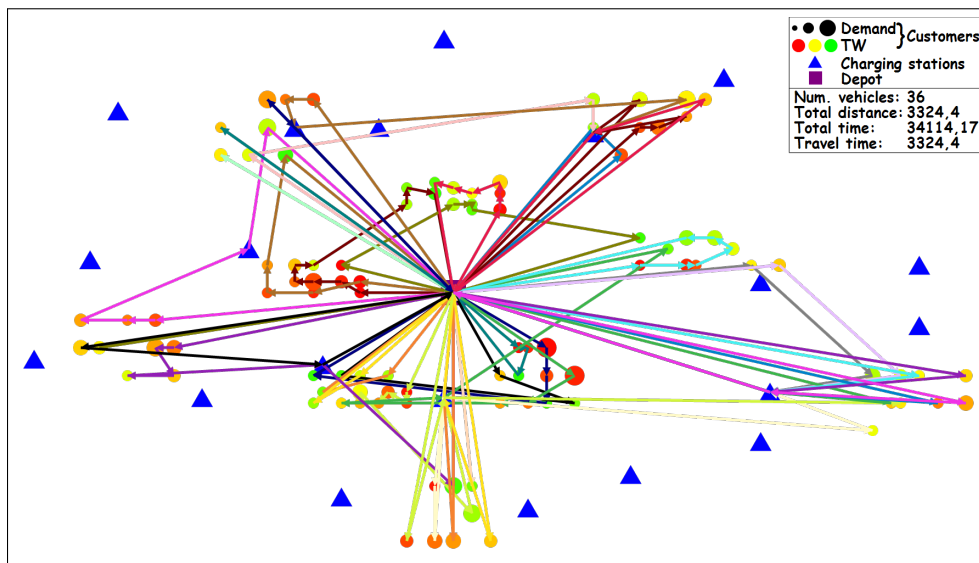


Figure 3.5: Example of nearest neighbor heuristic - C101: EVRPTW-FR

3.2 Local search

A great variety of researchers have performed LS procedures to intensify the search. They are often coupled with perturbation moves to escape the local optima. The used neighborhood operators try to find a change in the solution that will decrease the overall objective value. Often, the perturbation moves are similar to the neighborhood operators used in the LS phase. Most of the classical VRP neighborhood operators [3, 4] are used in the EVRP field, but still some additional problem-specific neighborhood operators have been developed. Depending on whether the operators perform on only one route or between the routes, they can be divided into intra route and inter route operators. Intra operators change the position of users within the route, while inter operators change the position of users between multiple routes. As LS neighborhood operators explore a large number of changes in the solution, the LS procedure tends to be time-consuming. Therefore, most of the operators applied in VRP have a constant time complexity $\mathcal{O}(1)$, but there are also the ones that are more complex, and therefore search a larger neighborhood solution space. For the list of most applied operators in the EVRP research field, the reader is referred to Erdelić et al. [21].

Often, there is a question of whether to make a first better or the best improvement move in the LS phase [106]. The first better strategy has a lower execution time but generally produces worse solutions than the best strategy. Some EVRP researchers perform one or the other, and some combine the two approaches, i.e., the best move of first 100 [9], or first 50 moves [29]. Usually, multiple neighborhood operators are applied to search the solution space. The strategy of using either the best improvement of all operators or the best improvement of each one significantly affects the overall execution time and the solution quality. It can be seen that the best ratio of solution quality and execution time in VRP is achieved when the best improvement value of each operator is used [26]. In such strategy, the order of operators is important. Common sense is to first use inter operators to reposition users between different routes and then to apply intra operators to improve each route.

3.2.1 Intra operators

As already mentioned, intra operators perform changes on a single route. Repositioning of users within a route affects arrival times at other users in the route. Therefore, in VRPTW and EVRPTW, the violation of time windows and battery capacity has to be checked. As changes are made on only one route, the load capacity violation does not need to be checked. The most common intra operators applied in both VRP and EVRP problems are presented in the following paragraphs. For operators initially applied on CVRP, the example of distance savings computation is presented.

Intra relocate operator

Intra relocate [107] operator aims to reposition one customer from its previous position in route to a new position. The example is presented in Figure 3.6 where customer b is repositioned between customers f and e . The depot is presented as a purple rectangle, while the customers are presented as black circles. The distance saving in CVRP problem can be computed by equation 3.1, with the $\mathcal{O}(1)$ complexity.

$$d_s = d_{ab} + d_{bc} + d_{ef} - (d_{ac} + d_{eb} + d_{bf}) \quad (3.1)$$

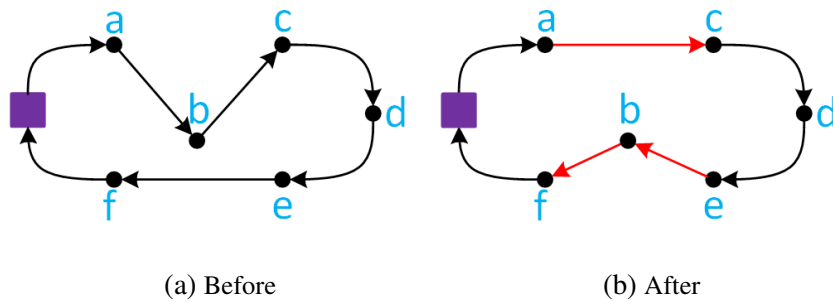


Figure 3.6: Intra relocate operator

Intra exchange operator

Intra exchange [107] operator aims to change the positions of two customers with each other. The example is presented in Figure 3.7 where customers b and f exchange their positions and are repositioned between customers e and g , and a and c , respectively. The distance saving in the CVRP problem can be computed by equation 3.2, with the $\mathcal{O}(1)$ complexity.

$$d_s = d_{ab} + d_{bc} + d_{ef} + d_{fg} - (d_{af} + d_{fc} + d_{eb} + d_{bg}) \quad (3.2)$$

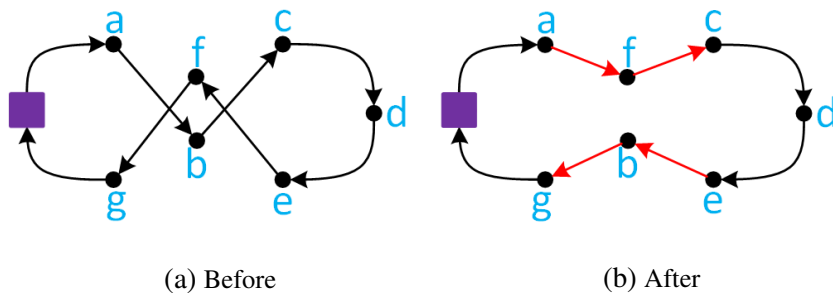


Figure 3.7: Intra exchange operator

Or-Opt operator

Or-Opt operator aims to delete three arcs in a route and replace it with three new ones [108]. The sequence of up to three customers is repositioned, meanwhile preserving the orientation of the subroutes. The Or-opt-1 changes the sequence of one customer and therefore is the same as intra relocate operator. The examples of Or-Opt-2 and Or-Opt-3 are presented in Figure 3.8. The distance saving in CVRP for Or-Opt-2 can be computed by equation 3.3, with the $\mathcal{O}(1)$ complexity.

$$d_s = d_{ab} + d_{cd} + d_{fg} - (d_{ad} + d_{fb} + d_{cg}) \quad (3.3)$$

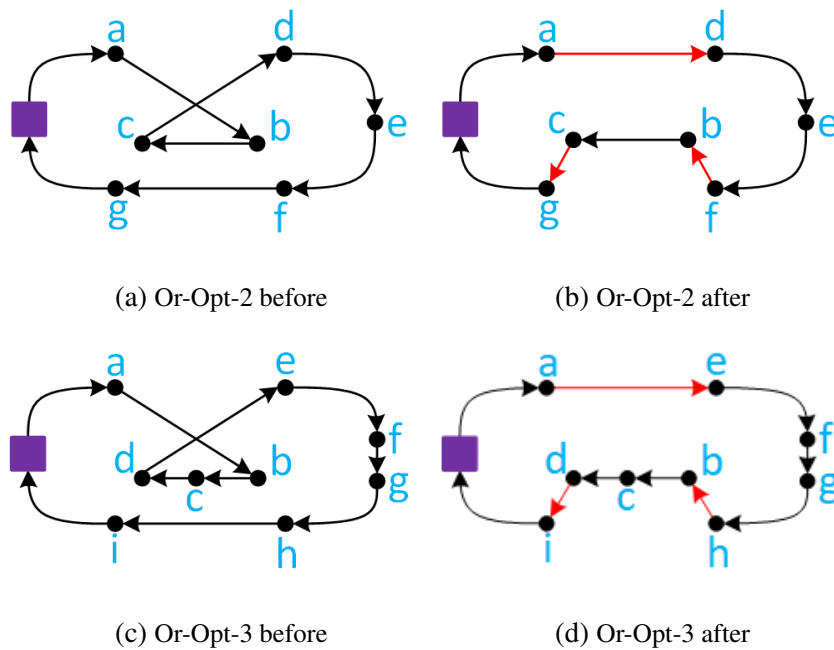


Figure 3.8: Or-Opt operator

2-Opt operator

2-Opt [109] operator replaces two arcs with two new ones, with a possibility of a sub-route reversal. The example is presented in Figure 3.9 where arcs ab and de are replaced by ad and be and the part of the route between customers b and d changed direction to $d \rightarrow c \rightarrow b$. The distance saving in the symmetric CVRP problem can be computed by equation 3.4, with the $\mathcal{O}(1)$ complexity. In comparison to the Or-opt-3 operator, it can be seen that for the same example, the 2-Opt produces a slightly worse solution. In asymmetric problems, the direction of the subroute has to be reversed, which increases the complexity of an operator to $\mathcal{O}(M)$,

where M is the number of customers in subroute.

$$d_s = d_{ab} + d_{de} - (d_{ad} + d_{be}) \quad (3.4)$$

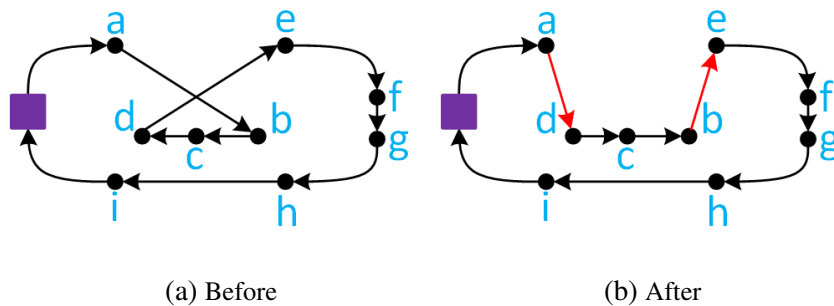


Figure 3.9: 2-Opt operator

Intra station in and out operators

Intra station in operator is specially designed for EVRP problems. The operator tries to insert CSs in a vehicle route. The intra station out does the opposite, as it tries to remove the CS from a vehicle route. These two operators can be merged together in one as StationInRe operator [18] in a way that if an arc to the CS is already in the solution, it is removed; otherwise, it is inserted. The example of Intra station in operator is presented in 3.10. This operator increases the overall distance, and it is mostly used to make the route energy feasible. The intra station out can be observed as inverse operation of intra station in operator. In the EVRP without time windows, the complexity of operator is $\mathcal{O}(1)$, while in the EVRPTW problem, the worst complexity is $\mathcal{O}(B)$, where B is the number of users from insertion position up to the latest CS in route.

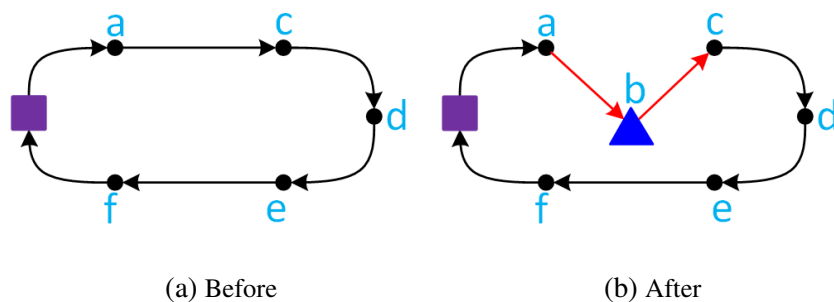


Figure 3.10: Intra station in operator

3.2.2 Inter operators

Inter operators perform changes between multiple routes. Most often, two routes are considered. In this case, the load capacities of vehicles affected by changes have to be checked, as well as time windows in VRPTW and EVRPTW problems. The most common inter operators applied in both VRP and EVRP problems are presented in the following paragraphs. For operators initially applied on CVRP, the example of distance saving computation is presented.

Inter relocate operator

Inter relocate operator [107] repositions a customer from its route to a new position in another route. The example is presented in Figure 3.11. The distance saving in CVRP can be computed by equation 3.5, with the $\mathcal{O}(1)$ complexity.

$$d_s = d_{ab} + d_{bc} + d_{ef} - (d_{ac} + d_{eb} + d_{bf}) \quad (3.5)$$

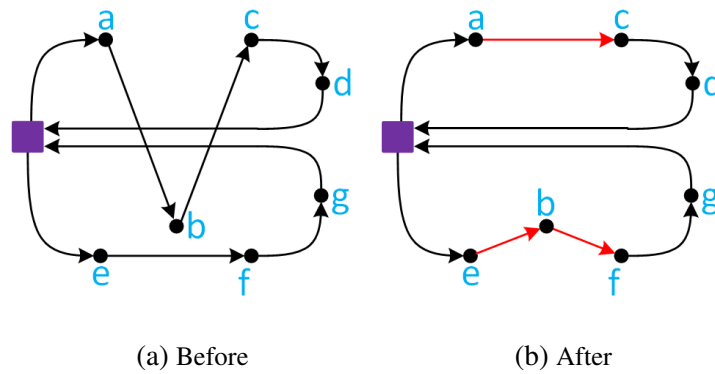


Figure 3.11: Inter relocate operator

Inter exchange operator

Inter exchange operator [107] swaps the positions of two customers in two different routes. The example is presented in Figure 3.12. The distance saving in CVRP can be computed by equation 3.6, while the complexity of the operator in such case is $\mathcal{O}(1)$.

$$d_s = d_{ab} + d_{bc} + d_{ef} + d_{fg} - (d_{af} + d_{fc} + d_{eb} + d_{bg}) \quad (3.6)$$

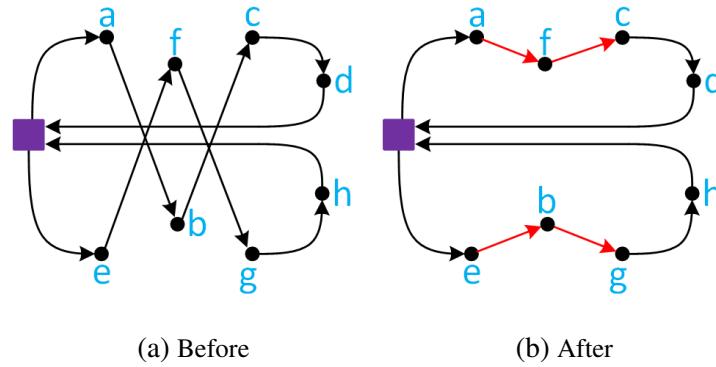


Figure 3.12: Inter exchange operator

Inter cross exchange operator

Inter cross exchange operator [110] swaps the sequences of up to k customers from one route with up to k customers from another route. The operator goes through all k^2 permutations of customer sequences swaps: 1-1, 1-2, \dots , 1- k , 2- k , \dots , k - k . The example of 3-2 swap is presented in Figure 3.13 while the distance saving in CVRP can be computed by equation 3.7. If the single combination cost evaluation can be determined in $\mathcal{O}(1)$ (as in the CVRP problem), then the complexity of operator is $\mathcal{O}(k^2)$.

$$d_s = d_{ab} + d_{cd} + d_{fg} + d_{ig} - (d_{ag} + d_{id} + d_{fb} + d_{cg}) \quad (3.7)$$

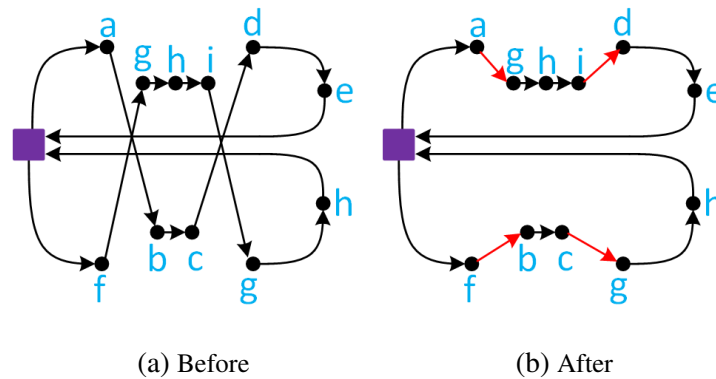


Figure 3.13: Inter cross exchange operator

Inter 2-Opt* operator

Inter 2-Opt* operator [111] swaps the partial route endings between two routes, meanwhile preserving routes' directions. The example is presented in Figure 3.14. The distance saving in CVRP can be computed by equation 3.8, with the $\mathcal{O}(1)$ complexity. As the operator avoids route reversal it is commonly applied on problems dealing with time windows.

$$d_s = d_{ag} + d_{ec} - (d_{ac} + d_{eg}) \quad (3.8)$$

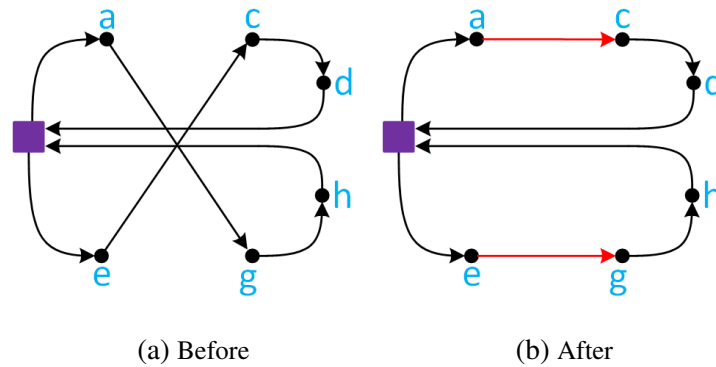


Figure 3.14: Inter 2-Opt* operator

3.3 Metaheuristics

Metaheuristics are used to explore a large search space, and therefore to escape the local optima. The escape from local optima is usually done by accepting solutions that are worse than the current solution. The most important part of any metaheuristic is the control of diversification and intensification, which affects both the solution quality and algorithm execution time. Diversification is used to search a larger solution space, while intensification is used to explore a particular solution space and find local optima. Over the years, a vast number of metaheuristics were developed for solving VRP problems [112, 113], with some of them mimicking a natural process such as evolutionary algorithm, ant-colony, particle swarm, simulated annealing, etc. Most often, metaheuristic and heuristic methods are combined together and adapted to solve the problem; therefore, the term hybrid is used in such occasions [114]. In this section, a brief overview of used metaheuristics for solving VRP and EVRP problems is presented, with a highlighting on the neighborhood-oriented metaheuristics.

3.3.1 Population based metaheuristics

Population metaheuristics are based on the natural selection to evolve a population and the principle of the survival of the fittest. They have been widely applied in the VRP field: genetic algorithm [115], scatter search [116], ant colony [117], bee colony [118], particle swarm optimization [119, 120], etc. Compared to the neighborhood-oriented metaheuristics, their application on EVRP problems is still scarce [21].

Genetic algorithm

Genetic Algorithm (GA) is the most applied metaheuristic used to solve various variants of the VRP problem [3, 17, 121, 122, 123]. The GA consists of a pool of individuals (VRP solutions) called a population, which goes through the process of evolution. Evolution is a robust process in which individuals in the population adapt to the conditions in nature - the so-called Darwin theory in which individuals better adapted to the environment have a higher chance of survival, and therefore with reproduction, transferring such good genes to their offsprings [124]. The GA has several important components presented in Figure 3.15: (i) solution representation, (ii) fitness function, (iii) selection, (iv) crossover, (v) mutation, and (vi) replacement of the old population with a new one. The solution to the VRP problem is usually represented as follows [125]: customer ID represents a gene, chromosome represents a vehicle - a sequence of customers between depots (depots have a zero ID), and individual - whole sequence of chromosomes representing a solution to the VRP problem. Each individual is evaluated through a fitness function which in VRP represents the objective function. Further on, to generate the offspring, two individuals (parents) are selected from the population for the crossover. Most often the roulette wheel selection or tournament selection is applied in such occasions [126]. The crossover procedure exchanges genetic material between the individuals and produces either one or multiple offsprings. A large number of crossover operators have been applied in the VRP field, and for detailed description, the reader is referred to Gendreau et al. [127]. The next step, mutation, is an important part of every evolutionary algorithm as it represents occasional and random changes in the individual's genes. Although, in most cases, mutation decreases the fitness value, it contributes to the diversification of genetic material in the population. For mutation operators, mostly some basic customer permutations are used, similar to the basic LS operators: relocate, exchange or route inversion [128]. In the EVRP research field, mostly non-efficient versions of GA are applied [39, 78, 129, 130, 131], which have the problem of determining a good CS placement and schedule. Only the hybrid GA proposed by Hiermann et al. [54] produced high-quality solutions and some of the BKSs on the EVRPTW-FR and EVRPTW-PR variants.

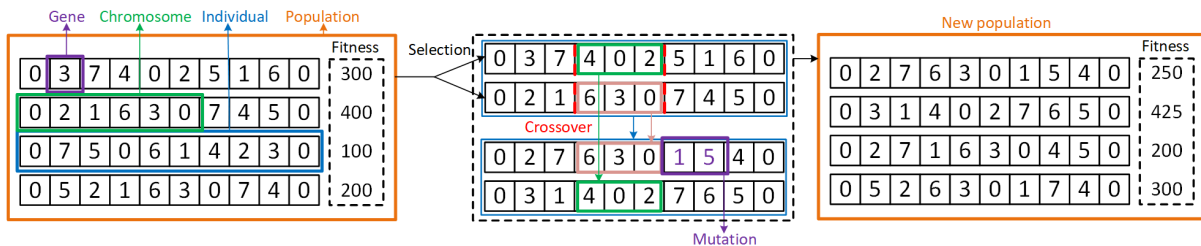


Figure 3.15: Genetic algorithm

Ant colony

The Ant Colony (AC) algorithm is based on the principle of indirect communication between individuals of the same species [117, 127]. The ants use indirect communication as they do not have highly developed eyesight and cannot see each other directly. This enables them to achieve complex self-organizing structures, although they do not have the ability to plan or memorize. As ants search for food, they leave pheromones (clues), which attract other ants. Over a time period, the pheromones accumulate over the most-used path between the nest and the food location. The example of finding the shortest path between the nest and food is presented in Figure 3.16. AC algorithm consists of three components: (i) pheromone initialization, (ii) pheromone evaporation, and (iii) pheromone update. For a detailed description of AC application in the VRP field, the reader is referred to [127, 132]. In the EVRP field, the AC algorithm has been rarely applied [131, 133].

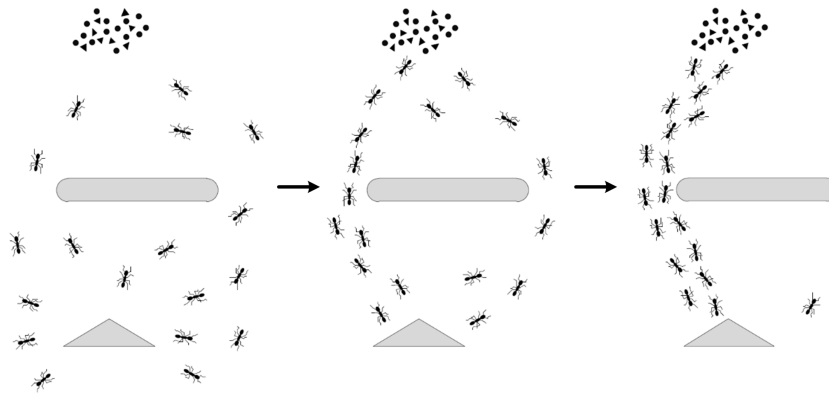


Figure 3.16: Ant colony example [132]

Particle swarm optimization

Particle Swarm Optimization (PSO) mimics the behavior of organisms such as fish schooling or bird flocking [119, 120]. In comparison to ants, fish and birds have highly developed eyesight and know not only the nearest individual but also where the socially nearest individual is in a flock. In PSO, each individual (solution) is represented as a particle and has a position and corresponding velocity in the swarm. These values are used to optimize the problem following the personal best solution and global best solution. The particle in the swarm does not know the exact location of the best solution, but it trusts its neighbor particles if they are close to the best solution. Further on, the well-balanced moves are applied to adapt the individual to the global and local exploration. The example of the solution space in PSO is presented in Figure 3.17, where particles are represented with blue circles, high fitness values are represented with red color, and low fitness values with dark blue color. In the EVRP field, the PSO is only applied in one research paper for fuel minimization [43].

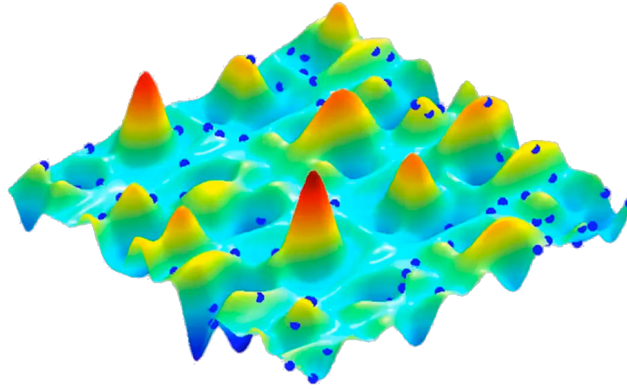


Figure 3.17: Particle swarm optimization solution space [134]

3.3.2 Simulated annealing

Simulated Annealing (SA) is one of the oldest metaheuristics used as an optimization technique. It mimics the thermodynamic cooling process of the material [127, 135, 136]. With annealing, the goal is to achieve the state of minimal internal energy of the material, which is achieved by atoms forming a regular crystal structure. The idea is to heat the metal and then to gradually cool it. At high temperatures, the atoms in the material move fast, and the internal structure of the material is stochastically organized. With fast cooling, the atoms form structures that represent local energy optima. With slow cooling (annealing), atoms have enough time to form stable crystal structures, which represents global optima.

In VRP problems, the diversification phase is performed when temperatures are high, while intensification is performed at low temperatures. The diversification considers accepting worse solutions and moving to the other solution space. The probability of accepting solution s_2 , which is worse than the current best solution s_1 , is computed by equation 3.9. This is the so-called Boltzmann function [138], where T is the temperature parameter and $f(s)$ is objective function value of solution s . The example of the SA cooling process is presented in Figure 3.18. The higher the temperature value is, the higher is the probability of accepting a worse solution.

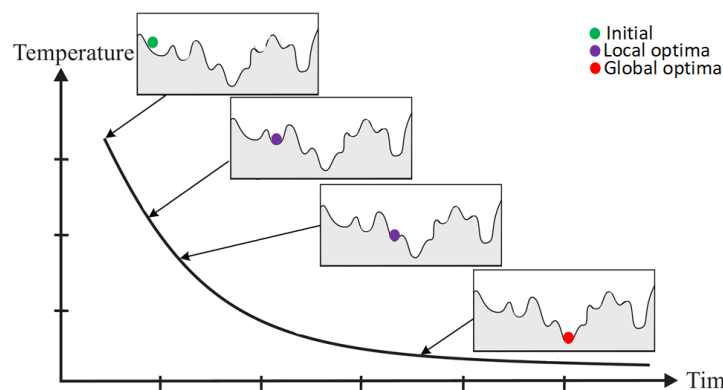


Figure 3.18: Simulated annealing cooling process [137]

At lower temperatures, only high-quality solutions have a chance to be accepted. At the end of each iteration k , the temperature T_k is decreased by coefficient α (equation 3.10, $\alpha < 1$), representing the temperature cooling rate coefficient. If observed over the whole execution period, the temperature is decreased following an exponential function. Another, important parameter is the initial temperature value T_0 which has to be high enough to accept worse solutions. Most often initial temperature T_0 is determined such that a $\mu\%$ worse solution (usually 30-40%) than the initial solution is accepted with a probability of 0.5, while the α value is set very close to 1 ($\alpha = 0.9994$) [1, 18, 20, 23]. SA is most often used as a control metaheuristic for accepting solutions produced by other heuristic and metaheuristic procedures.

$$p = e^{-\frac{f(s_1) - f(s_2)}{T}} \quad (3.9)$$

$$T_k = \alpha \cdot T_{k-1} \quad (3.10)$$

3.3.3 Tabu search

Tabu Search (TS) is a well-known metaheuristic used to solve VRP problems. It uses memory structures to forbid the exploration of the solution space that has already been explored [139, 140]. It escapes the local optima by accepting non-improving move if it is the best in the explored space. Arcs that are deleted from the solution are stored in the tabu-list, which prohibits the reinsertion of the deleted arcs into a specific part of the solution for a determined number of iterations - the tabu-tenure. A basic TS can be seen as a combination of LS operators with short-term memories. Sometimes the tabu-list is too powerful, as it may prohibit the selection of a good move, even when there is no danger of reusing the move or the search process stagnation. Therefore, in some occasions, it is advisable to remove a particular arc from the tabu-list - the so-called aspire criteria. Most used aspire criteria is the achievement of a new best solution [127]. In the EVRP field, TS with neighborhood operators is used to intensify the search process [18, 40, 65].

3.3.4 Variable neighborhood search

Variable Neighborhood Search (VNS) is a metaheuristic that systematically changes the neighborhood each time when there is no improvement in the LS phase [141]. The changes are based on the predefined neighborhood structures, which are similar to the ones used in the LS phase. The basic VNS framework is presented in Figure 3.19. In each step, the new neighborhood operator is used $N_1 \leq N_2 \leq \dots \leq N_k$, which increases the vicinity from the current solution. After each neighborhood perturbation, the LS procedure is called to find the local optima. The selection of neighborhood operators can be deterministic or probabilistic.

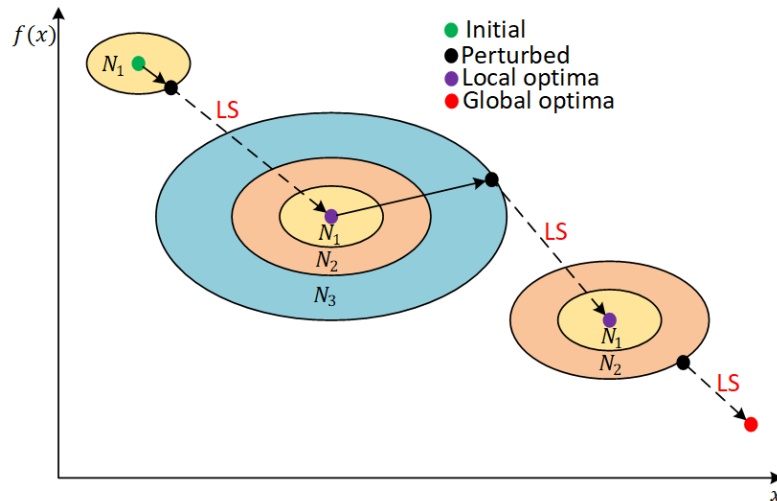


Figure 3.19: Variable neighborhood search

In the EVRP research field, the VNS was adapted for the diversification of the solution based on the cyclic exchange moves, with up to 15 different neighborhood operators [18, 91, 142]. The VNS was also combined with exact algorithms like branch-and-price [143], evolutionary algorithms [98], deterministic LS operators [56, 83], and adaptive mechanisms [9, 35, 91, 142] to further improve the solution quality.

3.3.5 Iterated local search

Iterated Local Search (ILS) [144] is a framework based on successively repeating LS on the current solution, given by Algorithm 3.1. When LS ends up in a local optima, a perturbation move is applied to escape the local optima. The essence of ILS can be given in a nut-shell: one iteratively builds a sequence of solutions generated by the embedded heuristic, leading to far better solutions than if one were to use repeated random trials of that heuristic [127]. The perturbation moves can be scaled dynamically to overcome the local optima. The effectiveness of the procedure highly depends on the used LS and perturbation operators. ILS was applied in several papers dealing with the EVRP problem [38], but usually it's coupled with some other metaheuristic: VNS [83], ant-colony [41], large neighborhood search [59, 66], or TS [57, 65].

Algorithm 3.1 Iterated local search [127]

- 1: $s_0 \leftarrow$ Generate initial solution
 - 2: $s^* \leftarrow$ Perform local search on s_0
 - 3: **while** termination criteria not met **do**
 - 4: $s' \leftarrow$ Perturb solution s^*
 - 5: $s'' \leftarrow$ Perform local search on s'
 - 6: $s^* \leftarrow$ Apply acceptance criteria on s''
 - 7: **end while**
-

3.3.6 Adaptive large neighborhood search

Neighborhood operators explore only the space of the immediate vicinity of the current solution, which often leads to local optima. In order to search a larger solution space, Shaw [145] proposed the Large Neighborhood Search (LNS) heuristic, which is based on destroying and repairing the solution - the ruin-recreate strategy. Heuristic efficiency depends on the implemented destroy and, especially, repair operators. The main drawback of the heuristic is the repeated use of the same destroy and repair operators, which can lead to local optima. The example of the LNS heuristic on the C101 instance and EVRPTW-FR problem is presented in Figure 3.20. The vehicle number remained the same, while the distance traveled decreased from 1085.55 before removal to 1053.88 after insertion. The removed customers are presented with red dashed circles in Figure 3.20b. The basic LNS has been successfully applied in several VRP and EVRP variants [41, 59, 65, 66, 93, 146].

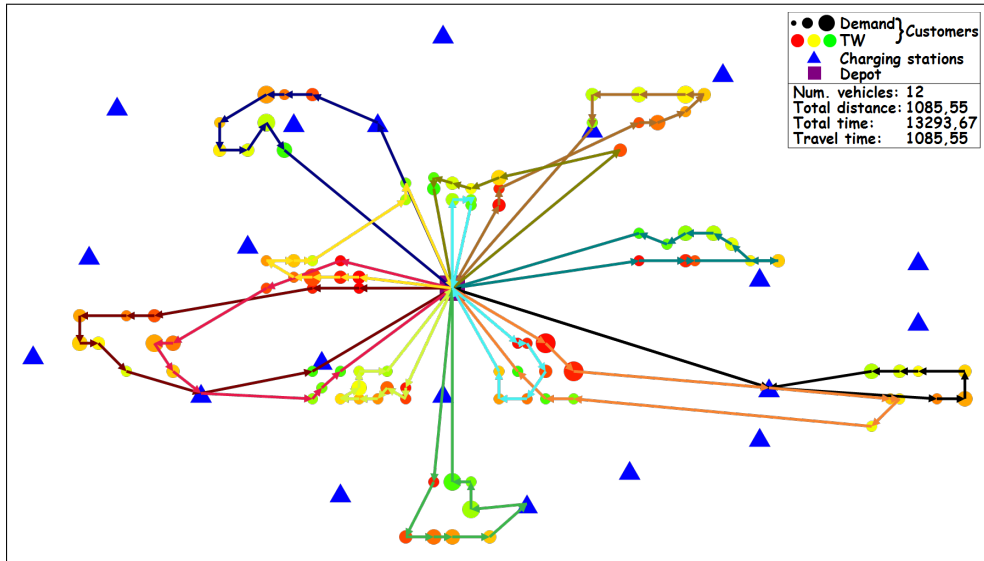
Adaptive Large Neighborhood Search (ALNS) [32, 147] is an extension of the LNS heuristic in which, throughout the search process, different destroy and repair operators are selected from respective D and R sets in an adaptive manner based on their past performance. The idea is to have versatile destroy and repair operators in order to escape the local optima. For each operator, the following attributes are stored:

- π_k - the score of the operator in period k ,
- θ_k - the number of times the operator was used in period k ,
- w_k - the weight of the operator in period k ,
- p_k - the probability of the operator in period k .

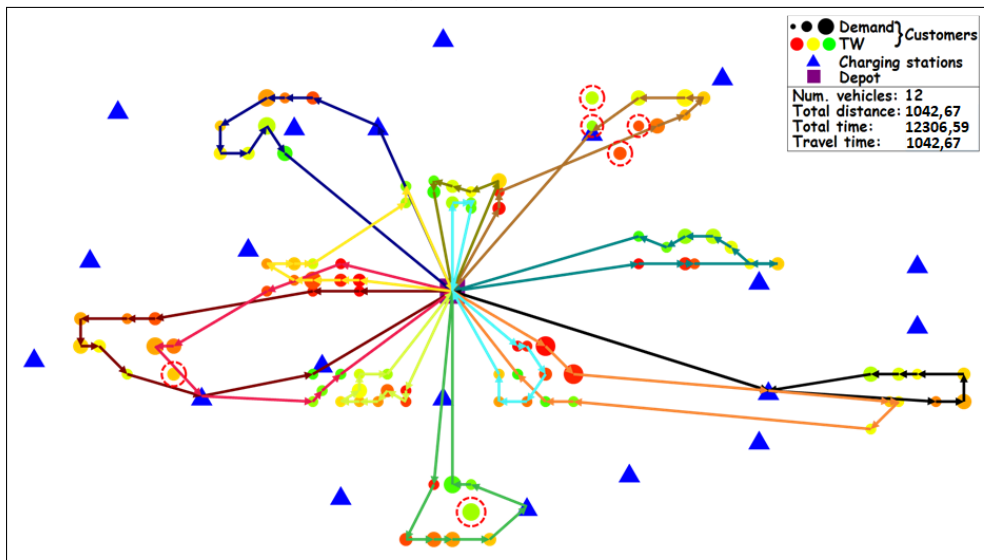
For each operator, the score value π_k is increased in the following order (scores $\sigma_1 \geq \sigma_2 \geq \sigma_3$):

- (i) σ_1 if a new best solution is found,
- (ii) σ_2 if the solution is better than the previous one,
- (iii) σ_3 if the solution is worse than the previous one but it is accepted due to the acceptance criteria,
- (iv) and no increase if the solution is not accepted.

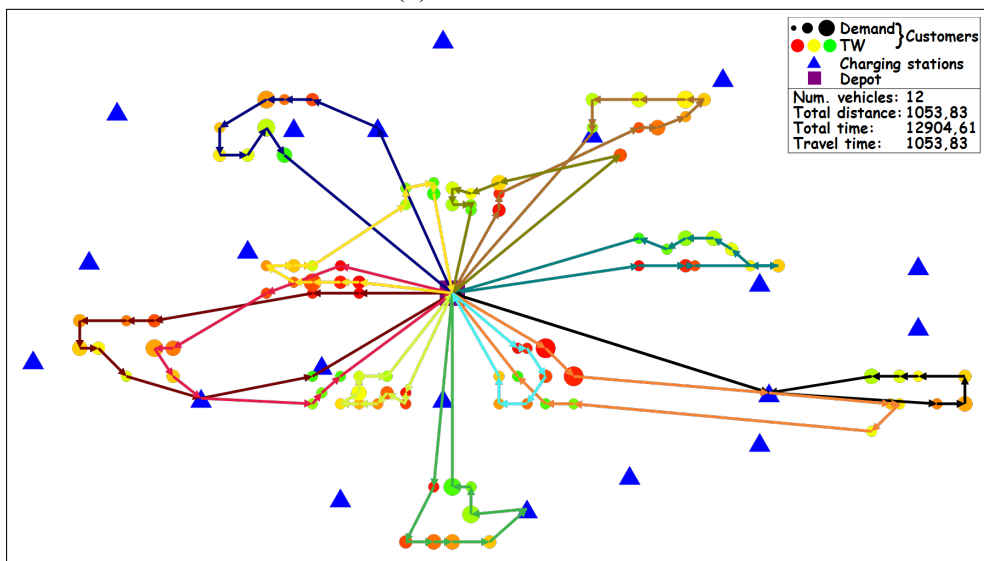
Some researchers use $\sigma_1 \geq \sigma_3 \geq \sigma_2$ order, to reward non-improved solutions that diversify the search [1, 32]. After each time period k in which operators are applied, the weights (equation 3.11) and the probabilities (equation 3.12) of operators are updated based on their previous weight values and new score values. The ρ factor controls the influence of past weight values for the computation of the new ones. Additionally, in each iteration, the scores and the number of times the operator was used are reset to zero. The repair operators probabilities are updated based on the operators in set R , while the destroy operators are updated based on the operators in set D . The operators in the current iteration are selected based on the Roulette Wheel Selection (RWS) [148], in which a probability of an operator selection is proportional to its weight value (equation 3.12). The set O represents destroy set D or repair set R . The example of RWS is



(a) Before removal



(b) After removal



(c) After insertion

Figure 3.20: LNS example: C101 - EVRPTW-FR

presented in Figure 3.21, where the area that the operator covers represents the probability that a randomly thrown ball will finish on it.

$$w_{k+1} = w_k(1 - \rho) + \rho \frac{\pi_k}{\theta_k} \quad (3.11)$$

$$p_k = \frac{w_k}{\sum_{o \in O} w_k^o} \quad (3.12)$$

The framework of the ALNS is given by Algorithm 3.2. ALNS is one of the most applied metaheuristic used to solve EVRP problems. Thus, a vast number of destroy and repair operators have been developed. Some of them originated from the original ALNS [32, 147], but also some new problem-specific ones regarding the CSs, have been developed [1, 20, 22, 29, 37, 53]. The comprehensive description of the applied destroy and repair operators in EVRP research field is presented by Erdelić et al. [21]. The ALNS operators tested during the development of the HALNS method are presented in section 4.4.

Algorithm 3.2 Adaptive large neighborhood search [32]

- 1: $s^* \leftarrow$ Generate initial solution
 - 2: **while** termination criteria not met **do**
 - 3: $d \leftarrow$ Select destroy operator from set D
 - 4: $r \leftarrow$ Select repair operator from set R
 - 5: $s' \leftarrow$ Destroy solution s^* with operator d
 - 6: $s'' \leftarrow$ Repair solution s' with operator r
 - 7: $s^* \leftarrow$ Apply acceptance criteria on s''
 - 8: **if** criteria for operators update met **then**
 - 9: Update operators' scores, weights and probabilities
 - 10: **end if**
 - 11: **end while**
-

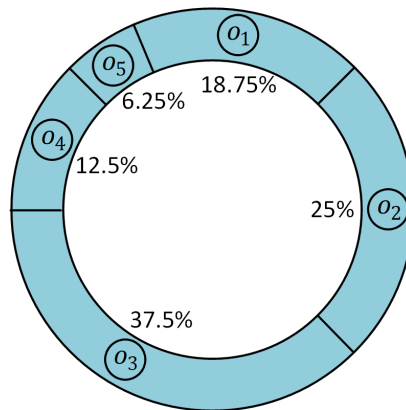


Figure 3.21: Roulette wheel selection

3.4 Feasibility

Regarding the feasibility during the search process, two types of strategies can be distinguished: (i) the ones that allow only feasible solutions, and therefore search only in a feasible solution space [1, 22, 23, 59, 65, 66, 69, 82, 83, 92], and (ii) the ones that allow infeasible solutions during the search process [9, 18, 20, 29, 53, 54, 91, 98]. The infeasible solution means that some users are served without satisfying all of the problem constraints, usually time-windows, load capacity and battery capacity constraints. The infeasible strategy generally broadens the search. In infeasible procedures, the objective function usually contains penalty coefficients for constraints violations, which are updated during the search process. At the beginning of the search, infeasible solutions are allowed in order to search a larger solution space. As the search process comes to an end, penalties for infeasible solutions increase in order to find feasible solutions. An example of such objective function used in infeasible search strategy is given by equation 3.13, where $f(s)$ is the total cost of the solution s , $P_{load}(s)$, $P_{tw}(s)$ and $P_{batt}(s)$ values of constraint violations, respectively, load capacity, time windows and battery capacity; and α , β and γ penalty coefficients [17, 18, 53, 54].

$$F(s) = f(s) + \alpha P_{load}(s) + \beta P_{tw}(s) + \gamma P_{batt}(s) \quad (3.13)$$

Although the idea of allowing infeasible solutions during the search process is not new, the efficient application of such strategy has been done only recently. The infeasible strategy has an advantage over a feasible one because it is easier to escape local optima, especially in problems with hard feasibility achievement such as EVRP problem. To better explain this concept, the example of feasible and infeasible search strategy is presented in Figure 3.22. The blue line represents the feasible solution space, while the red line represents the infeasible solution space. The current position marked with full blue circle represents a local optima. To overcome local optima and get to the global optima, a large solution space has to be escaped (blue dashed-dotted line). In many VRP variants, this usually means a solution with more than 50% difference in user schedule. In such cases, there are two strategies: explore a large solution space, which is time-consuming, or get lucky and hit the global optima by stochastic permutation. In an infeasible strategy, the local optima can be escaped by smaller moves in solution space (red dashed lines) and by the acceptance of worse solutions, which is common in the metaheuristic procedures. The difference between the infeasible and feasible strategy is smaller on problems that have large feasible solution space, like CVRP problem. On problems such as EVRPTW in which feasible solution space is much smaller, the infeasible strategy generally produces better solutions [18, 20, 29, 53, 54].

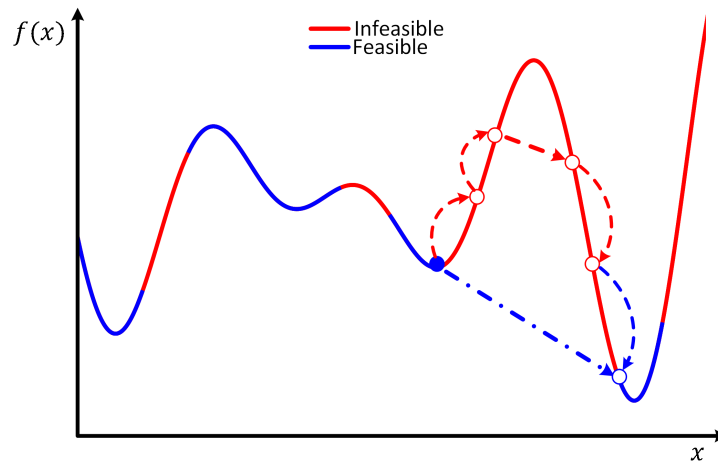


Figure 3.22: Difference between infeasible and feasible search strategy

3.5 Move evaluation

An important part of any VRP related search process is move evaluation. It is the evaluation of how good or bad a particular move is. The move is related to the performed LS, destroy, repair or perturb operators. Depending on the operator, this evaluation can be time-consuming. The goal is to perform the evaluation as fast as possible, and in the best case in constant time $\mathcal{O}(1)$. To highlight the importance of evaluation, the example of inter relocate evaluation is discussed. The assumption is that operator evaluation with $\mathcal{O}(1)$ complexity for one permutation is performed in 1 ms, while the evaluation with $\mathcal{O}(n)$ complexity for one permutation is performed in 10 ms, as there are in average roughly 10 customers per route in BKS of Solomon type 1 instances [31]. The number of inter relocate evaluations performed is equal to the number of permutations. With the assumption that each vehicle has 10 customers, one customer can be relocated to 90 other positions, and if repeated for each customer, the number of permutations is $90 \cdot 100 = 9000$. This results in total evaluation time of 9 s for $\mathcal{O}(1)$ complexity, and 90 s for $\mathcal{O}(n)$ complexity. If put in some practical conditions: (i) larger number of customers per route, (ii) different LS operators (at least 3-4) with different evaluation complexities, (iii) many other operators used in destroy, repair, perturb, or initial construction phases, and (iv) moderate assumption that such operators are applied at least 1000 times; it can be seen that execution time is highly affected and can be time-consuming. The evaluation time is a crucial part of any procedure used to solve a VRP problem.

The evaluation complexity of one move mainly depends on the observed problem variant and the objective function, but also, in some cases the type of the move itself. For example, the change in the objective function that solely depends on the total distance traveled can be computed in $\mathcal{O}(1)$ for all basic moves and almost all problems that do not consider time-dependent travel time. Such examples are presented for the computation of distance savings in CVRP

problem for most used LS operators in section 3.2. In the CVRP problem, all of the basic moves can be evaluated in $\mathcal{O}(1)$ if total traveled distance, total travel time, or total time objective functions are considered, as only load capacity is considered, which can be computed in $\mathcal{O}(1)$. In VRPTW, the total traveled distance and total travel time can be computed in $\mathcal{O}(1)$, but total time cannot be computed in $\mathcal{O}(1)$ as changes in route affect arrival times of customers further down the route. The begin times are nonlinear due to the time windows. In VRPTW, an additional problem is the feasibility check of time windows - the value of the time window violation, which in the most basic implementation cannot be computed in $\mathcal{O}(1)$, again due to the nonlinear arrival times. Fortunately, this can be done in $\mathcal{O}(1)$ by using additional variables, the latest arrival time, which is computed in a backward fashion, from route end to route start, while most of the other variables are computed in a forward fashion. The equations to compute forward variables for user i : arrival distance d_{a_i} , arrival time t_{a_i} and begin time t_{b_i} ; and backward variable latest arrival time t_{la_i} are given by equations 3.14-3.17. The arrival distance at user j is computed as the sum of arrival distance at user i and the traveled distance from i to j . The arrival time at user j is computed as the sum of the begin time of service at user i , the service time at user i , and the travel time from i to j . As in some cases, the arrival time at customer j is lower than the early (opening) time window, the vehicle has to wait for the user to open. Therefore, the begin time of service cannot start before the earliest begin time determined by the early time window, given by equation 3.16. If the arrival time is greater than the opening time window, the begin time is equal to the arrival time. The latest arrival time t_{la_i} represents the time by which the start of service has to begin; otherwise, the route is not time window feasible. It is computed in a similar way as the arrival time, but in a backward fashion from j to i , given by equation 3.17. Backward from user j to user i , the latest arrival time is computed as subtraction of latest arrival time at user j (the depot has the latest arrival time l_0), the travel time from i to j and the service time at user i . It is possible that t_{la_i} is greater than the late time window l_i , by which the service has to begin at user i . Therefore, in such occasion, the latest arrival time t_{la_i} is shifted to the late time window of user i by mathematical *min* operator.

$$d_{a_j} = d_{a_i} + d_{ij} \quad (3.14)$$

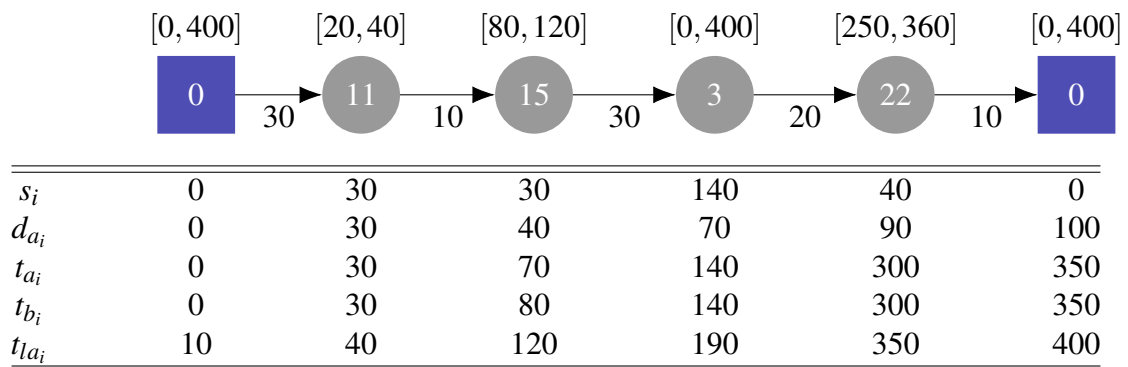
$$t_{a_j} = t_{b_i} + s_i + t_{ij} \quad (3.15)$$

$$t_{b_j} = \max(e_j, t_{a_j}) \quad (3.16)$$

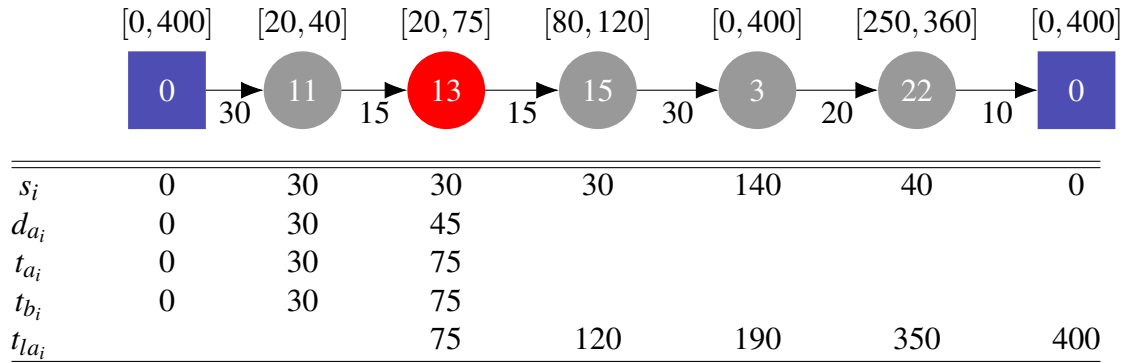
$$t_{la_i} = \min(t_{la_j} - s_i - t_{ij}, l_i) \quad (3.17)$$

3. Problem solving methods

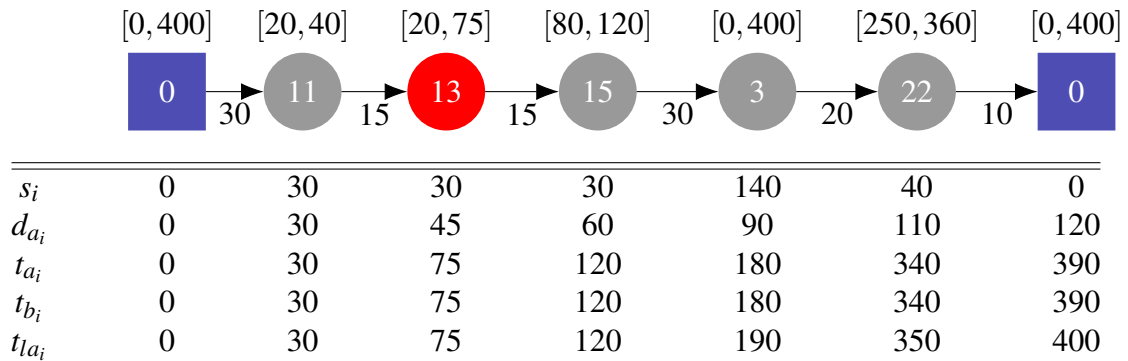
The example of computed variables for one VRPTW route is presented in Figure 3.23a. The depot is represented with the purple rectangle, while users are represented with gray circles. Each user has a time window $[e_i, l_i]$ (above user) and service time (in a row below). The arc distances between users are marked with arrows, and the assumption is that arc distance, travel time, and energy consumption have the same values ($d_{ij} = e_{ij} = t_{ij}$). It is interesting to note that the arrival time at customer 15 is 70, but the begin time of service is 80, as the vehicle has to wait for the opening of the customer to start a service. The example of the insertion of a customer 13 into a route is presented in Figure 3.23b. To evaluate the customer insertion, forward and



(a) Route evaluation before customer insertion - VRPTW



(b) Customer insertion - VRPTW



(c) Route evaluation after customer insertion - VRPTW

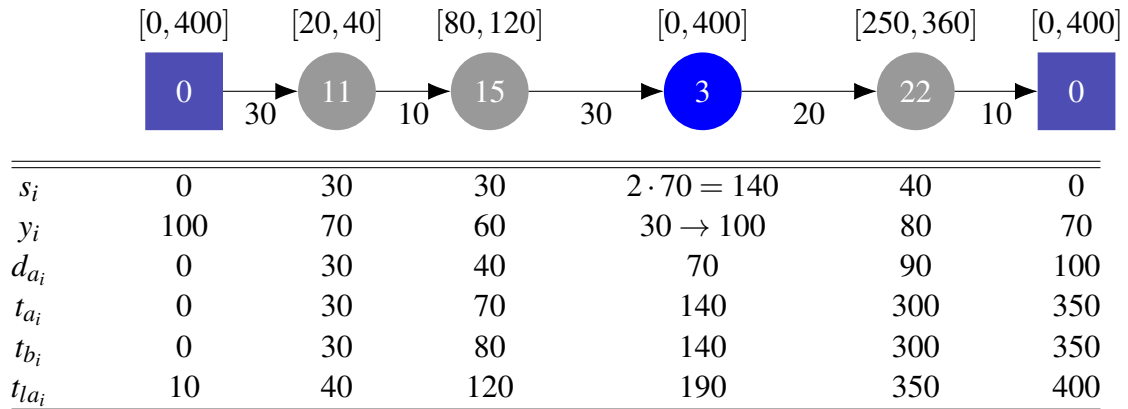
Figure 3.23: Move evaluations - VRPTW

backward variables have to be computed up to the inserted customer, which is performed in $\mathcal{O}(1)$. As condition $t_{b_i} \leq t_{l_{a_i}}$ ($75 \leq 75$) is satisfied, the customer insertion is feasible and the overall distance increase is $d = 15 + 15 - 10 = 20$. Afterward, if this customer insertion move is selected as the best move, the move is performed, and all variables are recomputed (Figure 3.23c).

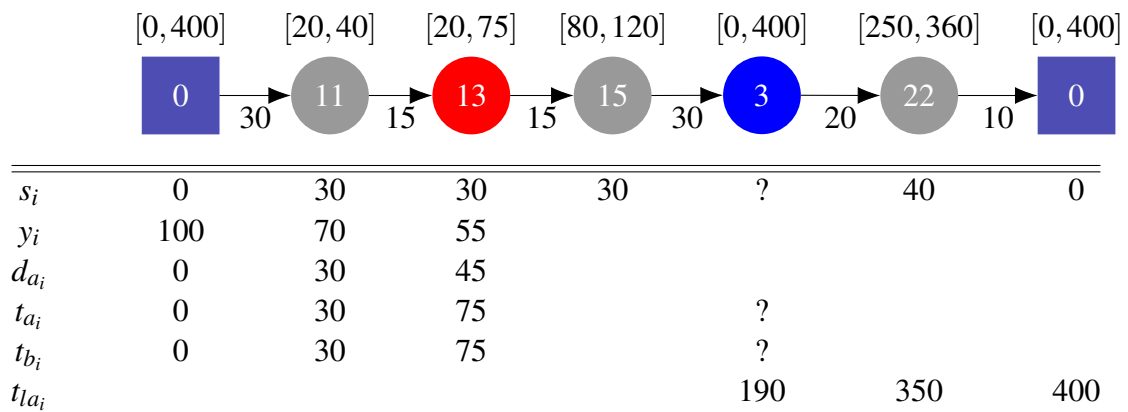
In the EVRPTW problem, beside the load and time windows constraints, there is also a constraint regarding the vehicle battery capacity (energy feasibility), which also influences the time window feasibility. Therefore, the time window evaluation is more complex. To present the example of route evaluation in EVRPTW, the forward rest battery capacity is added, given by equation 3.18.

$$y_j = y_i - e_{ij} \quad (3.18)$$

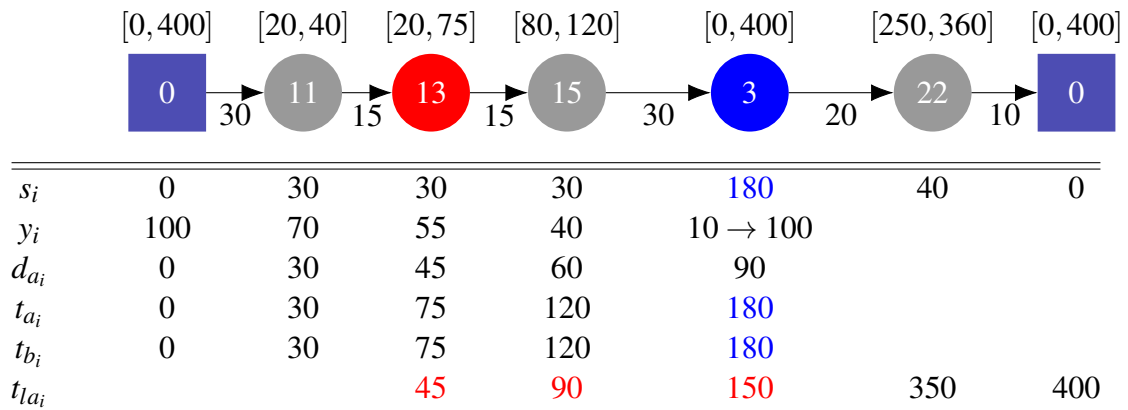
A similar example as for VRPTW is presented in Figure 3.24a for EVRPTW-FR, where customer 3 is changed to CS (blue circle) with charging time corresponding to same amount of service time as in VRPTW example, and with the recharging rate $g = 2$. To evaluate the move of inserting the customer 13, the forward and backward variables have to be computed for customer 13, but the problem is that the recharging amount is unknown at CS, and therefore the service time at CS is also unknown (3.24b). In problems where the energy consumption can be estimated from the distance increase, the additional charging amount can also be computed, which in this case is 20. Therefore the total charging time at CS is $140 + 20 \cdot 2 = 180$. But even with that, due to the nonlinear begin times caused by time windows, the arrival time at CS is unknown, or if looked in the opposite way, the latest arrival time at customer 13 is unknown. To know either of the values, the forward or backward variables have to be recomputed (propagated) forward up to the latest CS in route or backward from the latest CS in route up to the customer 13 (3.24c). In this case, at user 3 as the condition $t_{b_i} \leq t_{l_{a_i}}$ ($180 \leq 150$) is not satisfied, the time window violation of 30 occurs, and this customer insertion is not feasible. Therefore, the same example of customer insertion in VRPTW and EVRPTW-FR have different outcomes. Naturally, one could assume that the energy violation would occur, but actually, what usually happens is that an additional charging amount causes the time window violation. The complexity of such approach is $\mathcal{O}(B)$ where B is the number of users between the position of the insertion and the latest CS in route. It is not enough to extend the variables just to the next CS in route. If there is no CS in the second part of the route, recomputation is not needed. By storing additional information about the recharging position and its amount, and by the use of general concatenation operators, the basic neighborhood operators in EVRPTW can be evaluated in $\mathcal{O}(1)$ [29, 121]. A different approach is to use a specially defined surrogate cost function whose evaluation is less demanding and completely evaluate only a subset of best moves [20].



(a) Route evaluation before customer insertion - EVRPTW-FR



(b) Insertion of customer - EVRPTW-FR



(c) Propagation of variables after insertion - EVRPTW-FR

Figure 3.24: Move evaluations - EVRPTW-FR

Hybrid adaptive large neighborhood search method

In this chapter, the Hybrid Adaptive Large Neighborhood Search (HALNS) method for solving different EVRP variants is presented. The term hybrid is used as the ALNS method is coupled with the exact procedure for optimal CS placement. The proposed HALNS method includes the LS procedure and specially defined concatenation operators that allow the evaluation of some moves (local search moves) in $\mathcal{O}(1)$. The idea for the HALNS is based on the ALNS for Electric Location Routing Problem with Time Windows and Partial Recharging (ELRPTW-PR) presented by Schiffer et al. [53]. The authors formulate penalty functions and concatenation operators that allow evaluation of most of the basic moves in ELRPTW-PR problem in $\mathcal{O}(1)$. In this thesis, the proposed penalty functions and concatenations operators for ELRPTW-PR are simplified and applied to solve EVRPTW-PR. Further on, the new penalty functions and concatenation operators are presented for three problems: EVRPTW-FR, EVRPTWDCS-PR, and EVRPTWDCS with Full Recharge (EVRPTWDCS-FR). The EVRPTWDCS-FR problem has not been addressed in the literature by itself, as it is mostly observed as a sub-problem of the partial variant, but if looked at separately, this problem has different constraints than the partial variant.

The chapter is organized as follows. In section 4.1 the overview of the proposed HALNS method is presented. In section 4.2 the heuristic for creating the initial solution is given. Then the penalty functions and concatenations operators for four different EVRP variants are given in section 4.3. The destroy and repair operators tested during the development of HALNS method are presented in section 4.4, while the used route removal operators are presented in section 4.5. Then, the LS and exact procedures for improving the solution are described in sections 4.6 and 4.7. Further on, the techniques that significantly improve the execution time are described in section 4.8. The tuning of the HALNS parameters is presented in section 4.9. In the end, the

results of HALNS applied on both small and large EVRPTW instances are presented in section 4.10.

4.1 Framework

The general framework of the proposed HALNS method is given by Algorithm 4.1. First, the initial solution is created by special NNH that takes into account CS insertions presented in section 4.2. The produced initial solution s is completely feasible and copied to additional three solution instances: (i) temporal solution s_{temp} that stores previous iteration solution values (can be infeasible), (ii) best solution s_{best} that stores so far the best solution (can be infeasible), and (iii) the best feasible solution s_{best_feas} . The initial values of iteration i and last iteration in which the improvement happened, i_{last_imp} , are set to zero. In each iteration, the route removal criteria is checked, which if met, calls the procedure to minimize the number of vehicle routes, described in section 4.5. In each iteration, the current solution s is destroyed with the selected operator d , and repaired with the selected operator r , and as a result, a new solution s_{new} is produced. The destroy and repair operators used in HALNS are described in section 4.4. If the cost function of solution s_{new} is within Δ_{ls} percentage of solution s_{best} cost function, the LS procedure, described in section 4.6, is performed on solution s_{new} . Additionally, if the cost function of this new solution s_{new} after the LS procedure is within Δ_{exact} ($\Delta_{exact} < \Delta_{ls}$) percentage of s_{best} cost function, the exact procedure that determines optimal CS positions is performed on solution s_{new} . The exact procedure is described in section 4.7. Afterwards, regardless of the LS and exact procedures, the update of solutions instances occurs in lines 17-27. If a new solution s_{new} is better than the current temporal solution s_{temp} , the s_{new} is set as s_{temp} . If a new solution s_{new} is better than the best solution s_{best} , the s_{new} is set as s_{best} . As solution s_{best} can be infeasible the procedure to generate a feasible solution $s_{best_feas}^{new}$, if possible, is performed. The method for generating feasible solutions is presented in subsection 4.6.2. If this new solution $s_{best_feas}^{new}$ is feasible and better than the so far best feasible solution s_{best_feas} , the $s_{best_feas}^{new}$ is set as s_{best_feas} , and the last improvement i_{last_imp} is set to the current value of i . Further on, the scores of used destroy and repair operators are updated. Afterward, additional three updates are performed:

- (i) every μ_{rbks} iterations the temporal solution s_{temp} and best solution s_{best} are reset to the best feasible solution s_{best_feas} ,
- (ii) every μ_{uop} iterations the scores, weights and probabilities of destroy and repair operators are updated, and
- (iii) every μ_{upw} iterations the penalty coefficients are updated.

In the end, the iteration value i is increased by one and the temporal solution s_{temp} is set as solution s . The whole procedure (lines 4-40) is repeated until either a maximum value of

iteration μ_{max} is reached, or there was no improvement of the feasible solution for more than $\mu_{max}^{last_imp}$ iterations.

Algorithm 4.1 Hybrid adaptive large neighborhood search method

```

1:  $s \leftarrow$  Generate initial solution
2:  $s_{temp}, s_{best}, s_{best\_feas} \leftarrow$  Copy  $s$ 
3:  $i, i_{last\_imp} \leftarrow 0$ 
4: while  $i < \mu_{max}$  and  $i - i_{last\_imp} < \mu_{max}^{last\_imp}$  do
5:   if route removal criteria met then
6:      $s \leftarrow$  Perform route removal on  $s$ 
7:   end if
8:    $d, r \leftarrow$  Select destroy and repair operators
9:    $s_d \leftarrow$  Destroy solution  $s$  with operator  $d$ 
10:   $s_{new} \leftarrow$  Repair solution  $s_d$  with operator  $r$ 
11:  if  $f(s_{new}) < f(s_{best})(1 + \Delta_{ls})$  then
12:     $s_{new} \leftarrow$  Perform LS on  $s_{new}$ 
13:    if  $f(s_{new}) < f(s_{best})(1 + \Delta_{exact})$  then
14:       $s_{new} \leftarrow$  Perform exact procedure on  $s_{new}$ 
15:    end if
16:  end if
17:  if  $f(s_{new}) < f(s_{temp})$  then
18:     $s_{temp} \leftarrow s_{new}$ 
19:    if  $f(s_{new}) < f(s_{best})$  then
20:       $s_{best} \leftarrow s_{new}$ 
21:       $s_{best\_feas}^{new} \leftarrow$  Generate feasible solution from  $s_{new}$ 
22:      if  $s_{best\_feas}^{new}$  is feasible and  $f(s_{best\_feas}^{new}) < f(s_{best\_feas})$  then
23:         $s_{best\_feas} \leftarrow s_{best\_feas}^{new}$ 
24:         $i_{last\_imp} \leftarrow i$ 
25:      end if
26:    end if
27:  end if
28:  Update scores of selected destroy  $d$  and repair  $r$  operators
29:  if modulo( $i, \mu_{rbks}$ ) = 0 then
30:     $s_{temp}, s_{best} \leftarrow s_{best\_feas}$ 
31:  end if
32:  if modulo( $i, \mu_{uop}$ ) = 0 then
33:    Update destroy and repair operators scores, weights and probabilities
34:  end if
35:  if modulo( $i, \mu_{upw}$ ) = 0 then
36:    Update penalty weights
37:  end if
38:   $i \leftarrow i + 1$ 
39:   $s \leftarrow s_{temp}$ 
40: end while

```

Temporal solution s_{temp} is stored because sometimes the new solution s_{new} created by the destroy and repair operators can be much worse than the current temporal solution s_{temp} . In such cases, instead of propagating such bad solution to next iteration, the current solution s is set as the temporal solution s_{temp} . The solution s_{best} is used as it evaluates the infeasible solution space. The basic idea to store possibly infeasible solution is adopted from similar works that search in the infeasible solution space [53, 149]. By storing the s_{best} , which often is infeasible, the infeasible regions of the solution space are evaluated. As already explained in subsection 3.4, the small changes in infeasible search space can lead to a feasible solution.

4.2 Pseudo-greedy time-oriented nearest neighbor heuristic

To construct an initial solution on all observed EVRP variants, a new modified nearest neighbor heuristic is proposed: Pseudo-Greedy Time-Oriented Nearest Neighbor Heuristic (PGTONNH). The time-oriented construct heuristic selects the nearest customer in a timely manner which makes it suitable for the application on the problems that deal with time windows [31, 150]. In EVRP problems, the CSs have to be inserted to make the solution energy feasible. The proposed heuristic is described by Algorithm 4.2. Routes are constructed in a serial way, one by one, by selecting one customer from the set of all customers that satisfy all problem constraints. To add the customer j after the customer i in active route, the candidate customer j has to satisfy:

- (i) Vehicle load capacity given by equation 4.1, where u_i is the rest load capacity at user i and q_j demand of customer j (the first user in a route, the depot, has $u_0 = C$):

$$u_i - q_j \geq 0, \quad (4.1)$$

- (ii) Customer j late time window given by equation 4.2, where the arrival time at customer j is computed by equation 3.15:

$$t_{a_j} \leq l_j, \quad (4.2)$$

- (iii) The depot time window from customer j given by equation 4.3, where begin time at customer j is computed by equation 3.16:

$$t_{b_j} + s_j + t_{j0} \leq l_0, \quad (4.3)$$

- (iv) Rest vehicle battery capacity (energy) given by equation 4.4 (the first user in route, the depot, has $y_0 = Q$):

$$y_i - e_{ij} \geq 0. \quad (4.4)$$

The PGTONNH starts with the initialization of the solution s with an empty vehicle route list and a set of unserved customers U with all customers. Then the new vehicle route v is opened,

Algorithm 4.2 Pseudo-greedy time-oriented nearest neighbor heuristic

```

1:  $s \leftarrow$  Initialize solution with empty vehicle list
2:  $v \leftarrow$  Open a new vehicle
3:  $i \leftarrow$  Set the depot as last visited user
4:  $U \leftarrow$  Initialize set of unserved customers with all customers
5: while  $U$  is not empty do
6:    $C \leftarrow$  Initialize empty set
7:   for each customer  $j$  in  $U$  do
8:     if vehicle  $v$  has enough rest load capacity to accept  $j$  and  $j$  is reachable from  $i$  and the depot is reachable from  $j$  regarding the time windows then
9:       if  $j$  is reachable from  $i$  and the depot is reachable from  $j$  regarding the energy then
10:        Add  $j$  to set  $C$ 
11:       else
12:        Add  $j$  and depot to vehicle  $v$ 
13:         $v \leftarrow$  Perform MBSI
14:        if  $v$  is feasible then
15:          Add  $j$  with determined CS insertions to set  $C$ 
16:           $i \leftarrow j$  or succeeding CS
17:        else
18:          Remove  $j$  and last depot from vehicle  $v$ 
19:        end if
20:      end if
21:    end if
22:  end for
23:  if  $C$  is empty then
24:    Close the current vehicle  $v$  and add it to the vehicle list in solution  $s$ 
25:     $v \leftarrow$  Open a new vehicle
26:     $i \leftarrow$  Set the depot as last visited user
27:  else
28:    Sort customers (with possible CSs insertions) in set  $C$  based on the objective function given by equation 4.5
29:     $z \leftarrow$  From  $k$  customers that have the lowest objective function value select one using determinism factor  $k_{pgtonnh}$ 
30:    Add  $z$  together with the possible CSs insertions to the current vehicle  $v$ 
31:    Remove customer  $z$  from set of unserved customers  $U$ 
32:     $i \leftarrow z$  or succeeding CS
33:  end if
34: end while

```

and the depot is set as the last visited user i . In each outer loop iteration, a set of customer candidates C is initialized as empty. Each unserved customer j is checked if it satisfies the load and time-window constraints, but not the energy constraint, as either the customer j is not energy reachable or the depot from j is not energy reachable. If all constraints are satisfied, the customer j is added to set C ; otherwise, if only energy constraint is violated, the Multiple Best Station Insertion (MBSI) procedure [1] on the current vehicle is performed (line 13) to make the solution energy feasible, with the full recharge strategy. A detailed description of the MBSI

procedure is given in the next paragraph by Algorithm 4.3. After MBSI, if the vehicle route is feasible, the customer j together with the CSs and its positions is added to set C . If after checking all unserved customers, the set C is still empty, not a single customer could be added to a current vehicle route without a constraint violation. In such case, a current vehicle route is closed by adding a new instance of the depot at the end of the route, and the complete procedure is repeated from start until all customers are served (lines 5-34). If set C is not empty, then all of the candidates are evaluated by equation 4.5, which has three parts:

- (i) distance traveled between users i and j ,
- (ii) time between the begin of service at user j and end of service at user i (this time can include waiting time for time window opening at the customer j),
- (iii) spare time to the late time window of customer j .

δ_1 , δ_2 and δ_3 are coefficients that control the relation between the described parts. They have to be larger or equal to zero and have the sum equal to one (equation 4.6). The used equation also takes into account the possible inserted CSs within time and distance variables. Next, from k customers that have the lowest objective insertion value, one is selected based on the determinism factor $k_{pgtonnh}$ and added to the route together with appropriate CS insertions. This determinism factor favors customers with a lower objective function value and is more thoroughly described in subsection 4.4.1. The parameter value $k_{pgtonnh}$ controls the stochastic component of the algorithm. The values of $k_{pgtonnh} = 1$ (random) are suited for use in population metaheuristics where the goal is to have a versatile set of individuals in the initial population. In the end, the selected customer z is removed from the set of unserved customers U , the last visited user is updated to be either customer z or succeeding CS.

$$c_{ij} = \delta_1 d_{ij} + \delta_2 (t_{b_j} - (t_{b_i} + s_i)) + \delta_3 (l_j - (t_{b_i} + s_i + t_{ij})) \quad (4.5)$$

$$\delta_1 + \delta_2 + \delta_3 = 1, \delta_1 \geq 0, \delta_2 \geq 0, \delta_3 \geq 0 \quad (4.6)$$

The MBSI procedure is given by Algorithm 4.3. The first goal is to determine the position of the first energy violation in route and then insert one or more CSs somewhere before in route to make the route energy feasible. The tuple I_{best} , which contain CSs instances, their positions, and total cost value increase is stored. Next the backward loop from the energy violation position i to the depot position is performed (lines 3-26). In each iteration, the insertion of CSs between users at the current position is evaluated. If this insertion evaluation I_1 is feasible and it is better than the so-far best feasible insertion I_{best} , then I_1 is set as I_{best} . Otherwise, if it is time window feasible (the time window infeasible insertion cannot be improved by CS insertion), but not energy feasible, and the I_{best} is empty (meaning that no feasible solution was found so-far), then the evaluation of another CS insertion I_2 at the same position in the same way as in the previous

Algorithm 4.3 Multiple best station insertion

Input: Vehicle route v

- 1: $i \leftarrow$ Determine the first position of energy violation in route v (-1 if route v is feasible)
- 2: CS insertion tuple I_{best} : CSs, positions, value $\{cs, i_{cs}, f\} \leftarrow \{none, -1, max\}$
- 3: **while** $i \geq 1$ **do**
- 4: Current user $u_{cur} \leftarrow$ User at position i in vehicle route v
- 5: User before $u_{bef} \leftarrow$ User at position $i - 1$ in vehicle route v
- 6: **for each** possible CS cs_1 between u_{bef} and u_{cur} **do**
- 7: $I_1 \leftarrow$ Evaluate the insertion of CS cs_1 at position i in route v
- 8: **if** I_1 is completely feasible **then**
- 9: **if** $f(I_1) < f(I_{best})$ **then**
- 10: $I_{best} \leftarrow I_1$
- 11: **end if**
- 12: **else if** I_1 is time window feasible **and** not energy feasible **and** I_{best} is empty **then**
- 13: Insert cs_1 at position i in route v
- 14: **for each** possible CS cs_2 between u_{bef} and cs_1 **do**
- 15: $I_2 \leftarrow$ Evaluate the insertion of CS cs_2 at position i in route v
- 16: **if** I_2 is completely feasible **then**
- 17: **if** $f(I_1) + f(I_2) < f(I_{best})$ **then**
- 18: $I_{best} \leftarrow I_1$ and I_2
- 19: **end if**
- 20: **end if**
- 21: **end for**
- 22: Remove cs_1 at position i in route v
- 23: **end if**
- 24: **end for**
- 25: $i \leftarrow i - 1$
- 26: **end while**
- 27: **if** I_{best} is not empty **then**
- 28: Preform I_{best} on vehicle route v
- 29: **end if**

insertion is performed. This is needed as in EVRPTW instances at most two CS are necessary to reach the customer (section 2.2). Further on, if this new insertion I_2 produced a feasible solution and together with I_1 has a lower cost than the I_{best} , then both insertions I_1 and I_2 are stored as I_{best} . At the end of the while loop, the value of position i is decreased by one, and the procedure is repeated. In the end, if the best insertion I_{best} is not empty, it is performed to make the route v energy feasible.

The example of the EVRPTW-FR initial solution on instance C101 is presented in Figure 4.1. In the example, the following parameter values were used: $k_{pgtonnh} = 3$, $\delta_1 = 0.6$, $\delta_2 = 0.3$ and $\delta_3 = 0.1$. In comparison to the Sweep algorithm, CWS, and NNH of Felipe presented in section 3.1, the PGTONNH produces a completely feasible solution and significantly reduces the number of vehicles compared to the NNH algorithm.

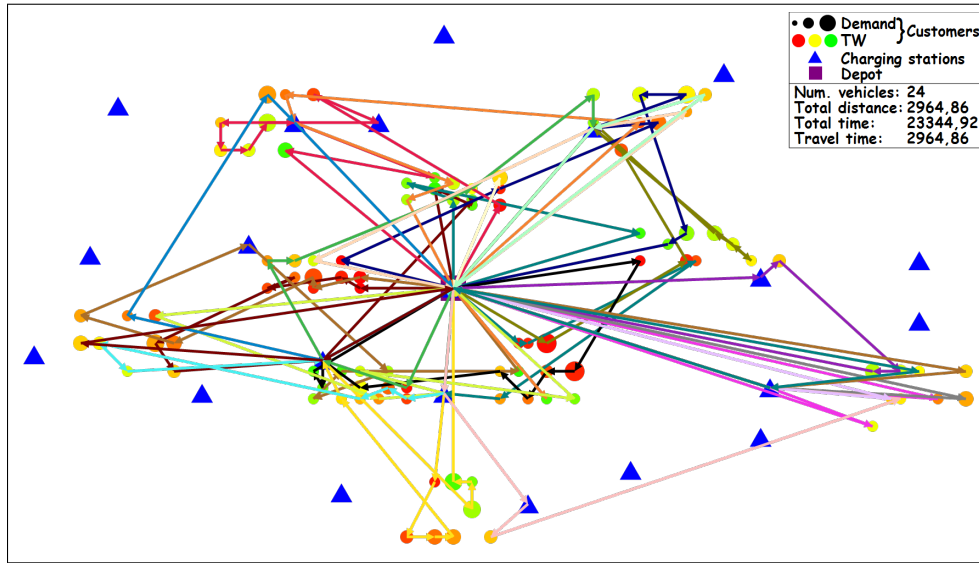


Figure 4.1: Initial solution constructed by PGTONNH - C101

4.3 Penalty functions

The proposed HALNS method allows infeasible solutions during the search phase. The advantages and drawbacks of searching in the infeasible solution space are already discussed in section 3.4. In HALNS, the generalized cost function given by equation 3.13 with penalties for time window, load capacity, and battery capacity is considered. As already mentioned, the penalty coefficients are labeled as α for load capacity, β for time window, and γ for battery capacity violation. The values of these coefficients control the ratio of diversification and intensification phases. In the beginning, these coefficients are initialized to the predefined starting values α_0 , β_0 , and γ_0 . To skip getting stuck in the over-intensification or over-diversification their values are bounded as $[\alpha_{min}, \alpha_{max}]$, $[\beta_{min}, \beta_{max}]$, and $[\gamma_{min}, \gamma_{max}]$. The penalty coefficients are updated in adaptive manner [53, 149], every μ_{upw} iterations, in the following way:

- (i) multiplied by ζ if in the last μ_{upw} iterations penalty in best solution s_{best} occurred, or
- (ii) divided by ζ if in the last μ_{upw} iterations penalty in the best solution s_{best} did not occur.

The penalty of the solution s is the sum of penalties per vehicle v in the solution $s = \{v_1, \dots, v_n\}$. As here, the focus will be only on the penalties and not the cost function, the penalty of the solution s can be computed by equation 4.7 as the sum of penalties per vehicle.

$$P(s) = \sum_{v \in s} P(v) = \sum_{v \in s} (\alpha P_{load}(v) + \beta P_{tw}(v) + \gamma P_{batt}(v)) \quad (4.7)$$

The example of a penalty coefficient function α during the search process when solving EVRPTW-FR on instance R101 is presented in Figure 4.2. The values of penalty coefficient α are plotted with black color and presented on the right y axis. On the left y axis are the values

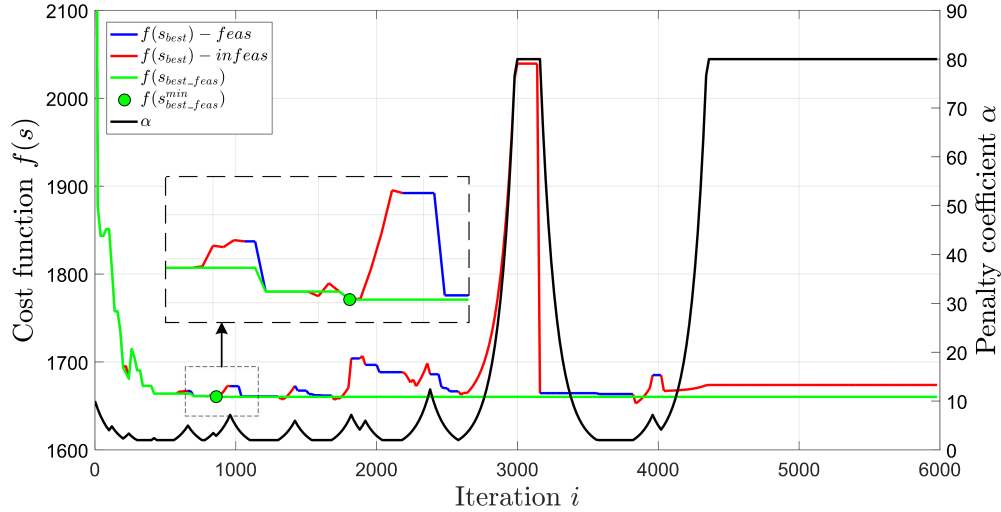


Figure 4.2: Penalty function value in search process - R101 - EVRPTW-FR

of the cost function for the best solution s_{best} (blue line for the feasible part and red line for the infeasible part), and best feasible solution s_{best_feas} (green line). The first occurrence of the best overall feasible solution $s_{best_feas}^{min}$ is represented with a green circle. The used minimum, maximum and starting alpha values are: $\alpha_{min} = 2$, $\alpha_{max} = 80$, and $\alpha_0 = 10$. The update value is $\zeta = 1.2$, and the penalties are updated every $\mu_{upw} = 20$ iterations. The cost value of the initial solution is 2829.84. It can be seen that the feasible solution is rapidly improved at the beginning of the search, with occasional spikes. These spikes represent that the feasible solution with a lower number of vehicles is found and that it is accepted although it has a worse cost function value. When the best solution s_{best} is infeasible but accepted due to the acceptance criteria (red lines), the penalty coefficient α increases, while the opposite occurs when the best solution s_{best} is feasible. In the zoomed part, it can be seen that allowing infeasible solutions can lead to better feasible solutions. Also, as minimum best feasible solution $s_{best_feas}^{min}$ is relatively quickly found ($i = 880$), in the rest of the search, the infeasible solutions are accepted, and penalties are updated to find a better solution.

As already mentioned in section 3.5 the important part of any VRP algorithm is the evaluation of the basic moves (changes) in the solution. As here, the cost function also includes penalties, the important aspect is also the efficient evaluation of the penalties in the solution.

The easiest penalty to compute is the vehicle load penalty. If a vehicle route v is represented as a sequence of users $v = (u_0, \dots, u_{N+1})$ the vehicle load penalty can be computed by equation 4.8. The penalty is equal to the amount of total vehicle overload: the load over the vehicle load capacity C . The evaluation of load violation between partial routes $\phi_1 = (u_0, u_1, \dots, x)$ and $\phi_2 = (y, \dots, u_{N+1})$ can be done in $\mathcal{O}(1)$ for most of the known move operators [18, 53, 151]. This requires tracking two forward variables for each user in route: rest load capacity z_i (in previous mathematical formulations, it was u_i , but to not confuse it with the user u_i here a

different symbol is used) and forward load violation p_{load} . For example, if the last user in the partial route ϕ_1 has rest load capacity z_x and load violation p_{load_x} , the first user in the partial route ϕ_2 has rest load capacity z_y and the last user in the partial route ϕ_2 has rest load capacity z_{N+1} , the load violation of vehicle route v combined from two partial routes ϕ_1 and ϕ_2 can be computed by equation 4.9. The way of storing variables for each user in route and evaluating the changes in solution by considering partial routes is called the concatenation strategy [3, 17].

$$P_{load}(v) = \max \left(\left(\sum_{u \in v} q_u \right) - C, 0 \right) \quad (4.8)$$

$$P_{load}(\phi_1 \otimes \phi_2) = p_{load_x} + \max(z_{N+1} - z_y + q_y - z_x, 0) \quad (4.9)$$

Due to the vehicle route's non-linear total travel time function, the evaluation of time windows is more complex. The example of computing the time window violation in a classical way is presented in Figure 4.3 by equation 4.10, where t_{a_u} is arrival time at user u and l_u is late time window at user u . The first violation occurs at user 15 as arrival time of 70 is larger than the late time window value of 50, and the time window violation is 20. Then, propagated further, the time window violation at user 3 occurs, as arrival time is 130, the late time window is 110, and the additional 20 is added to the total time window violation. In the end, one more violation occurs at the last user, resulting in the total time window violation value of 50.

$$P_{tw}(v) = \sum_{u \in v} \max(t_{a_u} - l_u, 0) \quad (4.10)$$

Nagata et al. [152] concluded that this way of computing time windows over-penalizes total time window violation. For example, the first penalty in Figure 4.3 occurs at user 15. As the vehicle is late at user 15, it is also possible that it will be late at other users after the user 15, and as a result, time window violation accumulates. Here, the problem is not in the sequence of users after the user 15, but rather only the user 15 produces the time window violation. The real question should be, if vehicle arrives before the late time window at user u , would there

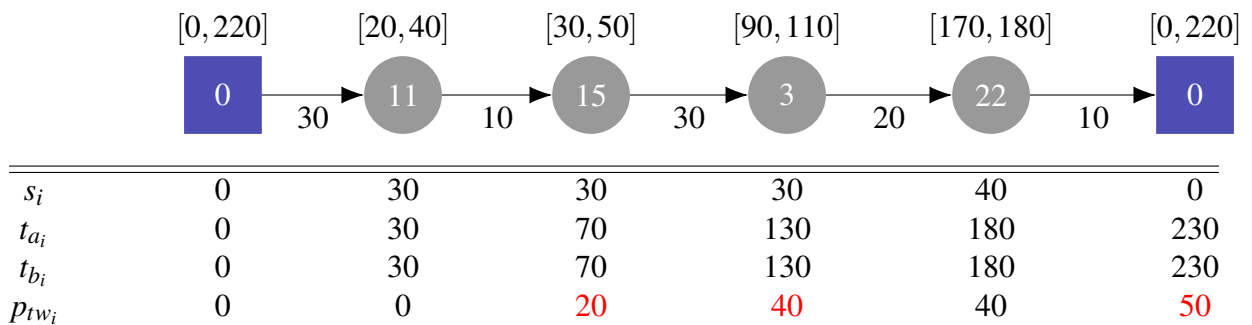
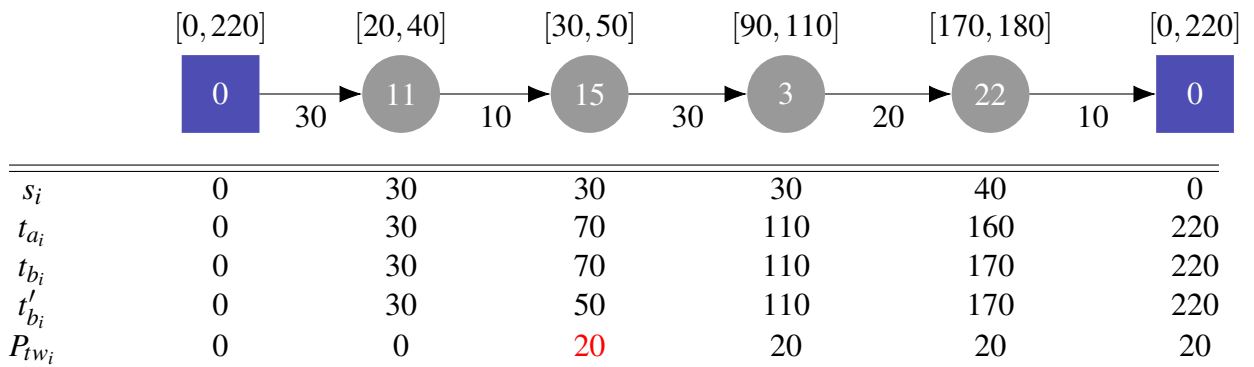
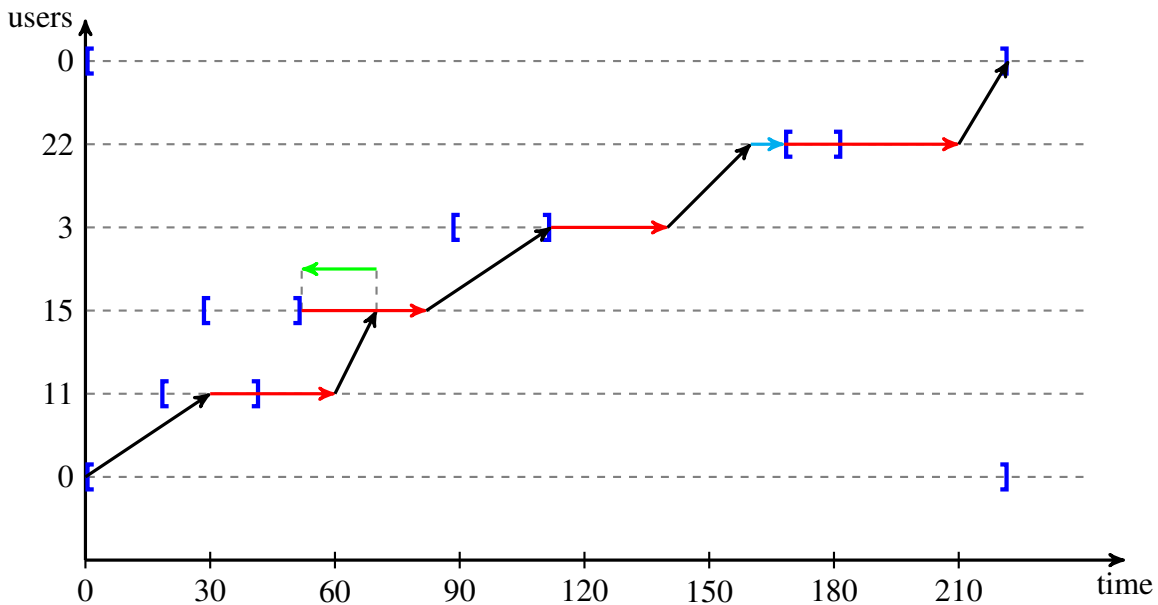


Figure 4.3: Computation of time window violations in a classical way in VRPTW

still be a time window penalty in the second part of the route? The answer is maybe yes, but certainly with lower or the same penalty value, depending on the user schedule. To reduce the over-penalization, Nagata et al. [152] proposed a time traveling technique, meaning that when the violation at user occurs, quantify the violation, and travel back in time, via time machine, to the latest feasible point in time. The example of computing time window violations with traveling back in time to the latest feasible point in time in VRPTW problem is presented in Figure 4.4. The same sequence of customers as in the previous example is used. One additional variable, the shifted begin time t'_{b_i} (latest feasible point in time) is computed by equation 4.11 and used in computation instead of a classical begin time t_{b_i} . The Figure 4.4b presents a graph of total time, where on the x axis is time and on the y axis are the users in the route. The time windows are represented by blue square brackets, the travel time between users with black arrows, the service time with red arrows, the waiting time for early time window opening with a cyan arrow, and the travel back in time to the latest feasible point with green arrow. It can be



(a) Computation of time window violations



(b) Travel times on a graph

Figure 4.4: Strategy of traveling back to the latest feasible point in time in VRPTW

seen that the time window violation occurs only at user 15, where the begin time of service is shifted to the late time window of user 15. Due to time-shifting, no more violations occur in the route, and the overall time window violation of the route is 20. This leads to a significantly lower time window violation value, which is more realistic than the one computed in a classical way.

$$t'_{b_i} = \min(t_{b_i}, l_i) \quad (4.11)$$

This concept includes specially defined penalty variables and concatenation operators for the evaluation of the most basic moves in the VRPTW solution (relocate, exchange, Or-Opt, 2-Opt*, etc.) [152]. Schneider et al. [153] noted that some concatenation operators produce a wrong evaluation values in some specific situations, i.e., an infeasible solution has no time-window penalty; and improved the formulation to account for the correct time window violation in such occasions. In the EVRPTW problem, this concept has to be modified to include the charging time at CS. Schneider et al. [18] applied this concept on the EVRPTW-FR problem with specially defined concatenation operators for full recharge strategy. They concluded that the evaluation of two partial routes ϕ_1 and ϕ_2 can be done in two ways depending on the situation: (i) in $\mathcal{O}(1)$ if the second route does not contain CS, and (ii) in $\mathcal{O}(B)$ if the second route contains at least one CS, where B is the number of users between the last user in partial route ϕ_1 and last CS in partial route ϕ_2 (already explained in section 3.5). For partial recharge strategy, this concept cannot be used since the amount of fuel is no longer fixed when the user is evaluated. The time window and battery (fuel) violations became linearly dependent on the amount of fuel replenished at preceding CSs. Schiffer et al. [53] proposed a corridor-based penalty approach for computing both time window and fuel violations in $\mathcal{O}(1)$ for most of the basic move operators in the ELRPTW-PR problem. This corridor-based approach consists of specially defined forward and backward penalty variables, and a specially defined concatenation operators. In this thesis, the concept of the corridor-based approach is used for the evaluation of moves in different EVRP variants. Here, the difference between the corridor-based approach proposed in this thesis and the one proposed by the Schiffer et al. [53] are highlighted:

- (i) EVRPTW-PR - the original corridor-based approach by Schiffer et al. [53] for the ELRPTW-PR problem with evaluation $\mathcal{O}(1)$ is slightly modified to not include CS location placement problem and service times while charging,
- (ii) EVRPTW-FR - modified forward variables for the evaluation of the surrogate function in $\mathcal{O}(1)$ and later evaluation of best LS moves in $\mathcal{O}(n)$,
- (iii) EVRPTWDCS-FR - new penalty variables and concatenation operators, that use surrogate function for evaluation in $\mathcal{O}(1)$ (for both compatible and incompatible charger types), and later evaluation of best LS moves in $\mathcal{O}(n)$,
- (iv) EVRPTWDCS-PR

- (a) if partial routes contain compatible charger types (or none at all) - the penalty variables and concatenation operators for EVRPTW-PR are used and computed in $\mathcal{O}(1)$,
- (b) if partial route contains different charger types - new penalty variables and concatenation operators for EVRPTWDCS-PR- in practice between $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$.

In the following subsections, the penalty variables and concatenation operators for different problem variants are described.

4.3.1 EVRPTW-PR

The original corridor-based approach for ELRPTW-PR [53] lacks a detailed description of penalty variables and concatenation operators to fully understand the proposed concept. Here, a detailed description is provided for the EVRPTW-PR problem. The proposed variables and operators for the ELRPTW-PR problem are modified for the EVRPTW-PR problem, as CS placement and service time at CSs are removed.

The good idea of the corridor-based approach is to express all variables in a unit of time [55]. The energy consumed on a path is expressed as the time needed to recharge the consumed energy $h_{ij} = ge_{ij} = grd_{ij}$, as in the EVRPTW-PR problem the consumed energy depends linearly on the total distance traveled (section 2.4). The battery capacity Q is expressed as the time $H = gQ$ needed to fully charge the battery. The variables indicating travel back in time to the latest feasible point in time are labeled with tilde (\sim) and determined based on the shifting rules.

Forward variables

The forward variables for EVRPTW-PR are presented in Table 4.1. First the slack time at user j is computed by equation 4.12 as the time between the arrival time at user j with minimum charging amount and early time window e_j . The arrival time at user j is computed as the sum of shifted begin time at previous user with minimum charging amount \tilde{a}_i^{min} , service time s_i and the travel time t_{ij} .

Next, the maximum recharging time a_j^{rt} at user j is computed by equation 4.13. The assumption is that no recharging has been performed at previous CSs except the minimum charging amount needed to visit users before j . The minimum charging amount can be added to previous CSs only if there exists such a CS, and there is enough free amount (time) to charge the vehicle (vehicle cannot be charged over Q). If the preceding user i is a CS, then the maximum recharging time a_i^{rt} at user i is first reduced by computed slack time a_i^{sl} and then increased by the time h_{ij} needed for recharging the energy consumed on arc (i, j) . More precisely, the possible spare slack time at CS preceding user j is always used for recharging as this is the "spare" time that would otherwise be wasted. Therefore, the a_j^{rt} is the maximum charging time at user j with consideration that all possible slack times are used for recharging and that, if needed and possible,

Table 4.1: Forward variables - EVRPTW-PR

Symbol	Description
a_j^{min}	Earliest begin time at user j with charging minimum amount as possible before j
a_j^{max}	Earliest begin time at user j with charging maximum amount as possible at preceding CS with assumption that minimum charging was performed before preceding CS
a_j^{rt}	Time needed to charge to maximum at user j with assumption that minimum charging was performed before j
a_{ij}^{sl}	Slack time at user j
a_{ij}^{add}	Additional charging time at user j that has to be added at preceding CSs
\tilde{a}_j^{min}	Shifted earliest begin time at user j when violation occurs with charging minimum amount as possible before j
\tilde{a}_j^{max}	Shifted earliest begin time at user j with charging maximum amount as possible at preceding CS with an assumption that minimum charging was performed before preceding CS
$a_{j \rightarrow CS_{bef}}^{ct}$	Charging time at CS preceding user j
a_j^{cn}	Cumulative charging time needed for traversing the route
a_j^{cr}	Cumulative charging time at CSs
$a_{j \rightarrow CS_{bef}}$	An instance of CS preceding user j

the minimum charging amount has already been charged before. If the slack time a_{ij}^{sl} is greater than a_i^{rt} , the a_i^{rt} is limited to zero, as the vehicle cannot be charged over its capacity, although there is more spare time. The computed limited difference is increased by h_{ij} , meaning that the maximum recharging time is increased by the amount of time needed for recharging the energy consumed on arc (i, j) , but again limited by vehicle capacity Q , i.e., total vehicle recharge time H . If the preceding user i is not a CS, then the same principle is used, but the available slack time for recharging is limited by the spare time between minimum charging begin time \tilde{a}_j^{min} and maximum charging begin time \tilde{a}_j^{max} . This also means that if there is more spare time than there is slack time, the spare charging time is limited to the slack time. The variable a_j^{rt} at user which is not CS is used only for the propagation of the values, and the actual charging does not occur at such user.

Next, the additional recharging time a_{ij}^{add} that needs to be accounted for at previous CSs is computed by equation 4.14. The equation is almost the same as the equation for maximum recharging time a_j^{rt} (4.13). Instead of limiting the maximum recharging time to H , the assumption is to charge at the previous user as max as possible, and if the unlimited maximum recharging time is greater than H , then this difference is the minimum additional charging time that has to be added at preceding CSs to reach the user j without fuel violation.

The earliest begin time at user j with charging minimum amount as possible a_j^{min} is computed by equation 4.15. It is computed as the sum of the shifted begin time \tilde{a}_i^{min} at previous user i , travel time t_{ij} between users and the service time s_j ; and it is limited by the early time

window e_j , as the service cannot start before that time. Additionally, if there is not enough energy expressed in time to reach the user j even if at previous CS the maximum charging is performed, the minimum additional recharging time a_{ij}^{add} is added. This is valid, as all of the slack times up to j (including) are already used for recharging, but some additional recharging is still needed. As there are no non-linear spare time holes, the added charging time increases linearly the earliest begin time at user j . Adding a_{ij}^{add} does not necessarily mean that the fuel violation occurs. It only means that to escape the fuel violation, the a_{ij}^{add} recharging time has to be added at preceding CSs. But if there is no such possibility, the violation occurs, which will be accounted for, in the equation for computing fuel violation.

The earliest begin time a_j^{max} at user j with charging maximum amount as possible at preceding CS with the assumption that minimum charging was performed before preceding CS is computed by equation 4.16. If the user i is not CS then the shifted earliest begin time \tilde{a}_i^{max} at preceding user i , the travel time t_{ij} and the service time s_i are summed, and limited by the early time window e_j . Otherwise, if user i is CS, then instead of \tilde{a}_i^{max} the \tilde{a}_i^{min} is used which is the shifted earliest begin time at user i with charging minimum amount as possible, and the maximum charging time a_i^{rt} and travel time t_{ij} are added. Again the result is limited to the early time window e_j , if the vehicle has to wait for the opening of a time window.

$$a_{ij}^{sl} = \max(0, e_j - \tilde{a}_i^{min} - t_{ij} - s_i) \quad (4.12)$$

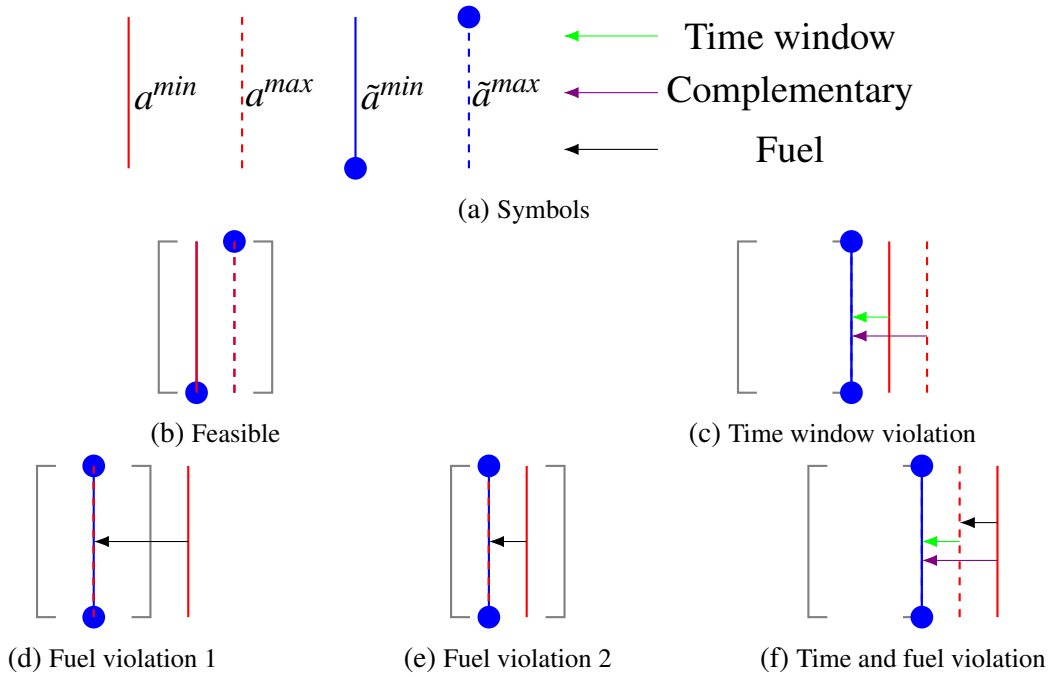
$$a_j^{rt} = \begin{cases} \min(H, \max(0, a_i^{rt} - a_{ij}^{sl}) + h_{ij}) & i \in F' \\ \min(H, \max(0, a_i^{rt} - \min(a_{ij}^{sl}, \tilde{a}_j^{max} - \tilde{a}_j^{min})) + h_{ij}) & else \end{cases} \quad (4.13)$$

$$a_{ij}^{add} = \begin{cases} \max(0, \max(0, a_i^{rt} - a_{ij}^{sl}) + h_{ij} - H) & i \in F' \\ \max(0, \max(0, a_i^{rt} - \min(a_{ij}^{sl}, \tilde{a}_j^{max} - \tilde{a}_j^{min})) + h_{ij} - H) & else \end{cases} \quad (4.14)$$

$$a_j^{min} = \max(e_j, \tilde{a}_i^{min} + t_{ij} + s_i) + a_{ij}^{add} \quad (4.15)$$

$$a_j^{max} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + a_i^{rt} + t_{ij}) & i \in F' \\ \max(e_j, \tilde{a}_i^{max} + t_{ij} + s_i) & else \end{cases} \quad (4.16)$$

To travel back in time to the latest feasible point when a violation occurs, the forward shifting rules are determined. The rules are presented in Figure 4.5. In total there are 5 shifting rules, but only 4 account for time and fuel violations (4.5c-4.5f). Figure 4.6a represents symbols used for a_{min} , a_{max} and their shifted counterparts \tilde{a}^{min} and \tilde{a}^{max} . Figure 4.5b represents feasible variables and no violation. Figure 4.5c represents time window violation (green arrow), when $a^{min} > l$, and a^{min} has to be shifted. The a^{max} is also shifted to the late time window


Figure 4.5: Forward shifting rules - EVRPTW-PR

- the complementary shift (magenta arrow). Figures 4.5d and 4.5e represent fuel violation when $a^{max} < a^{min}$ (black arrows). This happens when there is not enough fuel to reach a user. For example, in route without CSs, when there is not enough fuel the a^{add} value is additionally added to a^{min} (equation 4.15), while the variable a^{max} remains the same (without additional charging time). As a result condition $a^{max} < a^{min}$ is satisfied and a violation occurs. Figure 4.5f presents violation of both time window and fuel, as conditions $a^{min} > l$ and $a^{max} < a^{min}$ are satisfied. Also the complementary shift of a^{max} is presented. All of these shifting rules can be combined in two equations 4.17 and 4.18. The forward time window and fuel penalties of a route $\phi = (u_0, \dots, u_{N+1})$ are given by equations 4.19 and 4.20.

$$\tilde{a}_j^{min} = \min(a_j^{min}, a_j^{max}, l_j) \quad (4.17)$$

$$\tilde{a}_j^{max} = \min(l_j, a_j^{max}) \quad (4.18)$$

$$\vec{TW}(\phi) = \sum_{u \in \phi} \max(\min(a_u^{min}, a_u^{max}) - l_u, 0) \quad (4.19)$$

$$\vec{FL}(\phi) = \sum_{u \in \phi} \max(a_u^{min} - a_u^{max}, 0) \quad (4.20)$$

The proposed approach does not specifically determine how much charging is performed at each CS. The a_{ij}^{add} is computed as the additional recharging time needed at previous CSs, but

no scheduling is performed to determine how much to charge on each CS. To track the charging time at each CS, four new variables are proposed, which are not presented in the original corridor-based approach of Schiffer and Walter [53]. The $a_{j \rightarrow CS_{bef}}$ represents an instance of the latest CS before user j given by equation 4.21. The charging time $a_{j \rightarrow CS_{bef}}^{ct}$ at the latest CS before j is given by equation 4.22. The charging time at latest CS, if any, is computed as follows. If previous user i is a CS, the charging time is equal to the sum of slack time a_{ij}^{sl} limited by maximum charging amount a_i^{rt} and additional charging time a_{ij}^{add} . If the previous user i is not a CS, the charging time is equal to the sum of slack time a_{ij}^{sl} limited by the difference between minimum \tilde{a}_j^{min} and maximum shifted begin time \tilde{a}_j^{max} , additional charging time a_{ij}^{add} and previous charging time $a_{j \rightarrow CS_{bef}}^{ct}$. Last two variables represent cumulative route values: the cumulative recharging time at CSs a_j^{crt} (4.23) and a_j^{crtn} cumulative recharging time (energy, fuel) needed for traversing the route up to j (4.24). The charging time in CS is needed for determining the exact solution of the problem, with the whole routing plan and charging schedule or in cases where the objective function contains charging costs. The model assumes that whenever there is an available slack time, that it will be used for charging, but sometimes the feasibility can be achieved without considering all slack times for charging, which can reduce overall charging costs. Therefore, by limiting the total charging time to $\max(a_j^{crtn} - H, 0)$, which is the energy consumed on the entire route, the minimum amount of charging is performed in a route, and the vehicles ends the route with empty battery. This can be used to reduce the slack charging times and still to ensure feasibility. Initial values of forward variables at depot are all set to as $\tilde{a}_0^{min} = \tilde{a}_0^{max} = a_0^{min} = a_0^{max} = a_0^{rt} = a_0^{crt} = a_0^{crtn} = 0$, while the CS before depot is set as $a_{0 \rightarrow CS_{bef}}^{ct} = None$. The service time in all CSs ($i \in F'$) is set as $s_i = 0$, and therefore are excluded from equations related to CS only.

$$a_{j \rightarrow CS_{bef}} = \begin{cases} None & i = 0 \\ i & i \in F' \\ a_{i \rightarrow CS_{bef}} & else \end{cases} \quad (4.21)$$

$$a_{j \rightarrow CS_{bef}}^{ct} = \begin{cases} 0 & a_{j \rightarrow CS_{bef}} \text{ is } None \\ \min(a_{ij}^{sl}, a_i^{rt}) + a_{ij}^{add} & i \in F' \\ a_{j \rightarrow CS_{bef}}^{ct} + \min(a_{ij}^{sl}, \tilde{a}_j^{max} - \tilde{a}_j^{min}) + a_{ij}^{add} & else \end{cases} \quad (4.22)$$

$$a_j^{crt} = \begin{cases} a_i^{crt} + \min(a_{ij}^{sl}, a_i^{rt}) + a_{ij}^{add} & i \in F' \\ a_i^{crt} + \min(a_{ij}^{sl}, \tilde{a}_j^{max} - \tilde{a}_j^{min}) + a_{ij}^{add} & else \end{cases} \quad (4.23)$$

$$a_j^{crtn} = a_i^{crtn} + h_{ij} \quad (4.24)$$

Backward variables

The used corridor-based approach, besides the forward variables, also contains backward variables, which enable efficient penalty computation. The used backward variables are presented in Table 4.2 and are computed in a similar way as the forward variables. The minimum functions are replaced by maximum functions, and e_i is swapped by l_i . The main idea is to propagate variables backward, from the latest user in route (from depot late time window) to the first user in route, i.e., from user j to user i . The initial values for backward variables related to the time of the last user in route u_{N+1} are set to the late time window, $\tilde{b}_{N+1}^{min} = \tilde{b}_{N+1}^{max} = b_{N+1}^{min} = b_{N+1}^{max} = l_{N+1}$, while the fuel related variable is set to zero, $b_{N+1}^{rt} = 0$. The latest begin time b_i^{min} at user i with charging minimum amount as possible before i is computed by equation 4.28. The shifted latest begin time at user j , \tilde{b}_j^{min} is decreased by the travel time t_{ij} (forward travel time direction $i \rightarrow j$) and service time s_i at user i . Further on, it is limited by the late time window l_i as the latest arrival time cannot be after the late time window. Following the same idea as in forward variables, the additional recharging time is computed by equation 4.27 and is subtracted from the previous result. The value b_i^{min} represents the latest point in time by which the service has to start without the violation of constraints of all the users following user i in a forward way (towards the route end). The variables b_i^{rt} and b_i^{add} are computed in almost the same way as the forward variables (equations 4.26 and 4.27), with only difference being the order of subtraction $\tilde{b}_j^{min} - \tilde{b}_j^{max}$, as in backward manner \tilde{b}_j^{min} generally has a greater time value (later point in time) than \tilde{b}_j^{max} . It is important to note that these variables related to charging amount are computed in backward fashion considering that at the beginning (last user in route) the vehicle is charged to a full capacity Q as initial value of time needed to charge to maximum is set to zero, $b_{N+1}^{rt} = 0$. The b_{ij}^{sl} slack time is the amount of spare time up to the late time window l_i that can be spent on recharging (equation 4.25). The b_i^{max} is computed similar as the forward counterpart, given by equation 4.29. The backward shifting rules are presented in Figure 4.6, while the equations are given by 4.30 and 4.31. The time window violation (Figure 4.6c, green arrow) occurs when b^{min} value is lower than the early time window, and the b^{max} is complementary shifted to the early time window (magenta arrow). The fuel violation occurs when $b^{min} < b^{max}$, as in figures 4.6d and 4.6e. The case when both time window and fuel violation occur is presented in Figure 4.6f. The backward time window and fuel violations are given by equations 4.32 and 4.33.

$$b_{ij}^{sl} = \max(0, \tilde{b}_j^{min} - t_{ij} - s_i - l_i) \quad (4.25)$$

$$b_i^{rt} = \begin{cases} \min(H, \max(0, b_j^{rt} - b_{ij}^{sl}) + h_{ij}) & j \in F' \\ \min(H, \max(0, b_j^{rt} - \min(b_{ij}^{sl}, \tilde{b}_j^{min} - \tilde{b}_j^{max})) + h_{ij}) & else \end{cases} \quad (4.26)$$

Table 4.2: Backward variables - EVRPTW-PR

Symbol	Description
b_i^{min}	Latest begin time at user i with charging minimum amount as possible before i
b_i^{max}	Latest begin time at user i with charging maximum amount as possible at preceding CSs with the assumption that minimum charging was performed before preceding CS
b_i^{rt}	Time needed to charge to maximum at user i with charging minimum amount as possible before i
b_{ij}^{sl}	Slack time at user i
b_{ij}^{add}	Additional charging time at user i that has to be added at preceding CSs
\tilde{b}_i^{min}	Shifted latest begin time at user i when violation occurs with charging minimum amount as possible before i
\tilde{b}_i^{max}	Shifted latest begin time at user i when violation occurs with charging maximum amount as possible at preceding CS with assumption that minimum charging was performed before preceding CS

$$b_{ij}^{add} = \begin{cases} \max(0, \max(0, b_j^{rt} - b_{ij}^{sl}) + h_{ij} - H) & j \in F' \\ \max(0, \max(0, b_j^{rt} - \min(b_{ij}^{sl}, \tilde{b}_j^{min} - \tilde{b}_j^{max})) + h_{ij} - H) & else \end{cases} \quad (4.27)$$

$$b_i^{min} = \min(l_i, \tilde{b}_j^{min} - t_{ij} - s_i) - b_{ij}^{add} \quad (4.28)$$

$$b_i^{max} = \begin{cases} \min(l_i, \tilde{b}_j^{min} - b_j^{rt} - t_{ij} - s_i) & j \in F' \\ \min(l_i, \tilde{b}_j^{max} - t_{ij} - s_i) & else \end{cases} \quad (4.29)$$

$$\tilde{b}_i^{min} = \max(b_i^{min}, b_i^{max}, e_i) \quad (4.30)$$

$$\tilde{b}_j^{max} = \max(e_j, b_j^{max}) \quad (4.31)$$

$$\overleftarrow{TW}(\phi) = \sum_{u \in \phi} \max(e_u - \max(b_u^{min}, b_u^{max}), 0) \quad (4.32)$$

$$\overleftarrow{FL}(\phi) = \sum_{u \in \phi} \max(b_u^{max} - b_u^{min}, 0) \quad (4.33)$$

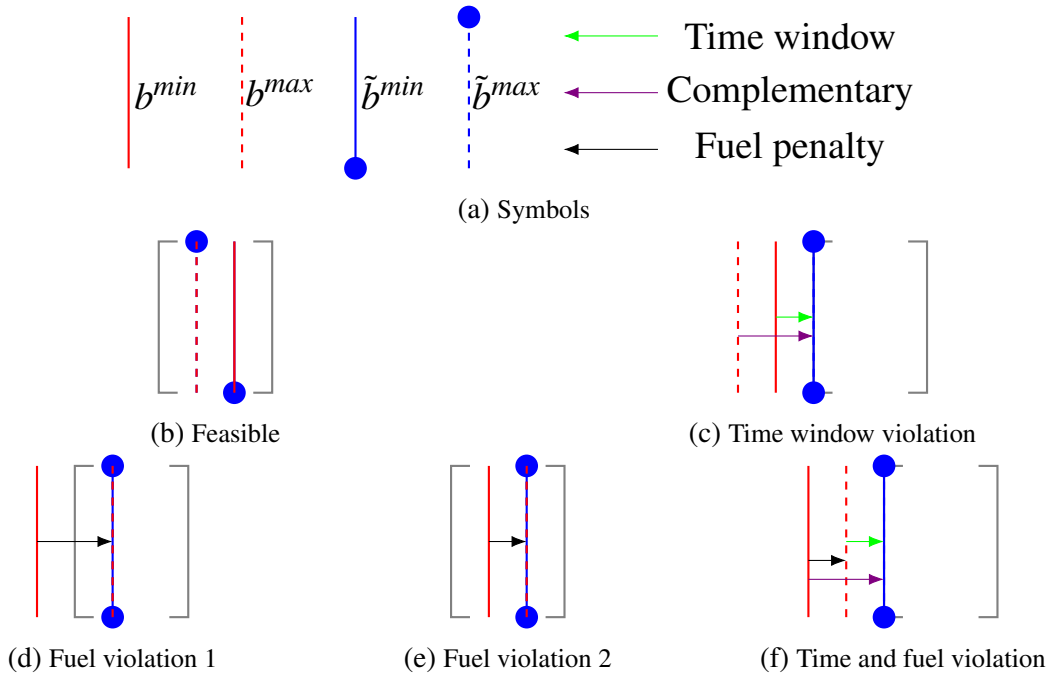


Figure 4.6: Backward shifting rules - EVRPTW-PR

Concatenation operators

In the following subsection, both forward and backward penalties are used to define equations representing concatenation operators. Most of the basic operators can be expressed as a concatenation between two partial routes $\phi_1 = (u_0, u_1, \dots, x)$ and $\phi_2 = (y, \dots, u_{N+1})$, and the result is route $\phi = \phi_1 \otimes \phi_2 = (u_0, u_1, \dots, x, y, \dots, u_{N+1})$. First, the forward variables are propagated to user y , based on the forward variables of user x . Then, the time window computation of concatenated routes is computed by equation 4.34 as the sum of forward time window violation $\overrightarrow{TW}(\phi_1)$ of partial route ϕ_1 , backward time window violation $\overleftarrow{TW}(\phi_2)$ of partial route ϕ_2 , and additional two parts, (1) and (2). The first one is the forward time window violation at user y (bracket 1 in equation 4.34) consisting of time window violation due to the late arrival at user y (bracket 1.1 in equation 4.34) which is reduced by the over-penalization value equal to the forward fuel penalty value at user y (bracket 1.2 in equation 4.34). The second one is the violation of the latest arrival time at user y (bracket 2 in equation 4.34) computed as the difference between earliest feasible arrival time (bracket 2.1 in equation 4.34) and latest arrival time with minimum charging amount b_y^{min} , further reduced by the over-penalization value equal to the backward fuel penalty at user y (bracket 2.2 in equation 4.34). As it can be seen, the general idea is to compute the time window violation in a typical way, but to reduce it by both forward and backward fuel violations, to escape the time window over-penalization. The fuel violation is computed by equation 4.35 as the sum of forward fuel violation $\overrightarrow{FL}(\phi_1)$ of route ϕ_1 , backward fuel violation $\overleftarrow{FL}(\phi_2)$ of route ϕ_2 , forward fuel violation at user y and additional value D computed by equation 4.36. The value D represents fuel penalty if a complete route

$\phi = \phi_1 \otimes \phi_2$ exceeds overall fuel capacity.

$$\begin{aligned}
 TW(\phi_1 \otimes \phi_2) = & \overrightarrow{TW}(\phi_1) + \overleftarrow{TW}(\phi_2) + \overbrace{\max(0, a_y^{min} - l_y - \max(0, a_y^{min} - a_y^{max}))}^{(1)} + \\
 & \overbrace{\max(0, \min(l_y, \max(e_y, a_y^{min})) - b_y^{min} - \max(b_y^{max} - b_y^{min}, 0))}^{(2)} \quad (4.34)
 \end{aligned}$$

$$FL(\phi_1 \otimes \phi_2) = \overrightarrow{FL}(\phi_1) + \overleftarrow{FL}(\phi_2) + \max(0, a_y^{min} - a_y^{max}) + D \quad (4.35)$$

$$D = \begin{cases} \overbrace{\max(0, a_y^{rt} + b_y^{rt} - H -}^{(3)} \\ \underbrace{\min(H, \min(\underbrace{\max(0, b_y^{min} - a_y^{min}}_{(5.1)}, \underbrace{\max(0, a_y^{max} - a_y^{min}}_{(5.2)} + \max(0, b_y^{min} - b_y^{max})))}_{(5)}}_{y \notin F'} \\ \overbrace{\max(0, a_y^{rt} + b_y^{rt} - H -}^{(6)} \\ \underbrace{\min(a_y^{rt}, \max(0, \underbrace{b_y^{min} - a_y^{min} - \max(0, a_y^{max} - b_y^{max}}_{(6.1)}, \underbrace{\max(0, a_y^{min} - b_y^{min}}_{(6.2)}))}_{(6)}}_{else} \end{cases} \quad (4.36)$$

The explanation of the computation of a D value is first provided for the case when the user y is not a CS. The bracket 3 (in equation 4.36) represents the maximum charging time at user y if previously all forward and backward slack times and minimum amounts of charging are considered for charging. The value in bracket 3 (in equation 4.36) could be larger than H , meaning that a fuel violation of the overall route can occur. This difference is further reduced by possible spare charging time (bracket 5 in equation 4.36) at user y to not over-penalize the fuel violation, as the goal of the approach is to use all spare time for charging. The spare time for charging is computed as the sum of the time between a_y^{max} and a_y^{min} and the time between b_y^{min} and b_y^{max} (bracket 5.2 in equation 4.36). The spare time values are further limited by the total spare time between the latest minimum begin time b_y^{min} and the earliest minimum begin a_y^{min} time at user y (bracket 5.1 in equation 4.36). The computed spare time cannot be larger than H if user y is not CS, and the overall result must be larger than zero; otherwise, no penalty occurs. If user y is CS, then the spare time (bracket 6 in equation 4.36) that can be used for charging cannot be larger than the maximum charging time a_y^{rt} or the time violation between

minimum charging $a_y^{min} - b_y^{min}$ (bracket 6.2 in equation 4.36). The spare value (bracket 6.1 in equation 4.36) is computed as spare time between b_y^{min} and a_y^{min} reduced by the time violation between maximum forward and backward charging $a_y^{max} - b_y^{max}$.

It is important to note that the time window and fuel violations are not exact, but rather an approximation of their values, as they can differ from values if the whole route is evaluated exactly. But the sum of approximated fuel and time window penalties are equal to the sum of exactly evaluated values, which means that it is hard to determine which penalties are caused by fuel violation and which by time window violation, because the fuel penalties also affect the time window penalties.

4.3.2 EVRPTW-FR

Instead of partial charging at CS, the EVRPTW-FR problem considers full recharge at each CS. For EVRPTW-FR, a similar corridor-based approach as in EVRPTW-PR is applied. All the variable names remain the same as for the EVRPTW-PR problem but have a slightly different description, presented in Table 4.3. The forward variables are computed by equations 4.37-4.42. Each time when a CS is visited, the full recharge is considered, i.e., the whole value of a_j^r is used. The a_{ij}^{add} variable accounts for the additional recharging time, which here directly corresponds to the fuel violation. The slack times are always zero, as there is no spare time to charge the vehicle further than its capacity Q . The forward shifting rules, time window penalties, and fuel penalties remain the same as in EVRPTW-PR, as well as all backward variables, shifting rules, and penalties. The backward variables remain the same because the solution to full recharge strategy in second partial route ϕ_2 , considers backward charging somewhere between minimum and maximum charging $[b_{min}^y, b_{max}^y]$. In EVRPTW-FR, the fuel violation occurs whenever the $a_{ij}^{add} > 0$, and it is equal to the difference between a_j^{min} and a_j^{max} .

Table 4.3: Forward variables - EVRPTW-FR

Symbol	Description
a_j^{min}	Earliest begin time at user j with charging maximum as possible before j plus the additional infeasible charging time
a_j^{max}	Earliest begin time at user j with charging maximum as possible before j
a_j^r	Time needed to charge to maximum at user j
a_{ij}^{add}	Additional charging time at user j that has to be added at preceding CSs
\tilde{a}_j^{min}	Shifted earliest begin time at user j with charging maximum as possible before j plus the additional infeasible charging time
\tilde{a}_j^{max}	Shifted earliest begin time at user j with charging maximum as possible before j
a_j^{crtn}	Cumulative charging time needed for traversing the route
a_j^{crt}	Cumulative charging time at CSs

$$a_{ij}^{sl} = 0 \quad (4.37)$$

$$a_j^{rt} = \begin{cases} \min(H, h_{ij}) & i \in F' \\ \min(H, a_i^{rt} + h_{ij}) & \text{else} \end{cases} \quad (4.38)$$

$$a_{ij}^{add} = \begin{cases} \max(0, h_{ij} - H) & i \in F' \\ \max(0, a_i^{rt} + h_{ij} - H) & \text{else} \end{cases} \quad (4.39)$$

$$a_j^{min} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + a_i^{rt} + t_{ij}) + a_{ij}^{add} & i \in F' \\ \max(e_j, \tilde{a}_i^{min} + t_{ij} + s_i) + a_{ij}^{add} & \text{else} \end{cases} \quad (4.40)$$

$$a_j^{max} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + a_i^{rt} + t_{ij}) & i \in F' \\ \max(e_j, \tilde{a}_i^{min} + t_{ij} + s_i) & \text{else} \end{cases} \quad (4.41)$$

$$a_j^{crt} = \begin{cases} a_i^{crt} + a_i^{rt} & i \in F' \\ a_i^{crt} & \text{else} \end{cases} \quad (4.42)$$

In EVRPTW-FR, most of the basic neighborhood operators can be evaluated in $\mathcal{O}(1)$. However, in some special circumstances, the approximated violations are not good and underestimate penalties. Specifically, these are the cases in which there is a CS in the second partial route ϕ_2 . In these cases, it is not possible to precisely estimate and propagate backward slack times for all special cases, although for most of the cases, the penalties give a good approximation. Schneider et al. [18] in such cases used tabu list and prevented LS from searching in that space. This is still not the solution because some good solutions can come from such a solution space. One other solution is to proceed with forward propagation until the last CS, which in the worst case has $\mathcal{O}(n)$ complexity [53]. Here, a different approach is used. All of the moves are approximated with such surrogate function and are considered valid in the LS phase. During the search, the n_{RCL}^{ls} best LS moves are stored in the restricted list. At the end of the LS operator, all the moves in the restricted list are evaluated exactly with the complexity of $\mathcal{O}(n)$, and the best one is selected. It is important to highlight that the proposed corridor-based variables and concatenation operators have not yet been applied in the literature to solve the EVRPTW-FR problem.

4.3.3 EVRPTWDCS-FR

The EVRPTWDCS-FR problem is an extension of EVRPTW-FR problem in which at each CS different charger types can be used. The EVRPTWDCS-FR problem has not yet been addressed in the literature, and here the variables and concatenation operators are provided to solve the problem efficiently. The EVRPTW-FR variables and concatenation operators cannot be used in the EVRPTWDCS-FR because the time to charge the vehicle to the fullest $H = gQ$, is not fixed, and it depends on the used charger type and starting SoC. For each user, four different variable sets are stored: the EVRPTW-FR forward and backward variables for each charger type (three different chargers) and new forward and backward variables for EVRPTWDCS-FR. The first three variable sets are computed under the assumption that the whole route is charged with the same charger type, while the last set considers different charger types. Some variables for EVRPTWDCS-FR are similar to the variables used in EVRPTW-FR but are differently computed, and some are the same. The variables related to fuel, a_j^{rt} (b_j^{rt}) and a_{ij}^{add} (b_{ij}^{add}), are expressed in the capacity (energy) unit. Forward and backward variables, which are different, are presented in Table 4.4. Forward variables are given by equations 4.43-4.50, and backward variables are given by equations 4.51-4.57. The equations for forward and backward time window violation and shifting rules are the same as in EVRPTW-FR problem. To fully assess the different charger types in the solution, the objective function cannot be purely distance or travel time based, as in such case the rapid charging is always selected as the best option. The objective function should include costs for charging with different charger types, as for example the objective function given by equation 2.23 (section 2.5). The initial values of the additional variables are set as $a_j^{rt} = b_j^{rt} = a_{ij}^{add} = b_{ij}^{add} = a_j^{rec} = a_j^m = b_j^m = 0$ and $m_j^{min} = None$.

As capacity unit is used for fuel (energy) consumption, in the equation for a_j^{min} (equation 4.46) the capacity needed to charge to maximum a_j^{rt} is multiplied with recharging rate g^{m_i} where $m \in \{1, 2, 3\}$; and $m = 1$ represents rapid charging technology, $m = 2$ fast charging technology and $m = 3$ slow charging technology. The problem occurs where there is not enough energy left ($a_{ij}^{add} > 0$), and fuel violation occurs. This value has to be expressed in unit of time and added to the variable a_j^{min} . The time value depends on the CSs used in preceding part of the route. If only one CS is used, the $g^{m_j^{min}}$ is set to the used charger type of that CS. But if multiple CSs are

Table 4.4: New forward and backward variables - EVRPTWDCS-FR

Symbol	Description
a_j^{rt} (b_j^{rt})	Capacity needed to charge to maximum at user j
a_{ij}^{add} (b_{ij}^{add})	Additional charging capacity at user j that has to be added at preceding CSs
a_j^m (b_j^m)	Charger types used up to user j
m_j^{min}	Fastest charger type used up to user j
a_j^{rec}	Cumulative charging costs at CSs

used the $g_j^{m^{min}}$ is set to the fastest charger type used (equation 4.45) to not over-penalize fuel violation. If CS is not present in preceding part of the route, the rapid charger is set as $m = 1$. This is the main reason why the fuel violation is computed by equation 4.50, as exact capacity violation, instead of corresponding time violation. The variable $a_j^m \in \{0, 1, 2, 3, 4\}$ computed by equation 4.49 is added to determine the partial route CS type. The value of 0 represents that there is no CS before, the value 4 that there are CSs with different charger types, and values 1, 2 and 3 specify that only that type of CS is present in the partial route before user j . Time window violations are computed in a same way as in EVRPTW-PR problem (equation 4.19).

For concatenation operators, the backward variables (equations 4.51-4.54) also have to be computed but only up to the latest CS in route (first CS in backward direction), except the $b_i^m = \{0, 1, 2, 3, 4\}$ (equation 4.53) which has to be computed for whole route, as it is used in concatenation operators to determine which set to use. The g^1 recharge rate is used in the equation 4.53 as there are not any CS in the observed backward partial route (up to the latest CS) and again to reduce the over-penalization.

$$a_j^{rt} = \begin{cases} \min(Q, e_{ij}) & i \in F' \\ \min(Q, a_i^{rt} + e_{ij}) & else \end{cases} \quad (4.43)$$

$$a_{ij}^{add} = \begin{cases} \max(0, e_{ij} - Q) & i \in F' \\ \max(0, a_i^{rt} + e_{ij} - Q) & else \end{cases} \quad (4.44)$$

$$m_j^{min} = \begin{cases} \min(m_i^{min}, m_i) & i \in F' \\ m_i^{min} & else \end{cases} \quad (4.45)$$

$$a_j^{min} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + g^{m_i} a_i^{rt} + t_{ij}) + g_j^{m_j^{min}} a_{ij}^{add} & i \in F' \\ \max(e_j, \tilde{a}_i^{min} + t_{ij} + s_i) + g_j^{m_j^{min}} a_{ij}^{add} & else \end{cases} \quad (4.46)$$

$$a_j^{max} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + g^{m_i} a_i^{rt} + t_{ij}) & i \in F' \\ \max(e_j, \tilde{a}_i^{min} + t_{ij} + s_i) & else \end{cases} \quad (4.47)$$

$$a_j^{rec} = \begin{cases} a_i^{rec} + c^{m_i} a_i^{rt} & i \in F' \\ a_i^{rec} & else \end{cases} \quad (4.48)$$

$$a_j^m = \begin{cases} a_i^m & j \notin F' \\ m_j & a_i^m == 0 \text{ or } a_i^m == m_j \\ 4 & else \end{cases} \quad (4.49)$$

$$\overrightarrow{FL}(\phi) = \sum_{u \in \phi} a_{uu}^{add} \quad (4.50)$$

$$b_i^{rt} = \min(Q, b_j^{rt} + e_{ij}) \quad (4.51)$$

$$b_{ij}^{add} = \max(0, b_j^{rt} + e_{ij} - Q) \quad (4.52)$$

$$b_i^{min} = \min(l_i, \tilde{b}_j^{min} - t_{ij} - s_i) - g^1 b_{ij}^{add} \quad (4.53)$$

$$b_j^{max} = \min(l_i, \tilde{b}_j^{max} - t_{ij} - s_i) \quad (4.54)$$

$$b_i^m = \begin{cases} b_j^m & i \notin F' \\ m_i & b_j^m == 0 \text{ or } b_j^m == m_i \\ 4 & else \end{cases} \quad (4.55)$$

$$\overleftarrow{TW}(\phi) = \sum_{u \in \phi} \max(e_u - b_u^{min}, 0) \quad (4.56)$$

$$\overleftarrow{FL}(\phi) = \sum_{u \in \phi} b_{uj}^{add} \quad (4.57)$$

Concatenation operators

The concatenation is again performed between two partial routes $\phi_1 = (u_0, u_1, \dots, x)$ and $\phi_2 = (y, \dots, u_{N+1})$, and the goal is to efficiently evaluate the concatenation. Depending on the CS chargers used in the partial routes ϕ_1 and ϕ_2 (variables a_x^m and b_y^m) two evaluation types are present:

- (i) expression 4.58 - the used chargers in the first partial route ϕ_1 are all of the same type and are equal to the charger types used in the second partial route ϕ_2 in which the corresponding user variable sets are used, and the evaluation is performed by EVRPTW-FR concatenation operators in $\mathcal{O}(1)$ (this also covers if some or both partial routes do not contain any CS) and later after the LS operator, the evaluation of best stored moves is performed in $\mathcal{O}(n)$,

$$(a_x^m \neq 4 \wedge b_y^m \neq 4) \wedge (a_x^m == b_y^m \vee (a_x^m == 0 \vee b_y^m == 0)), \quad (4.58)$$

- (ii) expression 4.59 - the used chargers in the partial route ϕ_1 and ϕ_2 are different, in which case the proposed EVRPTWDCS-FR variables are used for the evaluation in $\mathcal{O}(B)$, where B is the number of users between last user in partial route ϕ_1 and the latest CS in partial route ϕ_2 ,

$$a_x^m == 4 \vee b_y^m == 4 \vee (a_x^m \neq b_y^m \wedge (a_x^m \neq 0 \wedge b_y^m \neq 0)). \quad (4.59)$$

The time window and fuel concatenation operators in this case are given by the equations 4.61 and 4.60, where LCS is the latest CS in partial route ϕ_2 . It is important to note that the variables are propagated up to the latest CS, where the exact violations are computed.

$$TW(\phi_1 \otimes \phi_2) = \overrightarrow{TW}(LCS) + \overleftarrow{TW}(LCS) + \max(a_{LCS}^{min} - b_{LCS}^{min}, 0) \quad (4.60)$$

$$FL(\phi_1 \otimes \phi_2) = \overrightarrow{FL}(LCS) + \overleftarrow{FL}(LCS) + a_{i,LCS}^{add} \quad (4.61)$$

4.3.4 EVRPTWDCS-PR

The EVRPTWDCS-PR is an extension of EVRPTW-PR problem, where different charger types at CSs are considered. This problem variant, with linear charging at CSs has only been addressed by Keskin et al. [22]. The authors proposed a solution method to solve the problem with searching only in the feasible solution space but did not describe the variables used to determine the overall route values (the same goes for the EVRPTW-PR by Keskin et al. [1]). This problem of partial recharging is hard to solve without the use of a corridor-based approach. More precisely, the evaluation is hard to solve. In EVRPTW-FR problem, in the worst case, the evaluation can be performed in $\mathcal{O}(n)$, as at each CS the exact charging amount is known. In EVRPTW-PR problem, this is not the case because additional (best possible) charging schedule has to be determined to evaluate the route, although the total route consumption is known. This charge scheduling is quite demanding in both EVRPTW-PR and EVRPTWDCS-PR. This problem can be defined as Fixed-Route Vehicle-Charging Problem (FRVCP) as a variant of the Fixed-Route Vehicle-Refueling Problem (FRVRP) [83]. The FRVRP seeks the minimum-cost refueling policy (which fuel stations to visit and the refueling amount at each station) for a given origin-destination route. Since the FRVRP is NP-hard [154], the FRVCP is also NP-hard. Instead of the exact determination of charging decisions in the evaluation part, here the heuristic approach is proposed to determine these charging decisions, and later on in the improvement part (section 4.7) the exact procedure will be applied to further improve the CSs positions and the charging amount.

Similar as in EVRPTWDCS-FR problem, in total four variable sets are used: three when charger types are compatible and one when they are not. When chargers are compatible, the EVRPTW-PR variables are used that correspond to the specific charger type used. In this case,

the complexity of evaluation is $\mathcal{O}(1)$. The compatibility of chargers between partial routes is determined in the same way as in EVRPTWDCS-FR problem. As already mentioned in EVRPTWDCS-FR problem, the previously described variables and concatenation operators for EVRPTW-PR cannot be used for the EVRPTWDCS-PR problem when chargers are incompatible. The variables used in such case are similar to the variables used in EVRPTW-PR and EVRPTWDCS-FR problems but are differently computed. Only the variables that differ from the variables in EVRPTW-PR and EVRPTWDCS-FR are presented in Table 4.5. The variables related to fuel a_j^{rt} , a_{ij}^{add} and a_j^{cc} are expressed in the unit of capacity while the computation of variables is given by equations 4.62-4.73. The charging costs are added to diversify the solution with different charger types. If recharging costs are not included, then always the fastest charging option would be selected as it decreases overall charging times.

The slack times are computed in a similar way as in EVRPTW-PR given by equation 4.62. If the preceding user is a CS, the value a_i^{cc} that represents the charging amount at CS i is multiplied by recharge rate g^{m_i} , to get the charging time. Already performed charging in CS is not considered as slack time. Rest of the equation follows already described procedure for computation of slack time. In the beginning, all a_i^{cc} values are set to zero, and all slack times can be used for charging. The charging amounts are computed only for CSs and are not propagated to customers. That is why an instance of the latest CS before user j , CS_{bef}^j , is used, for which the charging capacity (the amount that will be recharged) is computed by equation 4.63. If preceding user i is CS ($CS_{bef}^j = i$) then the slack is expressed in capacity unit based on the used charger m_i , and it cannot be larger than the difference between the maximum possible charging amount a_i^{rt} and already determined charging amount a_i^{cc} at CS i . This can be confusing, and one could ask, "Isn't the maximum possible charging amount already reduced by determined charging amount?". The answer is no, because as in EVRPTW-PR variables, the actual update on CS charging time (amount) happens on the first user after CS. If preceding user i is not a CS

Table 4.5: Forward variables - EVRPTWDCS-PR

Symbol	Description
a_j^{rt}	Capacity needed to charge to maximum at user j with assumption that minimum charging was performed before j
a_{ij}^{sl}	Slack time at user j
a_{ij}^{add}	Additional charging amount at user j that has to be added at preceding CSs
a_{ij}^{addt}	Additional charging time at user j that has to be added at preceding CSs
a_{ij}^{addrec}	Additional charging cost at user j that has to be added at preceding CSs
a_j^{cc}	Charging amount at CS j
a_j^{rec}	Cumulative charging costs at CSs
m_j^{min}	Fastest CS charger type used before user j
CS_{bef}^j	Instance of latest CS before user j

($CS_{bef}^j \neq i$) and is not *None*, then the possible slack time can be used for charging at CS CS_{bef}^j . The possible slack capacity, which in this situation is limited by the difference between $\tilde{a}_i^{max} - \tilde{a}_i^{min}$ expressed in capacity units and the difference between the maximum possible charging amount $a_{CS_{bef}^j}^{rt}$ and already determined charging amount $a_{CS_{bef}^j}^c$ at the latest CS. In this case, the charging amount $a_{CS_{bef}^j}^{cc}$ is increased by computed slack capacity. If there is no CS in route the variable value is propagated to next user.

$$a_{ij}^{sl} = \begin{cases} \max(0, e_j - \tilde{a}_i^{min} - t_{ij} - g^{m_i} a_j^{cc}) & i \in F' \\ \max(0, e_j - \tilde{a}_i^{min} - t_{ij} - s_i) & else \end{cases} \quad (4.62)$$

Next, the maximum charging amount a^{rt} is computed by equation 4.64. If preceding user i is a CS, then a_i^{rt} is reduced by already determined slack charging amount, and increased by the capacity corresponding to the energy consumption e_{ij} . If preceding user i is not a CS, and there has been at least one CS before j , then the a_i^{rt} is reduced by the slack capacity that could be added to the latest CS before j , and increased by energy consumption e_{ij} . If preceding user i is not CS, and there has not been at least one CS before j , then the a_i^{rt} is increased by the energy consumption e_{ij} .

$$a_{CS_{bef}^j}^{cc} = \begin{cases} a_i^{cc} + \min\left(\frac{a_{ij}^{sl}}{g^{m_i}}, \max(a_i^{rt} - a_i^{cc}, 0)\right) & i \in F' \\ a_{CS_{bef}^j}^{cc} + \min\left(\frac{\min(a_{ij}^{sl}, \tilde{a}_i^{max} - \tilde{a}_i^{min})}{g_{CS_{bef}^j}^m}, \max(a_{CS_{bef}^j}^{rt} - a_{CS_{bef}^j}^{cc})\right) & CS_{bef}^j \neq None \\ a_{CS_{bef}^j}^{cc} & else \end{cases} \quad (4.63)$$

Using the same principle the charging amount a_{ij}^{add} that has to be accounted for at previous CSs is computed by equation 4.65. This is where the thing get more complex than in EVRPTW-PR. In EVRPTW-PR, the variable a_{ij}^{add} expressed in time could be added, knowing that it will linearly increase the overall earliest begin time due to the: (i) all slack times are used for charging, and any further increase in time linearly increases arrival times (in case when all slack times has been used for recharging the increase in time can be non-linear due to the time windows), and (ii) the exact value of additional time can easily be computed as all CSs have same charger type. In EVRPTWDCS-PR, this time cannot be easily computed if different CS types are present in a partial route before j , because scheduling of this additional charging amount per CS before j has to be determined. Here, the following heuristic procedure is applied in case when $a_{ij}^{add} > 0$. If there is some space to charge to the fullest at previous CS, use all of that possible charging amount to cover the a_{ij}^{add} amount, and store the result in the variable $a_{ij}^{add'}$ (equation 4.66). Then, the additional charging amount a_{ij}^{add} is reduced by this amount (equation

4.67), and the charging amount at latest CS $a_{CS_{bef}^j}^{cc}$ is increased by this amount (equation 4.68).

At this point, all of the forward variables from the latest CS, CS_{bef}^j , to user j have to be recomputed. However, there is also a possibility that there is not enough charging amount at CS_{bef}^j to cover the a_{ij}^{add} amount or to cover it partially. In such cases, the additional time a_{ij}^{addt} and additional cost a_{ij}^{addco} are computed by equations 4.70 and 4.71. In cases where there is a CS in route before user j , the charger type m_j^{min} is determined by equation 4.69, and used for the computation of additional time and cost. If there is not a CS in route before user j the additional time and costs are computed using rapid charger $m = 1$. In both cases, this is the approximation of time and cost value because the exact time and cost are unknown. Using the fastest possible charger in such occasions makes more sense as faster charging increases feasibility and reduces over-penalization.

$$a_j^{rt} = \begin{cases} \min(Q, \max(0, a_i^{rt} - a_i^{cc}) + e_{ij}) & i \in F' \\ \min\left(Q, \max\left(0, a_i^{rt} - \frac{\min(a_{ij}^{sl}, \bar{a}_i^{max} - \bar{a}_i^{min})}{g_{CS_{bef}^j}}\right) + e_{ij}\right) & a_{CS_{bef}^j} \neq None \\ \min(Q, a_i^{rt} + e_{ij}) & else \end{cases} \quad (4.64)$$

$$a_{ij}^{add} = \begin{cases} \max(0, \max(0, a_i^{rt} - a_i^{cc}) + e_{ij} - Q) & i \in F' \\ \max\left(0, \max\left(0, a_i^{rt} - \frac{\min(a_{ij}^{sl}, \bar{a}_i^{max} - \bar{a}_i^{min})}{g_{CS_{bef}^j}}\right) + e_{ij} - Q\right) & a_{j \rightarrow CS_{bef}^j} \neq None \\ \max(0, a_i^{rt} + e_{ij} - Q) & else \end{cases} \quad (4.65)$$

$$a_{ij}^{add'} = \begin{cases} \min(a_{ij}^{add}, \max(0, a_{j \rightarrow CS_{bef}^j}^{rt} - a_{CS_{bef}^j}^{cc})) & CS_{bef}^j \neq None \\ 0 & else \end{cases} \quad (4.66)$$

$$a_{ij}^{add} = a_{ij}^{add} - a_{ij}^{add'} \quad (4.67)$$

$$a_{j \rightarrow CS_{bef}^j}^{cc} = a_{j \rightarrow CS_{bef}^j}^{cc} + a_{ij}^{add'} \quad (4.68)$$

$$m_j^{min} = \begin{cases} \min(m_i^{min}, m_i) & i \in F' \\ m_i^{min} & else \end{cases} \quad (4.69)$$

$$a_{ij}^{addt} = \begin{cases} 0 & a_{ij}^{add} \leq 0 \\ a_{ij}^{add} g_j^{m^{min}} & CS_{bef}^j \neq None \\ a_{ij}^{add} g^1 & else \end{cases} \quad (4.70)$$

$$a_{ij}^{addrec} = \begin{cases} 0 & a_{ij}^{add} \leq 0 \\ a_{ij}^{add} c_j^{m^{min}} & CS_{bef}^j \neq None \\ a_{ij}^{add} c^1 & else \end{cases} \quad (4.71)$$

The earliest minimum and maximum begin time, a_j^{min} and a_j^{max} , are computed by equations 4.72 and 4.73, in a similar fashion as in the EVRPTW-PR problem. The only difference is in the computation of a_j^{min} where the determined charging time at previous CS, a_i^{cc} (if preceding user i is CS) is considered as minimum charging amount (which does not have to be truly minimum). The recharging costs a_j^{rec} are cumulatively increased every time when there is a change in the charging amount at particular CS, or when a_{ij}^{addrec} is added. The same shifting rules and violation equations are used as in EVRPTW-PR problem.

$$a_j^{min} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + t_{ij} + a_i^{cc} g^{m_j}) + a_{ij}^{addt} & i \in F' \\ \max(e_j, \tilde{a}_i^{min} + t_{ij} + s_i) + a_{ij}^{addt} & else \end{cases} \quad (4.72)$$

$$a_j^{max} = \begin{cases} \max(e_j, \tilde{a}_i^{min} + a_i^{rt} g^{m_j} + t_{ij}) & i \in F' \\ \max(e_j, \tilde{a}_i^{max} + t_{ij} + s_i) & else \end{cases} \quad (4.73)$$

The presented variables provide one solution to the charging schedule in route which can be unfeasible. The determined charging schedule is based on the principle that whenever there is a slack or there is a violation of fuel, add appropriate amount to previous CS in route, if possible. This is a myopic solution, and in most of the cases the charging schedule could be further improved, which could lead to better feasible solutions. Here, the following procedure is proposed to determine a feasible charging schedule, if possible. If at user j there is no fuel violation, but there is a time window violation, perhaps this time window violation is caused by long charging at previous CS. In such case, the charging amount at latest CS can perhaps be reduced, and added to CS before it (if it exists). The latest CS in current partial route is labeled as CS_{lat} and the first CS before the CS_{lat} is labeled CS_{bef} . This charge relocation, is performed only if there is no fuel violation, meaning that the total charging amount remains the same. The charge relocation can affect time window violation and increase feasibility. The forward time window violation \overrightarrow{TW}_j has to be expressed in capacity unit to know how much to reduce the charging amount at CS_{bef} . This is not an easy task as it seems. For example, let CS_{lat} use slow charger $m_{CS_{lat}} = 2$. Then, the amount of capacity Δq that needs to be reduced at CS_{lat} can be

computed as $\Delta q = \overrightarrow{TW}_j / g^2$. Let there be a CS, CS_{bef} with $m_{CS_{bef}} = 3$ on which this amount of recharge q' can be added. The total increased time would be $\Delta q \cdot g^3$ which is greater than the original \overrightarrow{TW}_j as slower charging technology is used. Due to the non-linear time windows, this value could be lower, but as all slack times are considered as charging times, the total time is linearly increased. This procedure cannot be used if whole charging amount at CS_{lat} is computed from slack time, as linearity is not ensured. Only the charging amount that was added in previous step or the additional charging amount can be considered for relocation. To know the exact amount of the reduced charging capacity Δq corresponding to the time window violation \overrightarrow{TW}_j the additional following assumption is used: if there is no CS before the CS_{lat} that has a faster charger type than CS_{bef} , no improvement can be done, as all slack times are already used for charging, and additional charging would even increase time window penalty. Based on this, linear equations of earliest arrival travel times can be used to determine the reduced charging capacity Δq . The linear equations of earliest arrival travel times are given by equation 4.74 for the example presented in Figure 4.7. For example, the time window violation occurs at user 4 and has the value \overrightarrow{TW}_4 . The goal is to find new charging amounts $a_1^{cc'}$ and $a_3^{cc'}$ such that the overall earliest arrival time a_4^{min} is decreased by \overrightarrow{TW}_4 . This can be expressed by equation 4.75. Equation 4.76 is a result of the subtraction and reorganization of previous equations, and it represents the time window violation. If the total charging of route has to remain the same (equation 4.77), and the Δq is set as $a_1^{cc'} - a_1^{cc}$, the final equation for Δq can be expressed by equation 4.78. With generalization to any CS, this approach can be used to determine the charging amount at previous CS, under assumptions listed previously. After the computation of Δq , all of the variables have to be recomputed up to the current user j . If there are slack times that cannot be used for charging, this procedure cannot be applied due to the nonlinear arrival times. The complexity of proposed procedure is in practice $O(n^2)$, as mostly just two times the variables are recomputed, but still the complexity increases with the number of time window violations in route.

$$\begin{aligned}
 a_1^{min} &= a_0^{min} + s_0 + t_{01} \\
 a_2^{min} &= a_1^{min} + a_1^{cc} g^{m_1} + t_{12} \\
 a_3^{min} &= a_2^{min} + s_2 + t_{23} \\
 a_4^{min} &= a_3^{min} + a_3^{cc} g^{m_3} + t_{23} = \\
 & a_0^{min} + s_0 + t_{01} + a_1^{cc} g^{m_1} + t_{12} + s_2 + t_{23} + a_3^{cc} g^{m_3} + t_{23}
 \end{aligned} \tag{4.74}$$

$$a_4^{min} - \overrightarrow{TW}_4 = a_0^{min} + s_0 + t_{01} + a_1^{cc'} g^{m_1} + t_{12} + s_2 + t_{23} + a_3^{cc'} g^{m_3} + t_{23} \tag{4.75}$$

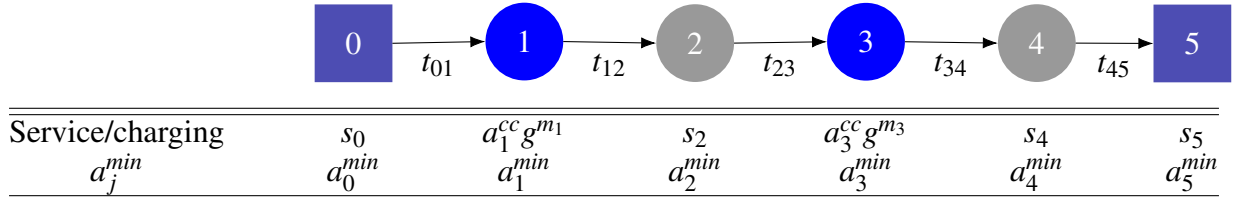


Figure 4.7: Example of route for determining charging schedule

$$\overrightarrow{TW}_4 = g^{m_1}(a_1^{cc} - a_1^{cc'}) + g^{m_3}(a_3^{cc} - a_3^{cc'}) \quad (4.76)$$

$$a_1^{cc} + a_3^{cc} = a_1^{cc'} + a_3^{cc'}$$

$$\underbrace{a_1^{cc'} - a_1^{cc}}_{\Delta q} = -(a_3^{cc'} - a_3^{cc}) \quad (4.77)$$

$$\overrightarrow{TW}_4 = g^{m_1}(\Delta q) + g^{m_3}(-\Delta q)$$

$$\Delta q = \frac{\overrightarrow{TW}_4}{g^{m_1} - g^{m_3}} \quad (4.78)$$

4.4 Destroy and repair operators

In this section, the destroy and repair operators used in the HALNS method are described. First, the destroy operators are used to remove the users from the solution, and then the repair operators are used to insert users back into the solution. The idea behind this procedure is to escape local optima and explore a new solution space. The operators differ in how much they change the solution. They usually produce a solution that is not better than the current solution, therefore the LS procedure is used to improve the solution. Keskin et al. [1] also applied destroy and repair operators, but their aim was to intensify the search and find better solutions without the LS phase. Such operators are often time-consuming, and several of them were tested as part of the proposed HALNS method but were rejected. The used destroy and repair operators are the following:

- Worst removal operator;
- Related removal operator;
- Shaw removal operator;
- CS vicinity operator;
- Sequential insertion operator;
- Sequential insertion operator with perturbed cost.

Before a destroy operator execution, a number of customers n_r to be removed from the solution has to be determined. The n_r is selected at random from the interval $[\underline{n}_r, \bar{n}_r]$, $\underline{n}_r \leq \bar{n}_r$, where \underline{n}_r and \bar{n}_r are computed by equations 4.79-4.80. The N_c represents the number of customers in the problem, while the parameters μ_{low} and μ_{high} are percentage threshold values, set in advance. The round function rounds the value to the nearest integer value. The \underline{n}_r cannot be lower than one, so the max function is added that limits the value to one. The threshold boundaries μ_{low} and μ_{high} are usually set in the $[0.1, 0.4]$ interval [1]. The values are rounded to the nearest integer values.

$$\underline{n}_r = \max(\text{round}(N_c \cdot \mu_{low}), 1) \quad (4.79)$$

$$\bar{n}_r = \text{round}(N_c \cdot \mu_{high}) \quad (4.80)$$

The rest of the section describes the tested destroy and repair operators.

4.4.1 Worst removal

The worst removal operator initially proposed by Pisinger and Ropke [32, 147], removes the customers that are "expensive", i.e., significantly increase the objective function value. The worst removal operator adapted on EVRP is described by Algorithm 4.4, similar to the ones described in [1, 20, 37, 53]. The operator removes only customers from the solution. The input values are solution s and number of customers to remove n_r . In each iteration of an algorithm, the list L_r , which contains customer candidates for removal, is initialized. Then, the operator loops through all unremoved customers in the solution s and computes the difference c_{diff} in objective cost value if the customer is removed from the solution and if the customer stays in the solution. This difference cost and observed customer are added as tuple to the list L_r . Afterward, the list L_r is sorted in descending order, from customers that have the highest savings to the ones that have the lowest savings value. Then, based on the determinism factor k_w , one customer is selected and removed from the solution, and the procedure is repeated. Depending on the objective function, the term worst distance removal or worst time removal operators can be found in the literature [1, 37].

To avoid selecting the same customers to be removed from the solution, the worst removal determinism factor k_w is used. The random value between $[0, 1)$ is used as a base in an exponential function, while the value of k_w is used as the exponent. The result is treated as a percentage that multiplied with the number of customers in list L_r (non-removed customers) gives a position i -th tuple (customer and cost) in the list L_r , that is going to be removed from the solution. The floor function was used to ensure that appropriate position in the list was selected. The

Algorithm 4.4 Worst removal operator

Input: Solution s and number of customers to remove n_r

- 1: $n \leftarrow 0$
- 2: **while** $n < n_r$ **do**
- 3: $L_r \leftarrow$ Initialize list of tuples containing customers and their costs
- 4: **for each** customer c in solution s **do**
- 5: $c_{in} \leftarrow$ Cost of vehicle route with customer c
- 6: $c_{out} \leftarrow$ Cost of vehicle route without customer c
- 7: $c_{diff} = c_{in} - c_{out}$
- 8: Add c_{diff} with customer c to list L_r
- 9: **end for**
- 10: Sort list L_r descending based on the c_{diff} value
- 11: $\lambda \leftarrow$ Random value in interval $[0, 1)$
- 12: $p \leftarrow \lambda^{k_w}$
- 13: $i \leftarrow \lfloor \text{length}(L_r) \cdot p \rfloor$
- 14: Remove the customer which cost is i -th in the list L_r from the solution s
- 15: $n \leftarrow n + 1$
- 16: **end while**

example of how the determinism factor influences the selection of customers is presented in Figure 4.8. In Figure 4.8a, the function $y = x^k$ (y-axis) is plotted for several k values. It can be seen that the larger the k value is, more the exponential function curves. The example of probability p for selecting a customer at i -th position for instance with 100 customers is presented in Figure 4.8b. Only the first 10 positions are presented as the rest have even lower probabilities. It can be seen that $k = 5$ significantly increases the probability of selecting the first customer in the list. The opposite holds for a low $k = 1$ value, as in this case, all customers have the same probability of selection, and high-quality removals could be rejected. The determinism factor k_w value in interval $[3, 6]$ is used in related researches [1, 37].

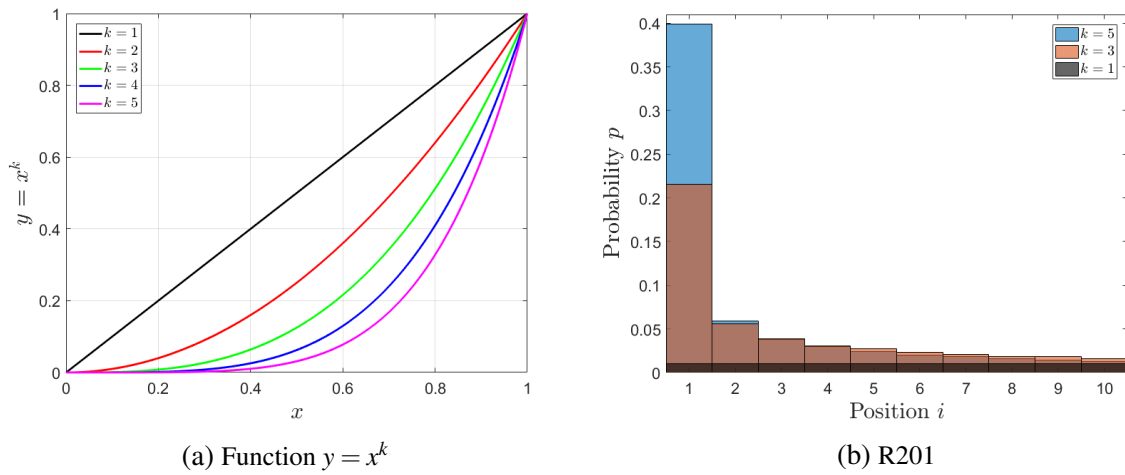


Figure 4.8: Worst removal determinism factor

The example of 10 customers (black dashed circles) removed by the worst removal operator on C101 instance with an infeasible EVRPTW-FR solution is presented in Figure 4.9. The used worst removal determinism factor k_w is set to 4. The objective function includes the values for total distance traveled and constraint violations. For comparison, the customers removed with the criteria of only distance minimization without the penalties for violation are represented with cyan dashed circles. These are the customers that have long detour costs, but also, some shorter arcs are present due to the added stochastic component in the operator (determinism factor k_w). It is also important to note that only one removed customer overlaps between the examples, showing the importance of using penalty violations in the objective function.

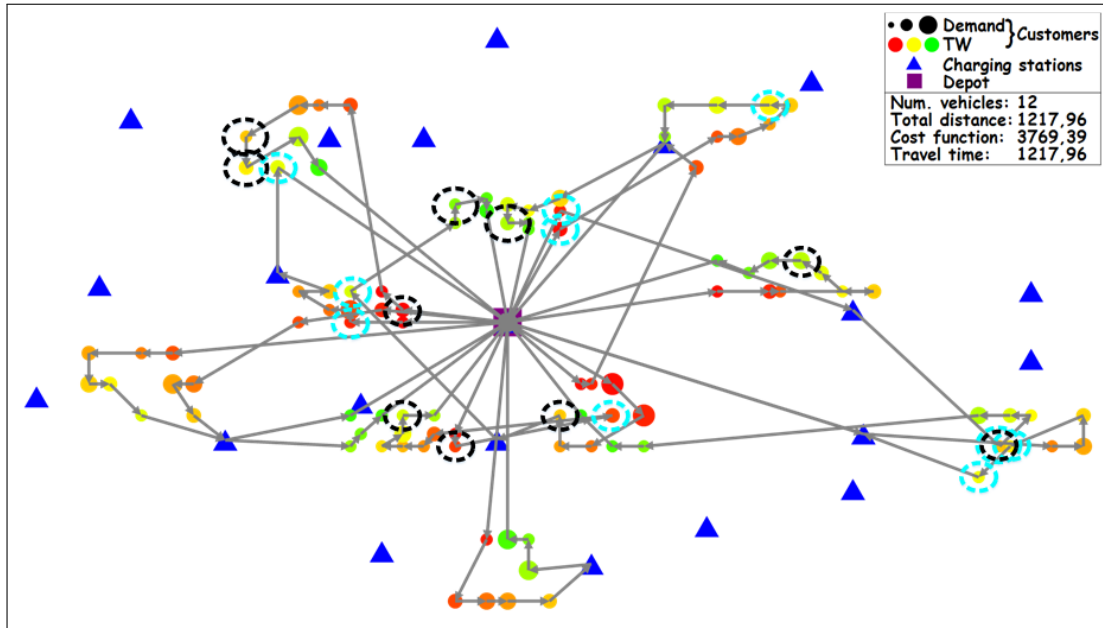


Figure 4.9: Example of worst removal operator

4.4.2 Related removal

Related removal operator removes a set of customers that are close to each other in terms of distance between them [147]. The first customer is selected at random or based on the detour distance cost. The next customers are selected in a deterministic way or by a roulette wheel selection based on the relatedness measure [29, 155]. The related removal operator is given by Algorithm 4.5. First, the two lists are initialized: list of removed customers R and list of unserved customers U containing all customers in the problem instance. Next, one customer is selected at random from list U , set as the current customer cur , and added to the list of removed customers R . Then the distance from customer cur to all other unserved customers is computed and together as a tuple added in the list L_r which represents candidates for the removal. The list L_r is sorted in ascending order, meaning that the customers that are closer to the customer cur are at the beginning of the list. Next, in the same way as in the worst removal operator, the stochastic component is added based on the determinism factor k_r (Figure 4.8a). The customer at i -th position in list L_r is removed from the list U and added to the list of removed customers R . The described procedure is repeated (lines 6-20) until n_r customers are added to the list R . In the end, all customers in the list R are removed from the solution s . This is the biggest difference between the worst and related removal, as in worst removal in each iteration of the main loop, a customer is removed from the solution, and the solution variables are recomputed (in total n_r times). In related removal, customers are added to the list R , and in the end, removed from the solution. Therefore, the solution variables are recomputed only once. Also, the used related removal operator is only distance based, while the worst removal operator takes into account the objective function value.

Algorithm 4.5 Related removal operator

Input: Solution s and number of customers to remove n_r

- 1: $R \leftarrow$ Initialize list of removed customers
- 2: $U \leftarrow$ Initialize list of unserved customers with all customers
- 3: $cur \leftarrow$ Select at random customer from the list U
- 4: Remove cur from U and add it to R
- 5: $n \leftarrow 1$
- 6: **while** $n < n_r$ **do**
- 7: $L_r \leftarrow$ Initialize list of tuples containing customers and distances to customer cur
- 8: **for each** customer c in list U **do**
- 9: $d \leftarrow$ Distance between cur and c
- 10: Add d with customer c to list L_r
- 11: **end for**
- 12: Sort list L_r ascending based on the distance value d
- 13: $\lambda \leftarrow$ Random value in interval $[0, 1)$
- 14: $p \leftarrow \lambda^{k_r}$
- 15: $i \leftarrow \lfloor \text{length}(L_r) \cdot p \rfloor$
- 16: Remove the customer c from U which distance is i -th in list L_r
- 17: Add customer c to R
- 18: $cur \leftarrow c$
- 19: $n \leftarrow n + 1$
- 20: **end while**
- 21: Remove all customers in the list R from the solution s

The example of 10 customers (black dashed circles) removed using related removal operator on C101 instance with an infeasible EVRPTW-FR solution is presented in Figure 4.10. The used worst removal determinism factor k_r is set to 6. As it can be seen, several groups of customers are removed from the solution. As the worst removal determinism factor is quite high, often the closest customers are removed from the solution, which, as a result, form a group. However, in some cases, it selects the n -th closest customer, which then represents a switch to another group.

4.4.3 Shaw removal

Shaw removal operator proposed by Shaw [145] tries to remove customers that are in some sense similar to each other by taking into account geographical distance, demand difference, earliest start time difference, and assigned route difference. The original objective function is modified for EVRP problems and given by equation 4.81, where the values of distance, early time windows, and demands are normalized [20]. d_{max} is the maximum distance between users in the problem, $e_{max} - e_{min}$ the difference between the maximum and minimum value of early time window, and $q_{max} - q_{min}$ difference between the maximum and minimum value of customer demand. Parameters χ_d , χ_e and χ_q control the contribution of each part to the related removal objective function. The special cases are proximity based ($\chi_e = \chi_q = 0$) [1, 37, 54], time based

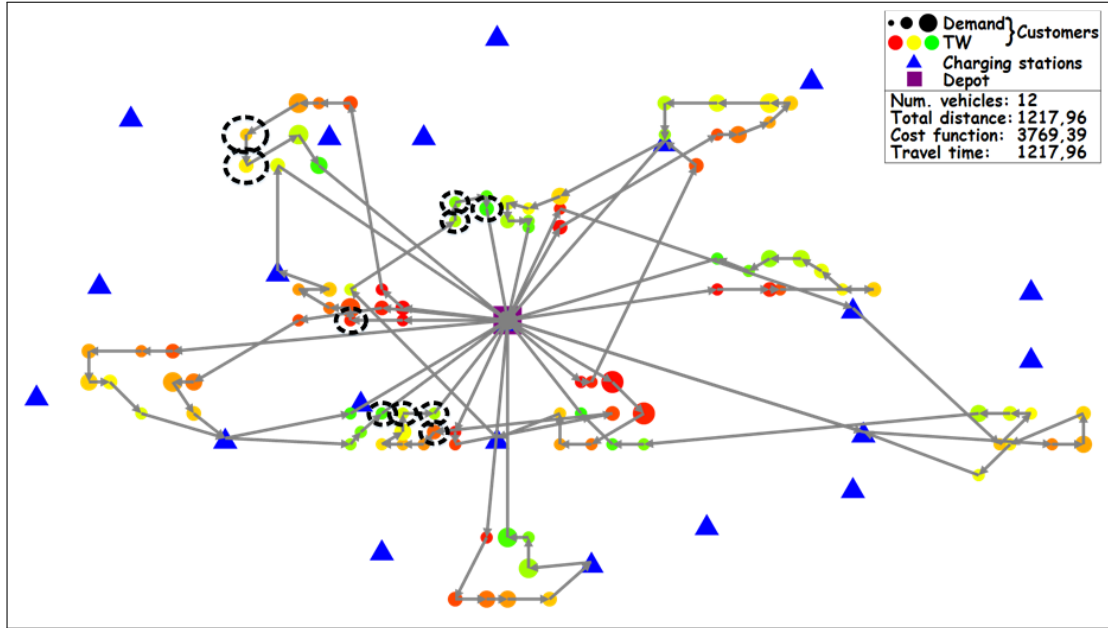


Figure 4.10: Example of related removal operator

($\chi_d = \chi_q = 0$) and demand based ($\chi_d = \chi_e = 0$) operators. The general algorithm of the Shaw removal is almost the same as for the related removal given by algorithm 4.5, with only few differences: (i) lines 9, 10 and 12 where instead of distance between the customers, the Shaw value given by equation 4.81 is used; and (ii) line 14 where the Shaw determinism factor k_s is used. The related removal can be observed as a special case of proximity based removal, where the maximum distance between users d_{max} is omitted.

$$c_{ij} = \chi_d \frac{d_{ij}}{d_{max}} + \chi_e \frac{|e_i - e_j|}{e_{max} - e_{min}} + \chi_q \frac{|q_i - q_j|}{q_{max} - q_{min}} \quad (4.81)$$

The example of 10 customers (black dashed circles) removed by Shaw removal operator on C101 instance with infeasible EVRPTW-FR solution is presented in Figure 4.11. The used Shaw determinism factor k_s is set to 6, while the objective function parameters are set to $\chi_d = 6$, $\chi_e = 5$ and $\chi_q = 4$.

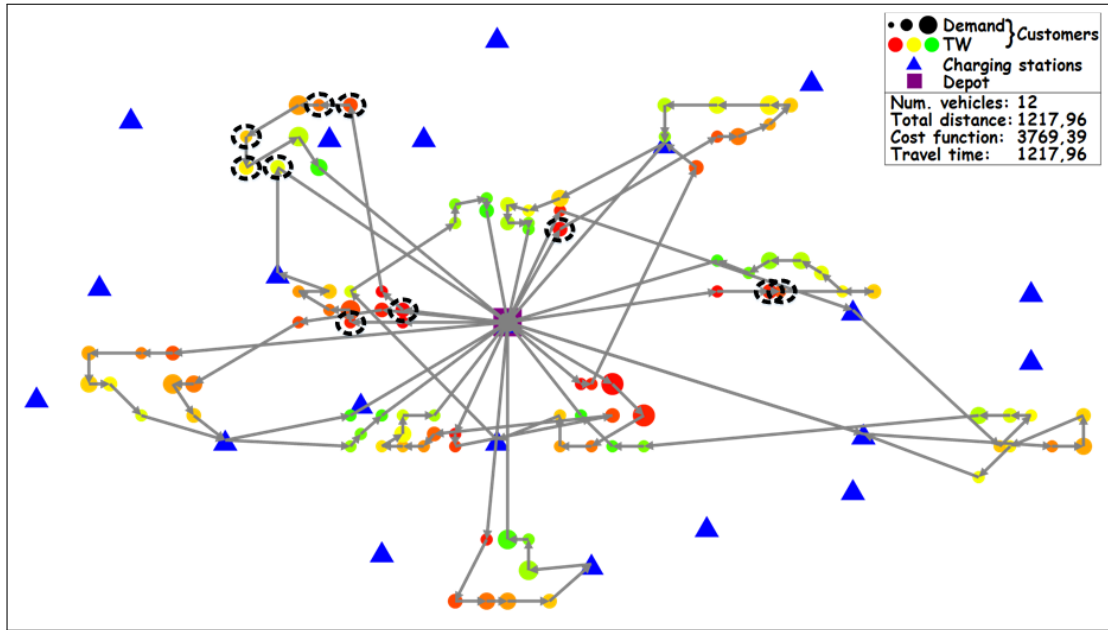


Figure 4.11: Example of Shaw removal operator

4.4.4 CS vicinity

CS vicinity operator removes customers and CSs in the radial vicinity of the selected CS [20]. The operator is given by Algorithm 4.6. First, two lists are initialized, a list of removed customers R and list L_{CS} containing all CSs in the solution s . Next the radius of vicinity r is determined as a random value within the interval $[\chi_{min}d_{max}, \chi_{max}d_{max}]$, where d_{max} is the maximum distance between two customers in the problem instance, and χ_{min} and χ_{max} are lower and upper threshold percentages for radius bound. Next, one CS is selected from the list L_{CS} , and all customers that are in the radial vicinity r of the selected CS are added to set V_r . The $|c - CS_{cur}|$ represents the Euclidean norm (equation 4.82). Afterward, each customer c in V_r , which has not yet been added to the list R , is added to R , and the selected CS is also removed from the list L_{CS} and solution s . In the end, in the same way as in the related removal operator, all customers in list R are removed from the solution s .

$$|c - CS_{cur}| = \sqrt{(c_x - CS_{curx})^2 + (c_y - CS_{cury})^2} \quad (4.82)$$

The application of the CS vicinity operator on C101 instance with infeasible EVRPTW-FR solution is presented in Figure 4.12. The parameters χ_{min} and χ_{max} are set to 0.05 and 0.15, respectively. Although the number of removed customers n_r is set to 10, in total, 11 customers were removed from the solution. This occurs in situations where the condition $n < n_r$ (line 7) is satisfied, and all customers in the vicinity of the selected CS are removed, which in the total number of removed customers can be greater than n_r . The red dashed ellipses (due to the distortion as figure width is greater than the figure height) represent the CS vicinity area,

in which customers (black dashed circles) are removed. In the end, the selected CSs are also removed from the solution.

Algorithm 4.6 CS vicinity

Input: Solution s and number of customers to remove n_r

- 1: $R \leftarrow$ Initialize empty list of removed customers
 - 2: $L_{CS} \leftarrow$ Initialize list with all CSs used in the solution s
 - 3: $\underline{r} \leftarrow \chi_{min} \cdot d_{max}$
 - 4: $\bar{r} \leftarrow \chi_{max} \cdot d_{max}$
 - 5: $r \leftarrow$ Select random number from interval $[\underline{r}, \bar{r}]$
 - 6: $n \leftarrow 0$
 - 7: **while** $n < n_r$ **do**
 - 8: $CS_{cur} \leftarrow$ Select one CS at random from the list L_{CS}
 - 9: $V_r \leftarrow$ All customers from s that satisfy $|c - CS_{cur}| \leq r$
 - 10: **for each** customer c in V_r **do**
 - 11: **if** c not in R **then**
 - 12: Add c to R
 - 13: $n \leftarrow n + 1$
 - 14: **end if**
 - 15: **end for**
 - 16: Remove CS_{cur} from list the L_{CS} and solution s
 - 17: **end while**
 - 18: Remove all customers in R from solution s
-

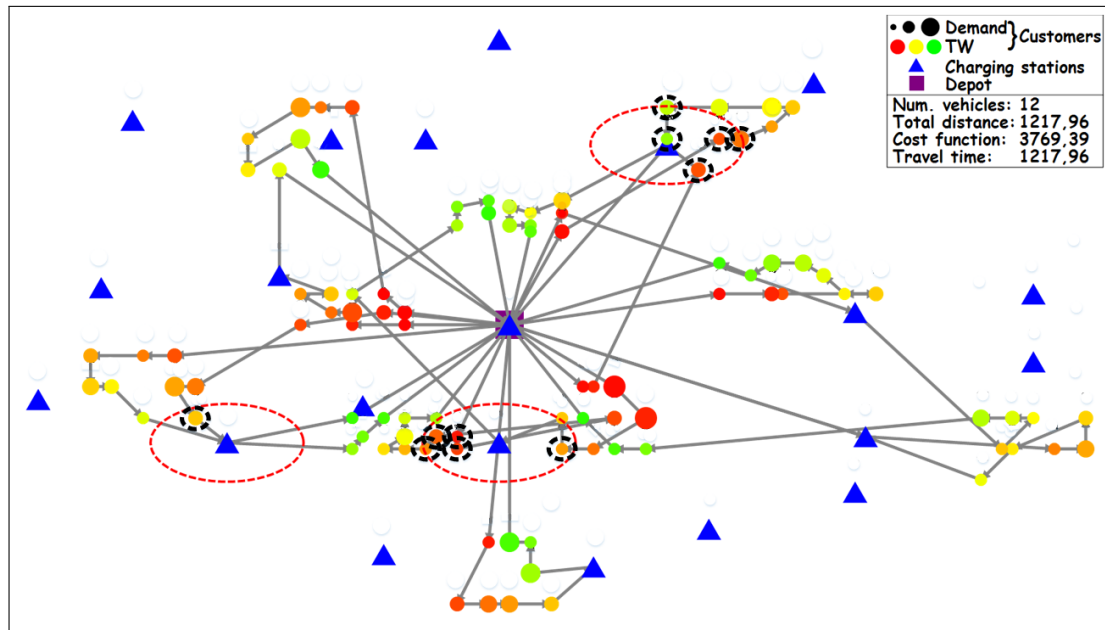


Figure 4.12: Example of CS vicinity operator

4.4.5 Sequential insertion operator

The sequential insertion operator inserts customers in the solution based on their removal order [29]. This can be interpreted as Last-In First-Out (LIFO) remove-insert strategy. The operator is given by Algorithm 4.7. First, the first customer c in list L_r , which contains all removed customers, is selected, and the list of tuple insertions L_i is initialized as empty. Then, two loops are nested, the first one for each vehicle v in the solution s , and the second one for each position i in the vehicle route v . The position of the first depot instance is skipped as the insertion of a customer at that position would mean that vehicle does not start from the depot. In each iteration, four insertions are evaluated:

- (i) customer only insertion, c_i ,
- (ii) customer with preceding CS, $c_{CS,i}$,
- (iii) customer with succeeding CS, $c_{i,CS}$,
- (iv) and customer with both preceding and succeeding CS, $c_{CS,i,CS}$.

The CSs that are inserted along with the customer are selected as the nearest CS for both succeeding and preceding cases. Then, the list L_i is restricted to the n_{RCL}^{si} best insertions. The

Algorithm 4.7 Sequential insertion operator

Input: Solution s and list of removed customers L_r

- 1: **while** L_r is not empty **do**
- 2: $c \leftarrow$ Select first customer from list L_r
- 3: $L_i \leftarrow$ Initialize empty list of tuples containing insertion information (position, cost, vehicle)
- 4: **for each** vehicle v in s **do**
- 5: **for each** position i in vehicle v after the first depot position **do**
- 6: $CS_{bef} \leftarrow$ Nearest CS between user at position $i - 1$ and customer c
- 7: $CS_{aft} \leftarrow$ Nearest CS between customer c and user at position $i + 1$
- 8: Cost $c_i \leftarrow$ Cost of insertion of customer c at position i in vehicle v
- 9: Cost $c_{CS,i} \leftarrow$ Cost of insertion of customer c with preceding CS_{bef} at position i in vehicle v
- 10: Cost $c_{i,CS} \leftarrow$ Cost of insertion of customer c with succeeding CS_{aft} at position i in vehicle v
- 11: Cost $c_{CS,i,CS} \leftarrow$ Cost of insertion of customer c with preceding CS_{bef} and succeeding CS_{aft} at position i in vehicle v
- 12: Add tuples of cost, position i and vehicle v to list L_i for each cost c_i , $c_{CS,i}$, $c_{i,CS}$ and $c_{CS,i,CS}$
- 13: Restrict candidate list L_i to the n_{RCL}^{si}
- 14: **end for**
- 15: **end for**
- 16: $c_{insert} \leftarrow$ Roulette wheel selection from list L_i based on the insertion cost
- 17: Perform insertion c_{insert} with possible CSs insertions
- 18: Remove c from list L_r
- 19: **end while**

idea is adopted from Hierman et al. [29] which store n_{RCL}^{si} best insertions of each customer in the current solution, and select one at random. Here, the same idea is adopted, but for the selection of the insertion from n_{RCL}^{si} best ones, the RWS selection is used. The RWS is already described in section 3.3.6, [148]. The idea is that the insertion that has the lowest insertion cost has the highest probability of being selected. The RWS is suitable in applications where the objective function used to order the entities has a strictly positive value. In EVRPTW, due to the violation penalties, there can be multiple insertions that have a negative value, meaning that they improve the current solution, i.e., CS insertion. In such cases, all cost values are shifted by the absolute value of the minimum objective cost increase. This process lowers the probability of selecting good insertions, but still, good insertions have higher probabilities than the bad ones. RWS is initially developed for selecting the entity that has a higher objective function, while here, the insertions that have lower values should have higher probabilities. Therefore, all objective costs are reversed in relation to the maximum objective cost in the restricted candidate list. After the selection of insertion c_{insert} , the insertion is performed and customer c is removed from the list L_r . The whole procedure is repeated until all customers are reinserted back into the solution (lines 1-19 in algorithm 4.7).

The example of sequential insertion operator on C101 instance with an infeasible EVRPTW-FR solution is presented in Figure 4.13. The insertion procedure continues on previous CS vicinity removal presented in Figure 4.12. The customers that are reinserted back in the solution are marked with black dashed circles. Red lines represent arcs that are inserted in the solution. It can be seen that in some cases, the CS is inserted together with the customer. The produced solution is worse than the solution before the destroy and repair operators but will be further improved in the LS phase.

The sequential insertion operator with perturbed cost has only one difference compared to the previous one. The insertion cost of c_{insert} is modified by the stochastic component. The cost is multiplied by a random value Γ selected within the interval $[\Gamma_{min}, \Gamma_{max}]$. This procedure helps to diversify the solution by artificially changing the insertion cost. The parameter values of Γ_{min} and Γ_{max} are usually within $[0.7, 1.3]$ interval, [29, 53].

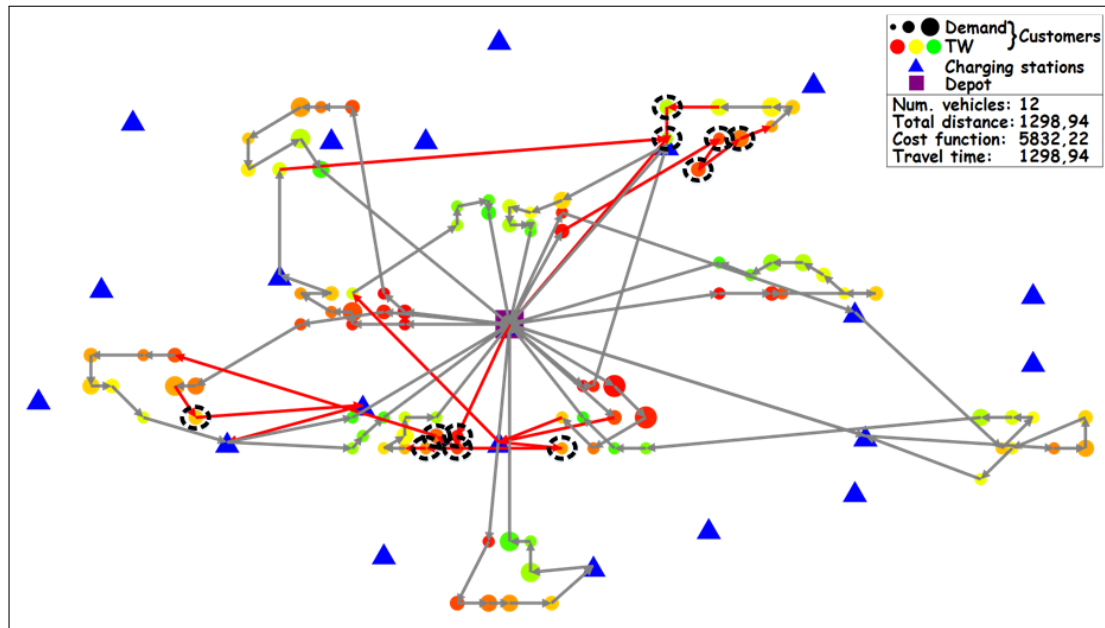


Figure 4.13: Example of sequential insertion operator

4.4.6 Other tested operators

During the development of the HALNS method, other removal and insertion operators have also been tested but have been rejected due to the high execution time and low contribution to the overall best solution. These are mostly the operators used in the ALNS presented by Keskin et al. [1], to which the HALNS method is directly compared in subsection 4.10.2.

Random customer removal

Random customer removal removes customers at random from the solution [1, 37]. Although randomness adds a stochastic component to the procedure and helps to escape the local optima, a completely random selection of users often leads to bad solutions, which are hard to repair.

Worst time and distance removal

Worst time and worst distance removals are a special case of the worst operator removal where the objective function is purely distance or time based [1, 37]. The worst distance operator removes customers that are distant from its neighboring customers in route. The worst time operator removes customers that have high value of $|\tau_i - e_i|$ (difference in time), where τ_i is the arrival time at customer i , and e_i is its early time window. The operators were rejected as they do not include the penalty violations.

Zone removal

the zone removal operator is based on the removal of users in predefined zones. The whole geographical instance space is divided into a fishnet quadratic grid. The operator at random selects one cell (zone) from the fishnet grid and removes all users from the selected zone. The example of zone removal is presented in Figure 4.14, where all customers in the red zone are removed from the solution.

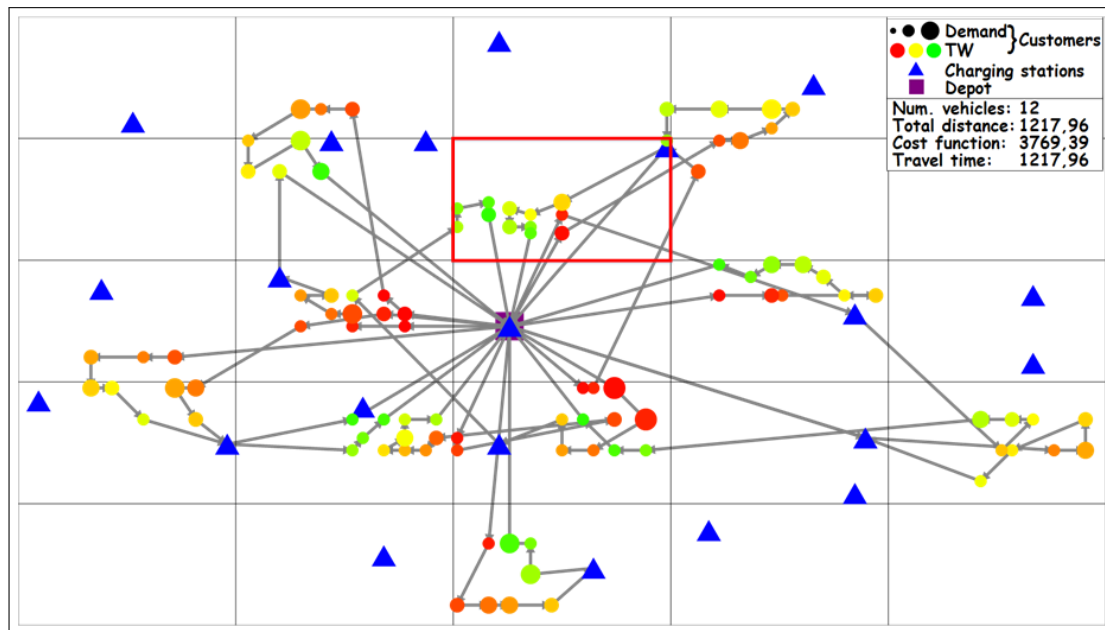


Figure 4.14: Example of zone removal operator

Removing customers with CSs

Removal of a customer with preceding CS removes customer with the preceding CS if it exists. The goal is to eliminate the visit to CSs as the vehicle no longer visits the removed customer, and recharging is not necessarily needed [1]. Removal of a customer with succeeding CS removes customer with the succeeding CS if it exists. The recharging may be needed after the departure from a customer in order to be able to reach the next user in the route. In that case, recharging is not necessarily needed as a customer is removed from the solution [1]. Both operators were rejected, as usually with a higher number of customers removed, almost all CSs in the solution are removed, which results in a solution far from the local optima.

CS removal

The random CS removal operator and worst distance CS removal operator are similar to their customer removal counterparts. The worst charge usage operator aims to remove CS visited with a high SoC value. The idea is to utilize the battery energy as much as possible and to avoid

short charging at CSs [1]. All of the mentioned CS removals were rejected as: (i) CS vicinity operator used in HALNS outperforms them, and (ii) in HALNS the removals of CSs are also performed in the improvement phase: LS and optimal CS placement.

Greedy insertion operator

Greedy insertion operator, from all removed customers selects the customer c , vehicle v and position i which minimize objective function the most [1, 20, 37, 147, 156]. The operator is given by Algorithm 4.8. The algorithm is similar to the sequential insertion operator, with the difference that instead of the first customer in the remove list L_r and RWS selection, always the best insertion is selected. The insertions of CSs with customers are also omitted. The greedy insertion operator tends to get stuck in local optima. The example of greedy customer insertion on the same example as for the sequential insertion operator is presented in Figure 4.15. It can be seen that the overall traveled distance is lower than in the sequential insertion example. The greedy insertion operator has a higher complexity due to the one additional loop for the iteration of customers in the list L_r (in order to know the best insertion).

Algorithm 4.8 Greedy insertion operator

Input: Solution s and list of removed customers L_r

- 1: **while** L_r is not empty **do**
- 2: $c_{best} \leftarrow$ Initialize a tuple of customer, vehicle, position and cost
- 3: **for each** customer c in L_r **do**
- 4: **for each** vehicle v in s **do**
- 5: **for each** position i in vehicle v after the first depot position **do**
- 6: Cost $c_i \leftarrow$ Cost of insertion of customer c at position i in vehicle v
- 7: **if** $c_i < f(c_{best})$ **then**
- 8: $c_{best} \leftarrow$ Tuple c, v, i and c_i
- 9: **end if**
- 10: **end for**
- 11: **end for**
- 12: **end for**
- 13: Perform insertion c_{best} and remove respective customer from the list L_r
- 14: **end while**

A special case of greedy insertion is time based insertion [1], in which the cost considers only the difference in total time of the route before and after insertion. Another special case of the greedy customer insertion is the zone insertion operator [1], which is a counterpart to the zone removal operator. The idea is to perform greedy customer insertion only within a randomly selected zone.

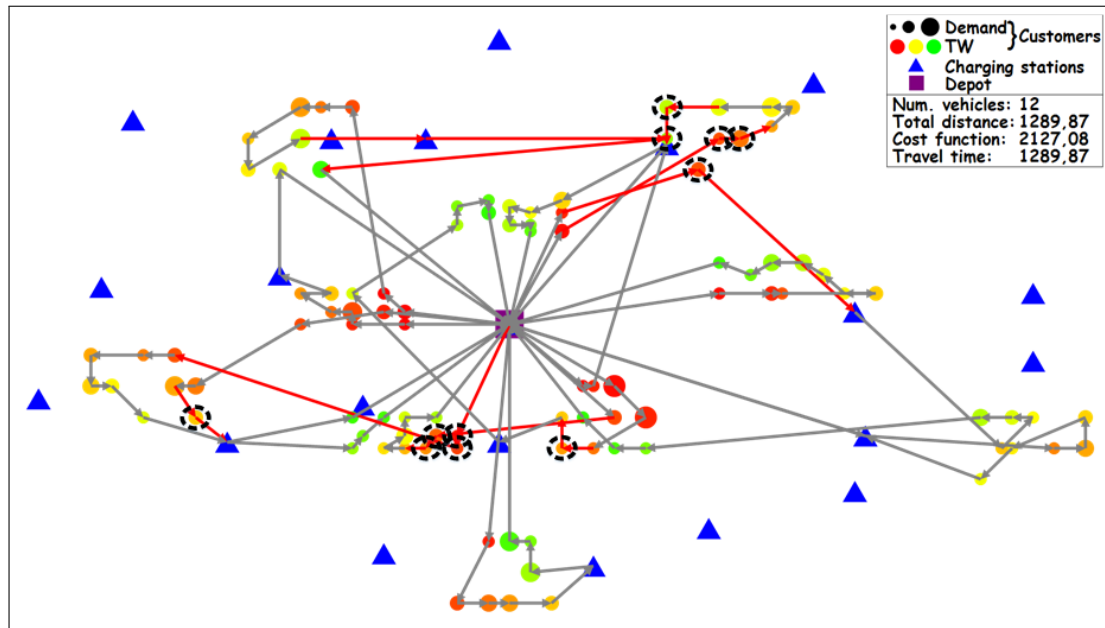


Figure 4.15: Example of greedy insertion operator

Regret insertion operator

Regret insertion operator is one of the most applied operators for solving various VRP problems [147]. The idea is to compute the difference between the best insertion and k -th best insertion. This difference is called the regret value. The customers with a higher regret value have a lower number of cost-effective insertions. Usually, the regret-2 and regret-3 variants are applied in the VRP field [37, 90]. Some variants of the problem consider the regret values between the best vehicle and k -th best vehicle, skipping the high regret values in the same vehicle [26, 157, 158]. The overview of the operator is given by Algorithm 4.9. The only difference to the greedy operator is that the regret operator stores all insertions in the list, and then based on the computed regret values, selects the customer insertion with vehicle and position that has the highest regret value. The application of the regret insertion operator has more sense in variants where feasible solution space is narrow, as there the regret values could be higher. The example of greedy customer insertion on the same example as for the sequential insertion is presented in Figure 4.16. The problem of both regret heuristic and greedy heuristic is that they are time consuming as they loop through all customers, vehicles, and positions and then find the best possible insertion, especially the regret heuristic which needs to sort regret insertions.

Algorithm 4.9 Regret insertion operator

Input: Solution s , list of removed customers L_r and regret value k

- 1: **while** L_r is not empty **do**
- 2: $c_{best} \leftarrow$ Initialize tuple containing customer, vehicle, position, and regret value initialized with minimum value
- 3: **for each** customer c in L_r **do**
- 4: $L_i \leftarrow$ Initialize empty list of tuples containing insertion information (vehicle, position, cost)
- 5: **for each** vehicle v in s **do**
- 6: **for each** position i in vehicle v after the first depot position **do**
- 7: Cost $c_i \leftarrow$ Cost of insertion of customer c at position i in vehicle v
- 8: Add tuple containing vehicle v , position i and cost c_i to list the L_i
- 9: **end for**
- 10: **end for**
- 11: Sort list L_i ascending based on the cost value
- 12: $r_c \leftarrow$ Compute regret value between the best insertion and k -th best insertion in the list L_i
- 13: **if** $r_c > \text{regret}(c_{best})$ **then**
- 14: $c_{best} \leftarrow$ First tuple in the list L_i coupled with the regret value r_c
- 15: **end if**
- 16: **end for**
- 17: Perform insertion c_{best} and remove respective customer from the list L_r
- 18: **end while**

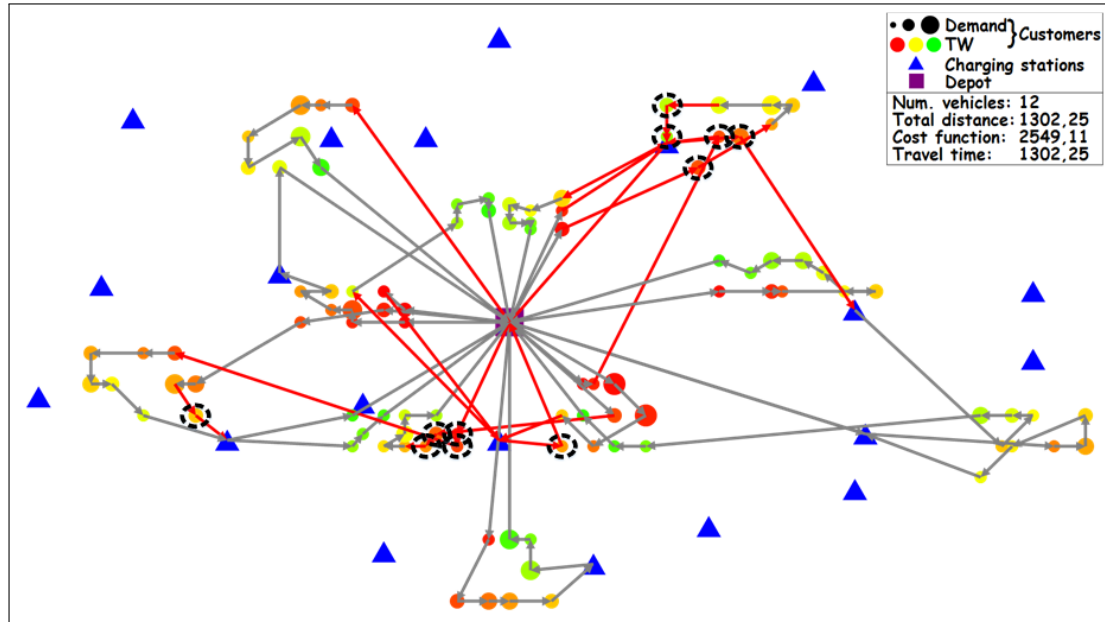


Figure 4.16: Example of regret insertion operator

4.4.7 ALNS example

The selection probabilities of the operators change depending on their performance. The example of how probabilities change for used operators in EVRPTW-PR on instance C101 is presented in Figure 4.17. The update of operator probabilities occurs every $\mu_{uop} = 50$ iterations. As it can be seen, different operators are preferred in different stages of the search. The related and CS vicinity destroy operators show lower probabilities than worst and Shaw destroy operators. The probability of related operator increased in the last phase, showing its benefits in the cases when the search is near the best found local optima. The sequential insertion operator is preferred in the early stages of the search when more frequently, better solutions are found. But later on, when the search gets stuck in the local optima, the perturbed variant shows its benefits.

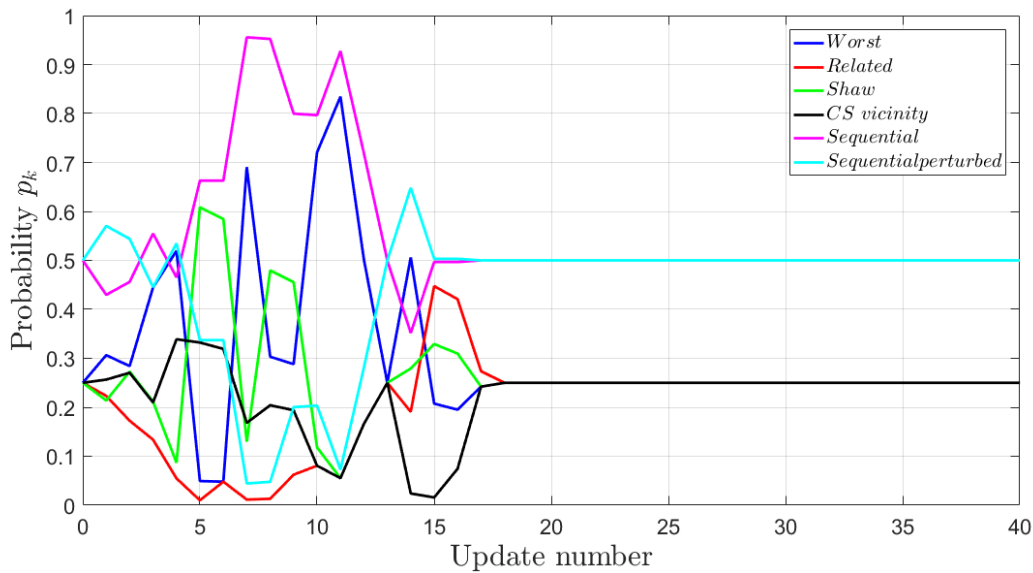


Figure 4.17: Example of operator probabilities: C101

4.5 Route removal operators

Removing a vehicle route from the solution means that all customers in the selected route are removed from the solution. The original ALNS framework for ELRPTW-PR proposed by Schiffer et al. [53] did not consider any route removal operator. By testing the route removal operators on EVRPTW problems, it was concluded that they significantly increase convergence to the best solution, as less time is spent on searching in the solution space with a higher number of vehicles. As it can be seen in Algorithm 4.1, line 6, the route removal criteria has to be met to perform the route removal. Instead of calling route removal operators every μ iterations, a different approach is used. The goal is to perform route removal operator more often at the beginning of the search, than at the end. To simulate such behavior the discrete function given by equation 4.83 and presented in Figure 4.18 is used. On the x -axis are the number of route removal calls, and on the y -axis is the iteration number in the HALNS. The value i represents the iteration at which the route removal is performed. After the creation of an initial solution, the route removal is immediately performed ($i = 0$), and afterwards it follows exponential function (25, 50, 100, 200, 400 and 800). Here, the upper limit on x is set to 6, as after the 800-th iteration, the focus is only on minimizing the costs with the determined number of vehicles. The vehicle number could still be reduced after the 800-th iteration, as in some cases in user removal operators, the whole route is removed (due to the stochastic components).

$$i = \begin{cases} 0 & x = 0 \\ 12.5 \cdot 2^x & 0 < x \leq 6 \end{cases} \quad (4.83)$$

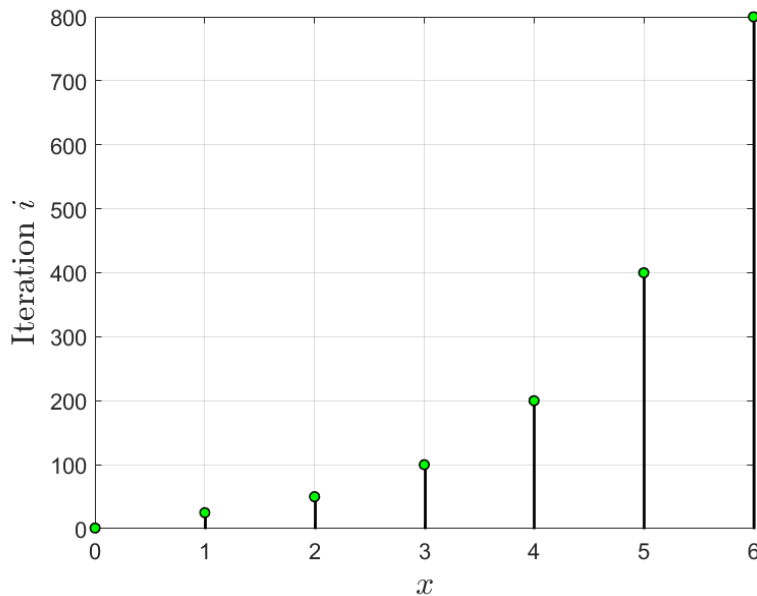


Figure 4.18: Route removal function

Five different route removal operators were tested: greedy route removal, random route removal, ALNS based route removal, repeated ALNS based route removal, and ejection pool. The greedy route removal and ejection pool have already been proposed in the literature, while the ALNS based route removal and repeated ALNS based route removal are newly proposed route removal operators.

4.5.1 Random and greedy route removal

Both random and greedy route removal operators remove ω routes from the solution, where ω is a random integer number between $[\underline{\omega}K, \overline{\omega}K]$, and K is the number of vehicle routes in the solution [1]. The value of ω is usually within 10-30% of the total number of vehicles. The random route removal operator removes ω routes at random, while the greedy route removal operator removes ω routes that have the lowest number of users. The example of removing two routes ($\omega = 2$) from the EVRPTW-FR solution on instance C101 is presented in Figure 4.19. The blue lines represent the random removal operator, while the red lines represent the greedy removal operator, in which two routes containing the lowest number of users, 9 and 10, are removed.

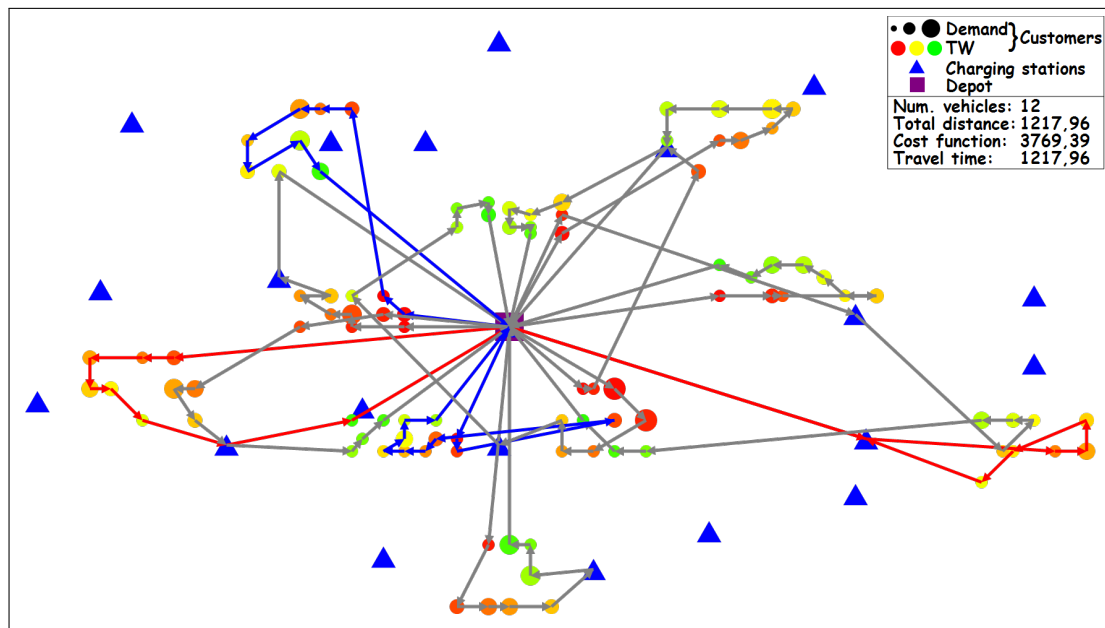


Figure 4.19: Random and greedy route removal operators

4.5.2 ALNS based route removal operator

The ALNS based Route Removal (ALNS-RR) operator follows a similar framework as the proposed HALNS method (Algorithm 4.1). The idea is to remove one route from the solution at random. Then perform HALNS on the solution, which has one vehicle route less, and try to find a feasible solution. The overview of the proposed operator is given by Algorithm 4.10, and follows the similar paradigm of HALNS. Here, only the differences will be highlighted. First, the current values of s_{temp} , s_{best} and penalties p_{cur} before route removal are stored in corresponding variables with added label *old*. All solution instances s_{temp} , s_{best} and s are initialized as the best feasible solution s_{best_feas} . The current penalty weights p_{cur} are set to the initial value for the start of route removal p_{start}^{rr} . Then, one route is removed at random from the solution s , and the repair operator r is selected to repair the solution s . Additionally, the flag *anyChange* is set to *False*, and the best solution s_{best} and temporal solution s_{temp} are set as current solution s after one route was removed. This flag is used at the end of the procedure as a check if there was at least one improvement of the vehicle number or not. If there was no improvement, the solution values are returned to their previous state for which the variables with label *old* are used (line 45). Otherwise, all solution instances are set to the best feasible solution s_{best_feas} . For μ_{max}^{rr} iterations the same steps as in HALNS are performed, but without route removal part: destroy, repair, LS, exact procedure, acceptance criteria, update of operator values, and update of penalty coefficients. Additional, two things are different. First, if the new solution is feasible, then it is set as the current best feasible solution s_{best_feas} and current solution s . Again, one vehicle route is removed from solution s , the solution s , is repaired and flag *anyChnage* is set to *True*. Second, a slightly different update procedure of penalty weights p_{cur} is used. As mentioned in section 4.3 the penalty coefficients in the HALNS are updated from minimum values p_{min} to maximum values p_{max} , depending on the feasibility of the best solution s_{best} . Here, the cycling procedure is used. As the search reaches the p_{min} values, the direction of the penalties update is reversed towards the p_{max} . And as the search reaches the p_{max} , the direction is reversed towards the p_{min} . This helps to overcome long computation time when penalties have a low value, as LS is frequently performed. Additionally, the penalty weight update for route removal is larger than in regular HALNS, $\zeta^{rr} > \zeta$, to ensure faster searching in solution space with a lower number of vehicles.

Algorithm 4.10 ALNS based route removal operator

Input: Temporal solution s_{temp} , best solution s_{best} , best feasible solution s_{best_feas} , current penalty coefficient values p_{cur}

```

1:  $\{s_{temp}^{old}, s_{best}^{old}, p_{old}\} \leftarrow \{s_{temp}, s_{best}, p_{cur}\}$ ,  $p_{cur} \leftarrow p_{start}^{rr}$ ,  $s_b \leftarrow s_{best\_feas}$ 
2:  $s_d \leftarrow$  Remove one vehicle route at random from solution  $s_b$ 
3:  $s \leftarrow$  Select repair operator  $r$  and repair solution  $s_d$ 
4:  $s_{best}, s_{temp} \leftarrow s$ 
5:  $anyChange \leftarrow False$ 
6:  $i \leftarrow 0$ 
7: while  $i < \mu_{max}^{rr}$  do
8:    $d, r \leftarrow$  Select destroy and repair operators
9:    $s_d \leftarrow$  Destroy solution  $s$  with operator  $d$ 
10:   $s_{new} \leftarrow$  Repair solution  $s_d$  with operator  $r$ 
11:  if  $f(s_{new}) < f(s_{best})(1 + \Delta_{ls}^{rr})$  then
12:     $s_{new} \leftarrow$  Perform LS on  $s_{new}$ 
13:    if  $f(s_{new}) < f(s_{best})(1 + \Delta_{exact}^{rr})$  then
14:       $s_{new} \leftarrow$  Perform exact procedure on  $s_{new}$ 
15:    end if
16:  end if
17:  if  $f(s_{new}) < f(s_{temp})$  then
18:     $s_{temp} \leftarrow s_{new}$ 
19:    if  $f(s_{new}) < f(s_{best})$  then
20:       $s_{best} \leftarrow s_{new}$ 
21:       $s_{best\_feas}^{new} \leftarrow$  Generate feasible solution from  $s_{new}$ 
22:      if  $s_{best\_feas}^{new}$  is feasible then
23:         $s_{best\_feas}, s \leftarrow s_{best\_feas}^{new}$ 
24:         $s_d \leftarrow$  Remove one vehicle route at random from solution  $s$ 
25:         $s \leftarrow$  Select repair operator  $r$  and repair solution  $s_d$ 
26:         $s_{best}, s_{temp} \leftarrow s$ 
27:         $anyChange \leftarrow True$ 
28:      end if
29:    end if
30:  end if
31:  Update scores of selected destroy  $d$  and repair  $r$  operators
32:  if modulo( $i, \mu_{uop}^{rr}$ ) = 0 then
33:    Update destroy and repair operators scores, weights and probabilities
34:  end if
35:  if modulo( $i, \mu_{upw}^{rr}$ ) = 0 then
36:    Update penalty coefficients  $p_{cur}$  for route removal
37:  end if
38:   $i \leftarrow i + 1$ 
39:   $s \leftarrow s_{temp}$ 
40: end while
41:  $p_{cur} \leftarrow p_{old}$ 
42: if  $anyChange$  then
43:   $s, s_{temp}, s_{best} \leftarrow s_{best\_feas}$ 
44: else
45:   $\{s, s_{temp}, s_{best}\} \leftarrow \{s_{temp}^{old}, s_{temp}^{old}, s_{best}^{old}\}$ 
46: end if

```


4.5.3 Repeated ALNS based route removal

In ALNS-RR, the first (or n -th) removed route is selected at random. This means that sometimes the best route, which could be part of the optimal solution, is also removed. In such cases, algorithm takes a long time, if ever, to rearrange the users in the solution to achieve a new feasible solution. Perhaps none of the new routes will be similar to the route that was removed. To overcome such drawback, the REPEATED ALNS based Route Removal (REP-ALNS-RR) is proposed, given by Algorithm 4.11. The goal is to perform ALNS-RR, i.e., random route removal multiple times μ_{max}^{rep-rr} . After each route removal, perform additional μ_{max}^{rr2} iterations of regular HALNS procedure without route removal part. Additionally, the parameter μ_{max}^{rr2} in REP-ALNS-RR is lower than the same parameter μ_{max}^{rr} in ALNS-RR. In this thesis, the expression $\mu_{max}^{rr2} \cdot \mu_{max}^{rep-rr} = \mu_{max}^{rr}$ is used to determine parameters for REP-ALNS-RR. This way, the execution times of ALNS-RR and REP-ALNS-RR are comparable.

Algorithm 4.11 Repeated ALNS based route removal operator

Input: Temporal solution s_{temp} , best solution s_{best} , best feasible solution s_{best_feas} , current penalty coefficient values p_{cur}

- 1: $j \leftarrow 0$
 - 2: **while** $j < \mu_{max}^{rep-rr}$ **do**
 - 3: Perform ALNS-RR
 - 4: $j \leftarrow j + 1$
 - 5: **end while**
-

4.5.4 Ejection pool

Ejection pool (EP) operator [159, 160] aims to remove all customers from one route and add them to the so-called ejection pool. The removed customers are reinserted in other routes only if the solution afterward is feasible. This algorithm performs searching only in the feasible solution space. If a customer from the ejection pool cannot be inserted anywhere in the current solution without the violation of constraints, the customer is given a penalty. In such case, additionally, up to k customers that minimize the custom objective function are removed from the solution and added to the ejection pool. The overall goal is to avoid removing customers that are hard to insert, i.e., have a large ejection penalty value.

The overview of the operator is given by Algorithm 4.12. First, each customer is initialized with the penalty value of $\varepsilon_c = 1$, and an empty ejection pool list L_{EP} is initialized. Then the route with the lowest number of customers is removed (greedy route removal operator) from the solution s , and the customers are added in the ejection pool list L_{EP} . The following procedure is repeated until the time limit criteria is not met: if list L_{EP} is not empty, the customer insertion-removal part is performed (lines 7-30); otherwise, the procedure succeeded in one route removal and the obtained solution s is set as best feasible solution s_{best_feas} , the penalties are reset and the

greedy route removal operator is performed again. At the end of each iteration, the improvement procedure (LS and exact procedure) is performed to improve the solution and find a better position of customers in the solution. The insertion-removal part starts with setting the last customer in the ejection pool list L_{EP} as the current customer c that will be inserted in the solution. This selection follows the LIFO strategy. If possible, the customer c is inserted in the best feasible position in the solution s and removed from the list L_{EP} . Otherwise, if it could not be inserted anywhere in the solution, the customer is marked as "troublesome", and the penalty of the customer ϵ_c is increased by one. Next, the goal is to find the position of insertion of customer c in the solution (in each vehicle v , in each position in vehicle i) with the removal of up to k customers from the same vehicle v that minimize the sum of penalties of selected k customers. Therefore, the forming tuple $\{v, i, L_r, \epsilon^{min}\}$ contains: vehicle v , position i , list of up to k removed customers L_r , and the minimum value of the sum of penalties ϵ^{min} . The removal of customers covers all permutations from only one customer removal to k customers removals in vehicle v but does not consider the removal of customer c that was just inserted. The insertion-removal part is marked as best so far c_{best} only if the result is a feasible vehicle route (does not consider that some customers are not inserted as they are waiting in the ejection pool list L_{EP}) and if it is better than the current best one c_{best} . The tuple c_{best} is initialized before insertion-removal part (line 13) with a maximum value for ϵ_{best}^{min} . The setting of the maximum value is standard in programming when minimizing some objective function, as then the first occurrence of a value lower than the maximum is stored as the minimum value. After the evaluation of the insert-removal part, the inserted customer c is removed from the vehicle at position i , and nested for loops are repeated for each vehicle and each position (without the position of the first depot instance). In the last part of the insert-removal procedure, if the tuple c_{best} is not empty, the insertion and removal is performed and the removed customers are added to list the L_{EP} . If the tuple c_{best} is empty, it means that the customer c could not be inserted anywhere in the solution s even with the removal of up to k customers from vehicle routes. In such case, the last customer in list L_{EP} is shifted at the beginning of the list, and the customer that was preceding him is set as the latest customer in the list.

Algorithm 4.12 Ejection pool operator

Input: Temporal solution s_{temp} , best solution s_{best} , best feasible solution s_{best_feas}

- 1: $s \leftarrow s_{best_feas}$
- 2: For each customer c initialize its penalty value ϵ_c to 1
- 3: $L_{EP} \leftarrow 0$ Initialize empty EP pool list
- 4: Perform one greedy route removal on s and add removed customers to L_{EP}
- 5: **while** time limit criteria is not met **do**
- 6: **if** L_{EP} is not empty **then**
- 7: $c \leftarrow$ Last customer in L_{EP}
- 8: $c_{insert} \leftarrow$ Try to insert customer at best feasible position in solution s
- 9: **if** c_{insert} is feasible **then**
- 10: Perform insertion c_{insert} and remove c from L_{EP}
- 11: **else**
- 12: Customer penalty $\epsilon_c \leftarrow \epsilon_c + 1$
- 13: Initialize best insertion-removal tuple c_{best} as $\{v_{best}, i_{best}, L_{r_best}, \epsilon_{best}^{min}\}$ with maximum value for ϵ_{best}^{min}
- 14: **for each** vehicle v in s **do**
- 15: **for each** position i in vehicle v without first depot position **and** customer at position $i \neq c$ **do**
- 16: Insert c at position i in vehicle v
- 17: Tuple $c_{new} \leftarrow$ Evaluate the removal of up to k customers c_1, \dots, c_k from vehicle v ($c_j \neq c, j = \{1, \dots, k\}$) that minimize the sum of penalties $\epsilon_1 + \dots + \epsilon_k$, and create the tuple c_{new}
- 18: **if** c_{new} is feasible **and** $f(c_{new}) < f(c_{best})$ **then**
- 19: $c_{best} \leftarrow c_{new}$
- 20: **end if**
- 21: Remove customer c at position i in vehicle v
- 22: **end for**
- 23: **end for**
- 24: **if** c_{best} is empty **then**
- 25: Replace the position of last customer in L_{EP} with previous one
- 26: **else**
- 27: Perform insertion of c in vehicle v_{best} at position i_{best}
- 28: Remove customers in list L_{r_best} from vehicle v_{best} and add them to L_{EP}
- 29: **end if**
- 30: **end if**
- 31: **else**
- 32: $s_{temp}, s_{best}, s_{best_feas} \leftarrow s$
- 33: For each customer c initialize its penalty value ϵ_c to 1
- 34: Perform one greedy route removal on s and add removed customers to L_{EP}
- 35: **end if**
- 36: Perform improvement procedure (LS and exact) on s
- 37: **end while**

4.5.5 Evaluation

The evaluation was conducted to determine which route removal operator to use. The EVRPTW-FR problem was considered. In total, 12 instances were used for evaluation, the 01 and 02 instances for each problem type: C1, C2, R1, R2, RC1, and RC2. All route removal operators started with the same initial feasible solution and were run for 800 iterations, except the EP, which was run for 10 minutes. The REP-ALNS-RR outer loop iteration parameter was set to $\mu_{max}^{rep-rr} = 10$, while the internal loop iteration parameter was set to $\mu_{max}^{rr2} = 80$ times. The EP k parameter was set to 4. The results are presented in Table 4.6 with total vehicle cumulative number $\sum K$, and total execution time t_e (min). The ALNS-RR produced a slightly lower number of vehicles than REP-ALNS-RR. The random and greedy route removal operators produce solutions with a slightly larger number of vehicles. The EP produced the worst results, as it was able to remove only a couple of vehicle routes. The problem is a very narrow feasible search space. As some customers are removed, they are penalized, but on most occasions not due to the fact that they are hard to insert, but rather due to the CS configuration in the solution. As a result, most of the customers are hard to insert, and the number of vehicles is hardly reduced. Based on the results, only the ALNS-RR is selected as the main route removal operator of the used HALNS method.

Table 4.6: Comparison of route removal operators

Measure	Random	Greedy	ALNS-RR	REP-ALNS-RR	EP
$\sum K$	113	116	109	111	201
t_e	170.54	91.59	53.11	52.94	125.60

4.6 Local search

To improve the solution generated by destroy and repair operators, local search operators are used. In the design phase, all of the local search operators presented in section 3.2 were considered: intra relocate, intra exchange, Or-opt, intra station in, intra station out, inter relocate, inter exchange, inter cross exchange ($k = 3$), and inter 2-Opt*. The intra 2-opt was not considered as 2-opt needs a reversal of route direction, which is time consuming in problems with time windows. The strategy of best move per LS operator is used, which then imposes on the determination of the order of operators.

Most of the evaluations in the inter operators can be done in $\mathcal{O}(1)$, as in each vehicle route the change is done at only one position in the route, and each change can be evaluated in $\mathcal{O}(1)$. Intra operators change the order of customers within the route. Usually, the changes happen on two position in the route, and the change positioned earlier in the route effects the variables of other downstream change. The variables of users between the changed positions have to be recomputed. Therefore, the complexity of such evaluation is $\mathcal{O}(B)$, where B is the number of users between the first and second position of change in the route.

4.6.1 Order and selection

To determine the order of the operators, as well as which operator to use and which not to use, a statistical test study was conducted. Again 12 instances were used, the 01 and 02 instances for each problem type: C1, C2, R1, R2, RC1, and RC2. This time the EVRPTW-PR variant was considered.

First, the order of all used operators is determined as an average of the best solutions in 10 runs using all LS operators. The value of best solutions in each run is computed by equation 4.84. The $K(r)$ represents the sum of vehicles, $f_{dist}(r)$ sum of total traveled distance, and $t_e(r)$ sum of total execution time in minutes in the run r . α_{veh} , β_{dist} , and γ_{time} are function coefficients, and are set to 1000, 1, and 5, respectively. The large value of α_{veh} is used to award solutions with a lower number of vehicles. The distance part has higher importance than the time as $\gamma_{time} = 5$, and the distance value has an order of 10^3 . Due to a large number of operator order permutations ($9! = 362880$), here the operator order is determined by experience as follows: first inter operators, then intra CS operators, and lastly, rest of the intra operators. Such order makes sense, as first, the users are exchanged between multiple routes. Then the CS configuration is improved, and lastly users within the same route are exchanged. The results are presented in Table 4.7. The measures used are the sum of all vehicles $\sum K$, sum of total traveled distances $\sum f_{dist}$ (in order of 10^3), sum of total execution time $\sum t_e[min]$ and average value of the evaluation function $\bar{\lambda}$ (in order of 10^3). First, the inter operator orders are determined, while the intra operators are fixed by experience as follows: intra station out, intra station in, Or-opt, intra

exchange, and intra relocate. The worst results are achieved for order 7 which has the highest vehicle number and also the highest execution time. The best orders turned out to be order 5 and order 8, which have similar objective function values. For the final order of the inter operators, the order 5 is selected: inter relocate, 2-Opt*, inter exchange and inter cross exchange, as it is faster than the order 8. In order 8, the inter cross exchange at the beginning consumes a lot of time. In the second step, the inter operators are fixed, and only the order of intra station in and intra station out are observed. As it can be seen, the order intra station out, intra station in (9) produces slightly better results than intra station in, intra station out (10). It can be seen, that assumed order of intra operators produced higher execution times, as after the change of order, the execution time significantly decreased. Again, this order is fixed for the third step in which additional 4 orders of intra operators are observed. The order intra exchange, Or-opt, intra relocate (13) produces the best results as it decreases the number of vehicles the most. As a result, the final order of the operators used in LS is: inter relocate, 2-Opt*, inter exchange, inter cross exchange, intra station out, intra station in, intra exchange, Or-opt and intra relocate.

Table 4.7: Order of LS operators

Num	Order	ΣK	$\Sigma f_{dist} \cdot 10^3$	$\Sigma t_e [min]$	$\lambda \cdot 10^3$
1	2-Opt*, inter exchange, inter relocate, inter cross exchange	1078	149.804	1012.98	127.845
2	Inter exchange, 2-Opt*, inter relocate, inter cross exchange	1078	150.065	1010.12	127.857
3	Inter relocate, inter exchange, 2-Opt*, inter cross exchange	1081	149.836	1013.37	128.151
4	Inter exchange, inter relocate, 2-Opt*, inter cross exchange	1079	150.146	1016.76	127.998
5	Inter relocate, 2-Opt*, inter exchange, inter cross exchange	1076	150.580	1007.61	127.696
6	2-Opt*, inter relocate, inter exchange, inter cross exchange	1078	150.047	1011.23	127.859
7	Inter cross exchange, 2-Opt*, inter exchange, inter relocate	1084	150.204	1049.65	128.668
8	Inter cross exchange, inter relocate, inter exchange, 2-Opt*	1075	149.658	1049.41	127.712
9	Intra station out, intra station in	1079	150.130	475.21	125.289
10	Intra station in, intra station out	1080	149.930	479.23	125.389
11	Intra relocate, Or-opt, intra exchange	1081	150.465	460.17	125.447
12	Intra relocate, intra exchange, Or-opt	1080	150.293	472.48	125.391
13	Intra exchange, Or-opt, intra relocate	1074	149.827	476.50	124.765
14	Intra exchange, intra relocate, Or-opt	1076	149.723	476.89	124.956

$$\lambda(r) = \alpha_{veh}K(r) + \beta_{dist}f_{dist}(r) + \gamma_{time}t_e(r) \quad (4.84)$$

After the operator order has been determined, additional tests were conducted to remove some operators if they do not contribute to the overall solution quality [53]. The same instances were used as in the previous test, and the same objective function given by equation 4.84. First, the reference value λ_{ref} was computed as an average of best solutions in 10 runs using all LS operators. Then, in each test, one LS operator is removed from the configuration, and again 10 runs were conducted. The relative difference between λ_{ref} and λ_i (i for each removed operator) is computed as $\Delta\lambda_i = \lambda_i - \lambda_{ref}$. The results of conducted tests are presented in Table 4.8. As

it can be seen, two operators could be removed: inter exchange and inter cross exchange as the quality of solution increased when they were removed. The inter cross exchange severely influences the execution time and therefore could be removed from the configuration. Inter exchange can be observed as a special case of inter cross exchange with $k = 1$. Removing both operators would decrease the overall solution quality; therefore, in the final version of LS configuration, only the inter cross exchange is removed due to the large execution time. Also, the intra CS out operator turned out to be the most important operator that significantly increases the overall execution time and solution quality. The order of operators based on their significance is the following: intra CS out, intra CS in, intra exchange, intra relocate, intra Or-Opt, inter relocate, inter exchange, and inter cross exchange.

Table 4.8: Removal of LS operators

Remove	ΣK	$\Sigma f_{dist} \cdot 10^3$	$\Sigma t_e [min]$	$\bar{\lambda} \cdot 10^3$	$\Delta \lambda$
No removal	1078	149.804	580.95	125.685	–
Inter exchange	1078	149.696	574.56	125.642	–42.82
Inter relocate	1080	150.260	537.34	125.712	27.55
Inter cross exchange	1080	149.938	445.21	125.219	–465.29
Intra CS in	1084	150.089	568.06	126.249	563.96
Intra CS out	1080	152.035	105.18	128.462	2777.53
Intra exchange	1083	149.903	582.34	126.202	516.82
Intra Or-Opt	1081	149.892	538.02	125.779	94.21
Intra relocate	1082	151.248	556.37	126.106	421.48

For EVRP variants dealing with different charger types at CSs, a new local search operator named intra CS change, is proposed. This is a new operator in the literature, as none of the researchers dealing with different charger types considered the LS phase to improve the solution. The intra CS change operator tries to change the charger type of a CS in a vehicle route. In LS, it is applied after the intra CS out operator, and before intra exchange operator.

4.6.2 Generate feasible solution with LS

In the HALNS, given by algorithm 4.1, when the solution s_{best} is infeasible a special procedure is called to make the solution feasible (line 21). This procedure is based on the LS with penalty coefficients multiplied by 100 [53, 121] to favor the moves that are feasible. If afterward, still no feasible solution is found, then the weights are additionally multiplied by 10, to a total of 1000, and LS is performed again. In the end, if the solution is still infeasible, no further steps are performed at this stage of the algorithm.

4.7 Optimal CS placement

As already reported by several researchers the configuration of CSs in a route significantly affects the overall routing cost [18, 22, 53, 83, 161]. Therefore, the aim is to determine the optimal CS configuration in a fixed route and also to determine the charging amount. As already mentioned, this is a FVRCP problem (NP-hard) [83], that seeks the minimum-cost refueling policy for a given fixed sequence of customers. To keep the computational times low, Schiffer. et al. [53] solved the Elementary Shortest Path Problem with Resource Constraints (ESPPRC) [162] for each route on a limited search tree by extending the fixed route with CSs. An example of a search tree is presented in Figure 4.20, where fixed route of customers (0, 1, 2, 0) is extended with CSs (blue circles). The tree is expanded by paths: (i) one path without CS, (ii) one path for each nearest CS, and (iii) one path for each combination of two nearest CSs. A path is further extended if it is feasible and is not dominated by another path.

In this section, different exact procedures are described for different observed problem variants.

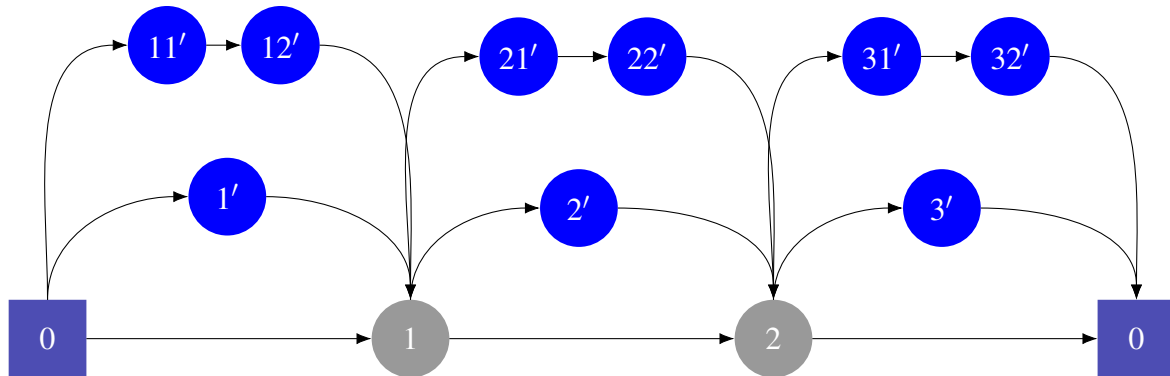


Figure 4.20: Search tree in ESPPRC

4.7.1 EVRPTW-PR

The exact procedure for optimal CS placement in EVRPTW-PR is based on solving ESPPRC with the Dynamic Programming (DP) technique proposed by Schiffer. et al. [53] (Figure 4.20) for ELRPTW-PR. The overview of the procedure is given by Algorithm 4.13. First, the CS visits are removed from each vehicle route v in the current solution s . Then the user before u_{bef} is set as the first user in the current vehicle route v , and the path p is initialized with user u_{bef} . Further on, the tree is initialized with path p . Then, the while loop is used to loop over users and gradually add them one by one in tree paths. First, the tree t is extended with path p_1 that does not contain a CS. Then, the tree is extended by paths p_i for each nearest station cs_i between the preceding user u_{bef} and current user u_{cur} . As in the EVRPTW test instances at maximum two

CSs are necessary to visit any customer, the tree is further extended by path p_j for each nearest station between the preceding CS cs_i and the current user u_{cur} . In the end, the best path from the tree is selected and set as vehicle route v .

Algorithm 4.13 Dynamic programming - ESPPRC

Input: Current solution s

- 1: **for each** vehicle route v in s **do**
- 2: $v \leftarrow$ Remove CSs visits from v
- 3: $u_{bef} \leftarrow$ First user in vehicle v
- 4: Add u_{bef} to initial path
- 5: Add path p to initial tree t
- 6: **while** vehicle v is not processed **do**
- 7: $u_{cur} \leftarrow$ Next user in vehicle v
- 8: **for each** path p in tree t **do**
- 9: $p_1 \leftarrow$ Extend path p with u_{cur}
- 10: Extend tree t with path p_1
- 11: $LCS_{u_{bef}, u_{cur}} \leftarrow$ List of possible CSs between u_{bef} and u_{cur}
- 12: **for each** CS cs_i in $LCS_{u_{bef}, u_{cur}}$ **do**
- 13: $p_i \leftarrow$ Extend path p with cs_i and u_{cur}
- 14: Extend tree t with path p_i and perform dominance rules
- 15: **if** path p_i is not dominated **then**
- 16: $LCS_{cs_i, u_{cur}} \leftarrow$ Set of possible CSs between cs_i and u_{cur}
- 17: **for each** CS cs_j in $LCS_{cs_i, u_{cur}}$ **do**
- 18: $p_k \leftarrow$ Extend path p with cs_i, cs_j and u_{cur}
- 19: Extend tree t with path p_k and perform dominance rules
- 20: **end for**
- 21: **end if**
- 22: **end for**
- 23: **end while**
- 24: **end for**
- 25: Vehicle route $v \leftarrow$ Best path in tree t
- 26: **end for**

To evaluate the paths in the tree, the Resource Extension Functions (REFs) are used. The original REFs for branch-price-and-cut framework for EVRPTW-PR and EVRPTW-FR have been proposed by Desaulniers et al. [55] and are presented by equations 4.85-4.91, while the corresponding variables are presented in Table 4.9. The presented variables are similar to the corridor-based variables for EVRPTW-PR 4.12-4.16, but there are some differences that limit their values to the feasible ones. First, the new variable T_j^F is used to track the number of CSs visited up to user j . Second, in cases where there was not a CS before user j the additional recharging time X_{ij} is not added to T_j^{tMin} and the value T_j^{rtMax} is updated only by value h_{ij} . Lastly, the T_j^{Max} is additionally limited to late time window l_j .

$$T_j^{cost} = T_i^{cost} + d_{ij} \quad (4.85)$$

Table 4.9: REFs - EVRPTW-PR variables

Symbol	Description
T^{cost}	Cost of a path
T_i^F	Number of CSs up to user i
T_i^{tMin}	Earliest arrival time at user i with charging the minimum amount as possible
T_i^{tMax}	Earliest arrival time at user j with charging the maximum amount as possible at preceding CS with assumption that minimum charging was performed before that
T_i^{rtMax}	Time required to charge at user j to maximum with assumption that minimum charging was performed before that
S_{ij}	Slack time at user j
X_{ij}	Additional charging time that needs to be added at preceding CSs

$$T_j^F = T_i^F + \begin{cases} 1 & j \in F' \\ 0 & else \end{cases} \quad (4.86)$$

$$T_j^{tMin} = \begin{cases} \max(e_j, T_i^{tMin} + t_{ij} + s_i) & T_i^F = 0 \\ \max(e_j, T_i^{tMin} + t_{ij} + s_i) + X_{ij} & else \end{cases} \quad (4.87)$$

$$X_{ij} = \begin{cases} \max(0, \max(0, T_i^{rtMax} - S_{ij}) + h_{ij} - H) & i \in F' \\ \max(0, \max(0, T_i^{rtMax} - \min(S_{ij}, T_i^{tMax} - T_i^{tMin})) + h_{ij} - H) & else \end{cases} \quad (4.88)$$

$$S_{ij} = \max(0, e_j - (T_i^{tMin} + t_{ij} + s_i)) \quad (4.89)$$

$$T_j^{tMax} = \begin{cases} \min(l_j, \max(e_j, T_i^{tMin} + T_i^{rtmax} + t_{ij})) & i \in F' \\ \min(l_j, \max(e_j, T_i^{tMax} + t_{ij} + s_i)) & else \end{cases} \quad (4.90)$$

$$T_j^{rtMax} = \begin{cases} T_i^{rtMax} + h_{ij} & T_i^F = 0 \\ \min(H, \max(0, T_i^{rtMax} - S_{ij}) + h_{ij}) & i \in F' \\ \min(H, \max(0, T_i^{rtMax} - \min(S_{ij}, T_i^{tMax} - T_i^{tMin})) + h_{ij}) & else \end{cases} \quad (4.91)$$

To speed up the search process in tree extension, the dominance rules play an important part. The basic idea is to associate each extended path with a label that contains basic values that describe a path. This so-called labeling technique is commonly used to perform dominance checks, and cuts in exact procedures [15, 16, 55, 101]. The dominance rules are performed on the labels for the extended tree paths, which results in the removal of paths that are dominated by other paths, meaning that they will always provide a worse solution than the other paths in the tree. In EVRPTW-PR, the label L_j at user j is defined by equation 4.92. The label L_j is

feasible if expression 4.93 is satisfied. Initial values of all variable are set to zero, $T_0^{cost} = T_0^F = T_0^{tMin} = X_{i0} = T_0^{tMax} = T_0^{rtMax} = 0$.

$$L_j = (T_j^{cost}, T_j^F, T_j^{tMin}, T_j^{tMax}, T_j^{rtMax}) \quad (4.92)$$

$$T_j^{tMin} \leq l_j \wedge T_j^{tMin} \leq T_j^{tMax} \wedge T_j^{rtMax} \leq H \quad (4.93)$$

To determine the dominance rules in partial recharge strategy, the relation between the maximum possible additional recharge time T_i^{max} and earliest service start time T_i^t has to be determined. The relation is presented in Figure 4.21 by four line segments, each corresponding to different labels L_1 to L_4 . The labels are associated with four paths in the tree ending at the same user and containing one visit to a CS. The slope of labels is the same and linear, as slack times were already considered for charging, and the same charging rate g is used in all CSs. If a path does not contain a CS, the earliest minimum begin time T_i^{tMin} and earliest maximum begin time T_i^{tMax} are equal, and the line reduces to a single point. For label L_1 there are two extremities: (i) if the minimum amount of recharging was performed at previous CS corresponding to the earliest service start time T_1^{tMin} , then the maximum possible additional recharge time is determined by the value of T_1^{rtMax} , and the first extremity is point $(T_1^{tMin}, T_1^{rtMax})$; and (ii) if maximum amount of recharging was performed at previous CS corresponding to the earliest service start time T_1^{tMax} , then in the ideal situation the maximum possible additional charging time is limited to $T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin})$, and the second extremity is point $(T_1^{tMax}, T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin}))$. To clarify this, the time T_i^{max} is the additional time that can be added to charge the battery to the fullest if the minimum or maximum charging amount was already performed. The slope is equal to one, as an increase of one time unit in charging time increases the value of T_i^t by one and decreases the value of T_i^{max} by one.

To apply dominance rules, all label components need to have nondecreasing functions (REFs) [55], as they have, but standard dominance rules cannot be applied due to the relation between T_i^{max} and T_i^t . A label L_1 dominates a label L_2 if $T_1^{cost} \leq T_2^{cost}$, $T_1^F \leq T_2^F$, $T_1^{tMin} \leq T_2^{tMin}$, and for every service start time $T_2 \in [T_2^{tMin}, T_2^{tMax}]$, there exists a service start time $T_1 \in [T_1^{tMin}, T_2]$ such that $T_1^{rtMax} - (T_1 - T_1^{tMin}) \leq T_2^{rtMax} - (T_2 - T_2^{tMin})$. This means that for every maximum possible additional charging time achievable by label L_2 the label L_1 can achieve same or lower maximum possible charging time at the same service time or earlier.

The example of different labels is presented in Figure 4.21. The label L_1 dominates L_4 as for every $T_4 \in [T_4^{tMin}, T_4^{tMax}]$ there exists a $T_1 \in [T_1^{tMin}, T_4]$ such that $T_1^{max} \leq T_4^{max}$. Graphically this means that the label L_4 has to be higher than the value at the extremity point $(T_1^{tMax}, T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin}))$ of label L_1 and right of the extremity point $(T_1^{tMin}, T_1^{rtMax})$ of label L_1 (it must be in the area between the red dashed lines). This means that label L_1 cannot dominate labels

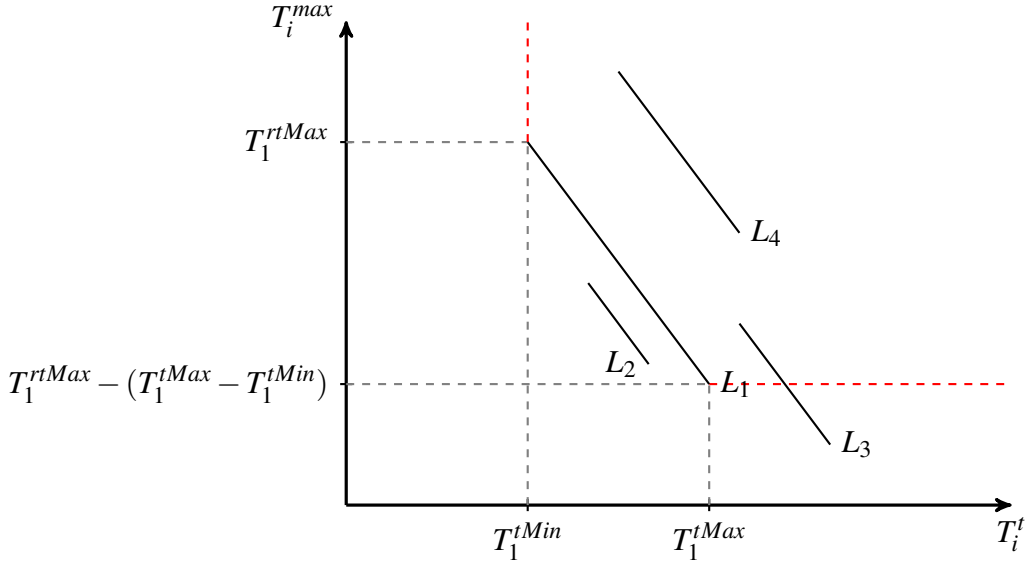


Figure 4.21: Maximum possible additional recharging time T_i^{max} in relation to earliest service start time T_i^t

L_2 and L_3 , as well as that none of the other labels can dominate or be dominated. L_1 cannot dominate L_2 as for $T_2 = T_2^{tMin}$, there does not exist $T_1 \in [T_1^{tMin}, T_1]$ such that $T_1^{rtMax} - (T_1 - T_1^{tMin}) \leq T_2^{rtMax} - (T_2 - T_2^{tMin})$. L_1 cannot dominate label L_3 , as the previous expression is valid only for the partial interval of $[T_3^{tMin}, T_3^{tMax}]$ (up to the point of crossover with red line). The final two dominance rules can be expressed by equations 4.94 and 4.95. The first equation is related to the lower extremity (horizontal red dashed line) as maximum service time T_i^{tMax} is considered for both labels L_1 and L_2 , then for L_1 to dominate L_2 , the maximum possible additional recharging time T_1^{max} has to be lower than T_2^{max} . The second equation is related to the higher extremity (vertical red dashed line) as the earliest service time T_1^t is set to T_2^{tMin} , then for L_1 to dominate L_2 , the maximum possible additional recharging time T_1^{max} at service time T_2^{tMin} has to be lower than $T_2^{max} = T_2^{rtMax} - (T_2^{tMin} - T_2^{tMin}) = T_2^{rtMax}$.

$$T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin}) \leq T_2^{rtMax} - (T_2^{tMax} - T_2^{tMin}) \quad (4.94)$$

$$T_1^{rtMax} - (T_2^{tMin} - T_1^{tMin}) \leq T_2^{rtMax} \quad (4.95)$$

The example of a search tree with labels and dominance rules for vehicle route $(0, 5, 2, 6, 0)$ (dark red line) in optimal EVRPTW-PR solution on instance C101-5 (Figure 4.22) is presented in Figure 4.23. The CSs are removed from the route, and the depot is set as the active user. The red border line that surrounds the user represents the part of the route that has been processed. The label values given by equation 4.92, are represented with the rectangle in the exact same order. In the beginning they are all set to zero and represented as rounded integers, although in a real application, they have decimal values. To simplify the search tree, only one CS insertion

is considered between two consecutive users. First, three new paths are generated: one without the CS, and two with CSs 2 and 3 inserted between users 0 and 5. Here, the path is not extended with the CS 1, as this CS is located at the depot, and there is no point in going to a CS immediately after the depot (the same goes for the ending depot instance). The second extended label is dominated by the first extended label as $38 \leq 85$ (T^{cost}), $0 \leq 1$ (T^F), $176 \leq 176$ (T^{Min}), $132 - (176 - 176) \leq 210 - (176 - 176)$ (equation 4.94), and $132 - (176 - 176) \leq 210$ (equation 4.95) are all satisfied. The first label does not dominate the third label, although it has a lower cost value due to the equation 4.94. Only the non-dominated paths are extended into the next iteration. In the end of the tree, two paths remain that are not dominated by each other, and the one that has a lower T^{cost} value is selected as the best path (red border color).

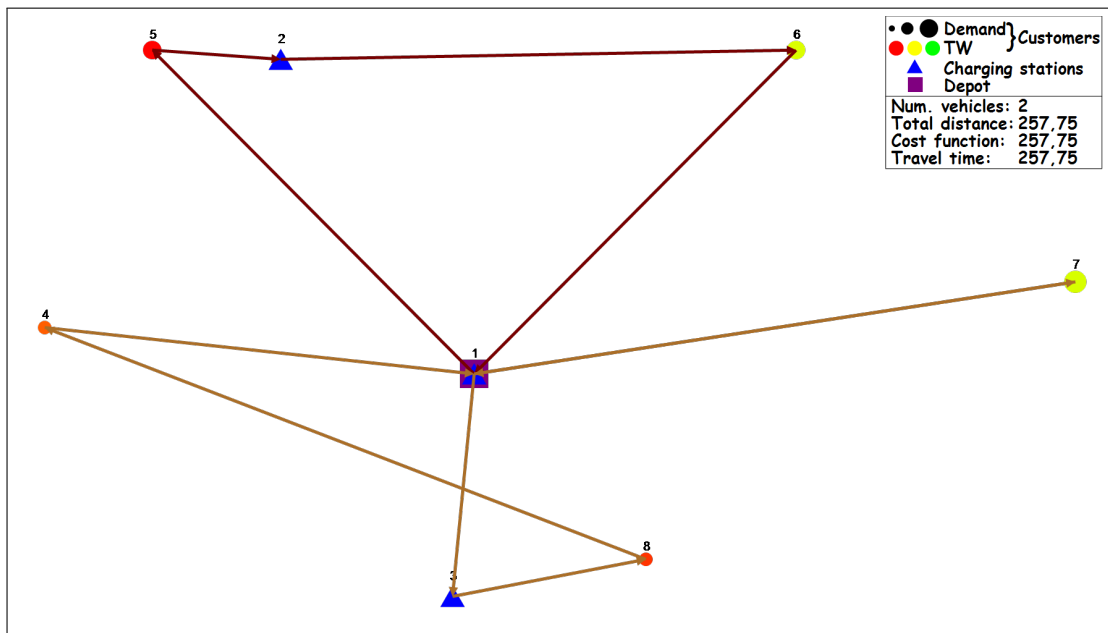


Figure 4.22: Optimal EVRPTW-PR solution for instance C101-5

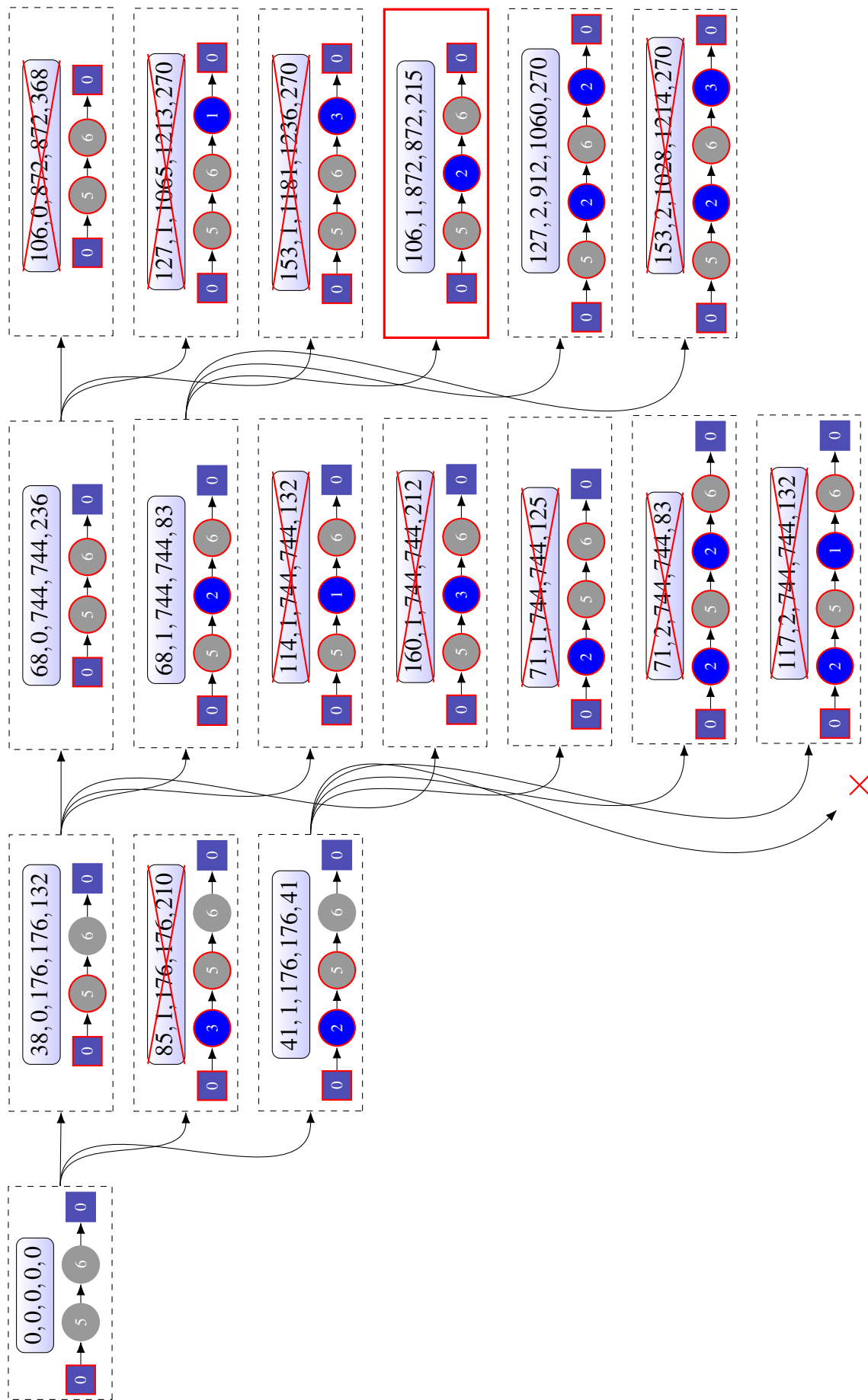


Figure 4.23: Example of tree with labels

4.7.2 EVRPTW-FR and EVRPTWDCS-FR

For EVRPTW-FR and EVRPTWDCS-FR problems, a similar procedure based on solving the ESPPRC with DP technique is used. To evaluate the paths in the tree, new REFs are proposed for EVRPTWDCS-FR. The EVRPTW-FR is considered as a special case of EVRPTWDCS-FR where all CSs have the same charger type. From variables presented in Table 4.9 for EVRPTW-PR, only the T^{cost} , T_i^{tMax} and T_i^{rtMax} are used, with T_i^{rtMax} representing the maximum possible charging amount expressed in the unit of capacity. The REFs are given by equations 4.96-4.98, the label is given by equation 4.99, the feasibility conditions are given by equation 4.100, and the dominance rules are given by equation 4.101. As there is no partial recharging, the rules are easier to determine. The rest of the procedure is completely the same as for the EVRPTW-PR problem.

$$T_j^{cost} = T_i^{cost} + d_{ij} \quad (4.96)$$

$$T_j^{rtMax} = \begin{cases} e_{ij} & i \in F' \\ T_i^{rtMax} + e_{ij} & else \end{cases} \quad (4.97)$$

$$T_j^{tMax} = \begin{cases} \max(e_j, T_i^{tMax} + t_{ij} + g^{m_i} T_i^{rtMax}) & i \in F' \\ \max(e_j, T_i^{tMax} + t_{ij} + s_i) & else \end{cases} \quad (4.98)$$

$$L_j = (T_j^{cost}, T_j^{tMax}, T_j^{rtMax}) \quad (4.99)$$

$$T_j^{tMax} \leq l_j \wedge T_j^{rtMax} \leq Q \quad (4.100)$$

$$T_1^{cost} \leq T_2^{cost} \wedge T_1^{tMax} \leq T_2^{tMax} \wedge T_1^{rtMax} \leq T_2^{rtMax} \quad (4.101)$$

4.7.3 EVRPTWDCS-PR

The previous procedure based on solving the ESPPRC for EVRPTW-PR cannot be applied for optimal CS placement in EVRPTWDCS-PR. The reason is similar to the already discussed additional charging time (a_{ij}^{add}) in section 4.3.4. Here this variable is represented as X_{ij} . The value X_{ij} could be easily computed and added in EVRPTW-PR, because the earliest begin time increases linearly, while in EVRPTWDCS-PR this value depends on the charger types used in preceding CSs, and the amount recharged with each charger. To determine the optimal CS placement and charging amount in EVRPTWDCS-PR, the procedure proposed by Keskin et al. [22] is used. The idea is to formulate the FVRCP as a MILP program and solve it with

commercially available software. The MILP program proposed by Keskin et al. [22] is based on the MILP program for EVRPTWDCS-PR presented in section 2.5 with a tighter formulation presented by Bruglieri et al. [143]. The proposed MILP program by Keskin et al. [22] contains several errors, which are all corrected and highlighted in the next paragraph.

First of all, all CSs are removed from vehicle route v . A set of customers in route without depots is labeled as $V = (1, \dots, N)$, while a set that contains a instance of depot (0 or $N + 1$) is labeled as V_0 or V_{N+1} . The variable $\Theta_{i,i+1}^m$ denotes the amount of energy recharged using charger type m if vehicles visit a CS between customers i and $i + 1$. The binary variable x_{ij} (4.116) indicates if arc from customer i to CS j (and then from j to $i + 1$) is used in the solution or not. Variables $a_{i,i+1}$ and $b_{i,i+1}$ (4.114) indicate the charger type used on arc from i to $i + 1$, which can also include a visit to CS ($i \rightarrow j \rightarrow i + 1$). The minimization function given by equation 4.102 consists of a straightforward part that includes charging costs per charger type in CSs, and a second part which represents the amount of used energy that was recharged at the depot with the slowest charger type before any routing begun. In the original formulation, the $i \in F$ is used, and not $i \in V_0$, which is impossible as arc variable x_{ij} is defined from $i \in V$ to $j \in F$. One additional equation that was missing in the original formulation, which ensures the stability of the model, is the limitation of the number of CS between customer i and $i + 1$ to 1 (equation 4.115). This equation needs to be added as $\Theta_{i,i+1}^m$ represents the amount of charging at a single CS, and not the multiple ones. The equation 4.104 ensures travel time flow, meaning that begin time at customer i plus the possible travel and charging times must be lower or equal the begin time at customer $i + 1$. In the original formulation, first instead of the proposed t_{jk} , the variable $t_{j,i+1}$ is actually needed, and second, the proposed equation is missing the part $\sum_{j \in F} t_{i,i+1}(1 - x_{ij})$ which ensures the travel time feasibility in cases when there is no CS between users. The equation 4.105 ensures travel time feasibility. The equation 4.106 ensures energy consumption flow. If CS j is used, rest battery capacity at user i is decreased by the energy consumed from user i to CS j , and from CS j to user i , and increased by the charging amount at CS. If CS j is not used, rest battery capacity at user i is decreased by the energy consumed from user i to user $i + 1$. Equation 4.107 ensures that there is enough rest battery capacity to reach CS j . Equations 4.108 and 4.109 ensure that vehicle cannot be charged more than Q . The starting vehicle rest battery capacity value is set to Q (equation 4.110). The equations 4.111-4.113 determine the charger type used. The equation 4.115, was not present in the original formulation, and here is added to limit the possible outcomes of the binary variables a_i and b_i to three cases: (i) $a_i = 1, b_i = 0$, (ii) $a_i = 0, b_i = 1$, and (iii) $a_i = 0, b_i = 0$. It is important to note that the proposed model considers only one CS insertion between the two consecutive customers.

$$\min \sum_{i \in V_0} \sum_{m \in M} (c^m \Theta_{i,i+1}^m) + c^3(Q - y_{N+1}) \quad (4.102)$$

$$\sum_{j \in F} x_{ij} \leq 1 \quad i \in V_0 \quad (4.103)$$

$$\tau_i + s_i + \sum_{j \in F} (t_{ij} + t_{j,i+1})x_{ij} + \sum_{j \in F} t_{i,i+1}(1 - x_{ij}) + \sum_{m \in M} g^m \Theta_{i,i+1}^m \leq \tau_{i+1}, \quad i \in V_0 \quad (4.104)$$

$$e_i \leq \tau_i \leq l_i, \quad i \in V_{N+1} \quad (4.105)$$

$$y_i - \left[e_{i,i+1} \left(1 - \sum_{j \in F} x_{ij} \right) + \sum_{j \in F} (e_{ij} + e_{j,i+1})x_{ij} \right] + \sum_{m \in M} \Theta_{i,i+1}^m = y_{i+1}, \quad i \in V_0 \quad (4.106)$$

$$y_i - \sum_{j \in F} e_{ij}x_{ij} \geq 0, \quad i \in V_0 \quad (4.107)$$

$$\sum_{m \in M} \Theta_{i,i+1}^m \leq Q \sum_{j \in F} x_{ij}, \quad i \in V_0 \quad (4.108)$$

$$\sum_{m \in M} \Theta_{i,i+1}^m \leq Q - \left(y_i - \sum_{j \in F} e_{ij}x_{ij} \right), \quad i \in V_0 \quad (4.109)$$

$$y_0 = Q \quad (4.110)$$

$$0 \leq \theta_{i,i+1}^1 \leq Qa_{i,i+1}, \quad \forall i \in V_0 \quad (4.111)$$

$$0 \leq \theta_{i,i+1}^2 \leq Qb_{i,i+1}, \quad \forall i \in V_0 \quad (4.112)$$

$$0 \leq \theta_{i,i+1}^3 \leq Q(1 - a_{i,i+1} - b_{i,i+1}), \quad \forall i \in V_0 \quad (4.113)$$

$$a_{i,i+1}, b_{i,i+1} \in \{0, 1\}, \quad \forall i \in V_0 \quad (4.114)$$

$$a_{i,i+1} + b_{i,i+1} \leq 1, \quad \forall i \in V_0 \quad (4.115)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in V_0, \quad \forall j \in F \quad (4.116)$$

To compare the execution time between the DP technique for EVRPTW-PR and solving a MILP for EVRPTWDCS-PR, the EVRPTWDCS-PR was considered with the same charger types and recharging costs equal to 1. The commercial software [CPLEX Optimizer IBM](#) was

used to solve the proposed MILP program. The EVRPTW instance C101 was considered for comparison with in total 2000 iterations. The results are presented in Table 4.10. In total, the DP was applied 394 times and the MIP solver 550 times. The a-priori assumption was that the DP would have a lower execution time, but the results show that the CPLEX has a slightly lower execution time. This occurs as a single-core implementation was used for DP, and CPLEX uses parallelization.

Table 4.10: Comparison between DP and MILP solver

	Applied	Total time [s]	Average time [s]
DP	394	14.56	0.037
MIP	550	17.56	0.032

4.8 Speed-up techniques and implementation

To improve the overall execution time, several speed up techniques are applied and described in this section. Besides the speed up techniques, the important part of the HALNS method is the implementation, which severely influences the execution time.

4.8.1 Infeasible arc removal

Some arcs in the instance are infeasible all the time as they violate problem constraints [163, 164]. As proposed by Schneider et al. [18] and Schiffer et al. [53] an arc (i, j) is considered to be infeasible if any of the constraints given by equations 4.117-4.120 are satisfied. The equation 4.117 represents vehicle capacity violation. The equation 4.118 represents time window violation as the service starts at the earliest possible time at user i and still the arrival time at user j is greater than late time window l_j . The equation 4.119 represents the violation of the depot time window, as there is not enough time to go from i to j and then, afterward, to arrive at the depot before its late time window l_0 . The equation 4.120 represents the battery capacity violation when there is not enough energy to traverse the arc $hd_{ij} > Q$. Additionally, it covers the case when the vehicle charges to full immediately before user i at CS m , and immediately after the user j at CS n , and there is not a combination of CSs m and n that can produce an energy feasible partial route $\phi = (m, i, j, n)$. During the development of the HALNS method, it was noted that removing infeasible arcs significantly reduces the search space and overall execution time. The number of infeasible arcs per EVRPTW instance groups is provided in Table 4.11. The column Total represents the total number of arcs in the instance group, while the Load, Time, and Battery columns represent the number of violated arcs by load capacity, time window, and battery capacity, respectively. The column Percentage represents the percentage of infeasible arcs in the total number of arcs in the instance group. As it can be seen, in instance

types 1 (with narrow time windows) larger number of arcs are infeasible (up to 35%), with most of them caused by time window violation. It can also be seen that not a single load capacity violation is present in all instances. The battery capacity is a cause of fewer infeasible arcs than time windows especially on problems with wide time windows (C2, R2, RC2).

$$i, j \in V \wedge q_i + q_j > C \quad (4.117)$$

$$i \in V'_0, j \in V'_{N+1} \wedge e_i + s_i + t_{ij} > l_j \quad (4.118)$$

$$i \in V'_0, j \in V' \wedge e_i + s_i + t_{ij} + s_j + t_{j,N+1} > l_0 \quad (4.119)$$

$$i, j \in V \wedge \forall m \in F'_0, n \in F'_{N+1} : h(d_{mi} + d_{ij} + d_{jn}) > Q \quad (4.120)$$

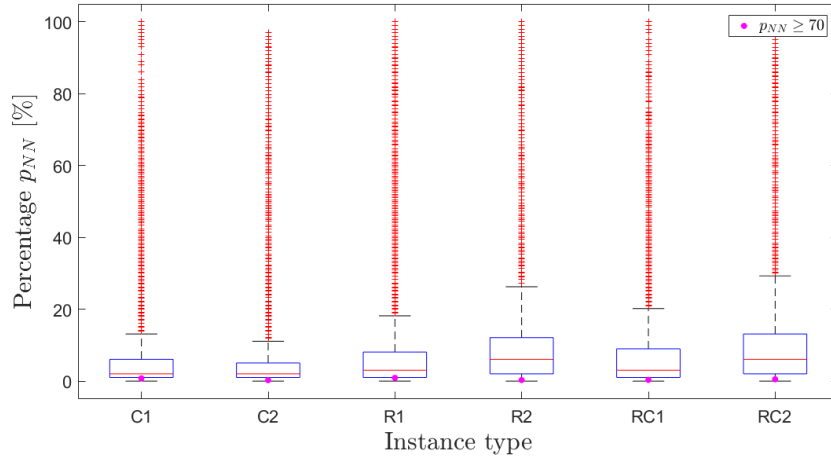
Table 4.11: Infeasible arcs

Instance	Total	Load	Time	Battery	Percentage
C1	132840	0	31703	10296	31.62
C2	118080	0	23562	13	19.96
R1	177120	0	36192	14655	28.71
R2	177120	0	17093	1707	10.61
RC1	118080	0	28002	12431	34.24
RC2	118080	0	15528	0	13.15

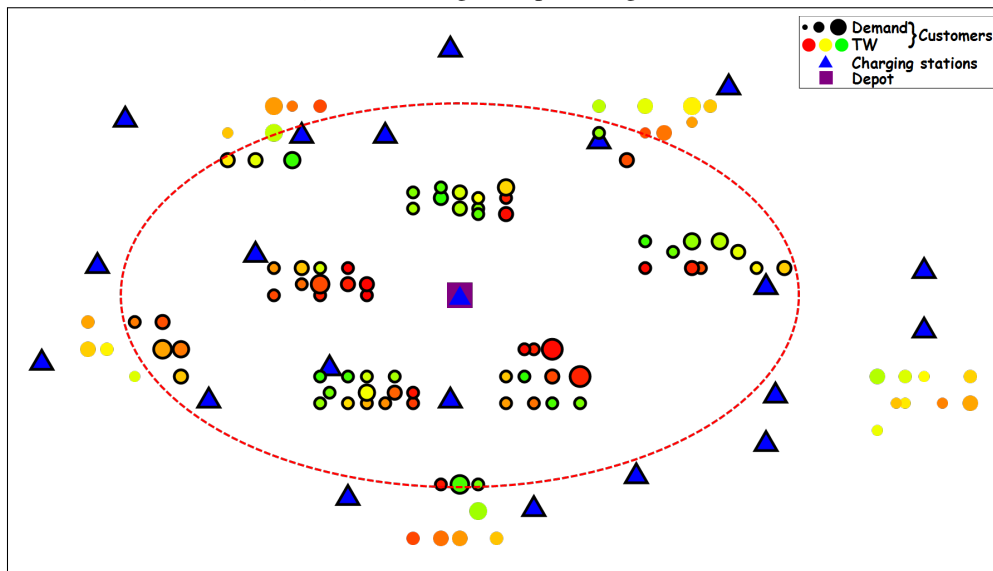
4.8.2 Nearest neighbors in local search

As noted by Christiaens et al. [146], in most BKS solutions, customers that are close to each other tend to be in the same vehicle route or in other close vehicle routes. This means that moves that lead to the improvement of the current solution often change the position of the customers that are close to each other. This could also be rephrased as: there is no point of inserting a customer between two "close" customers when this new customer is far away from both customers. Such moves almost never lead to a better solution even when time windows or battery capacity constraints are considered. Having that in mind, all of the LS operators are additionally limited to explore only the nearest neighbors' solution space. In the pre-processing step, for each arc (i, j) a list of limited nearest neighbors is created. This limited nearest neighbor list is determined as the threshold percentage p_{iNN} of all neighbors sorted in the ascending order.

Figure 4.24a presents the box plot example of computed nearest neighbor measures in the LS procedure for different EVRPTW-PR instance types. First of all, only the LS moves that lead to a better solution were considered. p_{NN} represents the percentage position of user move



(a) Nearest neighbors percentage in LS



(b) Nearest neighbors example on instance C101

Figure 4.24: Nearest neighbors example

(insertion or deletion) in the sorted list of all nearest neighbors. As it can be seen, all instance types have an upper whisker value up to 30%, while the lower whisker value is 0. Clustered instances have the lowest median and upper whisker value, while the randomly clustered instances have the largest median and upper whisker values. In regular statistical analysis, all of the red points can be considered as outliers, but in VRP problems, the threshold percentage p_{tNN} is usually determined in $[60, 100]\%$ interval. Such an interval is used, as some moves that include neighbors that are 60 – 80% far away, can still lead to good local optima solutions. To express how rare are such moves are, the magenta points are added. These values represent the percentage of moves in the total number of moves, in which neighbors are more than 70% away. The results are in $[0.26, 0.91]\%$ range. Figure 4.24b show an example of users nearest to the depot with $p_{tNN} = 70\%$ on EVRPTW instance C101. The users that are regarded as neighbors have a black color border. All of the customers within the red ellipse are considered as nearest

neighbors, as well as all CSs in instances, as CSs significantly influence the solution quality. The CS located at the depot is not considered as its nearest neighbor, as there is no point of going to this CS directly from the depot at the beginning of the route. The same goes for end depot instance, as there is no point in first visiting the CS located at the depot and then going to the depot.

4.8.3 Nearest CSs

In many steps of the HALNS method, CSs are inserted in the solution. In such cases, the selection of a CS that produces the lowest cost is often performed. Instead of looping through all CSs and selecting the best one, the set of possible CSs between users can be reduced. The criteria used for reducing the set is the distance between the CS and users. This can be done, as in all problem variants the energy consumption linearly depends on the traveled distance. Therefore, the closer CS would always produce a better solution in terms of total distance traveled than the CS that is further away. For each arc (i, j) , a set of nearest possible CSs is determined in the pre-processing step, based on the procedure proposed by Keskin et al. [22]. Let arc $(i, i + 1)$ be an arc that connects two consecutive users i and $i + 1$, and let the list of nearest possible CSs, L_{ij}^{nCS} be initialized as empty. In the beginning, all CSs in an instance ($j \in F$) are considered. The CS j is added to the list L_{ij}^{nCS} if it is not dominated by any CS k , already added in the list. If CS j dominates some of the CSs in the list, those CSs are removed from the list. The CS j is dominated by CS k if equation 4.121 is satisfied, meaning that CS k is both closer to user i and user $i + 1$. The example is shown in Figure 4.25 where dominated CSs have a red edge. None of the CSs j_1, j_2 and j_3 can dominate each other. The j_4 is dominated by j_1 and not j_3 , and j_5 is dominated by j_3 . This reduced CS set significantly reduces the search space and overall execution time in procedures oriented on CSs insertions and removals, i.e., solving ESPPRC with DP.

$$d_{ij} > d_{ik} \wedge d_{j,i+1} > d_{k,i+1} \quad (4.121)$$

4.8.4 LS inter-intra relation

The inter-intra relation is related to the order of the LS operators used. As already mentioned, first, the inter operators are performed that reposition users between different routes, and then intra operators are used to improve each vehicle route. The basic idea of inter-intra relation is the following: if in the current iteration not a single inter operator changed the vehicle route v , there is no point in performing intra operators on this route if the local optima with intra operators for a route v was already achieved in the previous iteration. Having that in mind, a binary flag indicating intra update or no-update of the vehicle route is added. This approach can

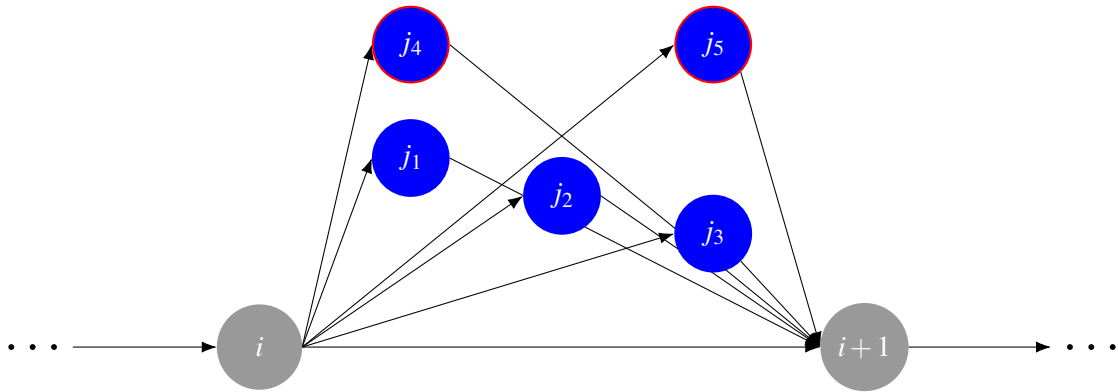


Figure 4.25: Nearest CSs between users

significantly reduce overall LS execution time, as it avoids searching in the same solution space without further improvements.

4.8.5 Minimum number of vehicles

For problems with load capacity, the theoretical lower bound on the number of vehicles can be determined by equation 2.17 in section 2.2, as the ceiling integer value of the ratio of the sum of customers' demand q_i and vehicle load capacity C . During the search, if the theoretical lowest number of vehicles is achieved, the operators for route removal finish their execution and are never called again. The search process afterward is only focused on searching in the solution space regarding the secondary objective: distance, travel time, total time or recharging cost.

4.8.6 Implementation

The HALNS is implemented as a single-thread code in the C# programming language. All tests were performed on a machine with Intel E5 processor and 32 GB of RAM. Throughout the code, the goal was to implement classes, methods, and variables to perform the execution as fast as possible. An overview of the stripped class diagram is presented in Figure 4.26, which contains the most basic classes: *User*, *Customer*, *Station*, *Vehicle*, *Solution* and *Params*. In the entire program, the comparison between real numbers is done with 10^{-7} precision. The implemented program contains 11365 lines of code, 209 classes and 3621 branches.

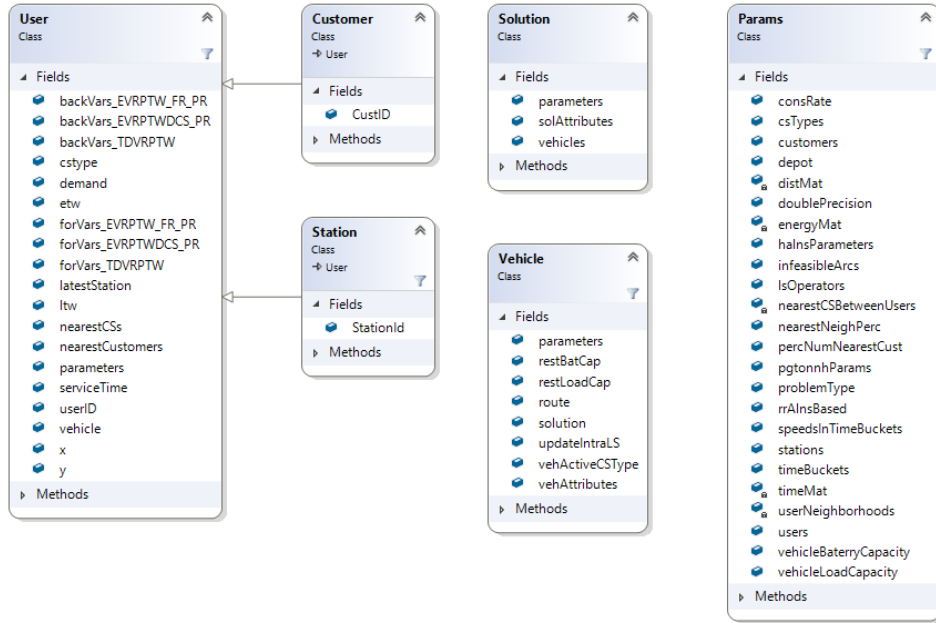


Figure 4.26: The most important classes in the implementation

4.9 Parameter tuning

Parameter tuning is an essential part of any algorithm, especially in cases where a large number of parameters is used, as in most metaheuristic methods. The tuning methodology is adopted from the literature [1, 32, 37, 147, 165]. First, a fair parameter setting is produced by an ad hoc trial-and-error phase. This parameter setting was found while developing the HALNS or has been adopted from the literature. Next, each parameter is allowed to take three values, while the rest of the parameters are kept fixed. For each parameter setting, the HALNS method is applied five times on the selected set of instances, and the setting that produced the best average deviation value from the best achieved solution is chosen. In the next step, the best parameter setting from the previous iteration is fixed, and the next parameter values are varied. This process continues until all parameters have been tuned. As there are many parameters in the HALNS method, and each parameter value is run five times, a following representative subset of instances is selected: R107, RC101, RC104, RC105, R205, and RC205 [1]. The instances C1 and C2 were omitted as they usually converge to the same solutions for different parameter values, and therefore do not provide information of how good the algorithm parameters values are. The EVRPTW-PR variant is considered as its evaluation is the fastest. The function used for the evaluation of solutions was already described in section 4.6, and it is given by equation 4.84. Here, the function values are set to $\alpha_{veh} = 200$, $\beta_{dist} = 1$, $\gamma_{time} = 1$. The lower value for α_{veh} is used as there are roughly 58 vehicles in one run. The HALNS parameters, their considered values, deviations, and final selected values are presented in Table 4.12. The final

selected parameter values are marked with bold font.

Table 4.12: Tuned parameters

Parameter	Initial	Values tested				Parameter	Initial	Values tested			
μ_{max}	V 4000 $\Delta\lambda$ —	1000	7000	15000	$(\delta_1, \delta_2, \delta_3)$	V (0.6, 0.3, 0.1) $\Delta\lambda$ —	(0.6, 0.3, 0.1)	(0.5, 0.4, 0.1)	(0.7, 0.2, 0.1)		
$\mu_{max}^{last_imp}$	V 2000 $\Delta\lambda$ —	1000	2000	4000	$n_{RCL}^{pgtonnh}$	V 5 $\Delta\lambda$ —	3	5	10		
μ_{rbks}	V 1500 $\Delta\lambda$ —	500	1000	2000	$k_{pgtonnh}$	V 3 $\Delta\lambda$ —	2	3	5		
μ_{uop}	V 50 $\Delta\lambda$ —	25	50	100	(μ_{low}, μ_{high})	V (0.1, 0.4) $\Delta\lambda$ —	(0.1, 0.4)	(0.2, 0.4)	(0.1, 0.6)		
μ_{upw}	V 50 $\Delta\lambda$ —	25	50	100	k_r	V 4 $\Delta\lambda$ —	2	4	6		
Δ_{Is}	V 0.2 $\Delta\lambda$ —	0.1	0.2	0.3	k_w	V 3 $\Delta\lambda$ —	2	4	6		
Δ_{exact}	V 0.02 $\Delta\lambda$ —	0.01	0.02	0.05	k_s	V 6 $\Delta\lambda$ —	2	4	6		
$(\alpha_0, \beta_0, \gamma_0)$	V 10 $\Delta\lambda$ —	10	100	1000	(χ_d, χ_e, χ_q)	V (6, 5, 4) $\Delta\lambda$ —	(4, 4, 2)	(6, 5, 4)	(8, 6, 3)		
$(\alpha_{min}, \beta_{min}, \gamma_{min})$	V 2 $\Delta\lambda$ —	0.1	1	10	(χ_{min}, χ_{max})	V (0.05, 0.15) $\Delta\lambda$ —	(0.02, 0.1)	(0.05, 0.15)	(0.1, 0.25)		
$(\alpha_{max}, \beta_{max}, \gamma_{max})$	V 80 $\Delta\lambda$ —	50	150	1000	n_{RCL}^{si}	V 5 $\Delta\lambda$ —	3	5	9		
ζ	V 1.2 $\Delta\lambda$ —	1.05	1.15	1.3	$(\Gamma_{min}, \Gamma_{max})$	V (0.8, 1.2) $\Delta\lambda$ —	(0.9, 1.1)	(0.8, 1.2)	(0.7, 1.3)		
$(\sigma_1, \sigma_2, \sigma_3)$	V (9, 4, 1) $\Delta\lambda$ —	(3, 2, 1)	(9, 4, 1)	(12, 6, 1)	μ_{max}^{rr}	V 200 $\Delta\lambda$ —	100	200	300		
ρ	V 0.8 $\Delta\lambda$ —	0.4	0.6	0.8	$(\mu_{uop}^{rr}, \mu_{upw}^{rr})$	V (20, 20) $\Delta\lambda$ —	(15, 15)	(25, 25)	(35, 35)		
w_0	V 0.1 $\Delta\lambda$ —	0.1	1	5	ζ^{rr}	V 2 $\Delta\lambda$ —	1.2	1.6	2		
P_{tNN}	V 0.9 $\Delta\lambda$ —	0.55	0.7	0.9	$(\alpha_0^{rr}, \alpha_{min}^{rr}, \alpha_{max}^{rr})$	V (15, 1, 50) $\Delta\lambda$ —	(10, 0.5, 50)	(100, 2, 100)	(100, 10, 150)		
n_{RCL}^{ls}	V 10 $\Delta\lambda$ —	10	20	50	$(\Delta_{Is}^{rr}, \Delta_{exact}^{rr})$	V (0.2, 0.02) $\Delta\lambda$ —	(0.2, 0.01)	(0.3, 0.02)	(0.4, 0.05)		

4.10 Results

In this section, the proposed HALNS method is tested and compared to other methods from the literature. First, the comparison is conducted on small instances formulated as MILP program and solved by commercial software CPLEX. Next, the HALNS method is compared to the ALNS method of Keskin et al. [1] (KESK-ALNS) on the EVRPTW-FR variant. In the end, HALNS is compared to the BKSs on different EVRP variants with different secondary objective functions: total traveled distance, total time, and recharging cost.

4.10.1 Small instances

As already mentioned in section 2.2, there are in total 36 small instances containing 5, 10 and 15 customers. First, EVRPTW problems for these instances are formulated as MILP programs and solved by the commercial software CPLEX. The MILP programs for the observed variants are already described in chapter 2: EVRPTW-FR - section 2.2, EVRPTW-PR - section 2.4 and EVRPTWDCS-PR - section 2.5. The commercial software MATLAB was also tested for solving the MILP programs, but CPLEX turned out to be a superior software. For details on solving the MILP for EVRPTW-FR problem with MATLAB, the reader is referred to Erdelić et al. [166]. In MILP formulations for EVRPTW-FR and EVRPTW-PR, the important aspect is parameter β which represents the number of virtual CSs per actual CS. This parameter affects the size of set F' , overall solution quality, and most important the execution time. The parameter β has not been discussed in any of the papers that deal with MILP and EVRPTW related problems [1, 18, 161]. Therefore, here the evaluation of parameter β will also be discussed. The EVRP problems use a hierarchical function, where solutions with a lower number of vehicles are preferred, although higher routing costs are achieved. This relation between primary and secondary objectives has not been discussed in any of the papers dealing with MILPs for EVRPTW. In this thesis, first, the MILP model is solved by minimizing the vehicle number, and then with found minimum vehicle number, the MILP model is solved again by minimizing the total traveled distance. In the second step, the found minimum number of vehicles is set as a constraint in the MILP program. For all MILP models, the maximum execution time was set to 2h. The results are presented in Table 4.13. For each instance, three β values were considered $\beta = 1, 2, 3$, and for each problem variant the following measures are used: number of vehicles K , CPLEX running time for the minimization of vehicle number t_e^K (min) and total traveled distance t_e^d (t_e^{rec}) (min), d total traveled distance or rec total recharging costs, and result R - indicating optimal O , feasible F or infeasible I solution for the respective minimization. The values with bold font represent either of the three things: (i) solutions that were not reported in the respective research papers, as the influence of β value has not been discussed, (ii) the value achieved is higher than the value in respective research papers meaning that it was reported

falsely, and (iii) improvement in partial recharge strategy to full recharge, or different charger types to single charger type.

It can be seen that limiting the number of visits to same CS $\beta = 1$, in 9 out of 36 increases the number of vehicles or total traveled distance. Additionally, on instance RC108-5 for EVRPTW-FR, the Schneider et al. [18] reported only one vehicle, whereas all of the reported optimal solutions contain two vehicles [166, 167]. On the instance RC204-15 for EVRPTW-FR, the CPLEX was able to found the BKSs of 384.86, which the MILP formulation in the original paper of Schneider et al. [18], was not able to find (the proposed metaheuristic was able to found this solution). In 4 instances, the partial strategy decreased the total traveled distance, while in additional 9 instances, the different charger types led to the better solutions in terms of recharging cost and vehicle number. The comparison of results on EVRPTWDCS-PR to the original research paper cannot be adequately done, as Keskin et al. [22] only reported cumulative sum of vehicles and total traveled distance and did not observe a different number of virtual CSs vertices.

Table 4.13: Solving small instances with CPLEX

Instance	β	EVRPTW-FR						EVRPTW-PR						EVRPTWDCS-PR					
		K	t_e^K	R	d	t_e^d	R	K	t_e^K	R	d	t_e^d	R	K	t_e^K	R	rec	t_e^{rec}	R
C101-5	1	2	0.01	O	257.75	0.02	O	2	0.01	O	257.75	0.01	O	2	0.01	O	253.29	0.01	O
	2	2	0.07	O	257.75	0.01	O	2	0.06	O	257.75	0.02	O	2	0.06	O	250.69	0.18	O
	3	2	0.09	O	257.75	0.12	O	2	0.47	O	257.75	0.21	O	2	0.36	O	250.69	0.73	O
C103-5	1	2	0.01	O	165.67	0.01	O	2	0.01	O	165.67	0.01	O	2	0.01	O	165.67	0.01	O
	2	1	0.01	O	176.05	0.01	O	1	0.01	O	175.37	0.15	O	1	0.02	O	175.37	0.07	O
	3	1	0.07	O	176.05	0.20	O	1	0.09	O	175.37	0.08	O	1	0.09	O	175.37	0.07	O
C206-5	1	1	0.01	O	245.34	0.01	O	1	0.02	O	244.37	0.01	O	1	0.01	O	244.37	0.01	O
	2	1	0.08	O	242.56	0.01	O	1	0.22	O	242.56	0.13	O	1	0.10	O	242.56	1.16	O
	3	1	7.42	O	242.56	0.15	O	1	27.07	O	242.56	0.12	O	1	14.275	O	242.56	66.59	O
C208-5	1	1	0.06	O	158.48	0.14	O	1	0.01	O	158.48	0.01	O	1	0.01	O	158.48	0.01	O
	2	1	0.40	O	158.48	0.20	O	1	0.18	O	158.48	0.02	O	1	0.01	O	158.48	0.05	O
	3	1	0.52	O	158.48	0.14	O	1	0.56	O	158.48	0.16	O	1	0.23	O	158.48	0.45	O
R104-5	1	2	0.13	O	136.69	0.12	O	2	0.17	O	136.69	0.01	O	2	0.01	O	136.69	0.01	O
	2	2	0.32	O	136.69	0.19	O	2	0.19	O	136.69	0.01	O	2	0.01	O	136.69	0.05	O
	3	2	0.39	O	136.69	0.15	O	2	0.82	O	136.69	0.03	O	2	0.23	O	136.69	0.19	O
R105-5	1	2	0.01	O	156.08	0.08	O	2	0.01	O	156.08	0.01	O	2	0.01	O	156.08	0.01	O
	2	2	0.33	O	156.08	0.21	O	2	0.15	O	156.08	0.01	O	2	0.01	O	156.08	0.03	O
	3	2	0.23	O	156.08	0.13	O	2	0.16	O	156.08	0.04	O	2	0.05	O	156.08	0.14	O
R202-5	1	1	0.25	O	128.78	0.21	O	1	0.02	O	128.78	0.01	O	1	0.01	O	128.78	0.01	O
	2	1	0.37	O	128.78	0.23	O	1	0.13	O	128.78	0.05	O	1	0.02	O	128.78	0.15	O
	3	1	1.47	O	128.78	0.11	O	1	1.76	O	128.78	0.03	O	1	0.51	O	128.78	2.75	O
R203-5	1	1	0.12	O	179.06	0.15	O	1	0.01	O	179.06	0.01	O	1	0.01	O	179.06	0.01	O
	2	1	0.78	O	179.06	0.18	O	1	0.45	O	179.06	0.01	O	1	0.09	O	179.06	0.31	O
	3	1	45.85	O	179.06	0.15	O	1	84.46	O	179.06	0.10	O	1	31.72	O	179.06	12.98	O
RC105-5	1	2	0.02	O	241.30	0.08	O	2	0.03	O	241.30	0.07	O	2	0.01	O	241.30	0.01	O
	2	2	0.38	O	241.30	0.17	O	2	0.68	O	233.77	0.14	O	2	0.17	O	233.77	0.49	O
	3	2	2.72	O	241.30	0.13	O	2	19.49	O	233.77	0.13	O	2	120	F	244.45	120	F
RC108-5	1	2	0.09	O	253.93	0.04	O	2	0.07	O	253.93	0.03	O	2	0.02	O	253.93	0.01	O
	2	2	0.11	O	253.93	0.03	O	2	0.46	O	253.93	0.01	O	2	0.12	O	253.93	0.26	O

Instance	β	EVRPTW-FR						EVRPTW-PR						EVRPTWDCS-PR					
		K	t_e^K	R	d	t_e^d	R	K	t_e^K	R	d	t_e^d	R	K	t_e^K	R	rec	t_e^{rec}	R
	3	2	120	<i>F</i>	383.62	1.33	<i>O</i>	2	120	<i>F</i>	383.62	1.90	<i>O</i>	1	120	<i>F</i>	–	120	<i>I</i>
C208-15	1	2	0.18	<i>O</i>	300.55	0.3	<i>O</i>	2	0.18	<i>O</i>	300.55	0.3	<i>O</i>	2	0.15	<i>O</i>	300.55	7.52	<i>O</i>
	2	2	120	<i>F</i>	300.55	0.12	<i>O</i>	2	120	<i>F</i>	300.55	0.07	<i>O</i>	1	11.96	<i>O</i>	339.21	120	<i>F</i>
	3	2	120	<i>F</i>	300.55	0.13	<i>O</i>	2	120	<i>F</i>	300.55	0.10	<i>O</i>	1	120	<i>F</i>	339.21	120	<i>F</i>
R102-15	1	5	9.43	<i>O</i>	420.47	0.27	<i>O</i>	5	43.94	<i>O</i>	419.32	0.33	<i>O</i>	5	120	<i>F</i>	433.33	120	<i>F</i>
	2	5	120	<i>F</i>	413.93	0.52	<i>O</i>	5	120	<i>F</i>	412.78	0.56	<i>O</i>	5	120	<i>F</i>	–	120	<i>I</i>
	3	6	120	<i>F</i>	419.64	0.66	<i>O</i>	5	120	<i>F</i>	412.78	2.39	<i>O</i>	–	120	<i>I</i>	–	120	<i>I</i>
R105-15	1	4	0.75	<i>O</i>	336.15	0.07	<i>O</i>	4	0.91	<i>O</i>	336.15	0.02	<i>O</i>	3	0.61	<i>O</i>	389.16	2.45	<i>O</i>
	2	4	120	<i>F</i>	336.15	0.07	<i>O</i>	4	120	<i>F</i>	336.15	0.09	<i>O</i>	3	120	<i>F</i>	–	120	<i>I</i>
	3	4	120	<i>F</i>	336.15	0.04	<i>O</i>	4	120	<i>F</i>	336.15	0.33	<i>O</i>	–	120	<i>I</i>	–	120	<i>I</i>
R202-15	1	2	0.33	<i>O</i>	384.60	0.06	<i>O</i>	2	1.11	<i>O</i>	384.60	0.20	<i>O</i>	2	0.46	<i>O</i>	400.91	10.88	<i>O</i>
	2	2	120	<i>F</i>	358.00	0.10	<i>O</i>	2	120	<i>F</i>	358.00	0.18	<i>O</i>	2	120	<i>F</i>	–	120	<i>I</i>
	3	2	120	<i>F</i>	358.00	0.36	<i>O</i>	2	120	<i>F</i>	358.00	0.39	<i>O</i>	2	120	<i>F</i>	410.92	120	<i>F</i>
R209-15	1	2	0.39	<i>O</i>	293.20	0.01	<i>O</i>	2	0.53	<i>O</i>	293.20	0.02	<i>O</i>	2	0.78	<i>O</i>	293.20	0.53	<i>O</i>
	2	1	120	<i>F</i>	313.24	0.85	<i>O</i>	1	120	<i>F</i>	313.24	0.59	<i>O</i>	1	120	<i>F</i>	331.24	120	<i>F</i>
	3	1	120	<i>F</i>	313.24	1.02	<i>O</i>	1	120	<i>F</i>	313.24	1.01	<i>O</i>	1	120	<i>F</i>	331.24	120	<i>F</i>
RC103-15	1	4	2.99	<i>O</i>	397.67	0.07	<i>O</i>	4	9.69	<i>O</i>	397.67	0.11	<i>O</i>	4	27.21	<i>O</i>	404.14	120	<i>F</i>
	2	4	120	<i>F</i>	397.67	0.23	<i>O</i>	4	120	<i>F</i>	397.67	0.46	<i>O</i>	4	120	<i>F</i>	404.14	120	<i>F</i>
	3	5	120	<i>F</i>	397.75	0.07	<i>O</i>	4	120	<i>F</i>	397.67	0.67	<i>O</i>	–	120	<i>I</i>	–	120	<i>I</i>
RC108-15	1	3	1.51	<i>O</i>	370.25	0.04	<i>O</i>	3	4.48	<i>O</i>	370.25	0.10	<i>O</i>	3	21.30	<i>O</i>	387.31	120	<i>F</i>
	2	3	120	<i>F</i>	370.25	0.10	<i>O</i>	3	120	<i>F</i>	370.25	0.15	<i>O</i>	–	120	<i>I</i>	–	120	<i>I</i>
	3	4	120	<i>F</i>	383.52	0.06	<i>O</i>	4	120	<i>F</i>	383.52	0.09	<i>O</i>	–	120	<i>I</i>	–	120	<i>I</i>
RC202-15	1	2	0.13	<i>O</i>	405.53	0.03	<i>O</i>	2	0.21	<i>O</i>	405.53	0.06	<i>O</i>	2	0.66	<i>O</i>	405.53	2.65	<i>O</i>
	2	2	120	<i>F</i>	394.39	0.03	<i>O</i>	2	120	<i>F</i>	394.39	0.06	<i>O</i>	2	120	<i>F</i>	400.52	120	<i>F</i>
	3	2	120	<i>F</i>	394.39	0.09	<i>O</i>	2	120	<i>F</i>	394.39	0.15	<i>O</i>	2	120	<i>F</i>	400.52	120	<i>F</i>
RC204-15	1	1	120	<i>F</i>	384.86	120	<i>F</i>	1	120	<i>F</i>	382.22	120	<i>F</i>	1	120	<i>F</i>	413.14	120	<i>F</i>
	2	1	120	<i>F</i>	384.86	120	<i>F</i>	1	120	<i>F</i>	382.22	120	<i>F</i>	1	120	<i>F</i>	413.14	120	<i>F</i>
	3	2	120	<i>F</i>	310.91	4.32	<i>O</i>	1	120	<i>F</i>	382.22	120	<i>F</i>	–	120	<i>I</i>	–	120	<i>I</i>

The comparison between solving a MILP model and proposed HALNS method on small instances is presented in Table 4.14. The vehicle number K , total distance traveled d , recharging cost rec , and execution time t_e (min) are reported for HALNS. The differences in used measures to CPLEX solutions are noted with ΔK , Δd , and Δt_e (min), respectively. The best CPLEX solution out of three possible ($\beta \in 1, 2, 3$) was used for comparison. The HALNS is able to achieve all optimal/non-optimal solutions found by CPLEX, and even in 13 cases (bold font) found a solution better than the non-optimal solution of CPLEX. In 5 instances, the used vehicle number was lower, while in 8 instances for the EVRPTWDCS-PR variant, the recharging cost was reduced. For CPLEX execution time, the sum of vehicle minimization time and secondary objective minimization was used. As it can be seen, the HALNS execution time is much lower than the execution time of CPLEX and shows all of the advantages of using a metaheuristic procedure instead of the exact procedure.

4. Hybrid adaptive large neighborhood search method

Table 4.14: Comparison between MILP program solved by CPLEX and HALNS method, on small EVRPTW instances

Instance	EVRPTW-FR						EVRPTW-PR						EVRPTWDCS-PR					
	<i>K</i>	<i>d</i>	<i>t_e</i>	ΔK	Δd	Δt_e	<i>K</i>	<i>d</i>	<i>t_e</i>	ΔK	Δd	Δt_e	<i>K</i>	<i>rec</i>	<i>t_e</i>	ΔK	Δrec	Δt_e
C101-5	2	257.75	0.03	0	0	-0.19	2	257.75	0.02	0	0	-0.66	2	250.69	0.14	0	0	-0.1
C103-5	1	176.05	0.02	0	0	-0.17	1	175.37	0.01	0	0	-0.16	1	175.37	0.1	0	0	0.01
C206-5	1	242.56	0.01	0	0	-7.57	1	242.56	0.01	0	0	-27.18	1	242.56	0.06	0	0	-1.21
C208-5	1	158.48	0.01	0	0	-0.65	1	158.48	0.01	0	0	-0.71	1	158.48	0.06	0	0	-0.01
R104-5	2	136.69	0.01	0	0	-0.52	2	136.69	0.01	0	0	-0.84	2	136.69	0.02	0	0	-0.04
R105-5	2	156.08	0.01	0	0	-0.35	2	156.08	0.01	0	0	-0.19	2	156.08	0.04	0	0	0.01
R202-5	1	128.78	0.01	0	0	-1.57	1	128.78	0.01	0	0	-1.78	1	128.78	0.05	0	0	-0.12
R203-5	1	179.06	0.01	0	0	-46.00	1	179.06	0.01	0	0	-84.55	1	179.06	0.05	0	0	-0.36
RC105-5	2	241.3	0.02	0	0	-2.83	2	233.77	0.01	0	0	-19.61	2	233.77	0.02	0	0	-0.65
RC108-5	2	253.93	0.02	0	0	-1.17	2	253.93	0.01	0	0	-8.73	2	253.93	0.07	0	0	-0.31
RC204-5	1	176.39	0.02	0	0	-120.16	1	176.93	0.01	0	0	-120.12	1	176.39	0.06	0	0	-3.47
RC208-5	1	167.98	0.02	0	0	-1.95	1	167.98	0.01	0	0	-4.52	1	167.98	0.06	0	0	-0.16
C101-10	3	393.76	0.09	0	0	-77.2	3	388.25	0.05	0	0	-120.31	3	382.93	0.2	0	0	-240.35
C104-10	2	273.93	0.08	0	0	-120.49	2	273.93	0.05	0	0	-120.12	1	267.6	0.14	0	0	-240.2
C202-10	1	304.06	0.07	0	0	-42.68	1	304.06	0.03	0	0	-120.35	1	304.06	0.15	0	0	-141.38
C205-10	2	228.28	0.06	0	0	-4.12	2	228.28	0.04	0	0	-13.36	1	283.29	0.1	0	0	-0.69
R102-10	3	249.19	0.04	0	0	-0.3	3	249.19	0.02	0	0	-4.27	3	249.19	0.06	0	0	-9.59
R103-10	2	207.05	0.07	0	0	-122.77	2	206.12	0.04	0	0	-120.90	2	206.12	0.12	0	0	-149.49
R201-10	1	241.51	0.07	0	0	-120.10	1	241.51	0.04	0	0	-120.09	1	241.25	0.12	0	-13.71	-122.13
R203-10	1	218.21	0.07	0	0	-119.96	1	218.21	0.03	0	0	-120.02	1	218.21	0.12	0	0	-240.06
RC102-10	4	423.51	0.04	0	0	-4.94	4	423.51	0.02	0	0	-115.76	4	415.99	0.06	0	-7.52	-16.73
RC108-10	3	345.93	0.05	0	0	-120.15	3	345.93	0.02	0	0	-120.18	3	345.93	0.08	0	0	-129.04
RC201-10	1	412.86	0.08	0	0	-92.80	1	412.86	0.04	0	0	-135.50	1	412.86	0.14	0	0	-2.37
RC205-10	2	325.98	0.08	0	0	-120.39	2	325.98	0.05	0	0	-120.34	1	590.82	0.14	-1	264.84	-121.22
C103-15	3	384.29	0.18	0	0	-128.16	3	348.46	0.11	0	0	-120.95	2	372.85	0.29	0	0	-119.84
C106-15	3	275.13	0.05	0	0	-2.13	3	275.13	0.04	0	0	-120	2	310.79	0.22	0	0	-123.32
C202-15	2	383.62	0.34	0	0	-122.94	2	383.62	0.29	0	0	-120.51	1	583.25	0.34	-1	202.02	-240.93
C208-15	2	300.55	0.12	0	0	-123.68	2	300.55	0.09	0	0	-120.22	1	339.21	0.18	0	0	-131.95
R102-15	5	413.93	0.10	0	0	-121.27	5	412.78	0.08	0	0	-120.65	5	420.15	0.17	0	-13.18	-121.72
R105-15	4	336.15	0.10	0	0	-119.99	4	336.15	0.05	0	0	-120.16	3	340.62	0.17	0	-48.54	-119.9
R202-15	2	358.00	0.36	0	0	-119.91	1	507.32	0.32	-1	149.32	-119.97	1	449.81	0.4	-1	65.21	-119.6
R209-15	1	313.24	0.21	0	0	-120.65	1	313.24	0.15	0	0	-120.45	1	313.24	0.43	0	-18	-239.58
RC103-15	4	397.67	0.12	0	0	-120.30	4	397.67	0.09	0	0	-120.59	4	397.67	0.15	0	-6.47	-241.04
RC108-15	3	370.25	0.09	0	0	-121.05	3	370.25	0.07	0	0	-120.26	3	370.25	0.15	0	-17.06	-141.15
RC202-15	2	394.39	0.23	0	0	-119.80	2	394.39	0.15	0	0	-120.08	1	648.05	0.41	-1	247.53	-242.2
RC204-15	1	384.86	0.30	0	0	-240	1	382.22	0.38	0	0	-239.93	1	382.22	0.34	0	-30.92	-241.33

4.10.2 Comparison between HALNS and KESK-ALNS

To validate the HALNS method, a direct comparison to the implementation of the KESK-ALNS [1] method is conducted. The initial solution is created in a similar way as in PGTONNH with CS placement immediately before or after the customer when an energy violation occurs. In PGTONNH, when a customer cannot be added to the current route, the optimization of CS placement with the best CS insertion strategy is conducted. In KESK-ALNS, customers are removed from the solution by one of the eight operators: random, worst distance, worst time, shaw, proximity-based, demand-based, time-based, and zone removal. Additionally, one of three options, for each customer removal operator, is selected at random: remove customer with the first preceding CS, remove customer with the first succeeding CS and remove customer only. For CS removal, two operators are applied: random and worst distance. For customer insertion five operators are applied: greedy, regret-2, regret-3, time-based, and zone; while for CS insertion operators greedy, best, and greedy with comparison are applied. To lower the number of vehicles the random route removal and greedy route removal operators are used. All of the removal and insertion operators have been tested as a part of the HALNS method, and described in section 4.4, while a more detailed description can be found in [1, 37, 166]. For detailed description of KESK-ALNS method applied to solve EVRPTW-FR problem, the reader is referred to Erdelić et al. [166].

Comparison of the proposed HALNS approach to the KESK-ALNS on EVRPTW-FR is presented in Table 4.15. Each method was run only once. The parameters presented by Keskin et al. [1] are used for KESK-ALNS. Values with bold font represent cases in which HALNS outperforms KESK-ALNS. In 41 out of 56 instances, HALNS produced a better solution, with in total 10 vehicles less and even further decrease in total traveled distance. The average execution time of HALNS is roughly 10 minutes lower than the average execution time of KESK-ALNS. The comparison shows the advantage of using exact procedure and objective function with penalties in terms of solution quality and overall execution time.

Table 4.15: EVRPTW-FR distance - comparison between HALNS and KESK-ALNS

Inst.	KESK-ALNS			HALNS			Init.	
	K	d	t_e	K	d	t_e	ΔK	Δd
C101	12	1053.83	19.79	12	1053.83	2.65	0	0.00
C102	11	1081.17	18.73	11	1058.85	4.34	0	-22.31
C103	11	1002.01	15.67	11	1001.81	5.98	0	-0.2
C104	11	1036.04	17	10	1006.55	5.65	-1	-29.49
C105	11	1094.87	19.08	11	1080.85	3.82	0	-14.03
C106	11	1066.88	19.33	11	1057.65	4.43	0	-9.23
C107	11	1093.57	19.75	11	1031.56	2.69	0	-62.01
C108	11	1015.68	16.08	10	1125.95	2.35	-1	110.27
C109	11	1042.96	15.3	10	1085.37	7.22	-1	42.41
C201	4	645.16	22.28	4	645.16	2.50	0	0.00
C202	4	648.84	19.65	4	645.16	4.7	0	-3.68
C203	4	645.55	20.91	4	644.98	6.61	0	-0.57
C204	4	638.09	20.52	4	638.53	7.92	0	0.44
C205	4	641.13	21.82	4	641.13	3.03	0	0.00
C206	4	638.17	20.76	4	638.17	4.18	0	0.00
C207	4	638.17	20.97	4	638.17	5.01	0	0.00
C208	4	638.17	21.93	4	638.17	4.75	0	0.00
R101	18	1691.45	16.93	18	1663.04	10.23	0	-28.42
R102	16	1497.46	14.67	16	1490.40	12.01	0	-7.06
R103	13	1363.38	15.21	13	1311.45	11.77	0	-51.92
R104	11	1100.84	14.76	11	1089.87	15.10	0	-10.97
R105	16	1404.25	15.84	15	1383.29	10.30	-1	-20.96
R106	14	1329.12	14.21	14	1305.55	7.71	0	-23.57
R107	12	1195.27	13.25	12	1176.90	10.12	0	-18.38
R108	11	1069.03	14.37	11	1046.68	5.24	0	-22.34
R109	14	1290.85	13.42	13	1214.47	12.14	-1	-76.38
R110	13	1167.93	13.18	12	1098.57	11.18	-1	-69.36
R111	12	1160.77	12.29	12	1103.92	9.16	0	-56.86
R112	11	1025.89	13.29	11	1016.63	15.65	0	-9.26
R201	3	1269.32	23.98	3	1269.61	11.19	0	0.29
R202	3	1062.14	23.85	3	1056.88	21.73	0	-5.26
R203	3	914.48	33.56	3	920.38	27.43	0	5.9
R204	2	818.99	34	2	801.02	36.48	0	-17.96
R205	3	1014.86	34.73	3	1000.30	14.38	0	-14.56
R206	3	940.93	35.58	3	922.19	18.42	0	-18.74
R207	3	822.77	36.06	2	875.96	25.13	-1	53.19
R208	2	739.65	43.84	2	748.39	52.97	0	8.74
R209	3	885.32	28.8	3	882.89	13.2	0	-2.43
R210	3	857.32	31.16	3	862.36	14.71	0	5.04
R211	3	803.46	34.73	2	840.16	15.51	-1	36.7
RC101	17	1761.90	15.04	16	1738.64	6.55	-1	-23.26
RC102	15	1596.65	14.5	15	1603.88	6.49	0	7.23
RC103	13	1399.51	15.23	13	1351.46	7.36	0	-48.05
RC104	12	1255.23	14.45	11	1228.69	10.56	-1	-26.54
RC105	14	1525.00	14.61	14	1493.43	9.77	0	-31.57
RC106	13	1445.62	14.55	13	1450.57	11.43	0	4.95
RC107	12	1275.89	13.46	12	1357.71	9.66	0	81.83
RC108	11	1197.66	14.35	11	1213.25	13.85	0	15.59
RC201	4	1460.51	21.51	4	1446.84	7.27	0	-13.67
RC202	3	1444.58	22.45	3	1421.47	16.06	0	-23.11
RC203	3	1132.66	24.99	3	1069.18	35.54	0	-63.48
RC204	3	902.42	22.43	3	901.82	32.17	0	-0.60
RC205	3	1351.36	85.87	3	1284.68	12.08	0	-66.68
RC206	3	1218.77	20.47	3	1218.01	21.84	0	-0.76
RC207	3	1055.57	20.69	3	1039.17	27.09	0	-16.40
RC208	3	867.38	22.97	3	836.29	21.40	0	-31.09
AVG	8.14	1088.15	21.77	7.96	1078.00	12.66	-0.18	-10.15
SUM	456	60936.48	1218.85	446	60367.91	708.71	-10	-568.58

4.10.3 EVRPTW-PR

The results on EVRPTW instances for EVRPTW-PR with distance minimization are presented in Table 4.16. The HALNS was run 10 times on each instance. The results are compared to the BKS of Hierman et al. [54] (HIER) and Keskin et al. [1] (KESK). Three columns are used to represent BKS values: name N , best vehicle number K , and total traveled distance d . The results of Shiffer et al. [53], which seem the best ones so far, were not considered. The authors stated that they compared their results to the results of Keskin et al. [1], but did not use the BKS values presented in the paper, but rather they used some different reference values which are not available in the literature. The problem is that the authors reported the number of vehicles as a difference to the used reference values, which are unknown. Therefore, an adequate comparison cannot be done. For the HALNS method, the following columns are presented: average vehicle number \bar{K} , best vehicle number K_{best} , the difference between BKS and HALNS best vehicle number ΔK , average total traveled distance \bar{d} , best total traveled distance d_{best} , relative difference Δd and percentage relative difference $\Delta_p d$ between BKS and best HALNS total traveled distance, total time of best HALNS solution tot_{best} , recharging cost of best HALNS solution rec_{best} , average execution time \bar{t}_e in minutes, and a number of CSs in the best solution, m . The relative percentage difference was computed as $\Delta_p d = \frac{d_{best} - d_{BKS}}{d_{BKS}} \cdot 100$. It is important to note that in distance difference computation, Δd and $\Delta_p d$, only the solutions that produced the minimum number of vehicles by HALNS method were considered, because otherwise, with a higher number of vehicles, the total traveled distance decreases which then influences the comparison. The summary of the results is presented in the last two rows as the average and sum values. Additionally, to evaluate the PGTNNH heuristic used to create an initial solution, average initial vehicle number \bar{K}_{ini} and average total distance traveled \bar{d}_{ini} columns are added. All rows in which the HALNS produced a better solution regarding the vehicle number or the total traveled distance are represented with bold font.

In 44 out of 56 instances, the HALNS found a better solution, with in total 9 vehicle less, and -0.18 average relative percentage difference of total traveled distance, meaning that even if the lower number of vehicles is found, the HALNS finds a better user configuration in terms of distance minimization. The average value of the average vehicle number \bar{K} has almost the same value as the average of the BKSs vehicle number, indicating the good average performance of HALNS. The columns for total time and recharging cost were added to show how much the total time increases, and to verify the recharging cost computation with a single charger type. In terms of average execution time, HALNS (6.61 min) outperforms both HIER (9.96 min) and KESK (11.14 min). It can be noted that Shiffer et al. [53], although hard to compare, produced some solutions that are both better and worse than HIER, KESK, or HALNS, with the lowest execution time of 4.31 minutes. The initial solution constructed by PGTNNH on average produced 6 vehicles more than the best solution, while in the overall sum, it produced

310 vehicles more. The initial total traveled distance is significantly higher than the best one. This is expected as a feasible solution in EVRPTW is hard to achieve, and therefore a larger number of vehicles has to be used. On average, 10 CSs are visited per instance, or 1.31 per vehicle route, with in total 560 CS visited in all instances.

The total time has significantly larger values than the total traveled distance. To show the importance of the total time objective function, in Table 4.17 the results for the minimization of the total time in EVRPTW-PR are presented. The difference in the implementation is the move evaluation in $\mathcal{O}(n)$, which is why an average execution time is significantly larger, resulting in 27.11 minutes. The columns for BKS solutions were not presented as the minimization of total time has not yet been addressed in the literature. Instead, the solutions from distance minimization are used as reference values for the comparison, vehicle number K_{best}^{dist} and total time tot_{best}^{dist} . It can be seen that the total minimum number of vehicles is the same as in distance minimization and that total time decreased in 53 out of 56 instances. On average the total time is reduced by 5.27% at the expense of the increase in total traveled distance. Interestingly, the total number of visited CSs increased from 560 in distance minimization to 741 in total time minimization, as the costs of traveling to CS are somewhat reduced. This mostly refers to the fact that spare time can be utilized more for charging at the expense of total traveled distance.

Table 4.17: EVRPTW-PR total time - results

Inst.	K_{best}^{dist}	tot_{best}^{dist}	\bar{K}	K_{best}	ΔK	\overline{tot}	tot_{best}	Δtot	$\Delta_p tot$	d_{best}	rec_{best}	\bar{t}_e	m
C101	12	12647.62	12	12	0	11747.5	11721.44	-926.18	-7.32	1214.74	1214.74	8.915	16
C102	10	11619.45	10.25	10	0	11435	11477.35	-142.1	-1.22	1160.41	1160.41	9.4	15
C103	10	11411.29	10	10	0	11108.79	11055.34	-355.95	-3.12	1078.5	1078.5	10.838	13
C104	10	11057.74	10	10	0	10465.17	10463.36	-594.38	-5.38	939.12	939.12	23.4	7
C105	10	11436.3	10	10	0	11335.96	11335.96	-100.34	-0.88	1083.22	1083.22	4.149	12
C106	10	11376.02	10	10	0	11264.54	11260.29	-115.73	-1.02	1086.74	1086.74	4.708	15
C107	10	11254.18	10	10	0	11229.58	11159.48	-94.7	-0.84	1105.85	1105.85	4.182	12
C108	10	11017.02	10	10	0	10992.04	10976.64	-40.38	-0.37	1040.68	1040.68	5.132	12
C109	10	10744.74	10	10	0	10609.13	10601.91	-142.83	-1.33	971.74	971.74	10.024	9
C201	4	10125.99	4	4	0	10027.26	10027.26	-98.73	-0.98	633.6	633.6	12.207	4
C202	4	10125.99	4	4	0	10027.26	10027.26	-98.73	-0.98	633.6	633.6	26.522	4
C203	4	10074.33	4	4	0	10020.08	10020.08	-54.25	-0.54	631.66	631.66	38.362	4
C204	4	10343.46	4	4	0	10016.92	10016.92	-326.54	-3.16	629.95	629.95	37.646	3
C205	4	10020.61	4	4	0	10020.61	10020.61	0	0	629.95	629.95	15.223	3
C206	4	10021.16	4	4	0	10015.92	10015.92	-5.24	-0.05	633.09	633.09	20.562	3
C207	4	10021.43	4	4	0	10016.2	10016.2	-5.23	-0.05	633.09	633.09	24.537	3
C208	4	10021.43	4	4	0	10013.31	10013.31	-8.12	-0.08	635.01	635.01	20.161	4
R101	17	3587.38	17	17	0	3476.27	3470.15	-117.23	-3.27	1869.62	1869.62	12.852	45
R102	15	3123.33	15.5	15	0	3023.42	2980.85	-142.48	-4.56	1584.32	1584.32	22.415	31
R103	12	2650.75	12.5	12	0	2624.45	2650.75	0	0	1304.24	1304.24	22.102	24
R104	11	2386.27	11	11	0	2268	2248.74	-137.53	-5.76	1078.86	1078.86	23.445	14
R105	14	2903.37	14	14	0	2818.16	2817.59	-85.78	-2.95	1441.54	1441.54	17.971	29
R106	13	2722.36	13	13	0	2592.78	2570.67	-151.69	-5.57	1298.63	1298.63	16.954	21
R107	11	2391.69	11	11	0	2351.18	2341.7	-49.99	-2.09	1147.96	1147.96	24.081	18
R108	10	2239.85	10.5	10	0	2227.49	2239.85	0	0	1030.44	1030.44	24.99	15
R109	12	2569.17	12	12	0	2504.22	2483.04	-86.13	-3.35	1239.39	1239.39	23.556	21
R110	11	2395.6	11	11	0	2279.56	2276.36	-119.24	-4.98	1103.77	1103.77	30.008	17
R111	11	2338.33	11	11	0	2347.16	2347.16	8.83	0.38	1140.51	1140.51	21.183	19
R112	11	2322.52	11	11	0	2202.02	2202.02	-120.5	-5.19	1043.17	1043.17	32.718	15
R201	3	2892.61	3	3	0	2826.55	2826.55	-66.06	-2.28	1609.59	1609.59	57.833	17
R202	3	2936.01	3	3	0	2739.8	2739.8	-196.21	-6.68	1578.75	1578.75	45.674	13
R203	3	2930.01	3	3	0	2146.67	2146.67	-783.34	-26.74	1051.38	1051.38	77.609	9
R204	2	1987.21	2	2	0	1886.77	1886.77	-100.44	-5.05	843.15	843.15	69.769	4
R205	3	2718.76	3	3	0	2430.51	2430.51	-288.25	-10.6	1283.76	1283.76	26.371	17
R206	3	2763.95	3	3	0	2163.32	2163.32	-600.63	-21.73	1064.45	1064.45	38.925	6
R207	2	1993.88	2	2	0	1891.3	1891.3	-102.58	-5.14	855.53	855.53	29.98	3
R208	2	1911.93	2	2	0	1786.94	1786.94	-124.99	-6.54	743.48	743.48	37.033	2
R209	3	2726.92	3	3	0	2147.3	2147.3	-579.62	-21.26	1055.56	1055.56	22.116	8
R210	3	2785.5	3	3	0	2100.14	2100.14	-685.36	-24.6	1010.57	1010.57	28.433	7
R211	2	1892.85	2	2	0	1885.58	1885.58	-7.27	-0.38	849.31	849.31	21.032	5
RC101	15	3193.93	15	15	0	3137.99	3137.08	-56.85	-1.78	1815.18	1815.18	9.921	28
RC102	14	2951.51	14	14	0	2870.18	2864.53	-86.98	-2.95	1602.56	1602.56	13.616	23
RC103	12	2705.8	12	12	0	2635.19	2593.36	-112.44	-4.16	1413.48	1413.48	10.602	20
RC104	11	2416.92	11	11	0	2348.18	2338.45	-78.47	-3.25	1198.84	1198.84	17.804	14
RC105	13	2861.33	13.5	13	0	2836.68	2818.79	-42.54	-1.49	1536.32	1536.32	7.679	22
RC106	13	2832.9	13	13	0	2660.66	2659.51	-173.39	-6.12	1441.78	1441.78	10.613	17
RC107	11	2448.38	11	11	0	2427.25	2427.25	-21.13	-0.86	1252.3	1252.3	12.768	17
RC108	11	2391.45	11	11	0	2312.82	2311.79	-79.66	-3.33	1187.96	1187.96	15.862	14
RC201	4	3650.43	4	4	0	3337.37	3323.58	-326.85	-8.95	2090.59	2090.59	26.697	19
RC202	3	2838.67	3	3	0	2722.59	2722.53	-116.14	-4.09	1612.01	1612.01	50.101	10
RC203	3	2708.69	3	3	0	2419.05	2411.74	-296.95	-10.96	1307.61	1307.61	66.175	7
RC204	3	2679.15	3	3	0	2153.38	2120.7	-558.45	-20.84	1011.86	1011.86	70.709	7
RC205	3	2661.49	3	3	0	2609.11	2601.62	-59.87	-2.25	1431.83	1431.83	32.645	14
RC206	3	2693.07	3	3	0	2558.23	2558.16	-134.91	-5.01	1441.48	1441.48	41.42	12
RC207	3	2515.67	3	3	0	2266.24	2266.24	-249.43	-9.92	1180.79	1180.79	53.504	10
RC208	3	2425.03	3	3	0	2097.31	2097.31	-327.72	-13.51	997.65	997.65	75.079	8
AVG	7.61	5115.42	7.65	7.61	0	4936.73	4930.28	-185.14	-5.27	1151.63	1151.63	27.11	13.23
SUM	426	286463.43	428.25	426	0	276457.09	276095.73	-10367.7	-295.13	64491.17	64491.17	1518.21	741

4.10.4 EVRPTW-FR

The results on EVRPTW instances for the EVRPTW-FR problem are presented in Table 4.18. The same columns as in EVRPTW-PR variant are used. The results are compared to the BKSs of: Schneider et al. [18] (SCHN), Goeke et al. [20] (GOEK), Hierman et al. [29] (HIE1), Hierman et al. [54] (HIE2), Keskin et al. [1] (KESK) and Schiffer et al. [53] (SCHI). It can be seen that HALNS was not able to reduce the number of vehicles on four instances: R105, R106, R110 and RC102. Nevertheless, nine new BKSs (bold fonts) are found for instances: C109, R104, R108, R211, RC105, RC107, RC201, RC205 and RC206. The total traveled distance is lower than in BKSs, due to the higher number of vehicles. Compared to the partial recharge strategy it can be seen, that partial recharging produced 18 vehicles less, with also lower total traveled distance and total time. As a result partial recharge strategy uses 48 CSs more. The selection of the recharging strategy depends on the delivery problem and decision makers. With partial strategy better results can be achieved, but in real life it is stressful for the driver to visit CSs more frequently, and not only that, but also to perform time-precise charging and leave a CS without a possibly fully charged battery. The initial values produced by PGTONNH have almost the same values as in the partial variant, as same heuristic is used.

The results on EVRPTW instances for EVRPTW-FR with total time minimization are presented in Table 4.19. The best number of vehicles K_{best}^{dist} and the best total times tot_{best}^{dist} achieved with distance minimization are used as reference values for the comparison. As the objective function changed, the total time improved in almost all instances, except the ones in which HALNS was not able to find a lower number of vehicles (three instances: C103, C108, and R109). Similar to the results of the partial strategy, here also the total traveled distance increased. The number of visited CS did not increase as much as in partial recharge strategy but still an increase is present. The average execution time increased significantly compared to the distance minimization due to the $\mathcal{O}(n)$ evaluation, while compared to the total time in partial strategy, the execution time is roughly the same.

Table 4.19: EVRPTW-FR total time - results

Inst.	K_{best}^{dist}	tot_{best}^{dist}	\bar{K}	K_{best}	ΔK	\bar{tot}	tot_{best}	Δtot	Δ_{ptot}	d_{best}	rec_{best}	m	\bar{t}_e
C101	12	12904.61	12	12	0	12013.6	11970.12	-934.49	-7.24	1198.22	1198.22	9	6.71
C102	11	12832.67	11	11	0	11827.1	11767.67	-1065	-8.3	1161.12	1161.12	11	13.61
C103	10	11818.14	11	11	1	11446.72	11378.78	-439.36	-3.72	1088.69	1088.69	7	27.28
C104	10	11575.87	10	10	0	11046.68	11015.17	-560.7	-4.84	1022.31	1022.31	7	21.53
C105	11	12039.54	11	11	0	11645.9	11636.47	-403.07	-3.35	1215.24	1215.24	10	8.24
C106	11	12181.07	11	11	0	11570.63	11539.57	-641.5	-5.27	1173.3	1173.3	9	12.04
C107	11	12167.43	11	11	0	11541.2	11343.52	-823.91	-6.77	1125.32	1125.32	8	8.46
C108	10	11899.76	11	11	1	11198.56	11171.53	-728.23	-6.12	1088.1	1088.1	7	14.88
C109	10	11781.58	10.4	10	0	11057.24	11132.86	-648.72	-5.51	1066.82	1066.82	7	11.91
C201	4	10288.23	4	4	0	10158.86	10158.86	-129.37	-1.26	653.51	653.51	3	6.45
C202	4	10288.23	4	4	0	10156.44	10156.44	-131.79	-1.28	661.69	661.69	3	20.19
C203	4	11114.16	4	4	0	10127.76	10127.76	-986.4	-8.88	659.08	659.08	3	31.44
C204	4	10770.32	4	4	0	10105.09	10104.72	-665.6	-6.18	654.28	654.28	3	49.62
C205	4	10289.5	4	4	0	10124.94	10124.94	-164.56	-1.6	664.95	664.95	3	9
C206	4	11506.96	4	4	0	10118.31	10115.85	-1391.11	-12.09	662.18	662.18	3	14.32
C207	4	11360.96	4	4	0	10124.78	10124.78	-1236.18	-10.88	664.9	664.9	4	19.13
C208	4	11382.96	4	4	0	10115.85	10115.85	-1267.11	-11.13	662.18	662.18	3	19.52
R101	18	3820.37	18	18	0	3709.23	3702.34	-118.03	-3.09	1860.1	1860.1	28	13.81
R102	16	3301.68	16.33	16	0	3159.48	3129.35	-172.33	-5.22	1594.45	1594.45	22	22.87
R103	13	2817.76	13.67	13	0	2797.65	2733.46	-84.3	-2.99	1350.54	1350.54	18	22.94
R104	11	2429.12	11	11	0	2403.84	2356.76	-72.36	-2.98	1134.97	1134.97	13	18.62
R105	15	3109.09	15	15	0	2988.98	2967.46	-141.63	-4.56	1535.98	1535.98	22	12.61
R106	14	2812.46	14	14	0	2691.03	2675.9	-136.56	-4.86	1364.33	1364.33	16	15.34
R107	12	2610.63	12	12	0	2458.36	2441.37	-169.26	-6.48	1220.88	1220.88	11	20.34
R108	11	2388.86	11	11	0	2311.15	2304	-84.86	-3.55	1082.21	1082.21	13	24.43
R109	12	2634.99	13	13	1	2605.29	2594.14	-40.85	-1.55	1310.27	1310.27	18	21.58
R110	12	2610.55	12	12	0	2386.92	2371.79	-238.76	-9.15	1156.74	1156.74	15	24.9
R111	12	2618.52	12	12	0	2449.53	2426.5	-192.02	-7.33	1181.55	1181.55	15	16.21
R112	11	2411.74	11	11	0	2315.01	2304.29	-107.45	-4.46	1093.57	1093.57	15	17.9
R201	3	2912.47	3	3	0	2829.74	2826.55	-85.92	-2.95	1624.69	1624.69	10	30.99
R202	3	2882.2	3	3	0	2739.8	2739.8	-142.4	-4.94	1425.79	1425.79	6	42.55
R203	3	2930.01	3	3	0	2156.13	2156.13	-773.88	-26.41	1060.98	1060.98	6	76.49
R204	2	1988.76	2	2	0	1884.7	1884.7	-104.06	-5.23	838.38	838.38	2	70.76
R205	3	2695.55	3	3	0	2634.99	2634.99	-60.56	-2.25	1393.78	1393.78	11	18.84
R206	3	2782.55	3	3	0	2142.09	2142.09	-640.46	-23.02	1039.29	1039.29	5	52.59
R207	2	1986.82	2	2	0	1910.73	1910.73	-76.09	-3.83	869.36	869.36	2	57.83
R208	2	1875.73	2	2	0	1817.31	1817.31	-58.42	-3.11	768.53	768.53	2	60
R209	3	2584.27	3	3	0	2129.19	2129.19	-455.08	-17.61	1021.49	1021.49	7	35.8
R210	3	2788.83	3	3	0	2095.76	2095.76	-693.07	-24.85	1005.94	1005.94	5	54.28
R211	2	1901.2	2	2	0	1891.5	1891.5	-9.7	-0.51	844.78	844.78	3	45.8
RC101	16	3445.33	16	16	0	3363.78	3278.03	-167.3	-4.86	1877.33	1877.33	21	10.11
RC102	15	3243.74	15	15	0	3016.07	2992.76	-250.98	-7.74	1670.39	1670.39	17	15.95
RC103	13	2821.98	13	13	0	2720.95	2688.33	-133.65	-4.74	1446.5	1446.5	13	18.72
RC104	11	2495.46	11.6	11	0	2504.03	2455.57	-39.89	-1.6	1243.18	1243.18	13	21.09
RC105	14	3078.57	14	14	0	2885.07	2879.73	-198.84	-6.46	1583.93	1583.93	15	12.56
RC106	13	2906.45	13.2	13	0	2767.84	2744.47	-161.98	-5.57	1465.07	1465.07	16	11.22
RC107	12	2691.41	12	12	0	2540.66	2535.33	-156.08	-5.8	1330.04	1330.04	13	14.64
RC108	11	2507.67	11.2	11	0	2433.18	2418.17	-89.5	-3.57	1225.31	1225.31	13	20.57
RC201	4	3649.46	4	4	0	3380.42	3269.71	-379.75	-10.41	2036.91	2036.91	14	23.03
RC202	3	2830.46	3	3	0	2723.2	2708.92	-121.54	-4.29	1610.41	1610.41	6	38.95
RC203	3	2728.63	3	3	0	2443.31	2443.31	-285.32	-10.46	1325.9	1325.9	6	61.69
RC204	3	2679.81	3	3	0	2130.12	2129.94	-549.87	-20.52	1018.71	1018.71	6	79.88
RC205	3	2701.78	3	3	0	2610.22	2605.79	-95.99	-3.55	1412.84	1412.84	8	26.66
RC206	3	2605.25	3	3	0	2558.16	2558.16	-47.09	-1.81	1425.78	1425.78	8	26.9
RC207	3	2572.44	3	3	0	2306.69	2306.69	-265.75	-10.33	1201.81	1201.81	7	50.29
RC208	3	2456.21	3	3	0	2127.54	2127.54	-328.67	-13.38	1023.38	1023.38	6	52.86
AVG	7.93	5401.98	8.02	7.98	0.05	5060.7	5042.2	-359.77	-6.97	1180.32	1180.32	9.77	27.80
SUM	444	302510.8	449.4	447	3	283399.31	282363.45	-20147.35	-390.44	66097.98	66097.98	547	1556.93

4.10.5 EVRPTWDCS-FR

The EVRPTWDCS-FR problem has not been addressed in the literature, although more complex problem EVRPTWDCS-PR has been studied in [22, 23]. Three charger types are considered, which recharge rate and recharge cost values are adopted from [22, 23] (although not clearly stated) and presented in Table 4.20. The value k_g represents the coefficient value with which the instance recharge rate g for a single charger type is multiplied to obtain the recharge rate for a particular charger type. The cost c , as already described, is the coefficient that multiplies the amount of energy recharged at CS.

Table 4.20: Charger type values in EVRPTW with different CSs

Type	m	k_g	c
Rapid	1	0.08	1.2
Fast	2	0.48	1.1
Slow	3	1	1

First, results on EVRPTW instances for distance minimization are presented in Table 4.21. Instead of columns for initial solution three new columns m_1 (rapid), m_2 (fast) and m_3 (slow) that represent the number of respective chargers in the best solution are presented. The selection of charger type has no effect when distance minimization is considered as always the fastest (rapid) charging option is selected, which increases the overall charging cost by 1.2. The reference values for comparison are selected as BKSs for EVRPTW-FR distance minimization, presented in the previous subsection (bold values represent better solutions), labeled as K_{best}^{FR} and d_{best}^{FR} . It can be seen that in 46 out of 56 instances, the better solutions were found, which includes the decrease of the number of vehicles by 38 and, on average, 1.57% decrease in total traveled distance. The sum of recharging cost is significantly higher than in EVRPTW-FR, while the sum of the total time is much lower than in EVRPTW-FR, as less time is used for charging. Interestingly, there is no difference in the number of visited CSs between the single and multiple charger type variants.

Table 4.22 presents results on EVRPTW instances for the EVRPTWDCS-FR problem with the minimization of recharging cost. The minimization of recharging costs for a full recharge strategy has not yet been addressed in the literature. The recharging cost value for the minimization of distance presented in Table 4.21, are used as reference values for the comparison, labeled as K_{best}^{dist} and rec_{best}^{dist} . It can be seen that using different charger types reduced charging costs in almost all instances, except the instance R108, where HALNS was not able to produce a solution with a lower number of vehicles. On average, the recharging costs are reduced by 7.06%. The most used charger type is fast charging (m_2) which is utilized 247 times and offers a balance between recharging cost and recharging time. Next, the slow recharging is utilized 240 times, and then the rapid charger type, which is utilized 40 times. In total, 547 CSs are used,

Table 4.21: EVRPTWDCS-FR distance - results

Inst.	K_{best}^{FR}	d_{best}^{FR}	\bar{K}	K_{best}	ΔK	\bar{d}	d_{best}	Δd	$\Delta_p d$	tot_{best}	rec_{best}	\bar{t}_e	m_1	m_2	m_3
C101	12	1053.83	12	12	0	1043.38	1043.38	-10.45	-0.99	12570.36	1117.15	2.672	8	0	0
C102	11	1051.38	10	10	-1	1011.52	1011.52	-39.86	-3.79	11401.1	1104.79	3.786	9	0	0
C103	10	1034.86	10	10	0	958.8	952.06	-82.8	-8	11424.4	1035.33	5.396	9	0	0
C104	10	951.57	10	10	0	874.78	874.73	-76.84	-8.08	10480.83	944.74	6.77	7	0	0
C105	11	1075.37	10	10	-1	1040.98	1039.33	-36.04	-3.35	11036.15	1133.37	2.135	9	0	0
C106	11	1057.65	10	10	-1	1037.61	1037.53	-20.12	-1.9	10964.76	1133.89	2.563	9	0	0
C107	11	1031.56	10	10	-1	1014.19	1009.55	-22.01	-2.13	11038.9	1104.77	3.213	9	0	0
C108	10	1095.66	10	10	0	991.64	991.03	-104.63	-9.55	10913.24	1090.05	4.094	9	0	0
C109	10	1027.1	10	10	0	920.91	920.91	-106.19	-10.34	10517.8	1003.8	3.558	8	0	0
C201	4	645.16	4	4	0	629.95	629.95	-15.21	-2.36	9912.93	686.43	5.634	3	0	0
C202	4	645.16	4	4	0	629.95	629.95	-15.21	-2.36	9912.93	686.43	13.435	3	0	0
C203	4	644.98	3	3	-1	888.62	850.89	205.91	31.93	10044.83	975.04	5.718	9	0	0
C204	4	636.43	3	3	-1	742.84	693.88	57.45	9.03	9911.85	786.36	9.15	5	0	0
C205	4	641.13	4	4	0	629.95	629.95	-11.18	-1.74	9722.37	686.43	6.44	3	0	0
C206	4	638.17	4	4	0	629.95	629.95	-8.22	-1.29	9683.38	686.43	8.248	3	0	0
C207	4	638.17	4	4	0	629.95	629.95	-8.22	-1.29	9769.18	686.43	10.164	3	0	0
C208	4	638.17	4	4	0	629.95	629.95	-8.22	-1.29	9683.38	686.43	7.528	3	0	0
R101	18	1663.04	16.67	16	-2	1624.65	1661.94	-1.1	-0.07	3368.25	1854.42	5.179	24	0	0
R102	16	1484.57	14.67	14	-2	1442.37	1496.98	12.41	0.84	2965.63	1678.4	9.498	20	0	0
R103	13	1268.88	12	12	-1	1194.02	1186.03	-82.85	-6.53	2518.87	1327.62	12.747	17	0	0
R104	11	1088.43	10	10	-1	1042.94	1019.94	-68.49	-6.29	2148.55	1130.45	14.849	14	0	0
R105	14	1442.35	12.83	12	-2	1351.99	1507.31	64.96	4.5	2667.92	1714.35	7.145	25	0	0
R106	13	1324.1	12	12	-1	1234.71	1229.47	-94.63	-7.15	2478.16	1369.74	11.362	18	0	0
R107	12	1148.38	10	10	-2	1081.66	1067.98	-80.4	-7	2172.35	1203.64	12.677	15	0	0
R108	11	1043.12	9.83	9	-2	993.8	1014.88	-28.24	-2.71	2038.1	1137.36	13.541	15	0	0
R109	12	1261.31	11	11	-1	1157.74	1129.6	-131.71	-10.44	2395.53	1266.17	8.041	15	0	0
R110	11	1119.5	10	10	-1	1034.92	1031.42	-88.08	-7.87	2160.2	1149.07	11	12	0	0
R111	12	1099.53	10	10	-2	1046.66	1027.64	-71.89	-6.54	2196.34	1144.18	11.216	14	0	0
R112	11	1016.63	10	10	-1	979.33	969.24	-47.39	-4.66	2180.18	1087.99	13.351	12	0	0
R201	3	1264.32	3	3	0	1273.74	1269.07	4.75	0.38	2890.02	1437.57	22.518	6	0	0
R202	3	1052.32	3	3	0	1059.19	1055.49	3.17	0.3	2912.43	1165.45	27.718	4	0	0
R203	3	895.54	3	3	0	917.82	906.72	11.18	1.25	2912.86	987.34	37.115	4	0	0
R204	2	779.49	2	2	0	784.66	779.57	0.08	0.01	1985.91	837.99	17.511	2	0	0
R205	3	987.36	3	3	0	994.93	989.98	2.62	0.27	2700.7	1105.89	20.939	4	0	0
R206	3	922.19	3	3	0	938.38	922.02	-0.17	-0.02	2762.28	1042.13	22.822	4	0	0
R207	2	843.2	2	2	0	856.63	848.67	5.47	0.65	1986.82	930.35	16.747	2	0	0
R208	2	736.12	2	2	0	749.45	741.85	5.73	0.78	1913.72	814.9	18.267	2	0	0
R209	3	867.05	2.5	2	-1	911.23	941.45	74.4	8.58	1990.87	1069.97	18.775	6	0	0
R210	3	843.65	2.17	2	-1	912.12	911.93	68.28	8.09	1965.21	1055.63	16.09	5	0	0
R211	2	827.29	2	2	0	839.72	828.43	1.14	0.14	1912.96	907.47	17.565	3	0	0
RC101	16	1723.79	14	14	-2	1680.19	1659.03	-64.76	-3.76	3031.27	1852.63	5.651	19	0	0
RC102	14	1659.53	13	13	-1	1538.75	1522.15	-137.38	-8.28	2827.6	1703.9	8.42	17	0	0
RC103	13	1350.09	11.67	11	-2	1328.12	1386.12	36.03	2.67	2503.44	1545.25	9.228	16	0	0
RC104	11	1227.25	10	10	-1	1159.45	1142.81	-84.44	-6.88	2230.3	1264.21	10.112	12	0	0
RC105	14	1471.87	12.17	12	-2	1461.02	1450.68	-21.19	-1.44	2654.86	1617.01	6.622	16	0	0
RC106	13	1423.27	12	12	-1	1369.68	1358.22	-65.05	-4.57	2580.09	1520.13	6.625	16	0	0
RC107	12	1274.25	10.83	10	-2	1201.39	1202.11	-72.14	-5.66	2296.45	1359.74	9.655	14	0	0
RC108	11	1197.41	10	10	-1	1143.97	1129.01	-68.4	-5.71	2263.67	1263.63	9.958	12	0	0
RC201	4	1441.47	4	4	0	1444.06	1430.58	-10.89	-0.76	3649.46	1581.17	14.914	6	0	0
RC202	3	1408.16	3	3	0	1415.43	1398.89	-9.27	-0.66	2854.23	1534.37	27.581	4	0	0
RC203	3	1055.19	3	3	0	1065.71	1050.58	-4.61	-0.44	2705.65	1185.33	35.093	5	0	0
RC204	3	884.53	3	3	0	885.57	878.88	-5.65	-0.64	2680.21	971.77	37.691	5	0	0
RC205	3	1256.09	3	3	0	1283.72	1231.95	-24.14	-1.92	2648.09	1398.37	23.433	6	0	0
RC206	3	1187.72	3	3	0	1214.3	1194.82	7.1	0.6	2662.55	1336.26	20.906	4	0	0
RC207	3	985.03	3	3	0	1024.59	996.44	11.41	1.16	2521.08	1089.33	26.16	4	0	0
RC208	3	836.29	3	3	0	842.3	828.99	-7.3	-0.87	2280.65	929.9	26.818	4	0	0
AVG	7.86	1069.05	7.27	7.18	-0.68	1049.75	1046.49	-22.56	-1.57	4939.32	1162.84	13.32	9.14	0	0
SUM	440	59866.66	407.34	402	-38	58785.88	58603.38	-1263.28	-88.15	276601.85	65118.95	746.05	512	0	0

which is on average $547/403 = 1.36$ CSs per vehicle route. Compared to the distance minimization when only rapid charger was utilized, the total traveled distance increased by 0.52%.

Results on EVRPTW instances for the minimization of total time in EVRPTWDCS-FR is presented in Table 4.23. The reference values are the total times achieved with distance minimization, labeled as K_{best}^{dist} and tot_{best}^{dist} . In all instances, the rapid charger type is used, and the total time is on average reduced by 5.84%, while again, the overall charging costs are increased. In total, 582 CSs are used, which is 35 CSs more than in the minimization of recharging costs and 70 CSs more than in distance minimization. On instance RC103 HALNS method was not able to produce a solution with minimal number of vehicles.

4.10.6 EVRPTWDCS-PR

The results on EVRPTW instances for EVRPTWDCS-PR with the minimization of recharging costs are presented in Table 4.24. The referenced values used for the comparison are the ones reported by Keskin et al. [22], which are the only ones that observed the problem so far. The bold values indicate that HALNS was able to find a better solution. In 16 out of 56 instances, the better solution was found, with in total 5 vehicles less, and an increase in recharging costs by in average 2.05% (as a solution with a lower number of vehicles was found). Compared to the full recharge strategy, the total number of visited CSs increased by 48, and recharging costs significantly decreased, as well as the total traveled distance and total time. It can be noted that HALNS was able to produce 3 vehicles less on a full variant of the problem than on the partial variant. Compared to the partial variant without different charger types, 20 vehicles less were used, with a slightly higher total recharging cost due to the rapid and fast recharging. The average execution time increased significantly compared to the partial variant without different charger types, as the search space increased with different charger types.

Time-dependent electric vehicle routing problem with time windows

In the real world, the travel time in the traffic network changes dynamically, as it depends on various factors, which can be predictable and unpredictable. The typical predictable factors are urbanization, weather, time, social events, road works, etc. The unpredictable factors cover the stochastic component of traffic, such as accidents and anomalies. In the urban environment, due to the high population density, traffic congestion occurs regularly on the road network. The congestion occurs due to the lack of infrastructure, which could not develop at the rate at which urbanization and vehicle market share developed in a last few decades [168]. The congestion has a direct impact on people's daily lives and transport activities as it causes longer travel time [169, 170], additional transportation costs [171, 172], noise, increased gas consumption [173], congested queues [174], and increased CO₂ emission [175, 176]. Predictable traffic factors cover almost 85% of congestion occurring in large urban centers [177]. Due to the recurrent nature of the traffic congestion, time-varying travel times on the road network can be estimated. Erdelić et al. [28] conducted a test study in which test vehicles drove designated routes, and the overall travel time was recorded. The travel times computed by a model that considers static traffic information (average speeds) underestimated travel times by roughly 30%, compared to the real measured times. The route travel time can significantly affect logistic operations, and if neglected, can lead to a significant increase in overall routing cost [74]. Integrating congestion in logistic operations led to the definition of TD-VRPTW problem [25, 75, 76], which was already discussed in section 2.6.

The time-dependent vehicle routing problem in the context of electric vehicles has only been studied by Shao et al. [130]. The authors defined the problem as EVRP-CTVTT, in which fixed full recharging time was considered, and not the charging dependent on the BEV SoC value. The authors proposed a genetic algorithm with an average execution time of 30 minutes to solve

real-world instances with up to 50 customers and did not provide results on commonly used benchmark EVRPTW instances. In this thesis, the Time-Dependent Electric Vehicle Routing Problem with Time Windows and Full Recharge (TD-EVRPTW-FR) is defined, and throughout the thesis, it is referred to as the TD-EVRPTW problem. The problem considers charging time that linearly depends on the SoC level when entering CS. The proposed problem has not yet been addressed in the literature. The HALNS method proposed in chapter 4 is applied to efficiently solve the problem with an average running time of roughly 13 minutes.

Additionally, two variants of the problem are formulated for the first time: (i) Time-Dependent Electric Vehicle Routing Problem with Time Windows, Different Charging Stations and Full Recharge TD-EVRPTWDCS-FR, and (ii) Time-Dependent Electric Vehicle Routing Problem with Time Windows and Partial Recharge TD-EVRPTW-PR. All of the problems are formulated as MIP programs, except the TD-EVRPTW-FR for which additionally the MILP formulation is presented.

The TD-EVRPTW-PR variant is not solved by HALNS as it requires significant changes in the penalty functions and variables, which are left for future research. In EVRPTW-PR, when the variable a_{ij}^{add} that represents the additional charging time that needs to be added at preceding CSs, is larger than zero, the computed value is exact as there are no different charger types or time-dependent travel times (subsection 4.3.1). In such cases, the charging amount can easily be computed. In EVRPTWDCS-PR, this problem was solved by complex forward-backward looping, as the additional charging amount that needs to be recharged at preceding CSs can be exactly computed, and optimization of the charging schedule can be conducted. In TD-EVRPTW-PR, when variable a_{ij}^{add} is larger than zero at user j , one could apply the similar idea as in EVRPTW-PR and add charging time at previous CSs. If this time is added at CS, due to the time-dependent times, the forward variables of users between the CS at user j change and need to be recomputed. However, even that is not the main problem. The problem is that these values will be completely different from the values before, meaning that although a_{ij}^{add} value was added as charging time at preceding CSs the arrival times did not increase by this value, and the value of a_{ij}^{add} at user j could still be $a_{ij}^{add} > 0$ as the travel time periods changed. More importantly, there is no guarantee that after the update, the value of a_{ij}^{add} will be equal to zero.

5.1 FIFO property and travel time computation

In time-dependent routing, the delivery period, which is usually one day, is often discretized in shorter time intervals (time buckets), where for each interval, the speed value is assigned. In test instances, this speed value is used to compute the travel time in each of the discretized time buckets based on the computed Euclidean distance and the assumption of uniform motion. The researchers that deal with benchmark time-dependent problems usually consider five speed values per planning horizon [25, 176, 178, 179, 180]. These five values usually represent the following periods in a day: morning free-flow, morning rush hours, transition period between morning and afternoon rush hours, afternoon rush hours, and evening free-flow. To get a more accurate estimation, the delivery period could be discretized into shorter time buckets, i.e., for each 5-minute interval in day, [24, 73]. In a real application, the travel time value between users i and j is a solution to the Time-Dependent Shortest Path Problem (TD-SPP), with a set departure time. The solution to the problem is the shortest travel time path on a road network. Calling TD-SPP every time when there is a need for the shortest travel time path between users when solving a TD-VRPTW problem would be extremely time-consuming. This is why the TD-SPP is usually solved for each customer pair in each discretized time bucket in advance. This procedure is still time-consuming, but it is done only once in the pre-processing step. The technique proposed to speed-up the computation of real world travel time matrices based on the data mining procedures is discussed in section 6.6. The important aspects of the time-dependent routing are the travel time computation method and First-In First-Out (FIFO) principle.

The example of a speed profile (blue line) for a 250 m road segment is presented in Figure 5.1. Figure 5.1 shows an example of constant speeds per discretized five-minute time buckets: 08:00-08:05, 08:05-08:10, and 08:10-08:15. Due to the discretization, the speed function is not continuous, and function break points appear. The travel times computed in a classical way, based only on the current speed value, are presented with red lines. The uniform motion is assumed for computing the travel times. Let a vehicle A enter the road segment at $t_{s_A} = 08:09:31$. The speed in that time bucket is 20 km/h, and the travel time of vehicle A would be $t_{t_A} = 250/(20/3.6) = 45$ s. The vehicle A would come to the end of the road segment at the time $t_{e_A} = 08:10:16$. Let a vehicle B enter the road segment at $t_{s_B} = 08:10:00$ (29 s later). The travel time of vehicle B would be $t_{t_B} = 250/(60/3.6) = 15$ s, and the vehicle would come to the end of the road segment at time $t_{e_A} = 08:10:15$. The vehicle A that started earlier finished the road segment later than vehicle B that started later. This is referred to as the violation of the FIFO principle. FIFO ensures that the vehicle that has entered the road segment first has to be the first one that exits it. Besides the FIFO violation, the computation of travel times is not a good approximation of real travel times as, for example, in vehicle A , the travel time function crosses a break point at 08:10:00, where the speed changes and this change is not accounted for.

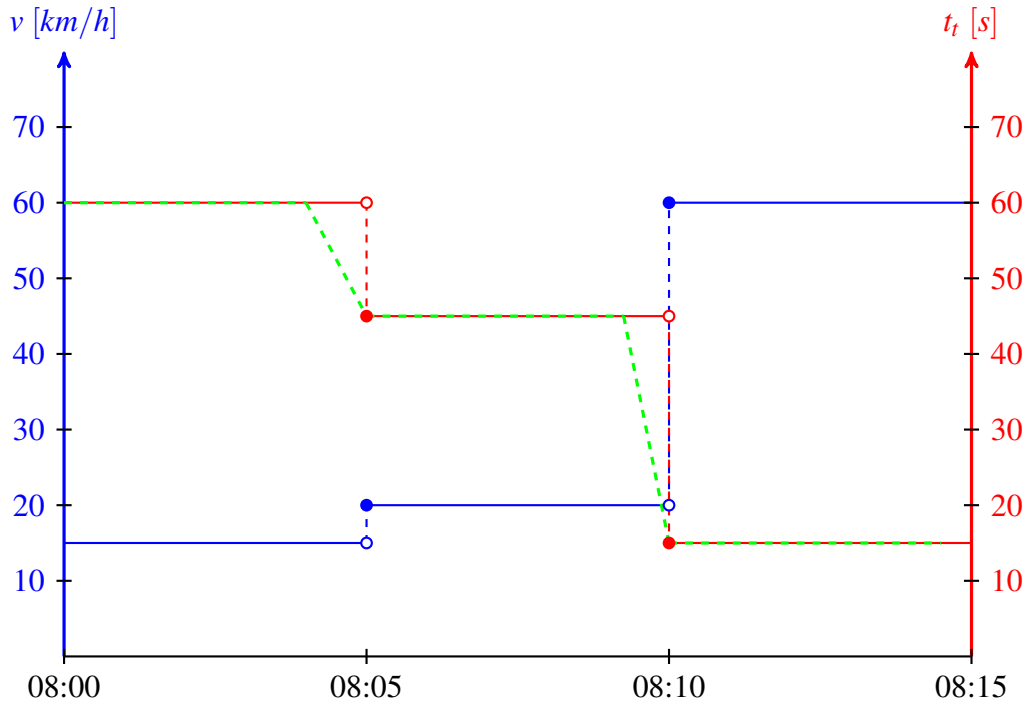


Figure 5.1: Speed and travel time profile

To ensure the FIFO property and provide a better approximation of travel times, the travel times can be linearized with piecewise functions [25, 181, 182]. Figliozzi [25] proposed the computation of travel times based on the discrete speed values that satisfy the FIFO property. The travel time t_{ij} between two given users i and j , or on a given road segment is computed using an iterative forward computation strategy, from the departure time at user i . The depot working time $[e_0, l_0]$ is discretized into p time buckets $T = \{T_1, \dots, T_p\}$. Each time bucket T_k ($k = \{1, \dots, p\}$) has an associated uniform speed v_k . The procedure is given by Algorithm 5.1. The algorithm assumes that the distance (path) between users i and j does not change and has a constant value d_{ij} . First the values of distance d_{ij} and departure time are stored in the corresponding current variables d_{cur} and $t_{d,cur}$. Next, the index of a time bucket in which departure t_d falls into is computed and set as k . Then, the final time t_f is computed as the sum of the departure time t_d and travel time computed based on distance d_{ij} and uniform speed v_k in time bucket k . Further on, the time bucket index of time t_f is computed and set as k_{next} . If k is equal to k_{next} , then there was no break point present in the travel time function, and the travel time is correctly computed. If k and k_{next} are not equal, the loop (lines 7-19) continues until they are. In the loop, first, the travel time t_k is computed as the amount of travel time spent in time bucket k , up to the break point T_k . Based on this travel time, the distance traversed in the time bucket k is computed as $t_k \cdot v_k$, and the overall distance d_{cur} that still has to be traversed is reduced by this amount. The new departure time $t_{d,cur}$ is set as a current break point time T_k , and the new final time is computed by using this new departure time and the travel time in time bucket k_{next} . Then, the k_{next} is set as k , and new value for k_{next} is computed. The *if* clause

Algorithm 5.1 Computation of forward travel times**Input:** Departure time t_d , distance d_{ij}

```

1:  $d_{cur} \leftarrow d_{ij}$ 
2:  $t_{d,cur} \leftarrow t_d$ 
3:  $k \leftarrow$  Get time bucket index of time  $t_d$ 
4:  $t_f \leftarrow t_d + \frac{d_{ij}}{v_k}$ 
5:  $k_{next} \leftarrow$  Get time bucket index of time  $t_f$ 
6: if  $k_{next} > k + 1$  then
7:    $k_{next} = k + 1$ 
8: end if
9: while  $k \neq k_{next}$  do
10:   $t_k \leftarrow T_k - t_{d,cur}$ 
11:   $d_{cur} \leftarrow d_{cur} - t_k \cdot v_k$ 
12:   $t_{d,cur} \leftarrow T_k$ 
13:   $t_f \leftarrow t_{d,cur} + \frac{d_{cur}}{v_{k_{next}}}$ 
14:   $k \leftarrow k_{next}$ 
15:   $k_{next} \leftarrow$  Get time bucket index of time  $t_f$ 
16:  if  $k_{next} > k + 1$  then
17:     $k_{next} = k + 1$ 
18:  end if
19: end while
20:  $t_{ij} \leftarrow t_f - t_d$ 

```

$k_{next} > k + 1$ covers the possibility that the next time bucket k_{next} might not be the succeeding time bucket to time bucket k (if travel time on a road segment is longer than the succeeding time bucket interval), in which case the succeeding time bucket is set as k_{next} . The example of computed travel times are presented in Figure 5.1 as green lines. It can be seen that a travel time function is linearized around break points. For previously observed example, the travel time for vehicle A would be $t_{t_A} = 29 + 5.33 = 34.33$ s, where 29 s is the time driven in the first time bucket with the speed of 20 km/h and covered distance of 161.11 m, and 5.33 s is the time driven in the second time bucket with the speed of 60 km/h and the covered distance of 88.89 m. As a result the vehicle A would come to the end at $t_{e_A} = 08:10:05.33$, and no FIFO violation would occur as $t_{e_A} < t_{e_B}$. In the travel time function t_{e_B} , there is no brake point in the observed period, so the travel time remains the same.

The other, not so common approach is to ensure FIFO within the data or the road segments in the digital map. For a road segment with length l to satisfy the FIFO between two vehicles A and B with departure times t_{s_A} and t_{s_B} ($t_{s_A} < t_{s_B}$), the equation 5.1 has to be satisfied [73]. From the equation 5.1, it can be seen that the lower the link length is, the lower the possibility for FIFO violation is. The delay between departure times and the difference in speed between two consecutive time buckets also significantly affect the FIFO property. The possibility of FIFO violation can be reduced by decreasing transition values between the speeds on the same or

different road segments, by averaging or smoothing [24, 28, 73, 183]. Although this procedure satisfies FIFO property, the travel time approximation is still not realistic, and in the real world, it is hard to satisfy FIFO through raw data on all road segments in a digital map.

$$l < \frac{v_{t_{sA}} \cdot v_{t_{sB}}}{v_{t_{sB}} - v_{t_{sA}}} (t_{sB} - t_{sA}) \quad (5.1)$$

5.2 MIP formulations

In the following subsections, MIP formulations for three TD-EVRPTW problems are presented.

5.2.1 TD-EVRPTW-FR

The TD-EVRPTW-FR problem considers travel time between users dependent on the departure time. The formulated MIP program for TD-EVRPTW-FR is an extension of the MILP program for EVRPTW-FR (section 2.2). The MIP program for TD-VRPWTW (section 2.6) was not considered, as it includes one additional variable k for the number of vehicles, which in a tighter formulation can be removed. The TD-EVRPTW-FR problem is defined as compact MIP program, where only the equations that differ from the original EVRPTW are given by equations 5.2-5.5. First of all, equation for vehicle minimization remains the same (equation 2.2). Instead of distance minimization two minimization functions are considered: (i) sum of travel time (equation 5.2) which does not include service or waiting times for start of the service, and (ii) total time which includes all routing times (equation 5.3). The equation for total time is modified to include several instances of the end depot (AD), in order to track arrival times at those instances. Figliozzi [25] who first proposed the TD-VRPTW formulation, considers the minimization of total times, but later on, in the numerical tests, the authors minimized the sum of travel times. The travel times are easier to evaluate than total times, as total times require whole route evaluation. The only two equations that changed compared to the EVRPTW MILP formulation are the ones that contain travel time t_{ij} for an arc (i, j) . Instead of a constant travel time, the travel time that depends on the departure time from user i is used $t_{ij}(\tau_i + s_i)$, where $\tau_i + s_i$ represents the departure time from user i , and τ_i is the begin time variable. The $t_{ij}(\tau_i + s_i)$ is a function of the departure time, which is often non-linear (as in Figure 5.1). Therefore the program is formulated as a MIP program and can not be solved by commercial solvers, such as CPLEX and MATLAB, which require a MILP formulation.

$$\min \sum_{i \in V_0 \cup F', j \in V_{N+1} \cup F', i \neq j} t_{ij}(\tau_i + s_i) x_{ij} \quad (5.2)$$

$$\min \sum_{i \in V \cup F'} \sum_{j \in AD} (\tau_j - \tau_i) x_{ij} \quad (5.3)$$

$$\tau_i + (t_{ij}(\tau_i + s_i) + s_i) x_{ij} - l_0 \cdot (1 - x_{ij}) \leq \tau_j, \forall i \in V_0, \forall j \in V_{N+1} \cup F', i \neq j \quad (5.4)$$

$$\tau_i + g(Q - y_i) + x_{ij} t_{ij}(\tau_i + s_i) - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j, \forall i \in F', \forall j \in V_{N+1} \cup F', i \neq j \quad (5.5)$$

An alternative MILP formulation of TD-EVRPTWFR can be obtained by discretizing the planning horizon $[e_0, l_0]$ into p time buckets $T = \{T_1, \dots, T_p\}$. The travel time between users i and j is then also discretized as $t_{ij} = \{t_{ij}^1, \dots, t_{ij}^p\}$. Further on, the travel times t_{ij}^k within each time bucket are linearized, $k \in P, P = \{1, \dots, p\}$. The travel time t_{ij}^k can be computed by equation 5.6, where θ_{ij}^k and η_{ij}^k are linear travel time function coefficients computed for the interval $T_k = [w_k, w_{k+1}]$, where w_k and w_{k+1} are consecutive time function break points.

$$t_{ij}^k(t_d) = \theta_{ij}^k t_d + \eta_{ij}^k, \forall t_d \in T_k, \quad (5.6)$$

To formulate the MILP model, the procedure proposed by Monter et al. [184] for time-dependent TSP is used. The binary variable x_{ij}^k is set to a value of one, if arc (i, j) is traversed within the time bucket k (equation 5.7), and zero otherwise. The travel time function $t_{ij}(\tau_i)$ in MILP program can be defined by equation 5.8. The new variable t_{ij}^k is used and defined by equation 5.9 to have a continuity in the model. The variable t_{ij}^k is constrained by equations 5.10 and 5.11. The first one ensures the correct relation between variables t_{ij}^k and x_{ij}^k , while the second ensures correct arrival times at each vertex and arc. The routing costs consist only of travel time on arcs (equation 5.13) but could easily be rewritten to include the total time. The rest of the equations follow the basic MILP model for EVRPTW-FR, with added p discretized time intervals, given by equations 5.12, and 5.14-5.23.

$$x_{ij}^k \in \{0, 1\}, \forall i \in V_0 \cup F', j \in V_{N+1} \cup F', i \neq j, \forall k \in P \quad (5.7)$$

$$t_{ij}(\tau_i) = \sum_{k \in P} (\theta_{ij}^k t_{ij}^k + \eta_{ij}^k x_{ij}^k) \quad (5.8)$$

$$t_{ij}^k = \begin{cases} \tau_i + s_i & x_{ij}^k = 1 \\ 0 & \text{else} \end{cases} \quad (5.9)$$

$$x_{ij}^k \max(e_i + s_i, w_k) \leq t_{ij}^k \leq x_{ij}^k \min(l_i + s_i, w_{k+1}), \forall i \in V_0 \cup F', j \in V_{N+1} \cup F', i \neq j, \forall k \in P \quad (5.10)$$

$$\tau_i = \sum_{j \in V_{N+1} \cup F', i \neq j} \sum_{k \in P} t_{ij}^k, \forall i \in V_0 \quad (5.11)$$

$$\min \sum_{k \in P} \sum_{j \in V \cup F'} x_{0j}^k \quad (5.12)$$

$$\min \sum_{k \in P} \sum_{i \in V_0 \cup F'} \sum_{j \in V_{N+1} \cup F', i \neq j} t_{ij}(\tau_i) x_{ij}^k \quad (5.13)$$

$$\sum_{k \in P} \sum_{j \in V_{N+1} \cup F', i \neq j} x_{ij}^k = 1, i \in V \quad (5.14)$$

$$\sum_{k \in P} \sum_{j \in V_{N+1} \cup F', i \neq j} x_{ij}^k \leq 1, i \in F' \quad (5.15)$$

$$\sum_{k \in P} \sum_{i \in V_{N+1} \cup F', i \neq j} x_{ji}^k - \sum_{k \in P} \sum_{i \in V_0 \cup F', i \neq j} x_{ij}^k = 0, j \in V \cup F' \quad (5.16)$$

$$t_{ij}^k + \theta_{ij}^k t_{ij}^k + \eta_{ij}^k x_{ij}^k - l_0 \cdot (1 - x_{ij}^k) \leq \tau_j, \forall i \in V_0, \forall j \in V_{N+1} \cup F', i \neq j, k \in P \quad (5.17)$$

$$g(Q - y_i) + t_{ij}^k + \theta_{ij}^k t_{ij}^k + \eta_{ij}^k x_{ij}^k - (l_0 + gQ)(1 - x_{ij}^k) \leq \tau_j, \forall i \in F', \forall j \in V_{N+1} \cup F', i \neq j, k \in P \quad (5.18)$$

$$e_j \leq \tau_j \leq l_j, \forall j \in V_{0, N+1} \cup F' \quad (5.19)$$

$$0 \leq u_j \leq u_i - x_{ij}^k (q_i + C) + C, \forall i \in V_0 \cup F', \forall j \in V_{N+1} \cup F', i \neq j, k \in P \quad (5.20)$$

$$0 \leq u_0 \leq C \quad (5.21)$$

$$0 \leq y_j \leq y_i - (e_{ij} + Q)x_{ij}^k + Q, \forall j \in V_{N+1} \cup F', \forall i \in V, i \neq j, k \in P \quad (5.22)$$

$$0 \leq y_j \leq Q - e_{ij} x_{ij}^k, \forall j \in V_{N+1} \cup F', \forall i \in 0 \cup F', i \neq j, k \in P \quad (5.23)$$

5.2.2 TD-EVRPTWDCS-FR

The TD-EVRPTWDCS-FR is an extension of TD-EVRPTW-FR variant which considers different charger types at CSs. Here, the methodology similar to the one for the EVRPTWDCS-PR presented in section 2.5, is used. Each virtual CS $i \in F'$ has a possibility to use only one charger type $m \in M$, with related charging rate and charging cost, g^m and c^m , respectively. As TD-EVRPTWDCS-FR MIP formulation is also an extension of EVRPTW-FR and EVRPTWDCS-PR MILP formulations, only the equations that differ are presented. The objective function considers minimizing total charging costs given by equation 5.24. As multiple depot end instances have to be considered, set AD is added to track the rest battery capacity variable at those instances. As there is no need for the variable that determines the charging amount at each CS, the objective is rewritten to include the charging costs for each selected charger type at CSs. The equations 5.27 and 5.28 represent the charger type selection based on the binary variables a_i and b_i . The only additional equations that differ from the original EVRPTW MILP formulation are equation for the preservation of travel time flow 5.25 and equation for the preservation of energy flow 5.26.

$$\min \sum_{i \in F'} (a_i c^1 + b_i c^2 + (1 - a_i - b_i) c^3) (Q - y_i) + c^3 \left(Q \sum_{j \in V'} x_{0j} - \sum_{i \in AD} y_i \right) \quad (5.24)$$

$$\tau_i + (t_{ij}(\tau_i + s_i) + s_i)x_{ij} - l_0 \cdot (1 - x_{ij}) \leq \tau_j, \forall i \in V_0, \forall j \in V_{AD} \cup F', i \neq j \quad (5.25)$$

$$\tau_i + (a_i g^1 + b_i g^2 + (1 - a_i - b_i) g^3) (Q - y_i) + x_{ij} t_{ij} (\tau_i + s_i) - (l_0 + g^3 Q) (1 - x_{ij}) \leq \tau_j, \\ \forall i \in F', \forall j \in V_{AD} \cup F', i \neq j \quad (5.26)$$

$$a_i, b_i \in \{0, 1\}, \forall i \in F' \quad (5.27)$$

$$a_i + b_i \leq 1, \forall i \in F' \quad (5.28)$$

5.2.3 TD-EVRPTW-PR

The MIP formulation of TD-EVRPTW-PR is an extension of a MILP formulation for EVRPTW-PR problem, presented in section 2.4. The minimization objective is the same as in equation 5.2, the preservation of travel time flow is given by equation 5.4, the only new equation is the equation for the preservation of energy flow given by 5.29.

$$\tau_i + x_{ij}t_{ij}(\tau_i + s_i) - (l_0 + gQ)(1 - x_{ij}) + g(Y_i - y_i) \leq \tau_j, \forall i \in F', \forall j \in V_{N+1} \cup F', i \neq j \quad (5.29)$$

5.3 HALNS modification

To efficiently solve the TD-EVRPTW-FR problem, the HALNS method developed in chapter 4 is used. To apply the HALNS method on TD-EVRPTW-FR, the computation of both forward and backward travel time has to be changed. The computation of forward travel times is given by Algorithm 5.1. The computation of backward travel times in a time-dependent context has not yet been addressed in the literature. The backward travel time computation follows a similar assumption as proposed forward travel time computation that the distance between users does not change. Instead of using departure time at user i , and computing the arrival time at user j , the backward travel time computation is given the latest arrival time at user j , and computes the latest departure time at user i . These two functions are inverse functions because if computed forward arrival travel time at user j based on the departure time from user i , is given as the latest arrival time at user j ; the backward travel time computation produces the latest arrival time at user i which is equal to the starting departure time at user i .

The computation of backward travel times is given by Algorithm 5.2. The inputs are the latest arrival time t_a and distance d_{ij} between users i and j . Instead of the sum of travel times, the travel times are subtracted from the latest arrival time to get the departure time at user i . Here, the travel time in bucket k is computed as the final time (departure time) minus the travel time brake point before, $T_{k_{bef}}$. If clauses are used to limit the skip to only the preceding time bucket.

The exact procedure based on DP can easily be used as in full recharge strategy the amount of charging is known when evaluating certain users in forward direction. Therefore, the time-dependent travel times can affect only the arrival times at users but are known exactly when evaluating users in a path.

Algorithm 5.2 Computation of backward travel times**Input:** Latest arrival time t_a , distance d_{ij}

```

1:  $d_{cur} \leftarrow d_{ij}$ 
2:  $t_{a,cur} \leftarrow t_a$ 
3:  $k \leftarrow$  Get time bucket index of time  $t_a$ 
4:  $t_f \leftarrow t_a - \frac{d_{ij}}{v_k}$ 
5:  $k_{bef} \leftarrow$  Get time bucket index of time  $t_f$ 
6: if  $k_{bef} < k - 1$  then
7:    $k_{bef} = k - 1$ 
8: end if
9: while  $k \neq k_{bef}$  do
10:   $t_k \leftarrow t_f - T_{k_{bef}}$ 
11:   $d_{cur} \leftarrow d_{cur} - t_k \cdot v_k$ 
12:   $t_{a,cur} \leftarrow T_{k_{bef}}$ 
13:   $t_f \leftarrow t_{a,cur} - \frac{d_{cur}}{v_{k_{bef}}}$ 
14:   $k \leftarrow k_{bef}$ 
15:   $k_{bef} \leftarrow$  Get time bucket index of time  $t_f$ 
16:  if  $k_{bef} < k - 1$  then
17:     $k_{bef} = k - 1$ 
18:  end if
19: end while
20:  $t_{ij} \leftarrow t_a - t_f$ 

```

5.4 Testing

To see if the proposed HALNS method can be used in time-dependent routing, two test instances are solved with HALNS: (i) Solomon VRPTW instances [31], and (ii) TD-VRPTW instances of Figliozzi [25]. For the observed cases, the simplified variables and operators are used without the charging procedure. As this procedure has not yet been applied to solve the TD-VRPTW problem, here variables and concatenation operators for the TD-VRPTW problem are presented. The variables are presented in Table 5.1, while the variables are given by equations 5.30-5.34. Similar non-shifted variables and equations were already introduced in section 3.5, but did not include variable travel times and shifting technique to the latest feasible point in time. The forward and backward time window violation are computed by equations 5.35 and 5.36, respectively. The time window violation of two concatenated partial routes $\phi_1 = (u_0, u_1, \dots, x)$ and $\phi_2 = (y, \dots, u_{N+1})$ can be computed by equation 5.37. First, the forward variables are computed for user y , based on the forward variables for user x . Then, the time window computation of concatenated routes is computed as the sum of forward time window violation $\overrightarrow{TW}(\phi_1)$ of route ϕ_1 , backward time window violation $\overleftarrow{TW}(\phi_2)$ of route ϕ_2 , and difference between arrival time at user y and latest shifted arrival time at user y .

Table 5.1: Variables in TD-VRPTW

Symbol	Description
t_{a_j}	Earliest arrival time at user j
t_{b_j}	Earliest begin time at user j
\tilde{t}_{b_j}	Shifted earliest begin time at user j
t_{la_i}	Latest arrival time at user i
\tilde{t}_{la_i}	Shifted latest arrival time at user i

$$t_{a_j} = \tilde{t}_{b_i} + s_i + t_{ij}(\tilde{t}_{b_i} + s_i) \quad (5.30)$$

$$t_{b_j} = \max(e_j, t_{a_j}) \quad (5.31)$$

$$\tilde{t}_{b_j} = \min(l_j, t_{b_j}) \quad (5.32)$$

$$t_{la_i} = \min(\tilde{t}_{la_j} - t_{ij}(\tilde{t}_{la_j}) - s_i, l_i) \quad (5.33)$$

$$\tilde{t}_{la_i} = \max(t_{la_i}, e_i) \quad (5.34)$$

$$\overrightarrow{TW}(\phi) = \sum_{u \in \phi} \max(t_{b_u} - l_u, 0) \quad (5.35)$$

$$\overleftarrow{TW}(\phi) = \sum_{u \in \phi} \max(e_u - t_{la_u}, 0) \quad (5.36)$$

$$TW(\phi_1 \otimes \phi_2) = \overrightarrow{TW}(\phi_1) + \overleftarrow{TW}(\phi_2) + \max(0, t_{a_y} - \tilde{t}_{la_y}) \quad (5.37)$$

The TD-VRPTW instances consider Solomon instances with in total 12 different speed configurations, from A1 to D3, presented in Table 2.3. The VRPTW instances can be observed as a special case of TD-VRPTW in which all speeds have a value of 1. The computation of forward and backward travel times have no effect in VRPTW instances, as time-dependent travel times are not considered. However, they are used for travel time computation to test the proposed method. The presented variables and concatenation operator evaluate most of the basic moves in $\mathcal{O}(1)$, but not in all cases. During the development, it was noted that in a small number of cases, due to the time-dependent backward travel time, the time-window penalty is accounted for, although if fully evaluated, the penalty does not exist. It can be noted that these special cases can be distinguished and included in the concatenation operator. In this thesis,

this drawback has been solved by the use of the restricted list of moves that will be evaluated in $\mathcal{O}(n)$. The inspection of special cases of time window violation and the development of appropriate concatenation operators is left for future research. If the objective function is purely distance-based, then this could be achieved, but if it is time-based, like travel time or total time, then still the $\mathcal{O}(1)$ evaluation can be used, but latter $\mathcal{O}(n)$ evaluation of best moves is needed, as due to the time-dependent travel times the exact total time is unknown until the whole route is evaluated.

5.4.1 VRPTW

The results on VRPTW instances are presented in Table 5.2. The same parameter values as for the original HALNS are used. The column Inst. represents the instance observed. The columns BKS represent the BKS values for the vehicle number K and distance value d , taken from the [SINTEF](#) page that acts as a central repository for storing BKS solutions in VRPTW. The columns \bar{K} , \bar{d} and \bar{tot} present the average result out of 10 runs for the number of vehicles, total traveled distance, and total time. The columns K_{best} and d_{best} represent the best values out of 10 runs for the number of vehicles and total traveled distance. The columns ΔK and Δd represent the relative difference between the HALNS values and BKS values, while the column $\Delta_p d$ represents the relative percentage distance difference between HALNS and BKS. The column \bar{t}_e represents the average execution time in minutes. The summary of the results is presented in the last two rows as average and cumulative values. The results indicate that HALNS is able to efficiently solve VRPTW instances with relatively fast execution time. In the sum of average runs, it produced 3 vehicles more than BKSs, as sometimes on R and RC instances, it was not able to achieve minimal vehicle number, for example in instance RC105. If only the best runs are analyzed, it can be seen that at least once the BKS solution with minimal vehicle number is achieved, resulting in the same total vehicle number. The overall average total distance percentage difference to the BKS solutions is 0.13. In VRPTW, the total distance traveled is equal to the sum of travel times, as constant speed values of 1 are used. To put it into the context of time, the average total time of achieved solutions is presented. It can be seen that total time is significantly larger than the travel time due to the service and waiting times. The average execution time of 5.44 minutes is not the best one, but it is comparable to other applied methods in the literature [132, 150, 152].

Table 5.2: VRPTW results - distance

Inst.	BKS					HALNS					
	K	d	\bar{K}	K_{best}	ΔK	\bar{d}	d_{best}	Δd	$\Delta_p d$	tot_{best}	\bar{t}_e
C101	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.007
C102	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.033
C103	10	828.06	10	10	0	828.06	828.06	0	0	9963.03	0.021
C104	10	824.78	10	10	0	824.78	824.78	0	0	10178.21	0.099
C105	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.006
C106	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.006
C107	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.007
C108	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.009
C109	10	828.94	10	10	0	828.94	828.94	0	0	9828.94	0.039
C201	3	591.56	3	3	0	591.56	591.56	0	0	9591.56	0.008
C202	3	591.56	3	3	0	591.56	591.56	0	0	9591.56	0.014
C203	3	591.17	3	3	0	591.17	591.17	0	0	9601.72	0.021
C204	3	590.6	3	3	0	590.6	590.6	0	0	9590.6	1.361
C205	3	588.88	3	3	0	588.88	588.88	0	0	9588.88	0.002
C206	3	588.49	3	3	0	588.49	588.49	0	0	9588.49	0.003
C207	3	588.29	3	3	0	588.29	588.29	0	0	9660.4	0.006
C208	3	588.32	3	3	0	588.32	588.32	0	0	9744.23	0.001
R101	19	1650.8	19	19	0	1650.8	1650.8	0	0	3599.45	1.2
R102	17	1486.12	17	17	0	1486.36	1486.12	0	0	3202.51	1.602
R103	13	1292.68	13.5	13	0	1238.17	1292.68	0	0	2755.79	2.816
R104	9	1007.31	9.4	9	0	999.08	1007.31	0	0	2055.48	7.271
R105	14	1377.11	14	14	0	1377.11	1377.11	0	0	2631.56	0.173
R106	12	1252.03	12	12	0	1256.18	1252.03	0	0	2360.62	1.793
R107	10	1104.66	10	10	0	1114.07	1104.66	0	0	2252.28	5.213
R108	9	960.88	9	9	0	963.51	960.88	0	0	2006.98	5.515
R109	11	1194.73	11	11	0	1204.52	1194.73	0	0	2256.67	3.376
R110	10	1118.84	10	10	0	1127.86	1118.84	0	0	2201.7	3.745
R111	10	1096.72	10	10	0	1096.73	1096.73	0.01	0	2198.5	10.156
R112	9	982.14	9.2	9	0	992.94	982.94	0.8	0.08	2001.37	7.611
R201	4	1252.37	4	4	0	1252.37	1252.37	0	0	3570.66	0.489
R202	3	1191.7	3	3	0	1194.43	1191.7	0	0	2744.86	7.833
R203	3	939.5	3	3	0	942.89	941.41	1.91	0.2	2772.01	13.273
R204	2	825.52	2	2	0	832.48	827.12	1.6	0.19	1971.53	17.652
R205	3	994.43	3	3	0	1005.47	994.43	0	0	2511.17	7.67
R206	3	906.14	3	3	0	915.01	906.14	0	0	2493.58	8.034
R207	2	890.61	2	2	0	895.57	890.61	0	0	1988.54	16.484
R208	2	726.82	2	2	0	732.86	726.82	0	0	1833.37	26.622
R209	3	909.16	3	3	0	913.44	909.16	0	0	2405.36	14.714
R210	3	939.37	3	3	0	942.73	939.37	0	0	2708.37	8.337
R211	2	885.71	2	2	0	906.39	891.89	6.18	0.7	1902.35	19.108
RC101	14	1696.95	14.6	14	0	1631.48	1697.43	0.48	0.03	2990.97	6.428
RC102	12	1554.75	12.5	12	0	1486.08	1562.96	8.21	0.53	2779.66	2.252
RC103	11	1261.67	11	11	0	1263.38	1261.67	0	0	2473.49	5.995
RC104	10	1135.48	10	10	0	1135.49	1135.48	0	0	2271.25	3.009
RC105	13	1629.44	13.7	13	0	1554.99	1661.65	32.21	1.98	2911.8	4.678
RC106	11	1424.73	11.4	11	0	1389.88	1470.7	45.97	3.23	2571.2	5.355
RC107	11	1230.48	11	11	0	1231.84	1230.54	0.06	0	2344.93	1.647
RC108	10	1139.82	10	10	0	1139.82	1139.82	0	0	2260.95	3.85
RC201	4	1406.94	4	4	0	1406.94	1406.94	0	0	3358.41	2.574
RC202	3	1365.65	3	3	0	1374	1365.65	0	0	2674.82	2.596
RC203	3	1049.62	3	3	0	1058.8	1051.82	2.2	0.21	2669.47	8.971
RC204	3	798.46	3	3	0	798.5	798.46	0	0	2515.93	14.614
RC205	4	1297.65	4	4	0	1297.65	1297.65	0	0	3464.61	7.325
RC206	3	1146.32	3	3	0	1146.99	1146.32	0	0	2447.95	5.868
RC207	3	1061.14	3	3	0	1066.06	1061.14	0	0	2416	13.988
RC208	3	828.14	3	3	0	834.31	829	0.86	0.1	2275.54	23.396
AVG	7.23	1021.19	7.29	7.23	0	1018.42	1022.99	1.79	0.13	4727.73	5.44
SUM	405	57186.88	408.3	405	0	57031.47	57287.37	100.49	7.25	264752.95	304.87

5.4.2 TD-VRPTW

The results on TD-VRPTW instances are presented in Table 5.3. The minimization objective is the sum of travel times (equation 2.58). As there is no benchmark website containing BKSs for TD-VRPTW, as there is for VRPTW, the BKS solution out of three research papers that dealt with TD-VRPTW is selected: CARI [26], RINC [180], and FIGL [25]. Due to the 12 different speed configurations, the TD-VRPTW researchers report only the cumulative sum of vehicle number K , total distance traveled d , and travel time tt . Here, similar measures per whole instance set are considered, with an additional measure of average and best cumulative travel time \bar{tt} and tt_{best} , cumulative best total time tot_{best} and cumulative average execution time \bar{t}_e which is not reported in any of the related researches. The difference between BKSs and HALNS are reported as difference in cumulative vehicle number ΔK , travel time Δtt , traveled distance Δd and percentage travel time $\Delta_p tt$. Detailed tables per configuration types are presented in appendix B, Tables 1-12. The HALNS method, in almost all configurations outperforms other methods from the literature, as it was able to further decrease the number of vehicles, with in total 43 vehicles less. Also, the HALNS was able to achieve significantly better configurations with a fewer vehicles in terms of total travel time, and total traveled distance. The average percentage decrease in travel time is 2.46%, while the average percentage decrease in the distance is around 1.56%. Results of conducted tests show that HALNS can be applied for solving time-dependent routing problems.

Table 5.3: TD-VRPTW results - travel time

Type	BKS				HALNS						Difference				
	Name	K	d	tt	K	K_{best}	\bar{tt}	tt_{best}	d_{best}	tot_{best}	\bar{t}_e	ΔK	Δtt	Δd	$\Delta_p tt$
A1	CARI	385	58780	47322	380.8	379	46622.17	46402.14	57443.08	249637.83	378.94	-6	-919.86	-1336.92	-1.94
A2	CARI	360	57969	39573	357.9	356	38914.28	38674.78	56680.61	241341.05	376.46	-4	-898.22	-1288.39	-2.27
A3	CARI	348	58447	34984	348.1	347	33866.45	33706.51	56620.33	237316.19	379.58	-1	-1277.49	-1826.67	-3.65
B1	CARI	399	59101	48293	400.3	395	47254.16	47213.29	57898.38	261555.8	397.33	-4	-1079.71	-1202.62	-2.24
B2	RINC	378	59179	41878	378.8	375	40614.67	40491.77	58803.03	253935.28	379.35	-3	-1386.23	-375.97	-3.31
B3	RINC	370	59018	37480	369.7	368	36228.49	36055.37	59603.35	252316.73	369.54	-2	-1424.63	585.35	-3.80
C1	RINC	387	57842	47051	382.9	380	46790.84	46592.45	57543.3	251727.98	385.8	-7	-458.55	-298.7	-0.97
C2	CARI	359	58524	40548	359.5	358	39422.34	39155.76	56540.24	244503.51	386.64	-1	-1392.24	-1983.76	-3.43
C3	CARI	350	58108	35780	349	348	35203.76	34984.84	56988.78	240408.03	384.38	-2	-795.16	-1119.22	-2.22
D1	RINC	401	57639	48841	401.4	398	48344.91	48155.52	57160.95	257659.12	385.37	-3	-685.48	-478.05	-1.40
D2	CARI	382	58476	43074	381.27	377	42410.77	42266.11	57584.72	250345.55	346.22	-5	-807.89	-891.28	-1.88
D3	RINC	375	58369	39473	372.92	370	38701.49	38501.64	57622.32	246895.73	357.35	-5	-971.36	-746.68	-2.46
AVG		374.5	58454.33	42024.75	373.55	370.92	41197.86	41016.68	57540.76	248970.23	377.25	-3.58	-1008.07	-913.58	-2.46
SUM		4494	701452	504297	4482.59	4451	494374.33	492200.18	690489.09	2987642.8	4526.96	-43	-12096.82	-10962.91	-29.57

Although the travel time objective is commonly used as a measure of time in TD-VRPTW, the minimization of total time can produce different results. The example of results on TD-VRPTW with total time minimization on A1, B1, C1 and D1 configurations are presented in Table 5.4, while the detailed tables are presented in Appendix section C, Tables 13-16. The reference values for the comparison are the best number of vehicles K_{tt}^{best} and best total times tot_{tt}^{best} achieved by the travel time minimization. It can be seen that the total time decreased between 2.34% and 4.8%, at the expense that HALNS was not able to produce the lowest number of vehicles on configurations A1, B1 and C1, resulting in total of 7 vehicles more. As

expected, the total traveled distance increased, as well as the execution time.

Table 5.4: TD-VRPTW results - total time

Type	K_{it}^{best}	tot_{it}^{best}	\bar{K}	K_{best}	\bar{tot}	tot_{best}	Δtot	$\Delta_p tot$	d_{best}	\bar{i}_e
A1	379	249637.83	382	382	243926.43	243787.73	-5850.1	-2.34	62127.56	1100.94
B1	395	261555.8	397.33	397	249353.33	249007.17	-12548.63	-4.8	64356.01	1058.98
C1	380	251727.98	383.18	382	246665.22	246276.96	-5451.02	-2.17	63685.24	1026.38
D1	398	257659.12	400.57	398	247072.11	246437.62	-11221.5	-4.36	61931.19	1009.19

5.5 TD-EVRPTW-FR and TD-EVRPTWDCS-FR

To solve the TD-EVRPTW-FR, the same HALNS as for EVRPTW-FR is used, with only difference in travel time computation. As this is a new problem, the instances for this problem were not presented so far. To overcome this problem, in this thesis, EVRPTW instances are coupled with TD-VRPTW to create instances for TD-EVRPTW. The configuration of users in EVRPTW instances is used as a base, and on top of that, the speed coefficients from TD-VRPTW instances are added. As already mentioned in section 2.6 the delivery period is discretized into five time buckets that represent typical urban traffic periods: morning free-flow, morning rush, transition period between morning and afternoon rush, afternoon rush, and evening free-flow. In the preprocessing step, all instance arcs per configuration were checked whether they satisfy all constraints or not, and whether the triangle inequality is valid for the travel times or not [185]. The summary of results per each of the 12 configuration types are presented in Table 5.5, while the detailed tables are presented in Appendix in section D, Tables 17-28. The reference values used for the comparison are the BKSs from EVRPTW-FR problem. It can be seen that in all instances, the vehicle number is decreased, as higher speed values are used than in EVRPTW-FR, with in total 415 vehicle less. Due to the same reason, the total travel time also decreased by 28.92%, as well as the total time. In all cases, the configuration types with label 3 (A3, B3, C3, and D3) produced the lowest number of vehicles as the highest speed values are used in those sets. The comparison between the configuration types A, B, C, and D shows that the best results are achieved with set A, which has the highest speed values in the morning and afternoon rush periods, and slightly lower speed values in transition period between morning and afternoon rush hours. The worst results are achieved for configuration D, which has higher speed values towards the end of the planning horizon and lower speed values at the beginning of the delivery. This happens, as in most instances, the customers that have an early late time window have to be visited right away, and a lot of time is consumed to visit such customers, especially if at the beginning the speed values are the lowest. Compared to the EVRPTW-FR without time-dependent travel times, in all configuration types, the number of visited CS increased, on average in all instances by 60 CSs. The reason is the average increase in total traveled distance

of 0.39%, and favoring shorter travel time routes instead of the shorter distance routes.

Table 5.5: TD-EVRPTW-FR results - travel time

Type	EVRPTW-FR		HALNS							Difference					
	K	d	K	K_{best}	\bar{t}	tt_{best}	d_{best}	tot_{best}	m	\bar{t}_e	ΔK	Δtt	$\Delta_p tt$	Δd	$\Delta_p d$
A1	440	59866.66	420.65	415	49097.54	49020.26	59869.34	284591.93	545	580.82	-25	-10846.4	-18.12	2.68	0.004
A2	440	59866.66	401.19	397	42259.03	42230.02	60104.02	275338.69	574	636.65	-43	-17636.64	-29.46	237.36	0.396
A3	440	59866.66	389.5	387	37647.91	37442.74	60243.05	272433.03	602	646.7	-53	-22423.92	-37.46	376.39	0.629
B1	440	59866.66	426.12	423	47624.37	47405.64	60079.81	291726.01	554	565.37	-17	-12461.02	-20.81	213.15	0.356
B2	440	59866.66	407.3	403	40159.52	39825.8	60624.32	284971.03	589	731.06	-37	-20040.86	-33.48	757.66	1.266
B3	440	59866.66	400.34	397	35535.5	35415.51	62140.05	284309.25	639	758.39	-43	-24451.15	-40.84	2273.39	3.797
C1	440	59866.66	423.17	422	48548.28	48421.24	59740.69	288780.47	542	724.31	-18	-11445.42	-19.12	-125.97	-0.21
C2	440	59866.66	405	405	41705.95	41579.87	59724.31	279456.15	567	781.08	-35	-18286.79	-30.55	-142.35	-0.238
C3	440	59866.66	395.16	391	37304.89	37244.26	59683.65	275183.33	578	869.14	-49	-22622.4	-37.79	-183.01	-0.306
D1	440	59866.66	427.91	425	49755.77	49507.08	59518.3	291664.43	526	847.82	-15	-10359.58	-17.3	-348.36	-0.582
D2	440	59866.66	408.75	407	43348.04	43268.85	59776.84	285391.7	562	762.37	-33	-16597.81	-27.72	-89.82	-0.15
D3	440	59866.66	397.05	393	39381.94	39269.81	59714.31	281938.26	583	623.85	-47	-20596.85	-34.4	-152.35	-0.254
AVG	440	59866.66	408.51	405.42	42697.4	42552.59	60101.56	282982.02	571.75	710.63	-34.58	-17314.07	-28.92	234.9	0.39
SUM	5280	718399.92	4902.14	4865	512368.74	510631.08	721218.69	3395784.28	6861	8527.56	-415	-207768.84	-347.05	2818.77	4.71

Additionally the total time in TD-EVRPTW-FR and recharging costs in TD-EVRPTWDCS-FR are minimized, with the results for four travel time configurations A1, B1, C1 and D1 presented in Table 5.6. The detailed tables for each configuration type are presented in Appendix in sections E and F, in Tables 29-36. The first part of the Table 5.6 is related to the total time minimization in TD-EVRPTW-FR, where reference values are the best total time value of TD-EVRPTW-FR with travel time minimization. It can be seen that HALNS was not able to achieve the same number of vehicles on the A1 configuration, as it produced three vehicles more, while on each of the other configurations, it produced one vehicle less. As expected, the total time was reduced in all configurations ranging from 4.56% to 6.67% decrease, at the expense of an increase in total traveled distance and total travel time. Interestingly, in all configuration types, the number of visits to CSs increased.

The second part of the table presents results for the recharging cost minimization with different charger types in CSs for TD-EVRPTWDCS-FR. The reference values for the comparison are the best recharging costs of TD-EVRPTW-FR with travel time minimization. The recharging costs increased up to 1.77% mostly due to the fact that rapid and fast charging are utilized much more, which led to a significant reduction in the number of vehicles, up to 41, which is similar to the reduction of vehicles between EVRPTW-FR and EVRPTWDCS-FR. The total distance traveled, and the number of visited CSs are also slightly reduced, at the expense of an increase in total travel time. At first glance, the 374 vehicles for A1 configuration in TD-EVRPTWDCS-FR seems impossible as in TD-VRPTW the best found solution for A1 is 379 vehicles, and TD-VRPTW has fewer constraints. The same goes for the other configuration types. The problem is that these two values are hardly directly comparable, as the used Solomon instances are not completely the same. More precisely, as Schneider et al. [18] pointed out, the locations of customers in the instance are the same, but the time-window values were relaxed in EVRPTW instances, as some instances were always infeasible in the original Solomon instances if BEVs are included. This problem does not come to the fore when a standard recharging rate,

set in EVRPTW instances, is used, but rather when rapid and fast charging rates are used, which then produce a solution configuration with a lower number of vehicles. If noted closely, this can also be seen in the comparison between VRPTW, where the BKSs contain 405 vehicles (Table 5.2), and the BKSs for EVRPTWDCS-PR which contain 402 vehicles [22] (Table 4.24), or the BKSs for EVRPTWDCS-FR with also up to 402 vehicles (Tables 4.21 and 4.22).

Table 5.6: TD-EVRPTW-FR total time and TD-EVRPTWDCS-FR recharging cost - results

Type	tt^{best}		HALNS								Difference			
	K	tot/rec	f	\bar{K}	K_{best}	\bar{f}	f_{best}	d_{best}	tt_{best}	m	\bar{i}_e	ΔK	Δf	$\Delta_p f$
A1	415	284591.93	<i>tot</i>	418	418	272122.84	271613.75	66069.57	54185.65	620	1339.93	3	-12978.18	-4.56
B1	423	291726.01	<i>tot</i>	422	422	271989.62	271472.73	65962.24	53344.87	613	1371.29	-1	-20253.28	-6.94
C1	422	288780.47	<i>tot</i>	422.1	421	274014.88	273120.12	67850.62	54366.82	619	1171	-1	-15660.35	-5.42
D1	425	291664.43	<i>tot</i>	424	424	272881.07	272207.06	64865.74	54890.51	583	1015.29	-1	-19457.37	-6.67
A1	415	59869.34	<i>rec</i>	378.7	374	60794.88	60437.62	58817.27	49419.61	549	727.61	-41	568.28	0.95
B1	423	60079.81	<i>rec</i>	391.4	390	60832.17	60475.88	59199.17	47980.01	552	696.62	-33	396.07	0.66
C1	422	59740.69	<i>rec</i>	384	384	60872.36	60799.26	59369.46	48861.39	557	639.93	-38	1058.57	1.77
D1	425	59518.3	<i>rec</i>	387.67	387	60816.46	60507.86	59065.21	49698.46	543	726.58	-38	989.56	1.66

Adapted real-world delivery problem

To show the application of the developed HALNS method in real-world delivery problems, one delivery problem from the real world is selected and adapted. The real-world problems often include additional special constraints by the company that performs the delivery. These constraints are most often simplified and added to the model of the problem. The selected real-world problem represents the post delivery problem in the City of Zagreb. In total, 225 customers are considered, representing ministries, city offices, and large companies that have a contract with the post service. The observed delivery problem is presented in Figure 6.1. The depot is represented by a black pin, while the customers are colored from red to green pins, with red indicating customers that need to be visited sooner due to the earlier closing service time and green indicating customers that can be visited later. The mail is delivered in bags, which are considered as a load unit size in the problem. The example of demand (bags) per customer is presented in Figure 6.2. The customers' demands range from zero to six bags, with zero representing the delivery demand of a couple of letters. The service time at each customer is 3 minutes, and the delivery is performed between 06:00 and 11:00. The fleet consists of 16 vehicles with equal load capacities of 20 bags. The position of each customer is geocoded. This process is important as a wrong location can have a significant impact on routing plans. In the original dataset, some customers were geocoded on links that are not accessible by the private vehicle, i.e., in a pedestrian zone, private company area, one-way roads, etc. In such occasions, the correction of the geographic coordinates was performed to improve the accuracy. In the real case, the delivery is performed by 16 vehicles, and the total traveled distance is 240.79 km. The used real world delivery data were collected on projects with industry conducted at the Faculty of Transport and Traffic Sciences, University of Zagreb, and cannot be used outside the institution.

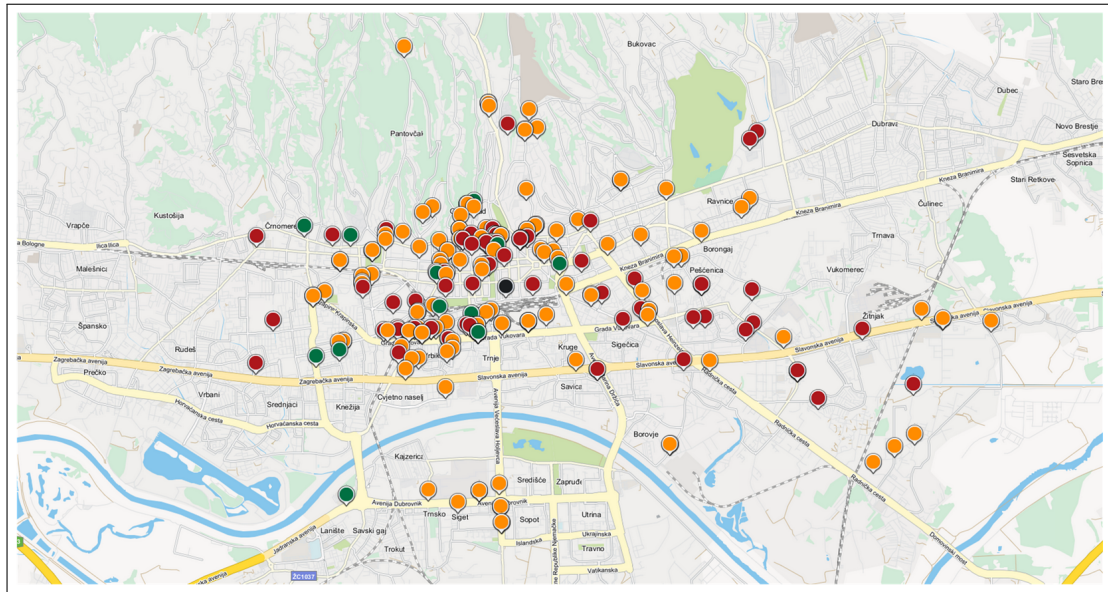


Figure 6.1: Observed delivery problem

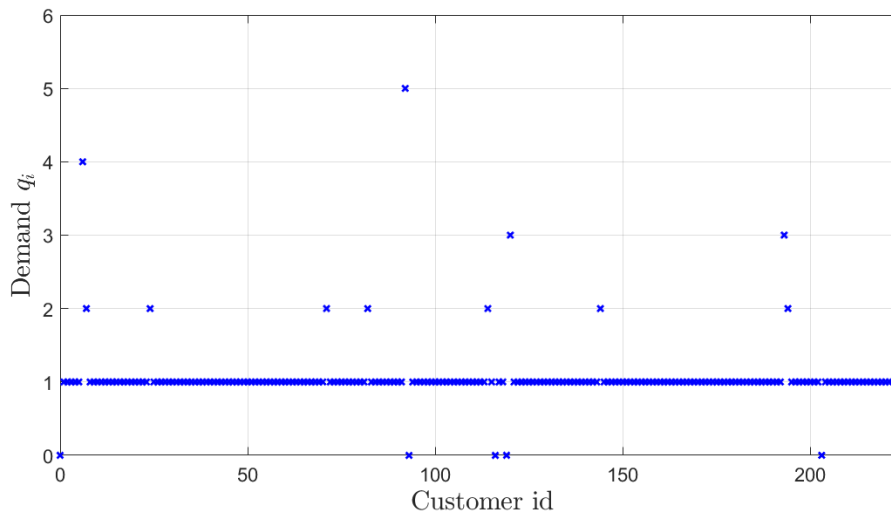


Figure 6.2: Demand per customers in the post delivery problem

6.1 Digital map

The digital road network map of the Republic of Croatia is provided by [Mireo Inc.](#). The digital map consists of 625702 road segments (links) covering 55049 km, with a median link length of 88 m. The link represents a segment of the road between two consecutive intersections. The line interpolation between the link start point and link end point is used to represent the real road segment. Each link contains a unique identifier, static speed and speed limit, all set by the digital map provider. Links are divided into seven categories, from highways to local streets. Directions for two-way roads are represented by separate links which have identical geometric representation but different orientation. Each link also contains the name of the street that it

belongs to.

The example of the digital traffic map for the City of Zagreb that will be used in this thesis is presented in Figure 6.3. The map consists of 66799 links which are colored based on their type. Highways are colored with orange color, main avenues and streets with dark blue color, main neighborhood streets with purple color, and local streets with light blue color. For background map representation as in Figure 6.1, the [web Mireo map](#) is used, coupled with JavaScript code for visual representation.

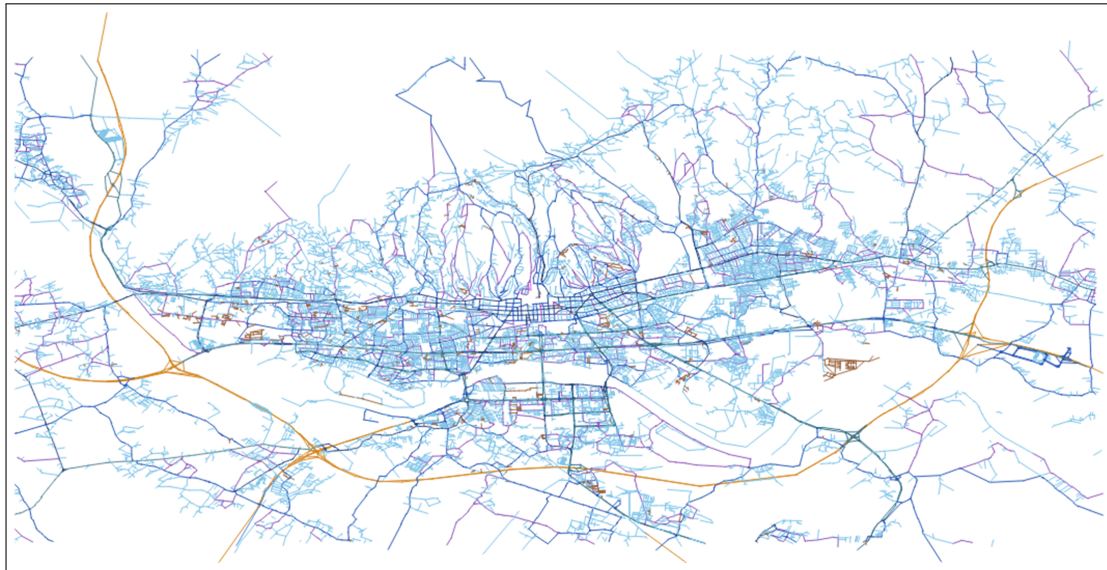


Figure 6.3: Road network in the City of Zagreb

To use the road links in the digital map for routing applications, commonly, the road traffic network is modeled as a graph. In this thesis, links in the road network are modeled as vertices, while the edges represent connections between links. The connection between two links for the same road segment that represent two different directions is removed to overcome the cycling in a graph. Such connections are only allowed for the starting link, for which most often there is a possibility to go on either of the links for different directions. This means that vehicle at the start can choose in which direction to go unless it is a one-way road, or there are two different links for different directions, if there is an unused road surface that divides directions, like a road fence that divides the avenues, or full white line.

6.2 VRPTW

The basic version of the problem can be modeled as the VRPTW problem. To solve the VRPTW problem the shortest path distance between each customer in the problem has to be computed, as well as the corresponding travel time. These values are commonly stored in the distance and time matrix $M = n \times n$, where n is the number of customers in the problem instance. In this case, the matrix has a size of $M = 226 \times 226$, as there are 225 customers and 1 depot in the observed problem. The matrices are asymmetric as different paths are traversed between customers in opposite directions. The shortest distance path between customers is expressed as a set of consecutive links in the digital map that minimize the total traveled distance. To compute the shortest path between customers, the most commonly applied algorithms are Dijkstra, Bellman-Ford, A^* , contraction hierarchies, etc. [186, 187]. In this thesis, the Dijkstra+s algorithm given by Algorithm 6.1 is used to compute the shortest distance path. First, all vertices in graph G are added to the list of unprocessed vertices Q . The first vertex is set to have zero value, while all the other vertices are set to have an infinity value. The preceding vertex of each vertex in the graph is set to None. Then, the algorithm loops through the list Q , and in each iteration, removes the vertex u that has the lowest value. If the distance is minimized, the vertex that has the lowest total traveled distance up to it is removed. After the vertex is removed, values and preceding vertices of neighboring unprocessed vertices of u are updated, only if the computed new value is lower than the current value of a vertex. The new value is computed as the current value of vertex u plus the arc weight $w_{u,v}$ representing the cost of traversing the arc. As a result, Dijkstra algorithm computes the shortest paths from starting vertex v_o to every other vertex in the graph G . The path from starting vertex v_o to any other vertex v_n is determined by backward looping of preceding vertices, starting from the vertex v_n and ending in vertex v_o . This means that Dijkstra algorithm does not need to be recomputed if a starting vertex in the graph does not change. Therefore, in VRP problems, the computation time can be reduced, as the Dijkstra algorithm needs to be computed for each customer, meaning that the shortest paths to all other customers are just restored from an already solved graph with a set starting vertex. The Dijkstra algorithm works on a graph that contains only non-negative arc weights, as otherwise, cycling in the graph can occur.

The most common implementation of the Dijkstra algorithm for a list of the unprocessed Q vertices uses an unsorted array, which has the complexity $\mathcal{O}(n)$ for extraction of the vertex with the lowest value. If $|V|$ represents the number of vertices in graph G , the complexity of such implementation would be $\mathcal{O}(|V| + |V|^2) = \mathcal{O}(|V|^2)$. Instead of an unsorted array, in this thesis, the minimum priority queue is used. The idea is based on the minimum heap structure, which stores the data based on their priority. Here, the minimum values are stored at the back of the priority queue. The vertex with the lowest value can be determined in $\mathcal{O}(1)$ time, while

the insertion of a vertex in such queue is performed in $\mathcal{O}(\log n)$ time, where n is the number of elements in the queue. Therefore, the complexity of such implementation of Dijkstra algorithm is $\mathcal{O}(|V|\log|V| + |E|\log|V|) = \mathcal{O}((|V| + |E|)\log|V|)$, where $|E|$ is the number of arcs in the graph G . If an assumption is made that is $|E| \approx |V|$, then the complexity of the Dijkstra algorithm is $\mathcal{O}(|V|\log|V|)$. This assumption is valid in traffic road network graphs as one vertex usually contains only several neighboring vertices. For a dense graph in which $|E| = |V|^2$ the complexity of the algorithm increases to $\mathcal{O}(|V|^2\log|V|)$.

Algorithm 6.1 Dijkstra algorithm

Input: Graph G and start vertex v_0

- 1: $Q \leftarrow$ Initialize list of unprocessed vertices
- 2: **for each** vertex v in graph G **do**
- 3: **if** $v = v_0$ **then**
- 4: $value(v) \leftarrow 0$
- 5: **else**
- 6: $value(v) \leftarrow \infty$
- 7: **end if**
- 8: $preceding(v) \leftarrow None$
- 9: Add v to Q
- 10: **end for**
- 11: **while** Q is not empty **do**
- 12: $u \leftarrow$ Remove vertex from Q with the lowest value
- 13: **for each** unprocessed neighboring vertex v of vertex u **do**
- 14: $val_{new} \leftarrow value(u) + w_{u,v}$
- 15: **if** $val_{new} < value(v)$ **then**
- 16: $value(v) \leftarrow val_{new}$
- 17: $preceding(v) \leftarrow u$
- 18: **end if**
- 19: **end for**
- 20: **end while**

The Dijkstra algorithm computes the shortest distance path between customers based on the sum of the links' lengths in the path. In VRPTW, beside the distance measure, the travel time on a path is needed to check the time window constraints. To compute the travel time on a path, first, the set link speed by the map provider is used. This is a static speed determined by the experience. As linearized travel times are not considered in VRPTW, the travel time on the link is computed as a ratio of link's length and speed. The result of running the Dijkstra algorithm on the problem are two matrices: distance (which is minimized) and travel time, and both are used as an input for the HALNS method. The average running time of an implemented Dijkstra algorithm on a graph for the observed road network of Zagreb (Figure 6.3) is 2.97 s. The graph contains 66799 vertices and 107736 arcs, which is in average 1.61 arc per vertex, meaning that the graph is not dense. It is important to note that there are faster implementations of the shortest path algorithm that include heuristic methods and searching in both directions,

start and end, but they were not considered in this thesis, as the purpose of this chapter is to show the applicability of HALNS in the real world.

The VRPTW solution of the delivery problem with used set digital map speeds, solved by HALNS, is presented in Table 6.1. The distances are expressed in kilometers, total and travel times in hours, and execution time in minutes. As in previous testings, the HALNS was run ten times; therefore, the average value of vehicle number and total distance traveled is presented. The best VRPTW solution is presented in Figure 6.4. In total, 12 vehicles are used for the delivery with the total traveled distance of 223.05 km in the best solution and 226.15 km on average. The total travel time in the best solution is 7.32 h, while the total time is 38.34 h. At first glance, total travel time seems relatively small, but if put into context of relatively close customers in the urban center of Zagreb, it can be seen that not much time is spent on driving, and more time is spent on the service and waiting times. If computed, the average driving speed would be 30.53 km/h. Compared to the real performed delivery, the vehicle number is reduced by 4 vehicles, and additionally, the total traveled distance is reduced by 17.74 km, which is a 7.37 percentage decrease. The results are hardly comparable to previous researches that dealt with this real-life problem [150, 158], as first of all, speeds on the road network changed from the time in which the studies were conducted, and second, slightly different Dijkstra computation is used, that approximates customer position with the whole closest link in the digital map, and not just the part of the link that is traversed. This generally means that results should be worse than the results in previous studies, but still, in some cases, better solutions were found, which shows the good performance of the HALNS method.

Table 6.1: VRPTW real example with set map speeds - results

Real		HALNS									
K	d	\bar{K}	K_{best}	ΔK	\bar{d}	d_{best}	Δd	$\Delta_p d$	t_{best}	tot_{best}	\bar{t}_e
16	240.79	12	12	-4	226.15	223.05	-17.74	-7.37	7.32	38.34	6.411

6.3 Speed profiles

To better estimate traffic conditions on a road network, Floating Car Data (FCD) can be used, as by tracking vehicles equipped with Global Navigation Satellite System (GNSS) devices, the real-time traffic data can be recorded [174, 188, 189, 190]. FCD act as location-based mobile sensors on the road network and record formatted data: geographic coordinates, timestamp, instantaneous velocity, heading, etc. Collecting FCD in a larger period of time results in voluminous and detailed traffic data, which then imposes on the application of data-mining procedures to extract spatio-temporal traffic patterns [188, 189, 191, 192, 193, 194].

In this thesis, FCD collected from vehicles equipped with GNSS devices are used to estimate

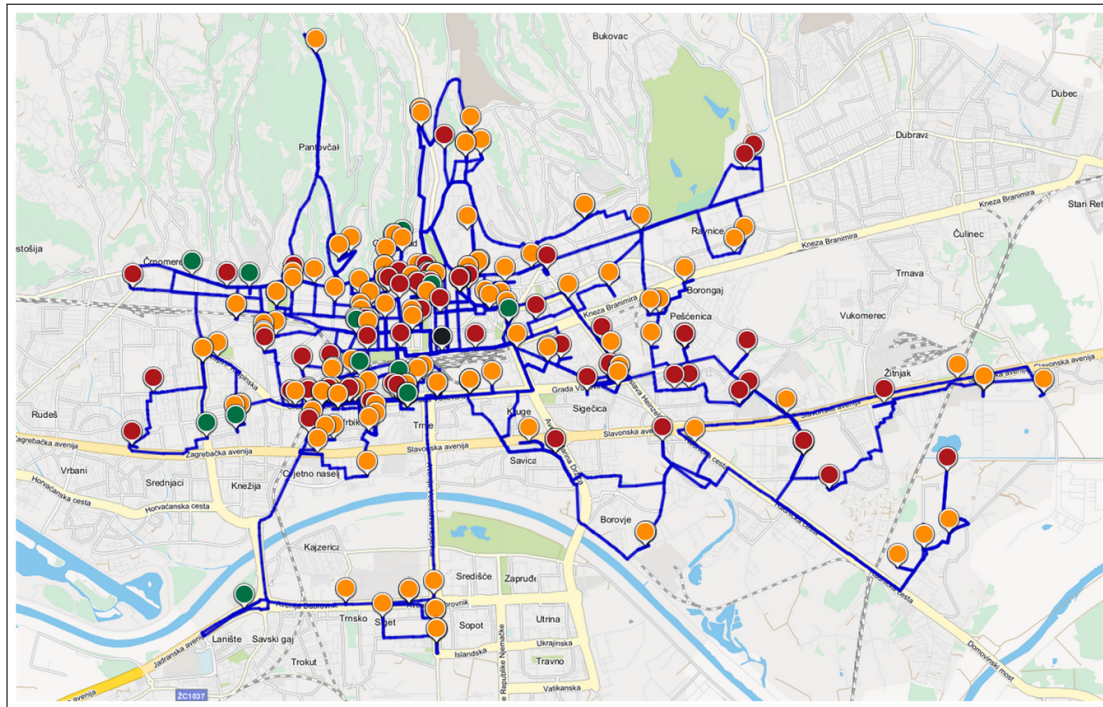


Figure 6.4: VRPTW solution - set map speeds

speed values on a road network. These data are collected as a part of the [SORDITO](#) project. The used historical FCD were recorded during a five-year period between August 2009 and October 2014 by 4908 vehicles. The GNSS tracking devices were installed in freight delivery vehicles, delivery vans, taxis, personal cars, etc. This resulted in 6.55 billion GNSS records, with a storage volume of 320 GB. The records for vehicles in the driving mode were sampled approximately every 100 m, while the records for stationary vehicles were sampled every 5 min. The FCD data were used to compute links' speed profiles. Road speed profile can be defined as expected vehicle speed for a particular link during the observed time period [195]. This way, the temporal behavior of recurrent traffic conditions can be taken into consideration.

Two features that influence recurrent congestion were taken into account when computing speed profiles: seasons and days of the week. First, the data were divided into two seasons: summer, containing July and August, and rest of the year. Due to the lower number of data during the night [175, 196], the one-day period was divided into two: daytime period (05:30-22:00 - working hours of vehicles in the dataset) and nighttime period (22:00-05:30). Further on, the daytime period was observed for different days in the week, from Monday to Sunday. A discretization of speed profiles in five-minute time buckets was selected as a balance between the estimation quality and the number of data [197, 198]. In the end for each time bucket t_k the space mean speed $v(t_k)$ was computed by equation 6.1. Usually, mean travel time speed is used to average speeds in the observed time bucket [191, 199], but in the project, the space mean speed was used as: (i) it is the arithmetic mean speed of all vehicles occupying a given link [198, 200], and (ii) it gives higher weight to slower vehicles which is suitable for quantifying

congestion [201]. For more details on the computation of speed profiles, the reader is referred to [24, 28, 158].

$$v(t_k) = \frac{n}{\sum_{i=1}^n \frac{1}{v_i(t_k)}} \quad (6.1)$$

In total, 564112 speed profiles were computed for the rest of the year season and 185708 for the summer season. 73.64% of the higher priority links (from minor city artery to motorway) were covered with the rest of the year speed profiles and 51.15% with the summer speed profiles. The example of computed absolute speed profiles through days of the week for one of the most frequently used link in Zagreb is presented in Figure 6.5a. On the x -axis is the time of day and on the y -axis is the value of speed in km/h. To better visualize speed profiles, the values are plotted as interpolated dotted lines between the discrete point values. Two distinct speed drops from 70 km/h to 20-30 km/h can be distinguished, corresponding to the morning and afternoon rush periods, with the afternoon rush period having higher intensity. During the weekend and between the rush periods, almost free-flow conditions are present. Also, the slight changes between the work-days are present. Monday and Tuesday show the largest congestion occurring in the morning after the weekend as: (i) at the beginning of the week, most of the job activities in the city have to be performed early in the morning, and (ii) many commuters from nearby cities return to the city. On the contrary, the largest afternoon congestion occurs on Friday as many people go on the vacation during the weekend or return to their homes in nearby cities.

Figure 6.5b presents seasonal variability between the summer and the rest of the year season on Monday for an already observed continental link in the City of Zagreb and link located near the City of Zadar on the Adriatic coast. It can be noted that there are almost no speed drops during the summer for the continental link, while during the the rest of the year, the speed drops significantly. The opposite holds for the link on the coast, where congestion occurs only during the summer season and is shifted later in the day.

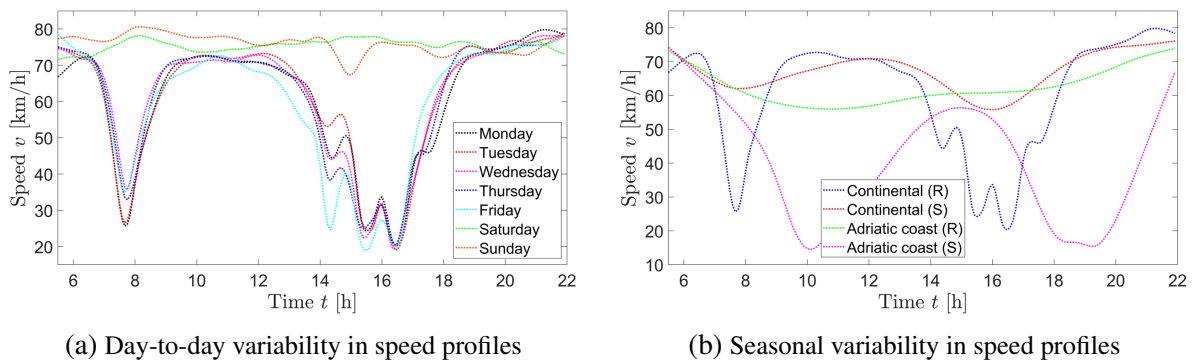


Figure 6.5: Speed profiles

One additional application of speed profiles is the spatio-temporal analysis on the micro

(link) level in different seasons for one of the most used route in the City of Zagreb on Monday, which is presented in Figure 6.6. On the x -axis is the time of day, and on the y -axis is the distance from the start. The route length is 20.95 km spreading from the east entry point to the west exit point of the city, with the route's links represented as horizontal lines. The color changes from red to green, with red representing congestion (relative speed value ≤ 60) and green representing free-flow conditions (relative speed value ≥ 100). The horizontal red lines represent links in the route at which traffic lights are located, and the congestion spreads in time (horizontal) and space (vertical), causing queues and waiting times [170, 174]. During the morning rush period for the rest of the year (Figure 6.6a) the part of the route leading to the city center (≈ 10 km) is heavily congested with congestion length up to 2.5 km and almost no congestion for part of the route leading out of the city. The opposite holds for the afternoon rush period and west-east direction, as the part of the route leading out of the city is heavily congested, and there is no congestion for part of the route leading into the city. Also, the summer season (Figure 6.6b) is significantly less congested than the rest of the year season (Figure 6.6a).

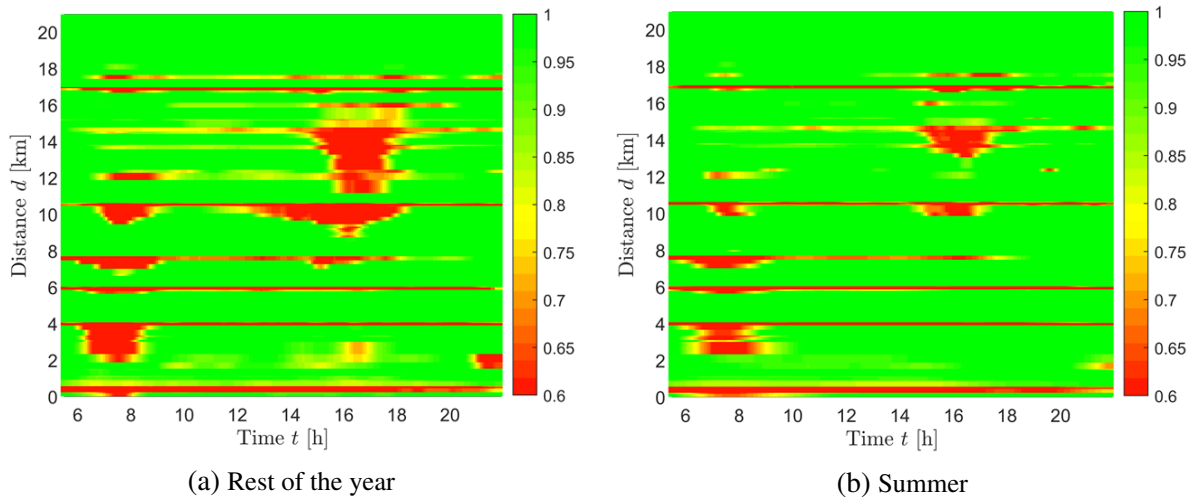


Figure 6.6: Spatio-temporal route analysis in different seasons

6.4 TD-VRPTW

First, to show the importance of using a more accurate speed, the previous VRPTW problem is solved by using average daily speeds per link computed from the speed profile and not the speed values set by the provider. The results are presented in Table 6.2 and the best solution is presented in Figure 6.7. Again in all runs, the solution with 12 vehicles is found. The distance in the best solution increased to 232.76 km, mostly due to the more realistic lower speed values, which as a consequence, has a harder time window completion. This means that if the previous solution with provider speeds is evaluated with average daily speeds, the produced solution

6. Adapted real-world delivery problem

would be infeasible, mostly due to the violation of time windows. The time values increased, as the total travel time in the best solution is 13.39 h, and the total time is 40.42 h. This means that vehicles spent 6.07 h more on the road network than in the case with set provider speed, which is in average 30 min more per vehicle. Here, the average routing speed dropped to 17.38 km/h. Again compared to the real delivery case, the vehicle number is reduced by 4 vehicles and the total traveled distance by 8 km, which is a 3.33% decrease.

Table 6.2: VRPTW real example with average daily speeds - results

Real		HALNS									
K	d	\bar{K}	K_{best}	ΔK	\bar{d}	d_{best}	Δd	$\Delta_p d$	tt_{best}	tot_{best}	\bar{t}_e
16	240.79	12	12	-4	235.08	232.76	-8.03	-3.33	13.39	40.42	7.304

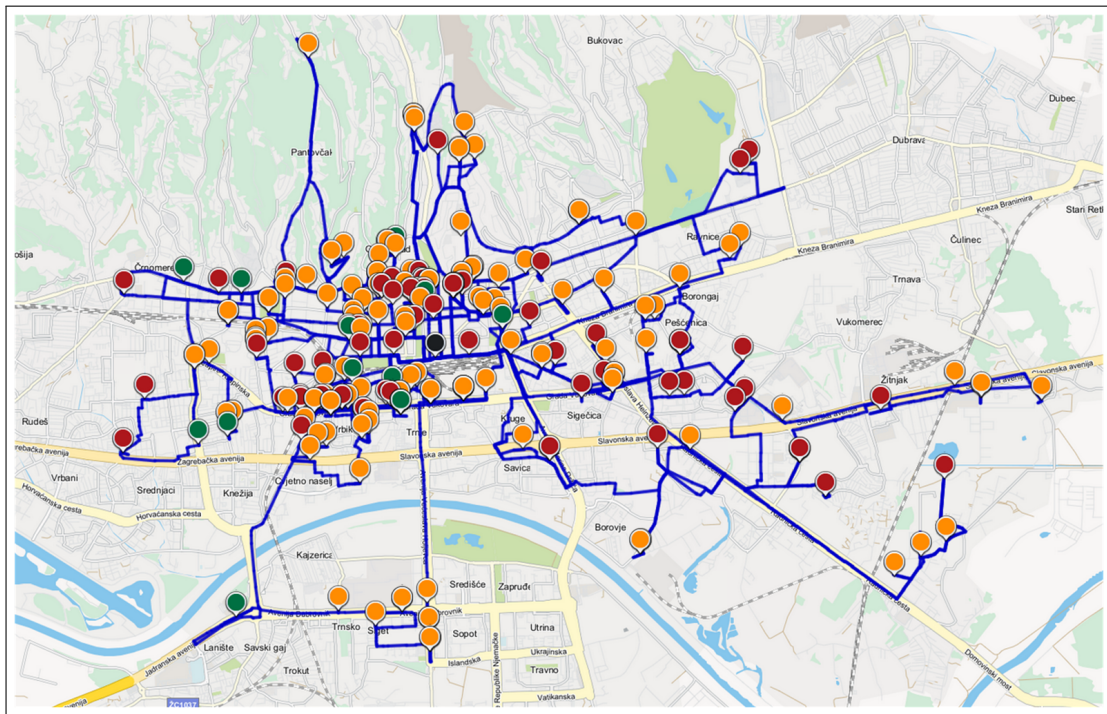


Figure 6.7: VRPTW solution with average daily speeds

To solve the TD-VRPTW problem, the distance and travel time matrices have to be computed for each time bucket. Therefore, these matrices can be represented as three-dimensional matrices $M = k \times n \times n$, where n is the number of customers in the problem instance, and k is the number of time buckets. The previously used Dijkstra's algorithm has to be modified to minimize travel times. In a time-dependent context, the travel time depends on the departure time, which led to the definition of TD-SPP [202, 203] and Time-Dependent Dijkstra algorithm (TDD) [26, 130, 178, 204]. The only difference, in the TDD algorithm, to the classical one is the computation of weights between vertices. Instead of a static distance weight in distance minimization, in travel time minimization, the weight is expressed as a travel time function that

depends on the departure time. As presented by the Algorithm 6.2 the TDD algorithm, additionally as an input, requires the start time t_0 for the start vertex v_0 . The edge weight between vertices is expressed as a function of the starting time $w_{u,v}(start_time(u))$, and when the update is conducted, additionally the future starting time is also updated (line 20). It is important to note that the travel time was computed in the same way as proposed by Figliozzi [25] given by Algorithm 5.1, which linearizes travels times and satisfies FIFO principle.

Algorithm 6.2 Time-dependent Dijkstra algorithm

Input: Graph G , start vertex v_0 and start time t_0

- 1: $Q \leftarrow$ Initialize list of unprocessed vertices
- 2: **for each** vertex v in graph G **do**
- 3: **if** $v = v_0$ **then**
- 4: $value(v) \leftarrow 0$
- 5: $start_time(v) \leftarrow t_0$
- 6: **else**
- 7: $value(v) \leftarrow \infty$
- 8: $start_time(v) \leftarrow \infty$
- 9: **end if**
- 10: $preceding(v) \leftarrow None$
- 11: Add v to Q
- 12: **end for**
- 13: **while** Q is not empty **do**
- 14: $u \leftarrow$ Remove vertex from Q with the lowest value
- 15: **for each** unprocessed neighboring vertex v of vertex u **do**
- 16: $val_{new} \leftarrow value(u) + w_{u,v}(start_time(u))$
- 17: **if** $val_{new} < value(v)$ **then**
- 18: $value(v) \leftarrow val_{new}$
- 19: $preceding(v) \leftarrow u$
- 20: $start_time(v) \leftarrow start_time(u) + val_{new}$
- 21: **end if**
- 22: **end for**
- 23: **end while**

To solve the real-world delivery problem, a $M = 199 \times 226 \times 226$ matrix is generated, as speed profiles are discretized in 198 five-minute intervals for a daytime period and 1 time interval for nighttime period. As there are speed profiles for different days in the week and seasons, for the analyzed delivery problem, the Monday and rest of the year season speed sets were selected. With an average Dijkstra running time of 2.79 s, the computation of such matrix took approximately 34.85 h. The results are presented in Table 6.3 for different minimization objectives: distance d , travel time tt and total time tot , with column \bar{f} representing average corresponding objective value. In all cases, the BKSs with 12 vehicles are achieved. As it can be seen, using time-dependent speeds for distance minimization, instead of a static average speed, further increased both the average and the best total traveled distance by approximately 3 km. The total travel time decreased by almost 2 h compared to the static average speed, while the

6. Adapted real-world delivery problem

difference in total time is only 3 min. Compared to the distance minimization, the minimization of travel times shows that travel times can be further decreased to at best 11.37 h, which is a decrease of 8.4 minutes, at the expense of the increase in total traveled distance by 3.72 km and decrease in total time by 47.4 min. The best solution with the minimization of total travel times is presented in Figure 6.8. The minimization of total times, which is the most complex, interestingly further reduced the total time by 2.73 h, at the expense of the increase in total traveled distance and total travel time by 121.31 km and 4.9 h, respectively. This means that the aim is to visit users that close soon and not leave them to the other vehicles. One explanation would be that vehicle visits one customer in the east part of the city, then one in the west part of the city, and then again goes back to the east part. As a result, vehicles spend much more time on the road network and cover larger distances but are able to finish delivery sooner. The example of the best solution with the minimization of total times is presented in Figure 6.9.

Table 6.3: TD-VRPTW real example with link speed profiles - results

Min.	Real		HALNS									
	K	d	\bar{K}	K_{best}	ΔK	\bar{f}	d_{best}	Δd	$\Delta_p d$	tt_{best}	tot_{best}	\bar{t}_e
d	16	240.79	12	12	-4	238.84	235.12	-5.67	-2.35	11.51	40.47	7.922
tt	16	240.79	12	12	-4	11.44	238.84	-1.95	-0.81	11.37	39.68	7.589
tot	16	240.79	12	12	-4	37.15	360.15	119.36	49.57	16.27	36.95	13.917

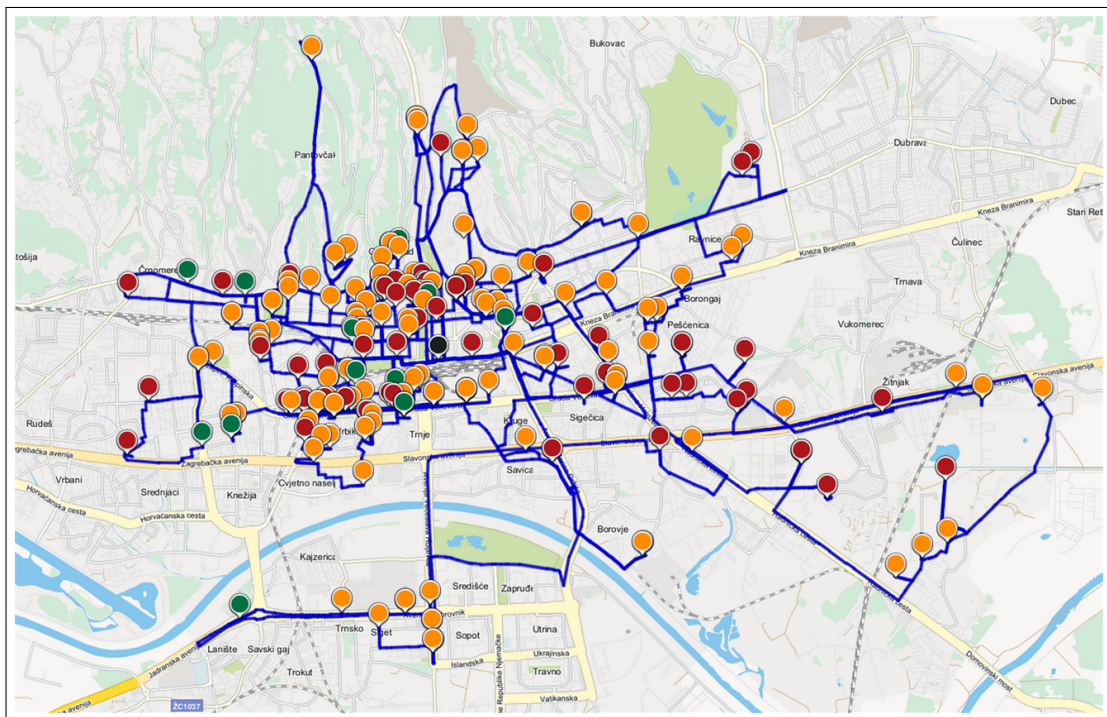


Figure 6.8: TD-VRPTW solution with link speed profiles - travel time

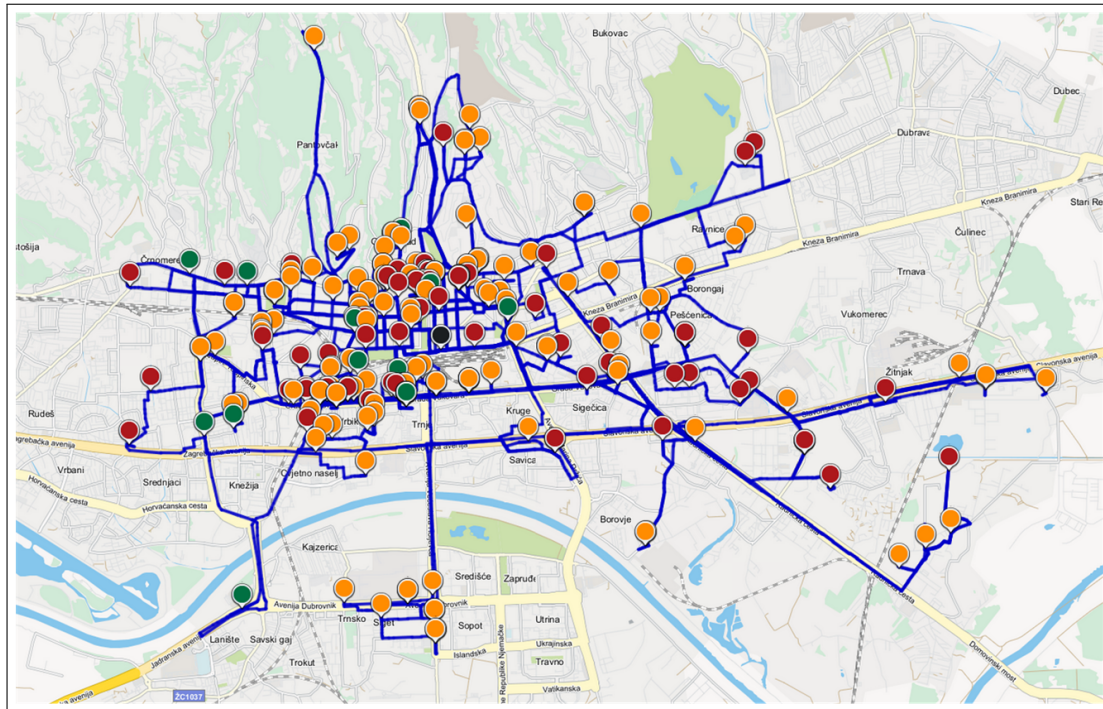


Figure 6.9: TD-VRPTW solution with link speed profiles - total time

6.5 EVRPTW

As the original delivery problem is not performed by BEVs, the problem is adapted to handle BEV routing constraints. This is manifested with the replacement of a purely ICEV fleet with a purely BEV fleet, as well as the determination of available CS in the road network of Zagreb [88].

6.5.1 Adaptation

The selected delivery problem is adapted to include EVs and CSs. First, the CSs in the urban area of Zagreb are extracted from [Google map](#), [Puni.hr](#) and [Charge map](#). Beside the extracted CSs, one additional CS is added at the depot location resulting in a total of 21 CSs. The list of all used CSs is presented in Table 6.4. In Figure 6.10 geographic distribution of CSs (blue pins) is presented.

Another important part of EVRP problem, is energy consumption related to BEVs. In the available literature regarding the application of BEVs in routing applications, energy consumption is often estimated using Longitudinal Dynamics Model (LDM). In this thesis, the LDM presented by Asamer et al. [205] is used. Force F needed to accelerate and to overcome resistances (grade, rolling and air) is given by equation 6.2, where m is the vehicle mass (mostly empty vehicle), a acceleration, v vehicle speed, g gravitational constant, f the inertia force of vehicle rotating parts (up to 5% of the total vehicle mass), α road slope, c_r rolling friction coef-

Table 6.4: CSs in the City of Zagreb

<i>ID</i>	<i>X</i>	<i>Y</i>	Name
1	15.981179	45.805281	Croatian post in Branimir street
2	15.96977005	45.81631713	Tuškanac parking garage
3	15.98241505	45.81448747	Ribnjak parking garage
4	15.99697897	45.81433979	Eugen Kvaternik square
5	15.99758927	45.81314805	Parking garage in Martićeva street
6	16.01115057	45.81537343	Road traffic school
7	15.9801913	45.80817855	Petrinjska parking garage
8	15.9890445	45.80367234	Strojarska street
9	15.97784908	45.80065808	City government 1
10	15.97780352	45.80138075	City government 2
11	16.00090236	45.80443909	Zaharova street
12	16.00108945	45.80283973	Green gold
13	16.00748604	45.80318021	Vukovarska street - Konzum
14	16.01693561	45.79540224	Capraška street
15	16.05000131	45.80194785	City Centar One East
16	15.9489664	45.79782361	Park Stara Trešnjevka
17	15.93297902	45.76928532	Blato - Kaufland
18	15.94357295	45.76712674	Station for technical inspection - Remetinec
19	15.96878552	45.77794117	Zagreb fair
20	15.9829782	45.76060619	Lidl in Oreškovićeve street
21	16.00381769	45.77051112	Travno - Konzum

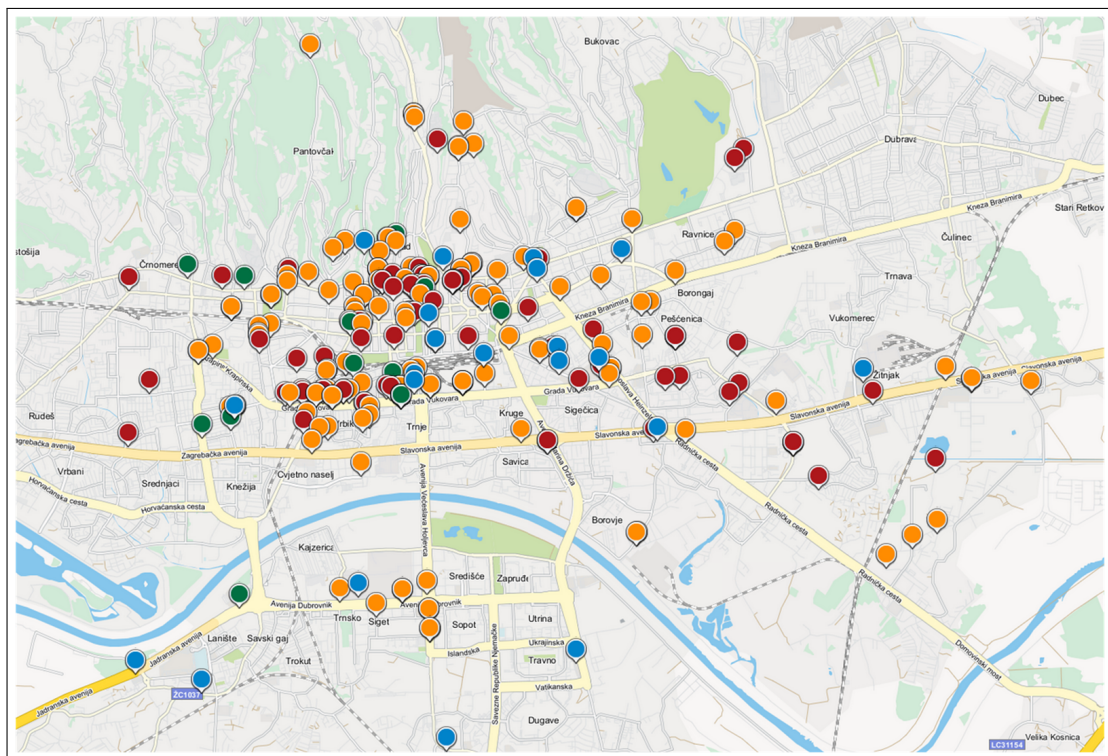


Figure 6.10: Post delivery problem with CSs

efficient, c_d air drag coefficient, ρ air density, and A vehicle frontal air surface. If condition $F \geq 0$ is satisfied, the vehicle is accelerating and power is needed for the movement of BEV (motor mode); otherwise, if condition $F < 0$ is satisfied, deceleration (braking) or driving downhill is occurring, and energy is returned into the BEV's battery as the electric engine has the ability to return the energy (generator mode). By process of regeneration, up to 15% of totally consumed energy can be returned [143, 186]. Electric power that comes from the battery is divided into the auxiliary power P_0 and mechanical power $P_m = Fv$. Auxiliary power is spent on the electronic devices in the vehicle: heating, ventilation, light, etc., which can shorten the BEV's range up to 30% [62]. Battery power P_b can be computed by equation 6.3, where μ_m is the transmission coefficient between the electric motor and drive-train, and μ_g is the conversion ratio from mechanical energy on wheels to chemical energy stored in the battery. Energy is returned into the battery only if the force F is lower than zero and speed is higher than the experimentally determined value v_{min} [205]. Instant energy consumption E can be computed as power multiplied by the elapsed time Δt , given by equation 6.4.

$$F = \underbrace{mg \sin \alpha}_{\text{Grade}} + \underbrace{c_r mg \cos \alpha}_{\text{Rolling}} + \underbrace{0.5c_d \rho A v^2}_{\text{Air}} + \underbrace{fma}_{\text{Acc.}} \quad (6.2)$$

$$P_b = \begin{cases} \frac{Fv}{\mu_m} + P_0, & \text{if } F \geq 0 \\ 0, & \text{if } v \leq v_{min} \\ \mu_g Fv + P_0, & \text{else} \end{cases}, \quad \text{if } F < 0 \quad (6.3)$$

$$E = P_b \Delta t \quad (6.4)$$

In Asamer et al. [205], Figure 6.11 is presented, which shows the relation between the average trip speed on x -axis and energy consumption in kWh per 100 km on y -axis. Additionally, each part of the LDM model is represented with different colors: rolling (gray), grade (red), air drag (green), acceleration (blue), and auxiliary (light blue). The energy consumption of BEVs corresponding to the grade resistance force is responsible for 5% to 10% of the total energy consumption on all vehicle speeds. This value can vary significantly depending on the terrain configuration and routing scenario. The acceleration force is the main cause of energy consumption at lower speeds, up to 40 km/h, where also the auxiliary power consumption has an important role. At speeds higher than 40 km/h, the air drag force is the dominant cause of energy consumption. Energy consumption corresponding to the rolling friction remains the same for all speed values.

To compute the energy consumption on a real road network, first, the BEV type used for the observed delivery problem needs to be selected. The [Nissan Leaf 2014](#) was selected for the delivery problem with specifications listed in Table 6.5. The Nissan Leaf was selected as it

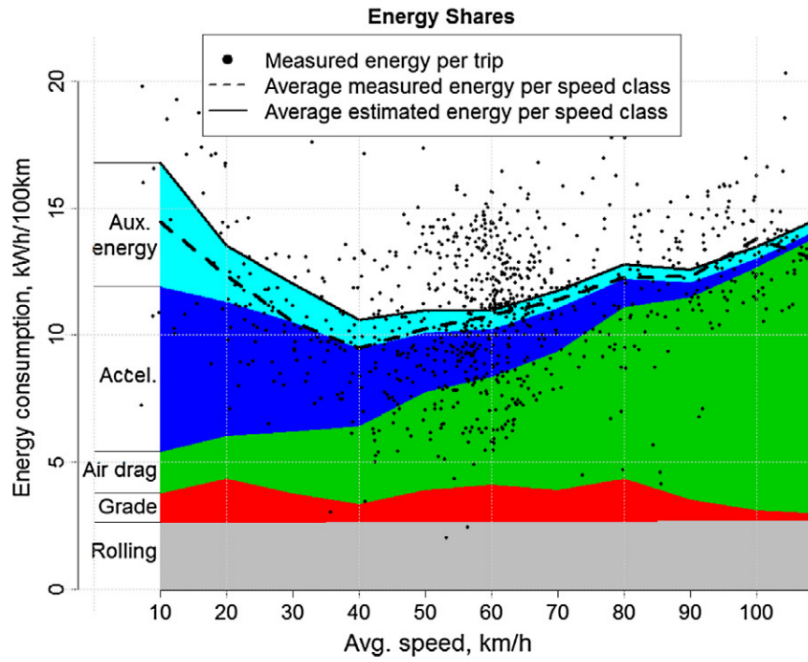


Figure 6.11: Average energy shares in dependence of average trip speed, [205]

was analyzed in several related researches that consider routing applications [13, 89, 98, 205, 206, 207, 208]. The same load capacity is used as for the conventional vehicles. Due to the specific delivery problem in which delivery is performed in a close urban environment, the total traveled distance of best solutions in different problem types is in the [220, 250] km range. As the used BEV has a range of 200 km [206], the whole delivery without load constraints could be made with one vehicle and one charging at CS. Due to the load capacity, the total number of used vehicles is 12, and as a result, in EVRPTW, no recharging at CS would be needed if battery capacity is set to 24 kWh. To validate the HALNS on the real-world example, the battery capacity of a BEV is reduced to 3 kWh, representing a range of roughly 25 km. This way, BEVs need to visit CSs during the routing. In basic EVRPTW variants, all CSs are considered to have an identical charging power of 16 kW as a balance between the rapid and slow recharging rate. For variants with different charger types, this value was considered as the fast charger type (middle value), while the rapid and slow charging powers are set to 50 kW and 11 kW, respectively. The considered charging cost coefficients are: (i) rapid - 0.192 €/kWh, (ii) fast - 0.176 €/kWh, and (iii) slow - 0.160 €/kWh, reported by Felipe et al. [23].

In equation 6.2, the air density is set to 1.2 kg/m^3 , while the gravitational acceleration g is set to 9.81 m/s^2 . The values of v , a , and α are not related to the vehicle specifications but rather to the characteristics of traversed road segment (link). For the computation of speed and acceleration values on links, the derived speed profiles from FCD are used. The example of the computed speed and acceleration profile for the road link in Figure 6.5a for Monday in the rest of the year season is presented in Figure 6.12a. As it can be seen, acceleration has low values due to the smoothed changes in the speed profile and does not reflect the real acceleration of

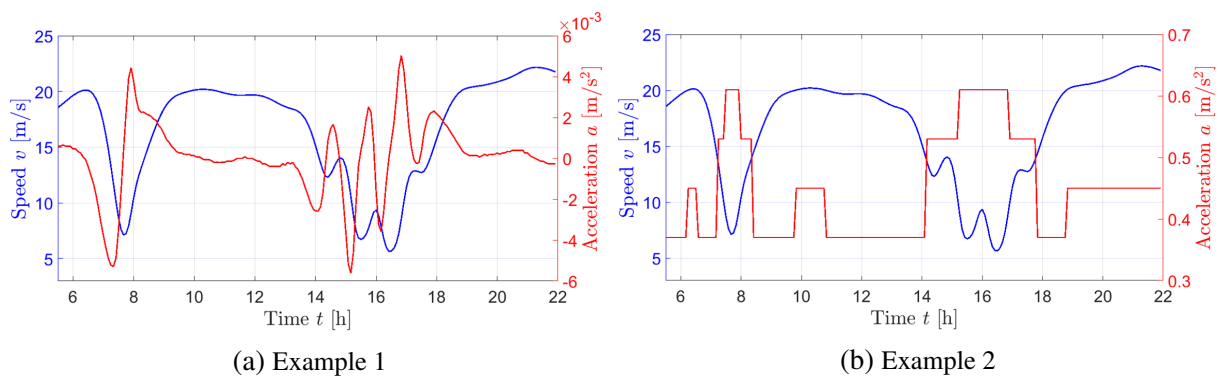
Table 6.5: Specifications of Nissan Leaf 2014

Attribute	Label	Value
Battery capacity	Q	24 kWh
Mass	m	1145 kg
Rolling friction coeff.	c_r	0.008
Air drag coeff.	c_d	0.35
Air frontal surface	A	1.9 m ²
Inertia force of rotating parts	f	1.01
Motor to drive-train coeff.	μ_m	0.9
Drive-train to motor coeff.	μ_g	0.8
Auxiliary power	P_0	450 W
Minimum speed for generator mode	v_{min}	2.78 m/s (10 km/h)

the vehicle. The speed from FCD could not be used for the acceleration computation as the recorded speed is the "spot" speed, and the sampling rate was quite large, on average, 100 m [28]. Instead, an average acceleration for Nissan Leaf vehicle per speed category is used [205], presented in Table 6.6 and Figure 6.12b for the observed speed profile example. Additionally, the acceleration on the observed link is limited to the maximum possible acceleration, computed with the assumption that the vehicle has the speed value of zero (0) at the beginning of the link, and average $v(t_k)$ speed at the end of the link, in time bucket t_k .

Table 6.6: Accelerations per speed intervals for Nissan Leaf 2014, [205]

Speed interval [km/h]	Avg. acceleration [m/s ²]
[0, 30)	0.61
[30, 51)	0.53
[51, 72)	0.37
[72, 93)	0.41
[93, 102)	0.28
[102, ∞)	0.05

**Figure 6.12:** Speed and acceleration profiles

6. Adapted real-world delivery problem

For the computation of road link grade, the digital elevation data by [EU Copernicus service](#) is used. The raster data is represented with 25 m resolution and vertical accuracy of ± 7 m. The example of data for the urban area of Zagreb is presented in Figure 6.13, where the highest points are represented with red color, and the lowest points with blue color and decreased opacity value.

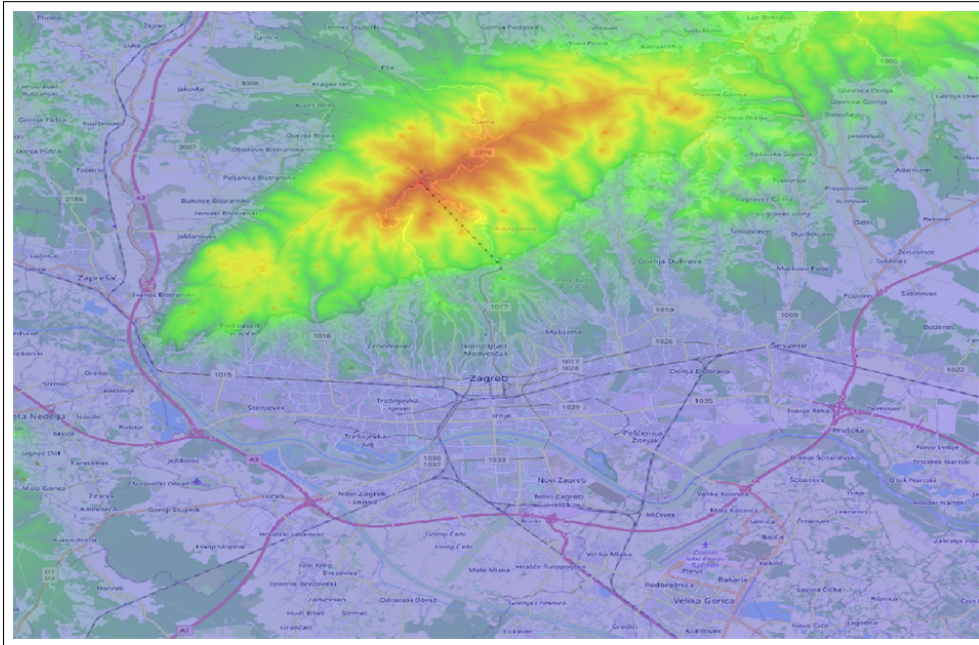
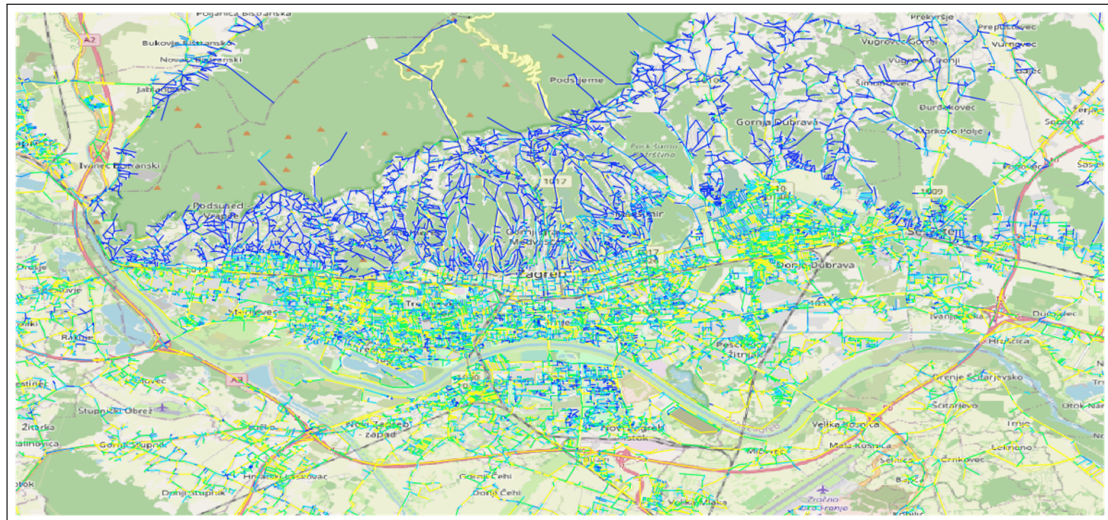


Figure 6.13: Copernicus elevation model

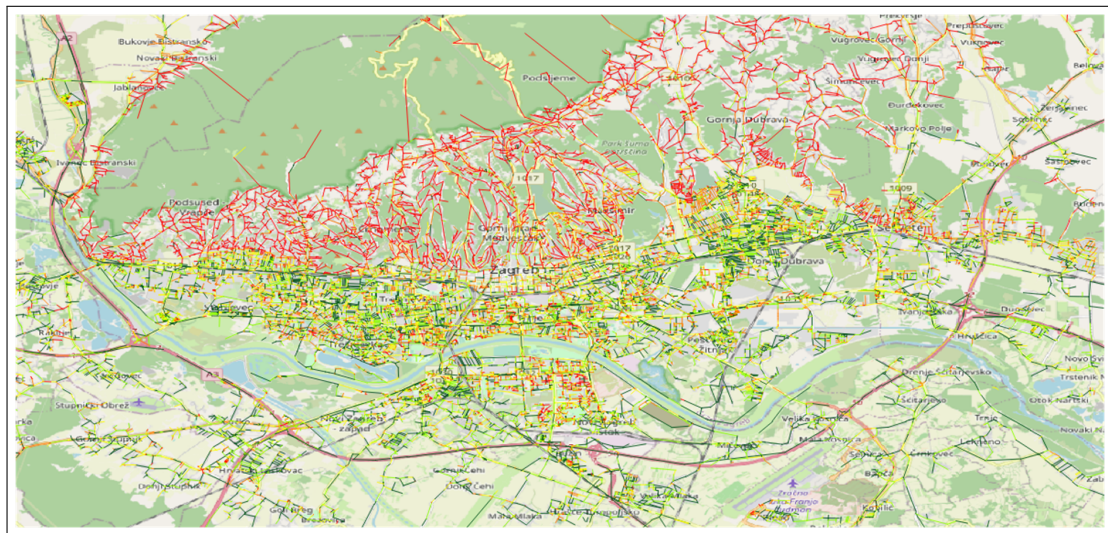
Each start and end points of links in the used road map of Zagreb are associated with the elevation value from the raster data. Based on the difference in the elevation Δh and aerial link distance d the grade of the link is computed by equation 6.5. The computed link grades are presented in Figure 6.14. The links with negative grades, representing downhill direction, are represented with color ranging from blue (the steepest) to yellow (the flattest) in Figure 6.14a, while the links with positive grades are represented with color ranging from red (the steepest) to green (the flattest) in Figure 6.14b.

$$\alpha = \arctan \frac{\Delta h}{d} \quad (6.5)$$

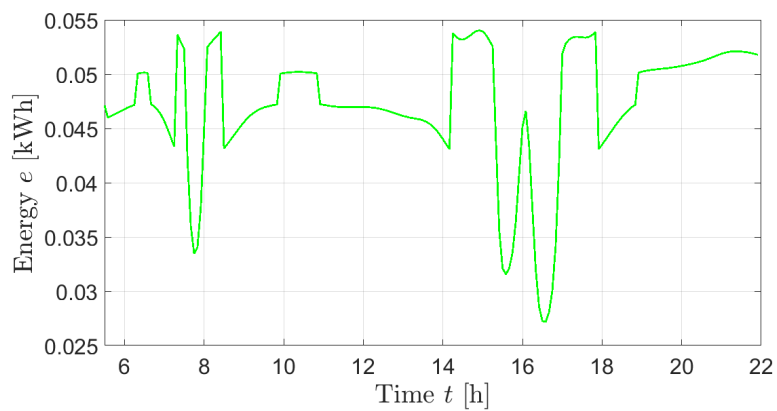
As a result of the previous computation, each link has an associated speed, acceleration, and energy consumption profile. The example of energy profile for already observed link in Figure 6.12, is presented in Figure 6.15.



(a) Negative grade



(b) Positive grade

Figure 6.14: Link grades for the road network in the City of Zagreb**Figure 6.15:** Example of energy consumption profile

6.5.2 Results

Results of applying the HALNS method on different EVRPTW variants are presented in Table 6.7. The columns for problem variant, best recharging cost (€) and configuration of CSs (m_1^b, m_2^b, m_3^b) in best solution are added, where m_1 represents the rapid charger type and m_3 the slowest charger type. The FR represents full recharge strategy, PR partial recharge strategy, DCS different charges types, and TD time-dependent variants. For comparison, the best VRPTW solution computed with the SORDITO average speed is selected. The non time-dependent variants are solved using average SORDITO speed values, while the time-dependent variants are solved using SORDITO speed profiles. In all variants, in best solutions, 12 vehicles are found. The FR strategy shows that a minimal number of vehicles is harder to achieve than in the VRPTW problem, as in some cases, the final solution contained 13 vehicles; therefore, the average vehicle number in FR strategy is 12.3 vehicles. Due to the visit to the CSs and additional recharging time, the total traveled distance, total travel time, and total time increased in all variants. In EVRPTW-FR, the total traveled distance increased by 15 km, the total travel time by 6.94 h and total time by 2.31 h. This means that on average, each vehicle spends roughly 30 min more on driving and ends its delivery 20 min later. The total recharging cost is 7.04 € recharged at CSs or used from the initial full battery capacity. The EVRPTW-PR again showed its benefits by decreasing all values, especially the total traveled distance by 5.6 km, as well as recharging costs by 0.22 €. Interestingly the number of visits to CSs decreased, which was not expected as reported in subsection 4.10.3. This occurred as HALNS was able to find a better customer and CS configuration for this particular problem. The example of the best solutions for EVRPTW-FR and EVRPTW-PR are presented in Figures 6.16 and 6.17. The problems with different charger types used distance minimization matrices with appropriate energy cost values. The EVRPTWDCS-FR and EVRPTWDCS-PR were able to decrease overall recharging costs at the expense of an increase in total traveled distance and total time. Lastly, the results for time-dependent variants are presented. In TD-EVRPTW-FR problem, the total travel time decreased to 11.91 h as well as recharging costs, although more CSs are visited. In TD-EVRPTWDCS-FR, the recharging cost is further decreased, but it is interesting that the total time is the lowest as the use of rapid charger enables easier completion of some customers.

Table 6.7: TD-/EVRPTW-DCS/PR/FR real example - results

Prob.	Min.	VRPTW		HALNS													
		K	d	\bar{K}	K_{best}	ΔK	\bar{f}	d_{best}	Δd	$\Delta_p d$	tt_{best}	tot_{best}	rec_{best}	m_1^b	m_2^b	m_3^b	\bar{t}_e
FR	d	12	232.76	12.3	12	0	249.87	246.97	13.99	6.01	14.26	40.65	7.64	—	8	—	11.85
PR	d	12	232.76	12	12	0	254.89	245.06	12.3	5.28	13.95	40.63	7.42	—	7	—	9.23
DCS-FR	rec	12	232.76	12	12	0	6.96	257.27	24.51	10.53	14.93	41.35	6.96	0	1	9	12.98
DCS-PR	rec	12	232.76	12	12	0	7.04	257.87	25.11	10.79	14.95	42.39	7.04	2	0	11	13.32
TD-FR	tt	12	232.76	12	12	0	11.91	249.38	16.62	7.14	11.79	40.76	7.09	—	12	—	15.36
TD-DCS-FR	rec	12	232.76	12	12	0	6.40	253.53	20.77	8.92	13.00	40.37	6.40	2	0	10	18.80

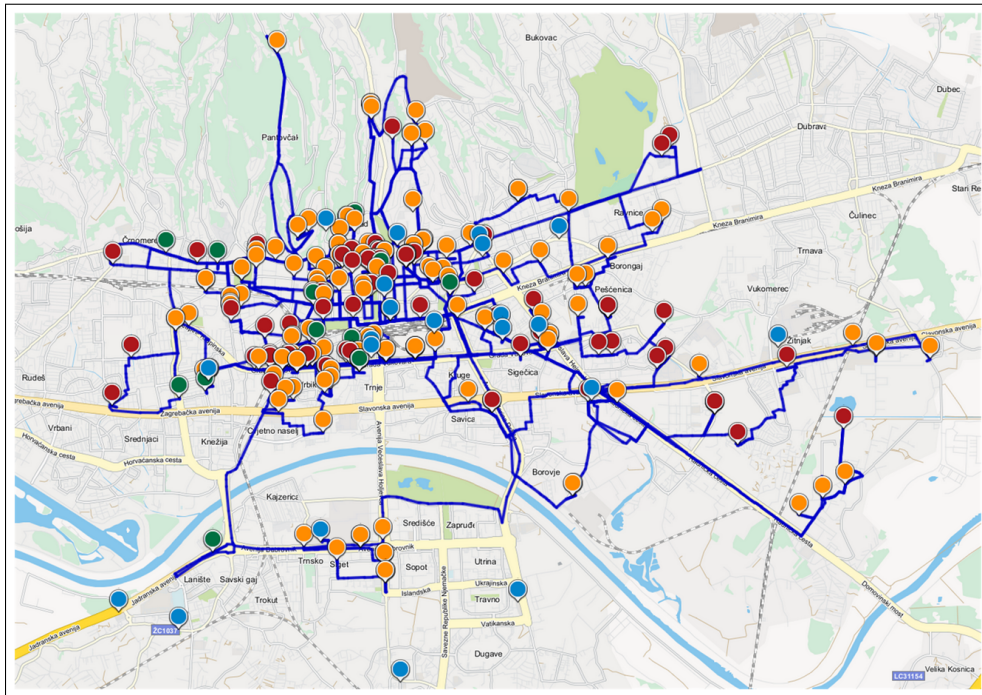


Figure 6.16: EVRPTW-FR solution with link speeds - total distance traveled

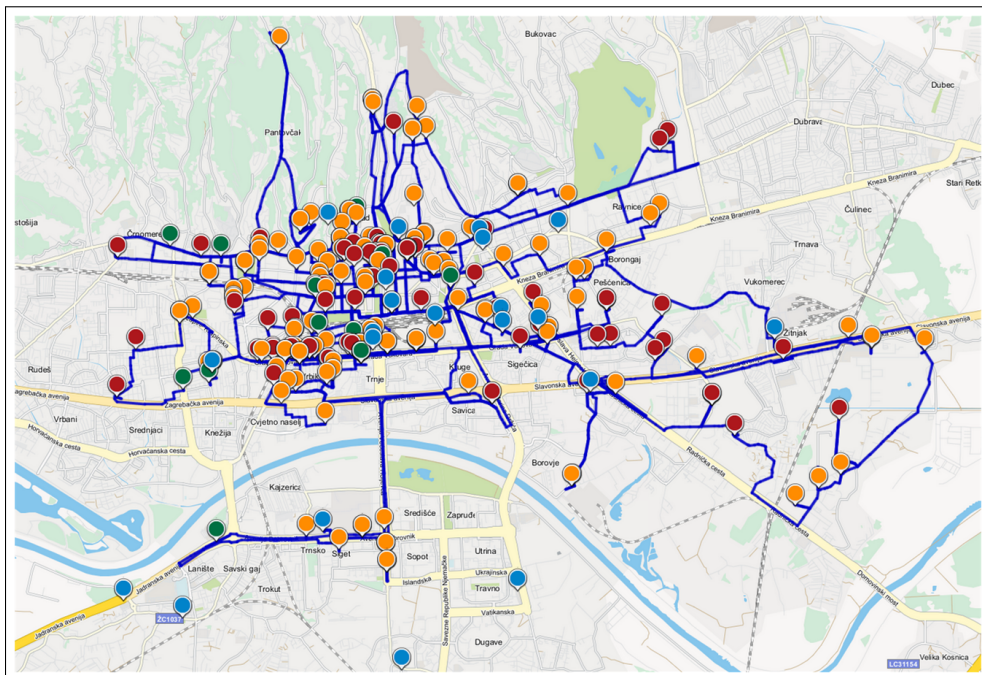


Figure 6.17: EVRPTW-PR solution with link speeds - total distance traveled

6.5.3 Minimization of energy consumption

None of the EVRP variants considered the minimization of the total energy consumed. The closest objective is the recharging cost minimization with different charger types in CSs. The main reason is that theoretically developed EVRP variants consider energy consumed on the arc as a linear function of the total traveled distance. Therefore, the minimization of the total traveled distance corresponds to the minimization of total energy consumed. In this thesis, to approximate the real energy consumption of the vehicle on the link level, the LDM model was used, and as a result, the total energy consumption does not depend linearly on the total distance traveled. Therefore, there is a need to determine the solution to the Energy Shortest Path Problem (ESPP) on a graph that can contain negative weights as energy is returned into the battery when the EV works in generator mode. Here the problem rises, as the Dijkstra's algorithm, cannot be applied on graphs with negative weights, more precisely on graphs where negative cycles can occur. The negative cycle means that the sum of cycle weights is lower than zero, which could happen in practice in directed graphs due to the discretization, double rounding errors, etc. The best-known algorithm for finding the shortest path in a directed graph with negative edge weights is the Bellman-Ford algorithm [209], given by Algorithm 6.3, which has $\mathcal{O}(|V| \cdot |E|)$ complexity. The only difference to Dijkstra algorithm is that it does not necessarily select the vertex with the lowest value to be removed from the list of the unprocessed vertices Q and that it searches through all neighboring vertices of u , and not only the unprocessed ones.

To further improve the efficiency of computing the shortest path on a negative graph cycle Johnson's technique can be used [210]. First, a new vertex q is added to the graph, connected by zero-weight edges to each other vertex. Then, the Bellman-Ford algorithm is run on such a graph, starting from vertex q . The result is a minimum weight path from vertex q to each other vertex in a graph. Next, the weights of the graph are recomputed based on the minimum path values between vertices u and v and previous weight value $w(u, v)$ given by equation 6.6, where BF_value represents the Bellman-Ford minimum path value. In the end, the vertex q and its edges are removed from the graph, and Dijkstra algorithm can be applied to find the shortest path from each starting vertex to each other vertex. The shortest path value produced by Dijkstra, has to be returned to the original range by reducing each edge weight in a path by the added value $BF_value(u) - BF_value(v)$. Although, Bellman-Ford algorithm shows larger complexity, this way it has to be run only once in the preprocessing step to prepare the graph values for Dijkstra algorithm.

$$w'_{u,v} = w_{u,v} + BF_value(u) - BF_value(v) \quad (6.6)$$

The described algorithm was used to determine the energy shortest path in a road network of Zagreb, with a similar running time as for the shortest distance and travel time paths. The only

Algorithm 6.3 Bellman-Ford algorithm

Input: Graph G and start vertex v_0

- 1: $Q \leftarrow$ Initialize list of unprocessed vertices
- 2: **for each** vertex v in graph G **do**
- 3: **if** $v = v_0$ **then**
- 4: $value(v) \leftarrow 0$
- 5: **else**
- 6: $value(v) \leftarrow \infty$
- 7: **end if**
- 8: $preceding(v) \leftarrow None$
- 9: Add v to Q
- 10: **end for**
- 11: **while** Q is not empty **do**
- 12: $u \leftarrow$ Remove vertex from Q
- 13: **for each** neighboring vertex v of vertex u **do**
- 14: $val_{new} \leftarrow value(u) + w_{u,v}$
- 15: **if** $val_{new} < value(v)$ **then**
- 16: $value(v) \leftarrow val_{new}$
- 17: $preceding(v) \leftarrow u$
- 18: **end if**
- 19: **end for**
- 20: **end while**

difference is the computation of Bellman-Ford values in advance, which took approximately 7.19 s. The result for the EVRPTW-FR and EVRPTW-PR problems are presented in Table 6.8. It can be seen that the energy shortest paths reduce the energy consumption as well as the recharging cost at the expense of an increase in total traveled distance and total time. The example of the solution for EVRPTW-FR is presented in Figure 6.18.

Table 6.8: EVRPTW-PR/FR with energy consumption minimization

Prob.	\bar{K}	K_{best}	\bar{e}	e_{best}	d_{best}	tt_{best}	tot_{best}	rec_{best}	m	\bar{t}_e
FR	12	12	40.05	39.12	256.88	15.39	40.76	5.60	7	9.23
PR	12	12	38.34	37.85	251.64	15.5	40.6	6.66	10	7.42

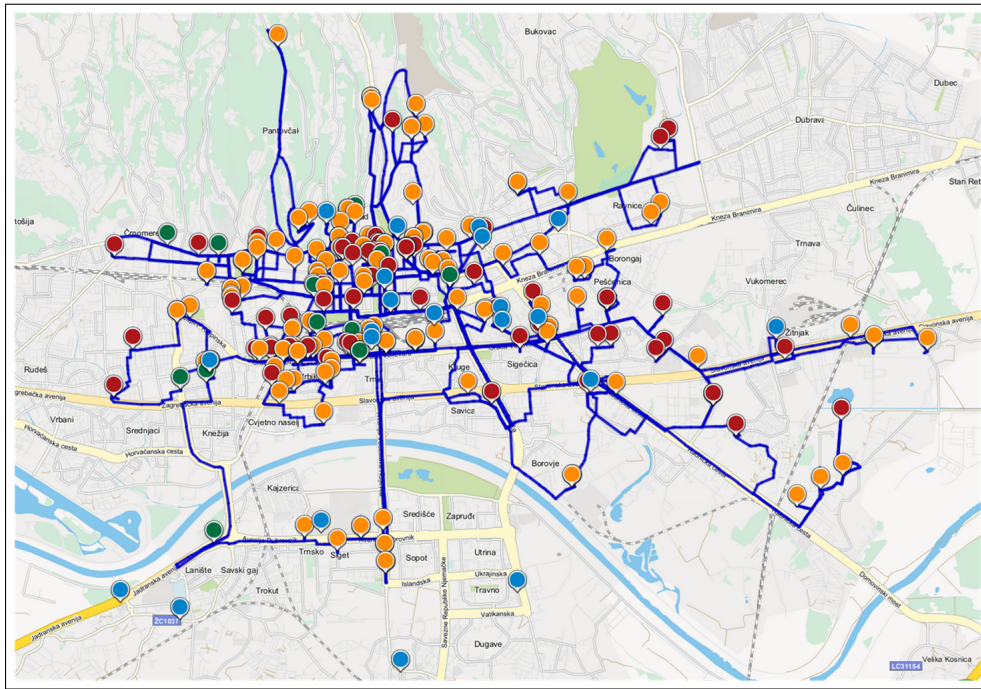


Figure 6.18: EVRPTW-FR solution with link speeds - energy consumption

6.6 Note on the computation of time-dependent matrices

Describing the traffic conditions with as little data as possible is important because it lowers memory requirements. The researchers that deal with benchmark time-dependent problems usually consider less than ten speed or travel time indexes per planning horizon [25, 176, 178, 179, 180, 211]. Others that tried to incorporate real-time traffic data into routing problems aggregated speed data into discretized time buckets (up to 15 min) and observed relatively small-sized road networks (up to 70000 links) [27, 204, 212, 213, 214].

In this thesis, a TDD algorithm was used to solve the TD-SPP with an average running time of 2.97 s. The time to compute one line of matrix is roughly the same on small problems, as the Dijkstra algorithm does not require the re-computation of graph values if the start vertex has not changed. For problems with a larger number of users, the computation of one line can last some additional time, depending on the problem, but here that time was not considered. As reported, the computation of matrices for 226 customers in 199 time buckets took approximately 34.85 h. Such time is too long in real applications, as sometimes the configuration of users changes frequently, which then requires matrices recomputation. One could say that used exact Dijkstra algorithm is slow and that much faster heuristic implementations can be applied. To prove that this computation time is still very long, the TDD algorithm developed as a part of the SORDITO project is used. The used TDD has an average running time of 0.724 s, determined based on the shortest path solution to 100000 random origin-destination pairs on the whole road network in Croatia, with on average 111 km distance between the origin and destination. Table 6.9

presents the pre-processing computation time in hours for a road network of Croatia by using a TDD algorithm that depends on the speed profiles with different discretization time periods ranging from 1 to 60 min. It can be seen that with the increase in the number of users, the computation time increases significantly.

Table 6.9: Pre-processing computation time [h] in time-dependent routing operations

Approach	Number of users		
	100	1000	10000
SP-60	0.48	4.83	48.26
SP-15	1.93	19.31	193.06
SP-5	5.79	57.92	579.2
SP-1	28.96	289.6	2896
TTI-7	0.14	1.41	14.08

To reduce the computation time, the congestion zones and time-varying Travel Time Indexes (TTIs) can be used, which are developed as a part of the SORDITO project, and here they will be briefly mentioned. For detailed description, the reader is referred to Erdelić et al. [28]. The congestion zones can be estimated based on the speed profiles for links in the digital map by fishnet divide of the observed area and the application of image processing methods. The example of extracted zones for whole Croatia is presented in Figure 6.19a, while the specific zone for the urban area of Zagreb is presented in Figure 6.19b. The fishnet cells are colored from red to green, representing congestion intensity. The congestion zone's borders are represented with blue line and the real city border with magenta line. Next, in each congestion zone, a Monte-Carlo approach was used to simulate 100000 routes, record their travel times and express

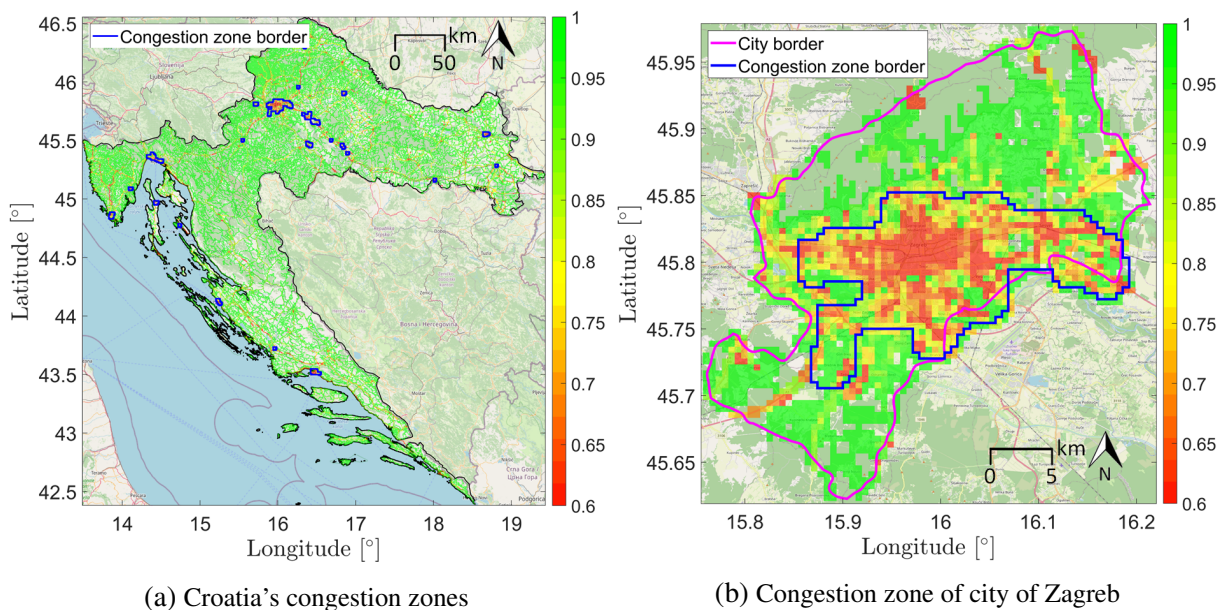


Figure 6.19: Congestion zones

it in relation to the free-flow route travel time. As a result, in several congestion zones, the temporal clustering could be performed to determine distinct time periods in day in which traffic conditions are similar, presented in Figure 6.20. Seven distinct time periods divided by black lines can be depicted. For each determined time period, the mean value is given in Table 6.10. As it can be seen, in the rush hours, travel time increases by almost 32%, compared to the free flow travel time. The developed TTIs were validated in the real driven test study, in which the TTIs produced a relative percentage error of 4.13%.

As developed TTIs use only seven mean values to represent daily traffic conditions the overall matrices computation time can be significantly reduced, as shown in the latest row in the Table 6.9. The computation time for 1000 users is 1.41 h. The further application of TTIs in time-dependent routing applications is presented in [26, 158].

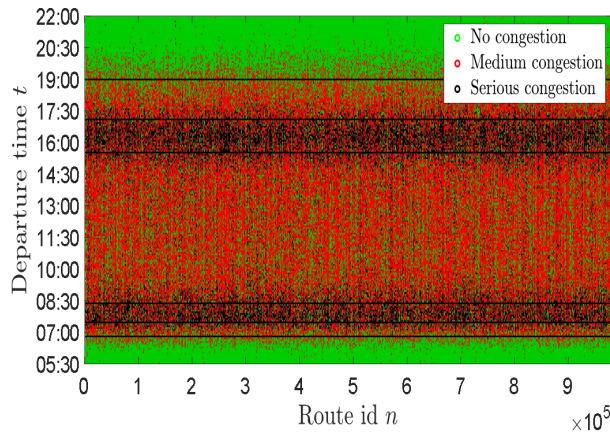


Figure 6.20: Clustered TTIs

Table 6.10: Mean TTI values

Period	TTI[%]
05:30-06:45	-0.09
06:45-07:25	17.90
07:25-08:20	31.66
08:20-15:30	18.08
15:30-17:05	31.58
17:05-19:00	16.24
19:00-22:00	1.42

Conclusion

The vehicle routing problem is a logistic optimization problem that considers the problem of everyday goods delivery from the depot to the customers. The inclusion of real-world constraints such as vehicle load capacity and customer time windows led to the definition of several basic variants of the problem. With the rise of ecological awareness and new regulations related to more sustainable and greener transport, especially in the EU, the research field of green logistics emerged. The important part of green logistics is the integration of electric vehicles in the distribution process. Electric vehicles are considered as a cleaner mode of transport than conventional vehicles with internal combustion engine due to the reduced greenhouse gas emission, minimal noise, refueling from renewable energy sources, and resistance to the fluctuating oil price and politically unstable countries. The integration of electric vehicles in logistics has led to the definition of a new variant of the vehicle routing problem that includes electric vehicles - the electric vehicle routing problem. As the vehicle routing problem is an NP-hard problem, meaning that a solution cannot be obtained in the polynomial time, a large number of heuristic and metaheuristic procedures were developed over the years to solve the problem.

The electric vehicle routing problem has to account for the limited battery capacity of electric vehicles, which severely shortens the driving range of electric vehicles compared to the conventional ones. This manifests in the need for the recharging at charging stations. Although the electric variant of the problem has been introduced only recently, there are already several variants of the problem that are mainly focused on the different recharging strategies or the fleet composition regarding the mix of conventional and electric vehicles. Most of the proposed solution procedures combine specially defined heuristic and metaheuristic procedures for each of the problem variants. Therefore, there is an open gap for the development of a more general metaheuristic that will efficiently solve different problem variants. The integration of exact procedures for optimal charging station placement, which improves the solution quality, has only been addressed in a few studies. The heuristic based on the ruin-recreate principle cou-

pled with exact procedures for optimal charging station placement can be generalized to apply it on several problem variants. The only method components that need to be changed between the problem variants are the applied penalty functions and the so-called concatenation operators that evaluate the changes in the solution. With slight modifications, the developed HALNS method was able to efficiently solve such problems and achieve several unpublished best known results.

Additionally, the time-dependent traffic conditions in the electric vehicle routing problem have not been addressed so far. It is a well-known fact that the travel time changes depending on the departure time, especially in the urban traffic environment where the traffic congestion occurs regularly. As traffic has a recurrent nature, the travel time on the road network can be predicted and integrated into the electric vehicle routing problem. If neglected, this can lead to a significant increase in routing cost due to the penalties for late deliveries in time-precise distribution. With slight modifications, the developed HALNS method was able to efficiently solve such problems and achieve several unpublished best known results.

Lastly, to show the application in real life, a real-world delivery problem was adapted to the specific EVRPTW model characteristics and solved with the developed HALNS method. The time-dependent traffic conditions on the road network were expressed as speed profiles derived from the historical GNSS data. Routing of electric vehicles on the real road network additionally required charging station locations, road grades, electric vehicle specifications, and energy consumption model. The results show the application of the developed method to solve real-world delivery problem modeled as several problem variants: VRPTW, TD-VRPTW, EVRPTW, and TD-EVRPTW.

7.1 Achieved contributions and main conclusions

The first objective of the thesis was to develop a hybrid metaheuristic procedure to efficiently solve different variants of the electric vehicle routing problem. This objective of the thesis was achieved by developing a method based on the ruin-recreate principle and coupled with the exact procedure for optimal charging station placement. Additionally, problem-specific variables, penalty functions, and concatenation operators were defined. It should be highlighted that although the proposed procedure has been tested out on only several problem variants, the procedure can be applied on other problem variants with only a change in problem-specific penalty functions and concatenation operators.

The second objective of the thesis was the formulation of a new time-dependent electric vehicle routing problem with time windows. This objective is achieved by mixed integer formulation of the basic time-dependent problem and several problem variants. The closely related third objective was to apply a developed hybrid metaheuristic to solve the newly defined prob-

lem. This was achieved by solving newly-defined test instances with the proposed solution method. The rest of the section lists major contributions achieved in this thesis, and gives a conclusion about each of the achieved contributions. The whole list of all thesis major and minor contributions is given in the Appendix section [A](#).

7.1.1 Hybrid ALNS method development for solving different existing variants of the electric vehicle routing problem

The hybrid adaptive large neighborhood search method was developed for solving different electric vehicle routing problems. The main contributions are the development and the application of new variables for several observed variants: full recharge, different charger types with full recharge, and different charger types with partial recharge. The problem with different charger types and full recharge has not been previously addressed in the literature. The contributions manifest in the new best known solutions on several instances and problem variants, as well as the relatively low method execution time.

Based on the obtained results, it can be concluded that the developed method is able to efficiently solve different variants of the electric vehicle routing problem, with the modification of only penalty functions and concatenation operators. Additionally, the proposed method shows stability when minimizing different objectives: total distance traveled, total time, and total recharging costs.

7.1.2 Development of the mixed integer program based model for time-dependent electric vehicle routing problem with time windows

As the time-dependent electric vehicle routing problem with time windows has not been previously addressed in the literature, the common approach when introducing a new vehicle routing problem is to formulate it as mixed integer program, or if possible, as a linear variant. The problem considers the travel time that depends on the departure time, which function in most occasions is nonlinear. Therefore, first, the mixed integer formulation for the basic problem variant with full recharge strategy is presented, as the extension of the mixed integer linear model for the electric vehicle routing problem with time windows and full recharge. Next, the linearized model, which considers the linearization of travel times in several time buckets, is presented. Lastly, the mixed integer formulations for the partial recharge strategy and different charger types with full recharge are presented as the extension of their non-time-dependent linear versions.

7.1.3 Solving the developed model of time-dependent electric vehicle routing problem with time windows using adapted hybrid ALNS method

The developed hybrid adaptive large neighborhood search method was applied to solve the newly defined time-dependent electric vehicle routing problem with time windows and full recharge. The piece-wise linearization of travel times was considered, which satisfies the FIFO principle and ensures algorithm stability. The observed delivery period was discretized into five time periods that represent typical daily urban traffic conditions. The problem was efficiently solved on a newly-defined problem instances, showing the impact of time-dependent travel time on routing decisions.

7.2 Future research

This thesis has addressed several open issues in the field of electric vehicle routing problem. Regardless, there are still many open research areas that need further consideration. This section will outline possible future research direction.

The first possible future research direction is the application of the developed method on other electric vehicle routing problem variants. What naturally comes to mind is the heterogeneous routing problem which considers a fleet of both electric and conventional vehicles or the problem, which considers non-linear charging function. The first one is the most logical extension, as most of the companies are gradually integrating electric vehicles in the existing fleet, while the second one takes into account the non-linear charging time after approximately 80% of the SOC value, which can seriously influence the routing decisions. Also, an extension of the problem that deals with capacitated charging stations, possible waiting times, and public or private CS property has received a little attention in the literature. The built charging stations usually have two to three chargers, which limits the number of vehicles charging simultaneously. Additionally, the highest demand for recharging occurs in similar time periods in a day, when drivers take a break. On that note, the charging reservation, as well as the compatibility of chargers, have received a little interest in the research community. To apply the developed method on such problems, new penalty functions and concatenation operators need to be developed as well as possibly new destroy, repair, and local search operators.

Second, the developed method itself can be further improved. The penalty functions and concatenation evaluation showed the best evaluation performance in $\mathcal{O}(1)$ for partial recharge strategy, while for the full recharge strategy, in some cases, the evaluation until the latest charging station in route in $\mathcal{O}(n)$, still has to be performed. Detecting all cases in which evaluation is not adequately determined could lead to new penalty functions and concatenation operators for full recharge with $\mathcal{O}(1)$ complexity. This also goes for the partial recharge strategy with

different charger types where the evaluation is not efficient if the charger types are not compatible. In the basic time-dependent vehicle routing problem, there are only a few cases in which evaluation is not adequate, which could be inspected and dealt with, in the future research. The time-dependent electric vehicle routing problem with partial recharge could not be solved as the exact procedure to evaluate the changes done in the solution has not been found yet, as charging time and travel time are dependent variables. Also the exact procedure for optimal charging station placement, which has been proven to increase the quality of the solution, has not yet been proposed for the problem variant. The most basic problems consider minimization of distance or travel times because they are the easiest to evaluate, but the literature lacks efficient evaluation procedures when total time is minimized.

Third, the exact procedure for optimal charging station placement can be viewed as the bottleneck of the proposed method. The procedure has proven to significantly improve the solution quality but has a long computation time compared to the other method components. To improve the computation time of the exact procedure, and generally all evaluations performed during the search process, the caching technique can be used, which stores evaluation values of a dozen of the latest evaluations performed during the search. For the application on large instances, the parallelization of the method should be considered to speed up the execution time.

Forth, in most of the problem variants, the energy consumption model that linearly depends on the total distance traveled is considered. This helps to strip some of the problem constraints and focus on the algorithm itself. However, for the application in the real world, a more accurate energy consumption model should be considered. In this thesis, for real-world energy consumption, the longitudinal dynamics model was used. The used model is better than the previously used linear model, but still, much better models for energy consumption exists in the literature which include machine learning procedures and simulation models.

Fifth, the application of population metaheuristics such as genetic algorithm, ant colony, or swarm particle optimization on the electric vehicle routing problem is scarce. Such algorithms could further provide more generic procedures for solving the observed problem variants.

Sixth, as the vehicle routing problem is a well-researched problem, the researchers established a benchmark website where the best known solutions are stored. This helps a research community to faster test their developed algorithms and to compare them to the other applied algorithms. For the electric vehicle routing problem, such center of benchmark data is not available, and future efforts could be made in that direction.

Bibliography

- [1] Keskin, M., Çatay, B., “Partial recharge strategies for the electric vehicle routing problem with time windows”, *Transportation Research Part C: Emerging Technologies*, Vol. 65, 2016, pp. 111 - 127, available at: <http://www.sciencedirect.com/science/article/pii/S0968090X16000322>
- [2] Laporte, G., “Fifty years of vehicle routing”, *Transportation Science*, Vol. 43, No. 4, 2009, pp. 408-416, available at: <https://doi.org/10.1287/trsc.1090.0301>
- [3] Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., “Heuristics for multi-attribute vehicle routing problems: A survey and synthesis”, *European Journal of Operational Research*, Vol. 231, No. 1, 2013, pp. 1 - 21, available at: <http://www.sciencedirect.com/science/article/pii/S0377221713002026>
- [4] Toth, P., Vigo, D., (ed.), *The Vehicle Routing Problem*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.
- [5] European Commission. (2009) Effort sharing: Member states’ emission targets. Accessed: 7. February 2021, available at: https://ec.europa.eu/clima/policies/effort_en
- [6] WEEA. (2015) Greenhouse gas data - emissions share by sector in EU28. Accessed: 31. August 2020, available at: <http://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases-viewer>
- [7] Sbihi, A., Eglese, R. W., “Combinatorial optimization and green logistics”, *4OR*, Vol. 5, No. 2, Jul 2007, pp. 99–116, available at: <https://doi.org/10.1007/s10288-007-0047-3>
- [8] Davis, B. A., Figliozzi, M. A., “A methodology to evaluate the competitiveness of electric delivery trucks”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 49, No. 1, 2013, pp. 8 - 23, available at: <http://www.sciencedirect.com/science/article/pii/S1366554512000658>
- [9] Schiffer, M., Stütz, S., Walther, G., “Are ECVs breaking even? : Competitiveness of electric commercial vehicles in retail logistics”, RWTH Aachen University, Aachen,

- Tech. Rep. G-2017-47, 2017, available at: <https://publications.rwth-aachen.de/record/691766>
- [10] Feng, W., Figliozzi, M., “An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market”, *Transportation Research Part C: Emerging Technologies*, Vol. 26, 2013, pp. 135 - 145, available at: <http://www.sciencedirect.com/science/article/pii/S0968090X12000897>
- [11] Young, K., Wang, C., Wang, L. Y., Strunz, K., *Electric Vehicle Battery Technologies*. New York, NY: Springer New York, 2013, pp. 15–56, available at: https://doi.org/10.1007/978-1-4614-0134-6_2
- [12] Li, J.-Q., “Transit bus scheduling with limited energy”, *Transportation Science*, Vol. 48, No. 4, 2014, pp. 521-539, available at: <https://doi.org/10.1287/trsc.2013.0468>
- [13] Martínez-Lao, J., Montoya, F. G., Montoya, M. G., Manzano-Agugliaro, F., “Electric vehicles in Spain: An overview of charging systems”, *Renewable and Sustainable Energy Reviews*, Vol. 77, 2017, pp. 970 - 983, available at: <http://www.sciencedirect.com/science/article/pii/S1364032116310152>
- [14] Gendreau, M., Tarantilis, C. D., “Solving large-scale vehicle routing problems with time windows: The state-of-the-art”, 2010, technical report 2010-04, available at: <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2010-04.pdf>
- [15] Pecin, D., Pessoa, A., Poggi, M., Uchoa, E., “Improved branch-cut-and-price for capacitated vehicle routing”, *Mathematical Programming Computation*, Vol. 9, No. 1, Mar 2017, pp. 61–100, available at: <https://doi.org/10.1007/s12532-016-0108-8>
- [16] Baldacci, R., Mingozzi, A., Roberti, R., “Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints”, *European Journal of Operational Research*, Vol. 218, No. 1, 2012, pp. 1 - 6, available at: <http://www.sciencedirect.com/science/article/pii/S0377221711006692>
- [17] Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., “A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows”, *Computers & Operations Research*, Vol. 40, No. 1, 2013, pp. 475 - 489, available at: <http://www.sciencedirect.com/science/article/pii/S0305054812001645>
- [18] Schneider, M., Stenger, A., Goeke, D., “The electric vehicle-routing problem with time windows and recharging stations”, *Transportation Science*, Vol. 48, No. 4, 2014, pp. 500-520, available at: <https://doi.org/10.1287/trsc.2013.0490>

- [19] Schiffer, M., Schneider, M., Walther, G., Laporte, G., “Vehicle routing and location routing with intermediate stops: A review”, *Transportation Science*, Vol. 53, No. 2, 2019, pp. 319-343, available at: <https://doi.org/10.1287/trsc.2018.0836>
- [20] Goeke, D., Schneider, M., “Routing a mixed fleet of electric and conventional vehicles”, *European Journal of Operational Research*, Vol. 245, No. 1, 2015, pp. 81 - 99, available at: <http://www.sciencedirect.com/science/article/pii/S0377221715000697>
- [21] Erdelić, T., Carić, T., “A survey on the electric vehicle routing problem: Variants and solution approaches”, *Journal of Advanced Transportation*, Vol. 2019, May 2019, pp. 1–48, available at: <https://www.hindawi.com/journals/jat/2019/5075671/>
- [22] Keskin, M., Çatay, B., “A matheuristic method for the electric vehicle routing problem with time windows and fast chargers”, *Computers & Operations Research*, Vol. 100, 2018, pp. 172 - 188, available at: <http://www.sciencedirect.com/science/article/pii/S0305054818301849>
- [23] Felipe, Á., Ortuño, M. T., Righini, G., Tirado, G., “A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 71, 2014, pp. 111 - 128, available at: <http://www.sciencedirect.com/science/article/pii/S1366554514001574>
- [24] Erdelić, T., Vrbančić, S., Rožić, L., “A model of speed profiles for urban road networks using g-means clustering”, in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1081-1086.
- [25] Figliozzi, A. M., “The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 48, No. 3, 2012, pp. 616–636.
- [26] Carić, T., Fosin, J., “Using congestion zones for solving the time dependent vehicle routing problem”, *PROMET - Traffic&Transportation*, Vol. 32, No. 1, 2020, pp. 25-38, available at: <https://traffic.fpz.hr/index.php/PROMTT/article/view/3296>
- [27] Ehmke, J. F., Steinert, A., Mattfeld, D. C., “Advanced routing for city logistics service providers based on time-dependent travel times”, *Journal of Computational Science*, Vol. 3, No. 4, 2012, pp. 193–205.

- [28] Erdelić, T., Carić, T., Erdelić, M., Tišljarić, L., Turković, A., Jelušić, N., “Estimating congestion zones and travel time indexes based on the floating car data”, *Computers, Environment and Urban Systems*, Vol. 87, 2021, pp. 101604, available at: <https://www.sciencedirect.com/science/article/pii/S0198971521000119>
- [29] Hiermann, G., Puchinger, J., Ropke, S., Hartl, R. F., “The electric fleet size and mix vehicle routing problem with time windows and recharging stations”, *European Journal of Operational Research*, Vol. 252, No. 3, 2016, pp. 995 - 1018, available at: <http://www.sciencedirect.com/science/article/pii/S0377221716000837>
- [30] Dantzig, G. B., Ramser, J. H., “The truck dispatching problem”, *Manage. Sci.*, Vol. 6, No. 1, Oct. 1959, pp. 80–91.
- [31] Solomon, M. M., “Algorithms for the vehicle routing and scheduling problems with time window constraints”, *Operations Research*, Vol. 35, No. 2, 1987, pp. 254-265, available at: <https://doi.org/10.1287/opre.35.2.254>
- [32] Ropke, S., Pisinger, D., “An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows”, *Transportation Science*, Vol. 40, No. 4, 2006, pp. 455-472, available at: <https://doi.org/10.1287/trsc.1050.0135>
- [33] Golden, B., Assad, A., Levy, L., Gheysens, F., “The fleet size and mix vehicle routing problem”, *Computers & Operations Research*, Vol. 11, No. 1, 1984, pp. 49 - 66, available at: <http://www.sciencedirect.com/science/article/pii/0305054884900078>
- [34] Bräysy, O., Gendreau, M., “Vehicle routing problem with time windows, part I: Route construction and local search algorithms”, *Transportation Science*, Vol. 39, No. 1, 2005, pp. 104-118, available at: <https://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0056>
- [35] Schiffer, M., Stütz, S., Walther, G., “Are ECVs breaking even? - Competitiveness of electric commercial vehicles in medium-duty logistics networks”, RWTH Aachen University, Aachen, Tech. Rep. Working Paper OM-02/2016, 2016, available at: <https://publications.rwth-aachen.de/record/689179>
- [36] Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S., “A Matheuristic for the Electric Vehicle Routing Problem with Time Windows”, *ArXiv e-prints*, May 2015, available at: <https://arxiv.org/abs/1506.00211>
- [37] Demir, E., Bektaş, T., Laporte, G., “An adaptive large neighborhood search heuristic for the pollution-routing problem”, *European Journal of Operational Research*, Vol. 223, No. 2, 2012, pp. 346 - 359, available at: <http://www.sciencedirect.com/science/article/pii/S0377221712004997>

- [38] Macrina, G., Pugliese, L. D. P., Guerriero, F., Laporte, G., “The green mixed fleet vehicle routing problem with partial battery recharging and time windows”, *Computers & Operations Research*, Vol. 101, 2019, pp. 183 - 199, available at: <http://www.sciencedirect.com/science/article/pii/S0305054818301965>
- [39] Barco, J., Guerra, A., Muñoz, L., Quijano, N., “Optimal routing and scheduling of charge for electric vehicles: Case study”, *Computing Research Repository*, Vol. abs/1310.0145, 2013, available at: <http://arxiv.org/abs/1310.0145>
- [40] Preis, H., Frank, S., Nachtigall, K., “Energy-optimized routing of electric vehicles in urban delivery systems”, in *Operations Research Proceedings 2012*, Helber, S., Breitner, M., Rösch, D., Schön, C., Graf von der Schulenburg, J.-M., Sibbertsen, P., Steinbach, M., Weber, S., Wolter, A., (ed.). Cham: Springer International Publishing, 2014, pp. 583–588.
- [41] Zhang, S., Gajpal, Y., Appadoo, S., Abdulkader, M., “Electric vehicle routing problem with recharging stations for minimizing energy consumption”, *International Journal of Production Economics*, Vol. 203, No. C, 2018, pp. 404-413.
- [42] Demir, E., Bektaş, T., Laporte, G., “The bi-objective pollution-routing problem”, *European Journal of Operational Research*, Vol. 232, No. 3, 2014, pp. 464 - 478, available at: <http://www.sciencedirect.com/science/article/pii/S0377221713006486>
- [43] Poonthalir, G., Nadarajan, R., “A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP)”, *Expert Systems with Applications*, Vol. 100, 2018, pp. 131 - 144, available at: <http://www.sciencedirect.com/science/article/pii/S0957417418300678>
- [44] Amiri, S. S., Jadid, S., Saboori, H., “Multi-objective optimum charging management of electric vehicles through battery swapping stations”, *Energy*, Vol. 165, 2018, pp. 549 - 562, available at: <http://www.sciencedirect.com/science/article/pii/S036054421831939X>
- [45] Grunditz, E. A., Thiringer, T., “Performance Analysis of Current BEVs Based on a Comprehensive Review of Specifications”, *IEEE Transactions on Transportation Electrification*, Vol. 2, No. 3, Sept 2016, pp. 270-289.
- [46] den Boer, E., Aarnink, S., Kleiner, F., Pagenkopf, J., “An overview of state-of-the-art technologies and their potential”, Germany, Tech. Rep., 2013, available at: https://www.theicct.org/sites/default/files/publications/CE_Delft_4841_Zero_emissions_trucks_Def.pdf

- [47] van Duin, J., Tavasszy, L., Quak, H., “Towards e(lectric)-urban freight: First promising steps in the electronic vehicle revolution”, *European Transport - Trasporti Europei*, Vol. 2013, No. 54, 2013, pp. 1–19.
- [48] Liao, C.-S., Lu, S.-H., Shen, Z.-J. M., “The electric vehicle touring problem”, *Transportation Research Part B: Methodological*, Vol. 86, 2016, pp. 163 - 180, available at: <http://www.sciencedirect.com/science/article/pii/S0191261516000278>
- [49] Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., “Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models”, *Transportation Research Part B: Methodological*, Vol. 103, 2017, pp. 158 - 187, *green Urban Transportation*, available at: <http://www.sciencedirect.com/science/article/pii/S0191261516303794>
- [50] UPS. (2013) UPS to rollout fleet of electric vehicles in California. Accessed: 15. December 2020, available at: <https://www.pressroom.ups.com/pressroom/news-assets/new-assets-subpage-landing.page?ConceptType=PressReleases>
- [51] DPDHL. (2014) Electric vehicles in inner city distribution traffic. Accessed: 15. December 2020, available at: https://www.haw-hamburg.de/fileadmin/user_upload/FakLS/07Forschung/FTZ-ALS/Veranstaltungen/Fuelling_the_Climate/Lohmeyer_FTC2014_VOE.pdf
- [52] Lin, J., Zhou, W., Wolfson, O., “Electric vehicle routing problem”, *Transportation Research Procedia*, Vol. 12, 2016, pp. 508 - 521, 10th International Conference on City Logistics 17-19 June 2015, Tenerife, Spain, available at: <http://www.sciencedirect.com/science/article/pii/S2352146516000089>
- [53] Schiffer, M., Walther, G., “An adaptive large neighborhood search for the location-routing problem with intra-route facilities”, *Transportation Science*, Vol. 52, No. 2, 2018, pp. 331-352, available at: <https://doi.org/10.1287/trsc.2017.0746>
- [54] Hiermann, G., Hartl, R. F., Puchinger, J., Vidal, T., “Routing a mix of conventional, plug-in hybrid, and electric vehicles”, *European Journal of Operational Research*, Vol. 272, No. 1, 2019, pp. 235 - 248, available at: <http://www.sciencedirect.com/science/article/pii/S0377221718305605>
- [55] Desaulniers, G., Errico, F., Irnich, S., Schneider, M., “Exact algorithms for electric vehicle-routing problems with time windows”, *Operations Research*, Vol. 64, No. 6, 2016, pp. 1388-1405, available at: <https://doi.org/10.1287/opre.2016.1535>

- [56] Roberti, R., Wen, M., “The electric traveling salesman problem with time windows”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 89, 2016, pp. 32 - 52, available at: <http://www.sciencedirect.com/science/article/pii/S1366554516000181>
- [57] Doppstadt, C., Koberstein, A., Vigo, D., “The hybrid electric vehicle – traveling salesman problem”, *European Journal of Operational Research*, Vol. 253, No. 3, 2016, pp. 825 - 842, available at: <http://www.sciencedirect.com/science/article/pii/S0377221716301163>
- [58] Sweda, T. M., Dolinskaya, I. S., Klabjan, D., “Adaptive routing and recharging policies for electric vehicles”, *Transportation Science*, Vol. 51, No. 4, 2017, pp. 1326-1348, available at: <https://doi.org/10.1287/trsc.2016.0724>
- [59] Sassi, O., Cherif, W. R., Oulamara, A., “Vehicle Routing Problem with Mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs”, Oct. 2014, working paper or preprint, available at: <https://hal.archives-ouvertes.fr/hal-01083966>
- [60] Barco, J., Guerra, A., Muñoz, L., Quijano, N., “Optimal routing and scheduling of charge for electric vehicles: Case study”, *Mathematical Problems in Engineering*, 2017, available at: <https://doi.org/10.1155/2017/8509783>
- [61] Keskin, M., Laporte, G., Çatay, B., “Electric vehicle routing problem with time dependent waiting times at recharging stations”, in *7th International Workshop on Freight Transportation and Logistics*, (ODYSSEUS 2018), 06 2018.
- [62] Restrepo, J., Rosero, J., Tellez, S., “Performance testing of electric vehicles on operating conditions in Bogota; DC, Colombia”, in *2014 IEEE PES Transmission Distribution Conference and Exposition - Latin America (PES T D-LA)*, Sept 2014, pp. 1-8.
- [63] Moghaddam, N. M., “The partially rechargeable electric vehicle routing problem with time windows and capacitated charging stations”, Master’s thesis, Clemson University, Clemson, South Carolina, 2015.
- [64] Yilmaz, M., Krein, P. T., “Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles”, *IEEE Transactions on Power Electronics*, Vol. 28, No. 5, May 2013, pp. 2151-2169.
- [65] Sassi, O., Cherif-Khettaf, W. R., Oulamara, A., “Iterated tabu search for the mix fleet vehicle routing problem with heterogenous electric vehicles”, in *Modelling, Computation and Optimization in Information Systems and Management Sciences*, Le Thi, H. A.,

- Pham Dinh, T., Nguyen, N. T., (ed.). Cham: Springer International Publishing, 2015, pp. 57–68.
- [66] Sassi, O., Ramdane Cherif-Khettaf, W., Oulamara, A., “Multi-Start Iterated Local Search for the Mixed fleet Vehicle Routing Problem with Heterogeneous Electric Vehicles”, in *Evolutionary Computation in Combinatorial Optimization*. Springer, 2015, Vol. 9026, pp. 138-149, available at: <https://hal.archives-ouvertes.fr/hal-01306896>
- [67] Pelletier, S., Jabali, O., Laporte, G., “Charge scheduling for electric freight vehicles”, *Transportation Research Part B: Methodological*, Vol. 115, 2018, pp. 246 - 269, available at: <http://www.sciencedirect.com/science/article/pii/S0191261517308871>
- [68] Kullman, N. D., Goodson, J., Mendoza, J. E., “Dynamic Electric Vehicle Routing: Heuristics and Dual Bounds”, Nov. 2018, working paper or preprint, available at: <https://hal.archives-ouvertes.fr/hal-01928730>
- [69] Froger, A., Mendoza, J. E., Jabali, O., Laporte, G., “A Matheuristic for the Electric Vehicle Routing Problem with Capacitated Charging Stations”, Centre interuniversitaire de recherche sur les reseaux d’entreprise, la logistique et le transport (CIRRELT), Research Report, Jun. 2017, available at: <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2017-31.pdf>
- [70] Çatay, B., Keskin, M., “The impact of quick charging stations on the route planning of electric vehicles”, in *2017 IEEE Symposium on Computers and Communications (ISCC)*, July 2017, pp. 152-157.
- [71] Güner, A. R., Murat, A., Chinnam, R. B., “Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information”, *Computers & Operations Research*, Vol. 39, No. 2, 2012, pp. 358 - 373, available at: <http://www.sciencedirect.com/science/article/pii/S0305054811001195>
- [72] Ehmke, J. F., Meisel, S., Mattfeld, D. C., “Floating car based travel times for city logistics”, *Transportation Research Part C: Emerging Technologies*, Vol. 21, No. 1, 2012, pp. 338 - 352, available at: <http://www.sciencedirect.com/science/article/pii/S0968090X11001562>
- [73] Erdelić, T., Ravlić, M., Carić, T., “Travel time prediction using speed profiles for road network of Croatia”, in *2016 International Symposium ELMAR*, Sept 2016, pp. 97-100.
- [74] Kellner, F., “Insights into the effect of traffic congestion on distribution network characteristics – a numerical analysis based on navigation service data”, *International Journal of Logistics Research and Applications*, Vol. 19, No. 5, 2016, pp. 395–423.

- [75] Malandraki, C., Daskin, M. S., “Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms”, *Transportation Science*, Vol. 26, No. 3, 1992, pp. 185-200.
- [76] Gendreau, M., Ghiani, G., Guerriero, E., “Time-dependent routing problems: A review”, *Computers and Operations Research*, Vol. 64, 2015, pp. 189 - 197.
- [77] Figliozzi, M., “Vehicle routing problem for emissions minimization”, *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2197, 2010, pp. 1-7, available at: <https://doi.org/10.3141/2197-01>
- [78] Shao, S., Guan, W., Ran, B., He, Z., Bi, J., “Electric vehicle routing problem with charging time and variable travel time”, *Mathematical Problems in Engineering*, 2017, pp. 13.
- [79] Mancini, S., “The hybrid vehicle routing problem”, *Transportation Research Part C: Emerging Technologies*, Vol. 78, 2017, pp. 1 - 12, available at: <http://www.sciencedirect.com/science/article/pii/S0968090X1730044X>
- [80] Abdallah, T., “The plug-in hybrid electric vehicle routing problem with time windows”, Master’s thesis, 2013, available at: <http://hdl.handle.net/10012/7582>
- [81] Vincent, F. Y., Redi, A. P., Hidayat, Y. A., Wibowo, O. J., “A simulated annealing heuristic for the hybrid vehicle routing problem”, *Applied Soft Computing*, Vol. 53, 2017, pp. 119 - 132, available at: <http://www.sciencedirect.com/science/article/pii/S1568494616306524>
- [82] Lebeau, P., de Cauwer, C., van Mierlo, J., Macharis, C., Verbeke, W., Coosemans, T., “Conventional, hybrid, or electric vehicles: Which technology for an urban distribution centre?”, *The Scientific World Journal*, 2015, pp. 11.
- [83] Montoya, A., Guéret, C., Mendoza, J. E., Villegas, J. G., “The electric vehicle routing problem with nonlinear charging function”, *Transportation Research Part B: Methodological*, Vol. 103, 2017, pp. 87 - 110, *green Urban Transportation*, available at: <http://www.sciencedirect.com/science/article/pii/S0191261516304556>
- [84] Zündorf, T., “Electric vehicle routing with realistic recharging models”, Master’s thesis, Karlsruhe Institute of Technology, Germany, 2014.
- [85] Froger, A., Mendoza, J. E., Jabali, O., Laporte, G., “Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions”, *Computers & Operations Research*, Vol. 104, 2019, pp. 256 - 294, available at: <http://www.sciencedirect.com/science/article/pii/S0305054818303253>

- [86] Prodhon, C., Prins, C., “A survey of recent research on location-routing problems”, *European Journal of Operational Research*, Vol. 238, No. 1, 2014, pp. 1 - 17, available at: <http://www.sciencedirect.com/science/article/pii/S0377221714000071>
- [87] Schiffer, M., Walther, G., “The electric location routing problem with time windows and partial recharging”, *European Journal of Operational Research*, Vol. 260, No. 3, 2017, pp. 995 - 1013, available at: <http://www.sciencedirect.com/science/article/pii/S0377221717300346>
- [88] Ravlić, M., Erdelić, T., Carić, T., “Optimizing charging station locations for fleet of electric vehicles using multi-source weber problem”, in 2016 International Symposium ELMAR, 2016, pp. 111-114, available at: <https://ieeexplore.ieee.org/document/7731766>
- [89] Adler, J. D., Mirchandani, P. B., “Online routing and battery reservations for electric vehicles with swappable batteries”, *Transportation Research Part B: Methodological*, Vol. 70, 2014, pp. 285 - 302, available at: <http://www.sciencedirect.com/science/article/pii/S0191261514001593>
- [90] Yang, J., Sun, H., “Battery swap station location-routing problem with capacitated electric vehicles”, *Computers & Operations Research*, Vol. 55, 2015, pp. 217 - 232, available at: <http://www.sciencedirect.com/science/article/pii/S0305054814001841>
- [91] Hof, J., Schneider, M., Goeke, D., “Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops”, *Transportation Research Part B: Methodological*, Vol. 97, 2017, pp. 102 - 112, available at: <http://www.sciencedirect.com/science/article/pii/S0191261516300285>
- [92] Erdođan, S., Miller-Hooks, E., “A green vehicle routing problem”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 48, No. 1, 2012, pp. 100 - 114, available at: <http://www.sciencedirect.com/science/article/pii/S1366554511001062>
- [93] Macrina, G., Laporte, G., Guerriero, F., Pugliese, L. D. P., “An energy-efficient green-vehicle routing problem with mixed vehicle fleet, partial battery recharging and time windows”, *European Journal of Operational Research*, 2019, available at: <http://www.sciencedirect.com/science/article/pii/S037722171930102X>
- [94] Bektaş, T., Laporte, G., “The pollution-routing problem”, *Transportation Research Part B: Methodological*, Vol. 45, No. 8, 2011, pp. 1232 - 1250, available at: <http://www.sciencedirect.com/science/article/pii/S019126151100018X>

- [95] Breunig, U., Baldacci, R., Hartl, R., Vidal, T., “The electric two-echelon vehicle routing problem”, *Computers & Operations Research*, Vol. 103, 2019, pp. 198 - 210, available at: <http://www.sciencedirect.com/science/article/pii/S0305054818302909>
- [96] Jie, W., Yang, J., Zhang, M., Huang, Y., “The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology”, *European Journal of Operational Research*, Vol. 272, No. 3, 2019, pp. 879 - 904, available at: <http://www.sciencedirect.com/science/article/pii/S0377221718306076>
- [97] Bruglieri, M., Pezzella, F., Pisacane, O., “A two-phase optimization method for a multiobjective vehicle relocation problem in electric carsharing systems”, *Journal of Combinatorial Optimization*, Vol. 36, No. 1, Jul 2018, pp. 162–193, available at: <https://doi.org/10.1007/s10878-018-0295-5>
- [98] Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., Cheikhrouhou, N., “The dial-a-ride problem with electric vehicles and battery swapping stations”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 118, 2018, pp. 392 - 420, available at: <http://www.sciencedirect.com/science/article/pii/S1366554517310086>
- [99] Paz, J., Granada-Echeverri, M., Escobar, J., “The multi-depot electric vehicle location routing problem with time windows”, *International Journal of Industrial Engineering Computations*, Vol. 9, 2018, pp. 123 - 136.
- [100] Schiffer, M., Walther, G., “Strategic planning of electric logistics fleet networks: A robust location-routing approach”, *Omega*, Vol. 80, 2018, pp. 31 - 42, available at: <http://www.sciencedirect.com/science/article/pii/S0305048317303079>
- [101] Ç. Koç, Karaoglan, I., “The green vehicle routing problem: A heuristic based exact solution approach”, *Applied Soft Computing*, Vol. 39, 2016, pp. 154-164, available at: <https://www.sciencedirect.com/science/article/pii/S1568494615007085>
- [102] Montoya, A., Guéret, C., Mendoza, J. E., Villegas, J. G., “A multi-space sampling heuristic for the green vehicle routing problem”, *Transportation Research Part C: Emerging Technologies*, Vol. 70, 2016, pp. 113 - 128, available at: <http://www.sciencedirect.com/science/article/pii/S0968090X15003320>
- [103] Pelletier, S., Jabali, O., Laporte, G., “50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives”, *Transportation Science*, Vol. 50, No. 1, 2016, pp. 3-22, available at: <https://doi.org/10.1287/trsc.2015.0646>

- [104] Gillett, B. E., Miller, L. R., “A heuristic algorithm for the vehicle-dispatch problem”, *Operations Research*, Vol. 22, No. 2, 1974, pp. 340-349, available at: <https://doi.org/10.1287/opre.22.2.340>
- [105] Clarke, G., Wright, J. W., “Scheduling of vehicles from a central depot to a number of delivery points”, *Operations Research*, Vol. 12, No. 4, 1964, pp. 568-581, available at: <https://doi.org/10.1287/opre.12.4.568>
- [106] Hansen, P., Mladenović, N., “First vs. best improvement: An empirical study”, *Discrete Applied Mathematics*, Vol. 154, No. 5, 2006, pp. 802 - 817, iV ALIO/EURO Workshop on Applied Combinatorial Optimization, available at: <http://www.sciencedirect.com/science/article/pii/S0166218X05003070>
- [107] Savelsbergh, M. W. P., “The vehicle routing problem with time windows: Minimizing route duration”, *ORSA Journal on Computing*, Vol. 4, No. 2, 1992, pp. 146-154, available at: <https://doi.org/10.1287/ijoc.4.2.146>
- [108] Or, I., “Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking”, 1976.
- [109] Lin, S., “Computer solutions of the traveling salesman problem”, *The Bell System Technical Journal*, Vol. 44, No. 10, Dec 1965, pp. 2245-2269.
- [110] Osman, I. H., “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem”, *Annals of Operations Research*, Vol. 41, No. 4, Dec 1993, pp. 421–451, available at: <https://doi.org/10.1007/BF02023004>
- [111] Potvin, J.-Y., Rousseau, J.-M., “An exchange heuristic for routeing problems with time windows”, *The Journal of the Operational Research Society*, Vol. 46, No. 12, 1995, pp. 1433–1446, available at: <http://www.jstor.org/stable/2584063>
- [112] Prodhon, C., Prins, C., *Metaheuristics for Vehicle Routing Problems*. Cham: Springer International Publishing, 2016, pp. 407–437, available at: https://doi.org/10.1007/978-3-319-45403-0_15
- [113] Gendreau, M., Potvin, J.-Y., Bräumlaysy, O., Hasle, G., Løkketangen, A., *Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography*. Boston, MA: Springer US, 2008, pp. 143–169, available at: https://doi.org/10.1007/978-0-387-77778-8_7
- [114] Blum, C., Blesa, M., Roli, A., Sampels, M., *Hybrid Metaheuristics: An Emerging Approach to Optimization*, January 2008.

- [115] Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
- [116] Resende, M. G., Ribeiro, C. C., Glover, F., Martí, R., *Scatter Search and Path-Relinking: Fundamentals, Advances, and Applications*. Boston, MA: Springer US, 2010, pp. 87–107, available at: https://doi.org/10.1007/978-1-4419-1665-5_4
- [117] Dorigo, M., Stützle, T., *Ant Colony Optimization*. Bradford book, 2004, available at: https://books.google.hr/books?id=_aefcpY8GiEC
- [118] Marinakis, Y., Marinaki, M., *Bumble Bees Mating Optimization Algorithm for the Vehicle Routing Problem*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 347–369, available at: https://doi.org/10.1007/978-3-642-17390-5_15
- [119] Kennedy, J., Eberhart, R., “Particle swarm optimization”, in *Proceedings of ICNN’95 - International Conference on Neural Networks*, Vol. 4, Nov 1995, pp. 1942-1948 vol.4.
- [120] Marinakis, Y., Marinaki, M., “A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem”, *Expert Systems with Applications*, Vol. 37, No. 2, 2010, pp. 1446 - 1455, available at: <http://www.sciencedirect.com/science/article/pii/S0957417409006460>
- [121] Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., “Timing problems and algorithms: Time decisions for sequences of activities”, *Networks*, Vol. 65, No. 2, 2015, pp. 102-128, available at: <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21587>
- [122] Nagata, Y., “Edge assembly crossover for the capacitated vehicle routing problem”, in *Evolutionary Computation in Combinatorial Optimization*, Cotta, C., van Hemert, J., (ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 142–153.
- [123] Potvin, J.-Y., “State-of-the art review—evolutionary algorithms for vehicle routing”, *INFORMS Journal on Computing*, Vol. 21, No. 4, 2009, pp. 518-548, available at: <https://doi.org/10.1287/ijoc.1080.0312>
- [124] Darwin, C., *On the Origin of Species by Means of Natural Selection*. London: Murray, 1859, or the *Preservation of Favored Races in the Struggle for Life*.
- [125] Alvarenga, G., Silva, R., Sampaio, R., “A hybrid algorithm for the vehicle routing problem with window”, *INFOCOMP Journal of Computer Science*, Vol. 4, No. 2, 2004, pp. 9–16, available at: <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/87>

- [126] Bräysy, O., Gendreau, M., “Genetic algorithms for the vehicle routing problem with time windows”, *Arpakannus*,(1), 2001, pp. 33–38.
- [127] Gendreau, M., Potvin, J.-Y., *Handbook of Metaheuristics*, 2nd ed. Springer Publishing Company, Incorporated, 2010.
- [128] Tan, K., Lee, L., Zhu, Q., Ou, K., “Heuristic methods for vehicle routing problem with time windows”, *Artificial Intelligence in Engineering*, Vol. 15, No. 3, 2001, pp. 281-295, available at: <https://www.sciencedirect.com/science/article/pii/S095418100100005X>
- [129] Alesiani, F., Maslekar, N., “Optimization of charging stops for fleet of electric vehicles: A genetic approach”, *IEEE Intelligent Transportation Systems Magazine*, Vol. 6, No. 3, Fall 2014, pp. 10-21.
- [130] Shao, S., Guan, W., Bi, J., “Electric vehicle-routing problem with charging demands and energy consumption”, *IET Intelligent Transport Systems*, Vol. 12, No. 3, 2018, pp. 202-212.
- [131] Masliakova, K., “Optimal routing and charging procedures for electric buses”, Master’s thesis, The Arctic University of Norway, Narvik, Norway, 2016.
- [132] Galić, A., “Solving the vehicle routing problem using hybrid cellular evolutionary algorithm”, Phd thesis, 2018, Faculty of Transport and Traffic Sciences, University of Zagreb (in Croatian).
- [133] Joo, H., Lim, Y., “Ant colony optimized routing strategy for electric vehicles”, *Journal of Advanced Transportation*, 2018, pp. 9.
- [134] Villacorta, P. J., “Swarm Intelligence Metaheuristics, part 2: Particle Swarm Optimization”, <https://blog.stratio.com/swarm-intelligence-metaheuristics-part-2-particle-swarm-optimization/>, 2019.
- [135] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., “Optimization by simulated annealing”, *Science*, Vol. 220, No. 4598, 1983, pp. 671–680, available at: <http://science.sciencemag.org/content/220/4598/671>
- [136] Černý, V., “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm”, *Journal of Optimization Theory and Applications*, Vol. 45, No. 1, 1985, pp. 41-51, available at: <https://app.dimensions.ai/details/publication/pub.1051937309>

- [137] Tsuzuki, M. S. G., (ed.), *Simulated Annealing - Advances, Applications and Hybridizations*, ser. Books. IntechOpen, June 2012, No. 2399, available at: <https://ideas.repec.org/b/ito/pbooks/2399.html>
- [138] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., “Equation of State Calculations by Fast Computing Machines”, *The Journal of Chemical Physics*, Vol. 21, Jun. 1953, pp. 1087-1092.
- [139] Glover, F., “Tabu search—part II”, *ORSA Journal on Computing*, Vol. 2, No. 1, 1990, pp. 4-32, available at: <https://doi.org/10.1287/ijoc.2.1.4>
- [140] Glover, F., “Tabu search—part I”, *ORSA Journal on Computing*, Vol. 1, No. 3, 1989, pp. 190-206, available at: <https://doi.org/10.1287/ijoc.1.3.190>
- [141] Mladenović, N., Hansen, P., “Variable neighborhood search”, *Computers & Operations Research*, Vol. 24, No. 11, 1997, pp. 1097 - 1100, available at: <http://www.sciencedirect.com/science/article/pii/S0305054897000312>
- [142] Schneider, M., Stenger, A., Hof, J., “An adaptive VNS algorithm for vehicle routing problems with intermediate stops”, *OR Spectrum*, Vol. 37, No. 2, Mar 2015, pp. 353–387.
- [143] Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S., “A variable neighborhood search branching for the electric vehicle routing problem with time windows”, *Electronic Notes in Discrete Mathematics*, Vol. 47, 2015, pp. 221 - 228, available at: <http://www.sciencedirect.com/science/article/pii/S1571065314000717>
- [144] Lourenço, H. R., Martin, O. C., Stützle, T., *Iterated Local Search: Framework and Applications*. Boston, MA: Springer US, 2010, pp. 363–397, available at: https://doi.org/10.1007/978-1-4419-1665-5_12
- [145] Shaw, P., “Using constraint programming and local search methods to solve vehicle routing problems”, in *Principles and Practice of Constraint Programming — CP98*, Maher, M., Puget, J.-F., (ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 417–431.
- [146] Christiaens, J., Vanden Berghe, G., “Slack Induction by String Removals for Vehicle Routing Problems”, KU Leuven, Department of Computer Science, CODeS & imec, Technical Report, 2018, available at: https://limo.libis.be/primo-explore/fulldisplay?docid=LIRIAS1988666&context=L&vid=Lirias&search_scope=Lirias&tab=default_tab&lang=en_US

- [147] Pisinger, D., Ropke, S., “A general heuristic for vehicle routing problems”, *Computers & Operations Research*, Vol. 34, No. 8, 2007, pp. 2403 - 2435, available at: <http://www.sciencedirect.com/science/article/pii/S0305054805003023>
- [148] Lipowski, A., Lipowska, D., “Roulette-wheel selection via stochastic acceptance”, *Physica A: Statistical Mechanics and its Applications*, Vol. 391, No. 6, Mar 2012, pp. 2193–2196, available at: <http://dx.doi.org/10.1016/j.physa.2011.12.004>
- [149] Cordeau, J.-F., Laporte, G., Mercier, A., “A unified tabu search heuristic for vehicle routing problems with time windows”, *Journal of the Operational Research Society*, Vol. 52, No. 8, 2001, pp. 928-936, available at: <https://doi.org/10.1057/palgrave.jors.2601163>
- [150] Galić, A., “Metaheuristic methods for solving vehicle routing problem with time windows”, pp. 126, M.Sc. thesis, Faculty of Transport and Traffic Sciences, University of Zagreb (in Croatian). 2012.
- [151] Kindervater, G. A. P., Savelsbergh, M. W. P., 10. Vehicle routing: handling edge exchanges. Princeton University Press, 2018, pp. 337–360, available at: <https://doi.org/10.1515/9780691187563-013>
- [152] Nagata, Y., Bräysy, O., Dullaert, W., “A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows”, *Computers & Operations Research*, Vol. 37, No. 4, 2010, pp. 724-737, available at: <https://www.sciencedirect.com/science/article/pii/S0305054809001762>
- [153] Schneider, M., Sand, B., Stenger, A., “A note on the time travel approach for handling time windows in vehicle routing problems”, *Computers & Operations Research*, Vol. 40, No. 10, 2013, pp. 2564 - 2568, available at: <http://www.sciencedirect.com/science/article/pii/S0305054813000348>
- [154] Suzuki, Y., “A variable-reduction technique for the fixed-route vehicle-refueling problem”, *Computers & Industrial Engineering*, Vol. 67, 2014, pp. 204-215, available at: <https://www.sciencedirect.com/science/article/pii/S0360835213003707>
- [155] Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A., “An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem”, *Computers and Operations Research*, Vol. 76, December 2016, pp. 73–83.
- [156] Feo, T. A., Resende, M. G. C., “Greedy randomized adaptive search procedures”, *Journal of Global Optimization*, Vol. 6, No. 2, Mar 1995, pp. 109–133, available at: <https://doi.org/10.1007/BF01096763>

- [157] Rožić, L., Fosin, J., Carić, T., “Solving the time dependent vehicle routing problem using real-world speed profiles”, Central European Conference on Information and Intelligent Systems, 2015, 2015.
- [158] Fosin, J., “Time dependent vehicle routing problem solving method based on speed profiles”, Phd thesis, 2016, Faculty of Transport and Traffic Sciences, University of Zagreb (in Croatian).
- [159] Nagata, Y., Bräysy, O., “A powerful route minimization heuristic for the vehicle routing problem with time windows”, *Operations Research Letters*, Vol. 37, No. 5, 2009, pp. 333-338, available at: <https://www.sciencedirect.com/science/article/pii/S0167637709000662>
- [160] Lim, A., Zhang, X., “A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows”, *INFORMS Journal on Computing*, Vol. 19, No. 3, 2007, pp. 443-457, available at: <https://doi.org/10.1287/ijoc.1060.0186>
- [161] Montoya, J.-A., “Electric vehicle routing problems : models and solution approaches”, Phd thesis, December 2016, Computation and Language [cs.CL]. Université d’Angers.
- [162] Irnich, S., Desaulniers, G., *Shortest Path Problems with Resource Constraints*. Boston, MA: Springer US, 2005, pp. 33–65, available at: https://doi.org/10.1007/0-387-25486-2_2
- [163] Psaraftis, H. N., “K-interchange procedures for local search in a precedence-constrained routing problem”, *European Journal of Operational Research*, Vol. 13, No. 4, 1983, pp. 391-402, available at: <https://www.sciencedirect.com/science/article/pii/0377221783900991>
- [164] Savelsbergh, M., “Local search in routing problems with time windows”, *Annals of Operations Research*, Vol. 4, 1984, pp. 285-305.
- [165] Emeç, U., Çatay, B., Bozkaya, B., “An adaptive large neighborhood search for an e-grocery delivery routing problem”, *Computers & Operations Research*, Vol. 69, 2016, pp. 109 - 125, available at: <http://www.sciencedirect.com/science/article/pii/S0305054815002683>
- [166] Erdelić, T., Carić, T., Erdelić, M., Tišljarić, L., “Electric vehicle routing problem with single or multiple recharges”, *Transportation Research Procedia*, Vol. 40, 2019, pp. 217 - 224, TRANSCOM 2019 13th International Scientific

- Conference on Sustainable, Modern and Safe Transport, available at: <http://www.sciencedirect.com/science/article/pii/S2352146519301942>
- [167] Taş, D., “Electric vehicle routing with flexible time windows: a column generation solution approach”, *Transportation Letters*, Vol. 13, No. 2, 2021, pp. 97-103, available at: <https://doi.org/10.1080/19427867.2020.1711581>
- [168] Jin, J., Rafferty, P., “Externalities of auto traffic congestion growth: Evidence from the residential property values in the US Great Lakes megaregion”, *Journal of Transport Geography*, Vol. 70, 2018, pp. 131 - 140.
- [169] Tang, L., Kan, Z., Zhang, X., Yang, X., Huang, F., Li, Q., “Travel time estimation at intersections based on low-frequency spatial-temporal GPS trajectory big data”, *Cartography and Geographic Information Science*, Vol. 43, No. 5, 2016, pp. 417-426.
- [170] Erdelić, T., Carić, T., Ravlić, M., “Predicting waiting times at intersections”, in 2017 International Symposium ELMAR, 2017, pp. 31-34.
- [171] Gavanas, N., Tsakalidis, A., Pitsiava-Latinopoulou, M., “Assessment of the marginal social cost due to congestion using the speed flow function”, in *Transportation Research Procedia*, Vol. 24. Elsevier B.V., 2017, pp. 250–258.
- [172] Chen, D., Ignatius, J., Sun, D., Goh, M., Zhan, S., “Impact of congestion pricing schemes on emissions and temporal shift of freight transport”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 118, 2018, pp. 77 - 105.
- [173] Alinaghian, M., Naderipour, M., “A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: A case study”, *Computers and Industrial Engineering*, Vol. 99, 2016, pp. 210–222.
- [174] Tišljarić, L., Erdelić, T., Carić, T., “Analysis of intersection queue lengths and level of service using GPS data”, in 2018 International Symposium ELMAR, 2018, pp. 43-46.
- [175] Gately, C. K., Hutyra, L. R., Peterson, S., Wing, I. S., “Urban emissions hotspots: Quantifying vehicle congestion and air pollution using mobile phone GPS data”, *Environmental Pollution*, Vol. 229, 2017, pp. 496 - 504.
- [176] Jabali, O., van Woensel, T., de Kok, A., “Analysis of travel times and CO2 emissions in time-dependent vehicle routing”, *Production and Operations Management*, Vol. 21, No. 6, 2012, pp. 1060-1074.

- [177] Chow, A. H., Santacreu, A., Tsapakis, I., Tanasaranond, G., Cheng, T., “Empirical assessment of urban traffic congestion”, *Journal of Advanced Transportation*, Vol. 48, No. 8, 2014, pp. 1000–1016.
- [178] Kok, A., Hans, E., Schutten, J., “Vehicle routing under time-dependent travel times: The impact of congestion avoidance”, *Computers and Operations Research*, Vol. 39, No. 5, 2012, pp. 910 - 918.
- [179] Liu, C., Kou, G., Zhou, X., Peng, Y., Sheng, H., Alsaadi, F. E., “Time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach”, *Knowledge-Based Systems*, 2019.
- [180] Rincon-Garcia, N., Waterson, B., Cherrett, T. J., Salazar-Arrieta, F., “A metaheuristic for the time-dependent vehicle routing problem considering driving hours regulations – an application in city logistics”, *Transportation Research Part A: Policy and Practice*, 2018.
- [181] Ichoua, S., Gendreau, M., Potvin, J.-Y., “Vehicle dispatching with time-dependent travel times”, *European Journal of Operational Research*, Vol. 144, No. 2, 2003, pp. 379 - 396, available at: <http://www.sciencedirect.com/science/article/pii/S0377221702001479>
- [182] Ehmke, J. F., Steinert, A., Mattfeld, D. C., “Advanced routing for city logistics service providers based on time-dependent travel times”, *Journal of Computational Science*, Vol. 3, No. 4, 2012, pp. 193 - 205, city Logistics, available at: <http://www.sciencedirect.com/science/article/pii/S1877750312000087>
- [183] Hill, A. V., Benton, W. C., “Modelling intra-city time-dependent travel speeds for vehicle scheduling problems”, *Journal of the Operational Research Society*, Vol. 43, No. 4, 1992, pp. 343-351, available at: <https://doi.org/10.1057/jors.1992.49>
- [184] Montero, A., Méndez-Díaz, I., Miranda-Bront, J. J., “An integer programming approach for the time-dependent traveling salesman problem with time windows”, *Computers & Operations Research*, Vol. 88, 2017, pp. 280-289, available at: <https://www.sciencedirect.com/science/article/pii/S0305054817301612>
- [185] Fleming, C. L., Griffis, S. E., Bell, J. E., “The effects of triangle inequality on the vehicle routing problem”, *European Journal of Operational Research*, Vol. 224, No. 1, 2013, pp. 1-7, available at: <https://www.sciencedirect.com/science/article/pii/S0377221712005267>
- [186] Artmeier, A., Haselmayr, J., Leucker, M., Sachenbacher, M., “The shortest path problem revisited: Optimal routing for electric vehicles”, in *KI 2010: Advances in Artificial*

- Intelligence, Dillmann, R., Beyerer, J., Hanebeck, U. D., Schultz, T., (ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 309–316.
- [187] Ortega-Arranz, H., Llanos, D., Gonzalez-Escribano, A., *The Shortest-Path Problem: Analysis and Comparison of Methods*, ser. Synthesis Lectures on Theoretical Computer Science. Morgan & Claypool Publishers, 2014, available at: <https://books.google.hr/books?id=WtzSBQAAQBAJ>
- [188] Yang, Y., Xu, Y., Han, J., Wang, E., Chen, W., Yue, L., “Efficient traffic congestion estimation using multiple spatio-temporal properties”, *Neurocomputing*, Vol. 267, 2017, pp. 344 - 353, available at: <http://www.sciencedirect.com/science/article/pii/S092523121731072X>
- [189] Keler, A., Krisp, J. M., Ding, L., “Detecting traffic congestion propagation in urban environments – a case study with Floating Taxi Data (FTD) in Shanghai”, *Journal of Location Based Services*, Vol. 11, No. 2, 2017, pp. 133-151.
- [190] D’Andrea, E., Marcelloni, F., “Detection of traffic congestion and incidents from GPS trace analysis”, *Expert Systems with Applications*, Vol. 73, 2017, pp. 43–56.
- [191] An, S., Yang, H., Wang, J., Cui, N., Cui, J., “Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data”, *Information Sciences*, Vol. 373, 2016, pp. 515–526.
- [192] Keler, A., Ding, L., Krisp, J. M., “Visualization of traffic congestion based on floating taxi data”, *Kartographische Nachrichten*, Vol. 66, 2016, pp. 7-13.
- [193] Tišljarić, L., Majstorović, Ž., Erdelić, T., Carić, T., “Measure for traffic anomaly detection on the urban roads using speed transition matrices”, in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 252-259.
- [194] Tišljarić, L., Carić, T., Erdelić, T., Erdelić, M., “Traffic state estimation using speed profiles and convolutional neural networks”, in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, Skala, K., (ed.). MIPRO Croatian Society, 2020, pp. 2147-2152.
- [195] Sunderrajan, A., Varadarajan, J., Lye, K. W., “Road speed profiling for upfront travel time estimation”, in *IEEE International Conference on Data Mining Workshops, ICDMW*, Vol. 2018-Novem. IEEE Computer Society, 2019, pp. 648–654.

- [196] Beliakov, G., Gagolewski, M., James, S., Pace, S., Pastorello, N., Thilliez, E., Vasa, R., “Measuring traffic congestion: An approach based on learning weighted inequality, spread and aggregation indices from comparison data”, *Applied Soft Computing Journal*, Vol. 67, 2018, pp. 910–919.
- [197] Sanaullah, I., Quddus, M., Enoch, M., “Developing travel time estimation methods using sparse GPS data”, *Journal of Intelligent Transportation Systems*, Vol. 20, No. 6, 2016, pp. 532-544.
- [198] Tulic, M., Bauer, D., Scherrer, W., “Link and route travel time prediction including the corresponding reliability in an urban network based on taxi floating car data”, *Transportation Research Record*, Vol. 2442, No. 1, 2014, pp. 140-149.
- [199] Laranjeiro, P. F., Merchán, D., Godoy, L. A., Giannotti, M., Yoshizaki, H. T., Winkenbach, M., Cunha, C. B., “Using GPS data to explore speed patterns and temporal fluctuations in urban logistics: The case of São Paulo, Brazil”, *Journal of Transport Geography*, Vol. 76, 2019, pp. 114–129.
- [200] Han, J., Polak, J. W., Barria, J., Krishnan, R., “On the estimation of space-mean-speed from inductive loop detector data”, *Transportation Planning and Technology*, Vol. 33, No. 1, 2010, pp. 91–104.
- [201] Knoop, V., Hoogendoorn, S. P., van Zuylen, H., “Empirical differences between time mean speed and space mean speed”, in *Traffic and Granular Flow '07*, Appert-Rolland, C., Chevoir, F., Gondret, P., Lassarre, S., Lebacque, J.-P., Schreckenberg, M., (ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 351–356.
- [202] Casey, B., Bhaskar, A., Guo, H., Chung, E., “Critical review of time-dependent shortest path algorithms: A multimodal trip planner perspective”, *Transport Reviews*, Vol. 34, No. 4, 2014, pp. 522-539, available at: <https://doi.org/10.1080/01441647.2014.921797>
- [203] Jigang, W., Jin, S., Ji, H., Srikanthan, T., “Algorithm for time-dependent shortest safe path on transportation networks”, *Procedia Computer Science*, Vol. 4, 2011, pp. 958-966, proceedings of the International Conference on Computational Science, ICCS 2011, available at: <https://www.sciencedirect.com/science/article/pii/S1877050911001591>
- [204] Kritzinger, S., Doerner, K. F., Hartl, R. F., Kiechle, G., Stadler, H., Manohar, S. S., “Using Traffic Information for Time-Dependent Vehicle Routing”, *Procedia - Social and Behavioral Sciences*, Vol. 39, 2012, pp. 217–229.
- [205] Asamer, J., Graser, A., Heilmann, B., Ruthmair, M., “Sensitivity analysis for energy demand estimation of electric vehicles”, *Transportation Research Part*

- D: *Transport and Environment*, Vol. 46, 2016, pp. 182 - 199, available at: <http://www.sciencedirect.com/science/article/pii/S1361920915300250>
- [206] De Cauwer, C., Verbeke, W., Coosemans, T., Faid, S., Van Mierlo, J., “A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions”, *Energies*, Vol. 10, May 2017, pp. 608.
- [207] De Cauwer, C., Van Mierlo, J., Coosemans, T., “Energy consumption prediction for electric vehicles based on real-world data”, *Energies*, Vol. 8, No. 8, 2015, pp. 8573–8593, available at: <http://www.mdpi.com/1996-1073/8/8/8573>
- [208] Álvarez Fernández, R., “A more realistic approach to electric vehicle contribution to greenhouse gas emissions in the city”, *Journal of Cleaner Production*, Vol. 172, 2018, pp. 949 - 959, available at: <http://www.sciencedirect.com/science/article/pii/S0959652617324654>
- [209] Bellman, R., “On a routing problem”, *Quarterly of Applied Mathematics*, Vol. 16, 1958, pp. 87–90.
- [210] Johnson, D. B., “Efficient algorithms for shortest paths in sparse networks”, *Journal of the ACM*, Vol. 24, No. 1, Jan. 1977, pp. 1–13, available at: <http://doi.acm.org/10.1145/321992.321993>
- [211] Xu, Z., Elomri, A., Pokharel, S., Mutlu, F., “A model for capacitated green vehicle routing problem with the time-varying vehicle speed and soft time windows”, *Computers & Industrial Engineering*, Vol. 137, 2019, pp. 106011.
- [212] Ehmke, J. F., Campbell, A. M., Thomas, B. W., “Data-driven approaches for emissions-minimized paths in urban areas”, *Computers and Operations Research*, Vol. 67, 2016, pp. 34–47.
- [213] Ehmke, J. F., Campbell, A. M., Thomas, B. W., “Vehicle routing to minimize time-dependent emissions in urban areas”, *European Journal of Operational Research*, Vol. 251, No. 2, 2016, pp. 478–494.
- [214] Alvarez, P., Lerga, I., Serrano-Hernandez, A., Faulin, J., “The impact of traffic congestion when optimising delivery routes in real time. A case study in Spain”, *International Journal of Logistics Research and Applications*, Vol. 21, No. 5, 2018, pp. 529–541.

List of Figures

2.1. Examples of TSP instance with 15 customers	8
2.2. Examples of TSP instance with 100 customers	9
2.3. Examples of EVRPTW instances	15
2.4. C101: EVRPTW-FR, BKS	16
2.5. C202-15: ETSPTW-FR, optimal	17
2.6. C101: EVRPTW-PR, BKS	18
2.7. R101: EVRPTWDCS-PR, BKS	22
2.8. R101: TD-VRPTW, BKSs	25
3.1. Optimization methods	32
3.2. Examples of Sweep algorithm	33
3.3. Clark & Wright savings algorithm	34
3.4. Examples of Clark & Wright savings algorithm	34
3.5. Example of nearest neighbor heuristic - C101: EVRPTW-FR	35
3.6. Intra relocate operator	37
3.7. Intra exchange operator	37
3.8. Or-Opt operator	38
3.9. 2-Opt operator	39
3.10. Intra station in operator	39
3.11. Inter relocate operator	40
3.12. Inter exchange operator	41
3.13. Inter cross exchange operator	41
3.14. Inter 2-Opt* operator	42
3.15. Genetic algorithm	43
3.16. Ant colony example [132]	44
3.17. Particle swarm optimization solution space [134]	45
3.18. Simulated annealing cooling process [137]	45
3.19. Variable neighborhood search	47
3.20. LNS example: C101 - EVRPTW-FR	49

3.21. Roulette wheel selection	50
3.22. Difference between infeasible and feasible search strategy	52
3.23. Move evaluations - VRPTW	54
3.24. Move evaluations - EVRPTW-FR	56
4.1. Initial solution constructed by PGTONNH - C101	64
4.2. Penalty function value in search process - R101 - EVRPTW-FR	65
4.3. Computation of time window violations in a classical way in VRPTW	66
4.4. Strategy of traveling back to the latest feasible point in time in VRPTW	67
4.5. Forward shifting rules - EVRPTW-PR	72
4.6. Backward shifting rules - EVRPTW-PR	76
4.7. Example of route for determining charging schedule	89
4.8. Worst removal determinism factor	92
4.9. Example of worst removal operator	93
4.10. Example of related removal operator	95
4.11. Example of Shaw removal operator	96
4.12. Example of CS vicinity operator	97
4.13. Example of sequential insertion operator	100
4.14. Example of zone removal operator	101
4.15. Example of greedy insertion operator	103
4.16. Example of regret insertion operator	104
4.17. Example of operator probabilities: C101	105
4.18. Route removal function	106
4.19. Random and greedy route removal operators	107
4.20. Search tree in ESPPRC	117
4.21. Maximum possible additional recharging time T_i^{max} in relation to earliest service start time T_i^t	121
4.22. Optimal EVRPTW-PR solution for instance C101-5	122
4.23. Example of tree with labels	123
4.24. Nearest neighbors example	129
4.25. Nearest CSs between users	131
4.26. The most important classes in the implementation	132
5.1. Speed and travel time profile	158
6.1. Observed delivery problem	174
6.2. Demand per customers in the post delivery problem	174
6.3. Road network in the City of Zagreb	175
6.4. VRPTW solution - set map speeds	179

6.5. Speed profiles	180
6.6. Spatio-temporal route analysis in different seasons	181
6.7. VRPTW solution with average daily speeds	182
6.8. TD-VRPTW solution with link speed profiles - travel time	184
6.9. TD-VRPTW solution with link speed profiles - total time	185
6.10. Post delivery problem with CSs	186
6.11. Average energy shares in dependence of average trip speed, [205]	188
6.12. Speed and acceleration profiles	189
6.13. Copernicus elevation model	190
6.14. Link grades for the road network in the City of Zagreb	191
6.15. Example of energy consumption profile	191
6.16. EVRPTW-FR solution with link speeds - total distance traveled	193
6.17. EVRPTW-PR solution with link speeds - total distance traveled	193
6.18. EVRPTW-FR solution with link speeds - energy consumption	196
6.19. Congestion zones	197
6.20. Clustered TTIs	198

List of Tables

2.1. Notation used in EVRP problems	12
2.2. EVRPTW instances - group values	16
2.3. Figliozzi speeds in time buckets	24
4.1. Forward variables - EVRPTW-PR	70
4.2. Backward variables - EVRPTW-PR	75
4.3. Forward variables - EVRPTW-FR	78
4.4. New forward and backward variables - EVRPTWDCS-FR	80
4.5. Forward variables - EVRPTWDCS-PR	84
4.6. Comparison of route removal operators	113
4.7. Order of LS operators	115
4.8. Removal of LS operators	116
4.9. REFs - EVRPTW-PR variables	119
4.10. Comparison between DP and MILP solver	127
4.11. Infeasible arcs	128
4.12. Tuned parameters	133
4.13. Solving small instances with CPLEX	135
4.14. Comparison between MILP program solved by CPLEX and HALNS method, on small EVRPTW instances	138
4.15. EVRPTW-FR distance - comparison between HALNS and KESK-ALNS	140
4.16. EVRPTW-PR distance - results	142
4.17. EVRPTW-PR total time - results	144
4.18. EVRPTW-FR distance - results	146
4.19. EVRPTW-FR total time - results	147
4.20. Charger type values in EVRPTW with different CSs	148
4.21. EVRPTWDCS-FR distance - results	149
4.22. EVRPTWDCS-FR recharging cost - results	150
4.23. EVRPTWDCS-FR total time - results	152
4.24. EVRPTWDCS-PR recharging cost - results	154

5.1. Variables in TD-VRPTW	166
5.2. VRPTW results - distance	168
5.3. TD-VRPTW results - travel time	169
5.4. TD-VRPTW results - total time	170
5.5. TD-EVRPTW-FR results - travel time	171
5.6. TD-EVRPTW-FR total time and TD-EVRPTWDCS-FR recharging cost - results	172
6.1. VRPTW real example with set map speeds - results	178
6.2. VRPTW real example with average daily speeds - results	182
6.3. TD-VRPTW real example with link speed profiles - results	184
6.4. CSs in the City of Zagreb	186
6.5. Specifications of Nissan Leaf 2014	189
6.6. Accelerations per speed intervals for Nissan Leaf 2014, [205]	189
6.7. TD-/EVRPTW-DCS/PR/FR real example - results	192
6.8. EVRPTW-PR/FR with energy consumption minimization	195
6.9. Pre-processing computation time [h] in time-dependent routing operations . . .	197
6.10. Mean TTI values	198
1. TDVRPTW - A1 - travel time - results	243
2. TDVRPTW - A2 - travel time - results	244
3. TDVRPTW - A3 - travel time - results	245
4. TDVRPTW - B1 - travel time - results	246
5. TDVRPTW - B2 - travel time - results	247
6. TDVRPTW - B3 - travel time - results	248
7. TDVRPTW - C1 - travel time - results	249
8. TDVRPTW - C2 - travel time - results	250
9. TDVRPTW - C3 - travel time - results	251
10. TDVRPTW - D1 - travel time - results	252
11. TDVRPTW - D2 - travel time - results	253
12. TDVRPTW - D3 - travel time - results	254
13. TD-VRPTW - A1 - total time - results	255
14. TD-VRPTW - B1 - total time - results	256
15. TD-VRPTW - C1 - total time - results	257
16. TD-VRPTW - D1 - total time - results	258
17. TD-EVRPTW - A1 - travel time - results	259
18. TD-EVRPTW - A2 - travel time - results	260
19. TD-EVRPTW - A3 - travel time - results	261
20. TD-EVRPTW - B1 - travel time - results	262

21.	TD-EVRPTW - B2 - travel time - results	263
22.	TD-EVRPTW - B3 - travel time - results	264
23.	TD-EVRPTW - C1 - travel time - results	265
24.	TD-EVRPTW - C2 - travel time - results	266
25.	TD-EVRPTW - C3 - travel time - results	267
26.	TD-EVRPTW - D1 - travel time - results	268
27.	TD-EVRPTW - D2 - travel time - results	269
28.	TD-EVRPTW - D3 - travel time - results	270
29.	TD-EVRPTW-FR - A1 - total time - results	271
30.	TD-EVRPTW-FR - B1 - total time - results	272
31.	TD-EVRPTW-FR - C1 - total time - results	273
32.	TD-EVRPTW-FR - D1 - total time - results	274
33.	TD-EVRPTWDCS-FR - A1 - recharging cost - results	275
34.	TD-EVRPTWDCS-FR - B1 - recharging cost - results	276
35.	TD-EVRPTWDCS-FR - C1 - recharging cost - results	277
36.	TD-EVRPTWDCS-FR - D1 - recharging cost - results	278

List of Algorithms

3.1. Iterated local search [127]	47
3.2. Adaptive large neighborhood search [32]	50
4.1. Hybrid adaptive large neighborhood search method	59
4.2. Pseudo-greedy time-oriented nearest neighbor heuristic	61
4.3. Multiple best station insertion	63
4.4. Worst removal operator	91
4.5. Related removal operator	94
4.6. CS vicinity	97
4.7. Sequential insertion operator	98
4.8. Greedy insertion operator	102
4.9. Regret insertion operator	104
4.10. ALNS based route removal operator	109
4.11. Repeated ALNS based route removal operator	110
4.12. Ejection pool operator	112
4.13. Dynamic programming - ESPPRC	118
5.1. Computation of forward travel times	159
5.2. Computation of backward travel times	165
6.1. Dijkstra algorithm	177
6.2. Time-dependent Dijkstra algorithm	183
6.3. Bellman-Ford algorithm	195

Nomenclature

AC Ant colony algorithm

AFV Alternative fuel vehicle

ALNS Adaptive large neighborhood search

ALNS-RR Adaptive large neighborhood search based route removal

BEV Battery electric vehicle

BKS Best-known solution

BSS Battery swap station

CC-CV Constant current, constant voltage charging process

CS Charging station

CVRP Capacitated vehicle routing problem

CWS Clark & Wright savings algorithm

DARP-EV Dial-a-ride problem with electric vehicles and battery swapping stations

DP Dynamic programming

E2-EVRP Two-echelon electric vehicle routing problem

EFV-CSP Electric freight vehicle charge scheduling problem

ELRP Electric location routing problem

ELRPTW-PR Electric Location Routing Problem with Time Windows and Partial Recharging

ELRPTWPR Electric location routing problem with time windows and partial recharges

ESPP Energy shortest path problem

ESPPRC Elementary shortest path problem with resource constraints

ETSP Electric traveling salesman problem

ETSPTW-FR Electric traveling salesman problem with time windows and full recharge

EV Electric vehicle

EVRP Electric vehicle routing problem

EVRP-CTVTT Electric vehicle routing problem with charging time and variable travel time

EVRP-NL Electric vehicle routing problem with nonlinear charging functions

EVRPTW Electric vehicle routing problem with time windows

EVRPTW-FC Electric vehicle routing problem with time windows and fast charging

EVRPTW-FR Electric vehicle routing problem with time windows and full recharge

EVRPTWDCS-FR Electric vehicle routing problem with time windows, different charging stations and full recharge

EVRPTWDCS-PR Electric vehicle routing problem with time windows, different charging stations and partial recharge

EVRPTWMF Electric vehicle routing problem with time windows and mixed fleet

EVRPTWPR Electric vehicle routing problem with time windows with partial recharging

FCD Floating car data

FIFO First-in first-out

FRVCP Fixed-route vehicle-charging problem

FRVRP Fixed-route vehicle-refueling problem

GA Genetic algorithm

GHG Greenhouse gas

GNSS Global navigation satellite system

GVRP Green vehicle routing problem

HALNS Hybrid adaptive large neighborhood search

HEV Hybrid electric vehicle

HVRP Hybrid vehicle routing problem

ICEV Internal combustion engine vehicle

ILS Iterated local search

LDM Longitudinal dynamics model

LIFO Last-in first-Out

LNS Large neighborhood search

LRP Location routing problem

LS Local search

MBSI Multiple best station insertion

MDEVLRTW Multi-depot electric vehicle location routing problem with time windows

MDVRP Multi-depot vehicle routing problem

MFVRP Mixed fleet vehicle routing problem

MILP Mixed integer linear program

MIP Mixed integer program

NNH Nearest neighbor heuristic

PGTONNH Pseudo-greedy time oriented nearest neighbor heuristic

PHEV Plug-in hybrid electric vehicle

PRP Pollution routing problem

PSO Particle swarm optimization

REF Resource extension function

RELRP Robust electric location routing problem

REP-ALNS-RR Repeated adaptive large neighborhood search route removal

RWS Roulette wheel selection

SA Simulated annealing

SoC State of charge

TD-EVRPTW Time-dependent electric vehicle routing problem with time windows

TD-EVRPTW-FR Time-dependent electric vehicle routing problem with time windows and full recharge

TD-EVRPTW-PR Time-dependent electric vehicle routing problem with time windows and partial recharge

TD-EVRPTWDCS-FR Time-dependent electric vehicle routing problem with time windows, different charging stations and full recharge

TD-SPP Time-dependent shortest path problem

TD-VRP Time-dependent vehicle routing problem

TD-VRPTW Time-dependent vehicle routing problem with time windows

TDD Time-dependent Dijkstra

TS Tabu search

TSP Traveling salesman problem

TTI Travel time index

VNS Variable neighborhood search

VRP Vehicle routing problem

VRPPD Vehicle routing problem with pickup and delivery

VRPTW Vehicle routing problem with time windows

Appendix

A List of all thesis contributions

- New Pseudo-Greedy Time-Oriented Nearest Neighbor Heuristic for the creation of the initial EVRPTW solution
- Slightly modified penalty functions for EVRPTW-PR, which has not been presented in the literature
- Four new variables in penalty functions for EVRPTW-PR that are used to determine the charging amount
- Slightly modified penalty functions for EVRPTW-FR, with the strategy for the evaluation in $\mathcal{O}(1)$
- Introducing a new variant of the EVRP problem - the EVRPTWDCS-FR, with proposed penalty functions and the strategy for the evaluation in $\mathcal{O}(1)$
- New penalty functions for EVRPTWDCS-PR with the strategy for the evaluation in $\mathcal{O}(1)$ if chargers are compatible and $\mathcal{O}(n^2)$ (in practice) if chargers are incompatible
- New ALNS based route removal operator
- Comparing different route removal operators for solving EVRPTW variants
- New local search operator for problems with different charger types
- New resource extension functions in DP for EVRPTWDCS-FR
- Improvement in the MILP formulation for the optimal CS placement in EVRPTWDCS-PR
- Analyzing the impact of virtual CS number in MILP formulations for EVRPTW-FR, EVRPTW-PR, and EVRPTWDCS-PR
- Solving EVRPTW-FR, EVRPTW-PR, EVRPTWDCS-FR, and EVRPTWDCS-PR problems using the HALNS method
- Possibly 44 new BKSs on benchmark instances for EVRPTW-PR - the actual number is lower, but the direct comparison could not be made
- Minimization of the total routing time in EVRPTW-FR, EVRPTW-PR, EVRPTWDCS-FR, and EVRPTWDCS-PR problems
- 9 new BKSs on benchmark instances for EVRPTW-FR

- 16 new BKSs on benchmark instances for EVRPTWDCS-PR
- MIP formulations for TD-EVRPTW-FR, TD-EVRPTW-PR, and TD-EVRPTWDCS-FR problems
- MILP formulation for TD-EVRPTW-FR problem
- Backward travel time computation in time-dependent problems
- New penalty functions and concatenation operators for TD-VRPTW problem
- Solving VRPTW and TD-VRPTW problem instances using the HALNS method
- Outperforming procedures applied to solve the TD-VRPTW problem in the literature and providing several BKSs on each of the 12 configuration types of TD-VRPTW benchmark instances
- Developing energy consumption model for routing purpose that includes data derived from GNSS data and digital elevation model
- Using the HALNS method to solve the adapted real-world delivery problems modeled as VRPTW, TD-VRPTW, EVRPTW, and TD-EVRPTW
- Solving a real world EVRPTW problem with the minimization of energy consumption
- Shortest energy paths for BEVs on the road network of the City of Zagreb

B TD-VRPTW travel time

Table 1: TDVRPTW - A1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	685.24	685.24	828.94	828.94	9685.24	9685.24	0.19
C102	10	10	683.26	683.26	835.46	835.46	9683.26	9683.26	0.72
C103	10	10	681.88	681.88	835.46	835.46	9750.96	9750.96	2.17
C104	10	10	679.94	679.94	833.88	833.88	9816.21	9816.21	4.32
C105	10	10	682.83	682.83	833.01	833.01	9763.24	9763.24	0.37
C106	10	10	685.24	685.24	828.94	828.94	9685.24	9685.24	0.44
C107	10	10	682.83	682.83	833.01	833.01	9697.6	9697.6	0.88
C108	10	10	680.74	680.74	833.03	833.03	9684.38	9684.38	1.4
C109	10	10	680.74	680.74	833.03	833.03	9680.74	9680.74	2.48
C201	3	3	502.9	502.9	591.56	591.56	9502.9	9502.9	0.44
C202	3	3	502.9	502.9	591.56	591.56	9502.9	9502.9	2.13
C203	3	3	502.05	502.05	591.17	591.17	9539.88	9539.88	4.32
C204	3	3	500.84	500.84	590.97	590.97	8550.76	9500.84	7.86
C205	3	3	500.34	500.34	588.88	588.88	9500.34	9500.34	1.13
C206	3	3	499.49	499.49	588.49	588.49	9499.49	9499.49	1.72
C207	3	3	498.33	498.33	588.32	588.32	9500.54	9500.54	2.1
C208	3	3	499.49	499.49	588.49	588.49	9499.49	9499.49	2.31
R101	18	18	1324.59	1312.04	1606.85	1582.12	3019.26	3370.77	1.85
R102	16	16	1185.05	1185.05	1432.68	1432.68	2753.78	3059.76	4.16
R103	13	13	1020.18	1020.18	1240.92	1240.92	2568.37	2568.37	7.4
R104	9	9	805.78	801.51	977.24	975.32	1724.12	1935.73	8.91
R105	12	12	1096.05	1084.71	1378.03	1370.65	2084.81	2306.89	2.44
R106	10.9	10	1012.42	1158.35	1261.94	1408.43	1920.04	2091.51	5.82
R107	9	9	925.45	908.92	1123.51	1102.49	1770.81	1960.62	6.22
R108	8.6	8	775.51	792.34	939.39	948.59	1644.68	1798.77	8.9
R109	10	10	943.06	930.22	1172.31	1162.84	1813.47	2008.31	4.96
R110	9	9	918.85	898.65	1116.19	1098	1767.69	1943.07	5.18
R111	9	9	886.68	872.9	1085.29	1068.02	1756.38	1942.09	5.43
R112	9	9	759.72	754.75	939.79	935.21	1668.01	1842.62	9.26
R201	3	3	1116.14	1115.8	1387.1	1382.83	2395.32	2661.46	4.92
R202	3	3	919.7	919.7	1141.19	1141.19	2451.46	2723.84	12.68
R203	2	2	905.73	901.7	1136.26	1134.1	1744.47	1931.76	15.87
R204	2	2	648.95	637.12	827.59	813.12	1660.66	1897.78	22.18
R205	3	3	754.95	747.37	1023.95	1020.64	2170.52	2404.95	10.46
R206	2	2	799.88	773.15	1009.73	983.01	1686.88	1844.31	10.59
R207	2	2	697.96	687.29	891.14	886.85	1652.84	1848.31	14.99
R208	2	2	573.35	566.64	747.26	742.7	1561.85	1709.08	17.73
R209	2.1	2	858.95	851.51	1077.05	1060.62	1742.95	1860.48	8.96
R210	2	2	884.16	872.28	1094.92	1080.35	1724.68	1901.54	10.43
R211	2	2	675.37	666.18	864.68	846.25	1562.71	1710.63	17.19
RC101	13.2	13	1324.59	1304.23	1653.55	1620.11	2345.13	2557.91	2.03
RC102	11	11	1190.53	1188.38	1458.1	1448.92	2123.3	2368.31	2.88
RC103	10	10	997.86	991.06	1225.56	1213.43	1923.83	2142.61	6.21
RC104	9	9	917.94	912.05	1111.93	1101.93	1798.45	2005.38	5.22
RC105	13	13	1204.23	1180.03	1483.01	1448.09	2364.09	2656.73	3.37
RC106	11	11	1045.17	1043.52	1323.07	1320.51	2013.35	2247.98	4.04
RC107	10	10	982.69	969.53	1223.36	1215	1885.39	2079.54	7.01
RC108	9	9	932.81	899.44	1137.94	1106.94	1791.42	1969.88	3.36
RC201	3	3	1259.92	1250.11	1573.55	1566.25	2343.87	2604.29	4.53
RC202	3	3	1015.96	1013.46	1282.59	1302.55	2348.64	2609.6	12.44
RC203	3	3	811.28	800.3	1017.51	998.17	2363.03	2665.36	14.98
RC204	2	2	734.67	725.18	927.44	916.09	1686.5	1915.11	16.8
RC205	3	3	1145.56	1137.53	1445.49	1417.19	2409.49	2669.7	9.46
RC206	3	3	852.53	848.99	1138.92	1138.9	2126.74	2326.66	10.2
RC207	3	3	782.97	772.43	1057.33	1036.03	2021.83	2235.27	14.85
RC208	2	2	785.94	758.5	985.19	961.84	1617.44	1767.6	10.05
AVG	6.8	6.77	832.54	828.61	1030.96	1025.77	4295.56	4457.82	6.77
SUM	380.8	379	46622.17	46402.14	57733.75	57443.08	240551.43	249637.83	378.94

Table 2: TDVRPTW - A2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	615.28	615.28	828.94	828.94	9615.28	9615.28	0.19
C102	10	10	607.92	607.92	833.03	833.03	9607.92	9607.92	0.66
C103	10	10	606.61	606.61	833.03	833.03	9676.87	9676.87	1.88
C104	10	10	605.31	605.31	860.76	860.76	9670.56	9670.56	4.87
C105	10	10	608.39	608.39	834.99	834.99	9752.95	9752.95	0.37
C106	10	10	612.37	612.37	831.68	831.68	9685.69	9685.69	0.5
C107	10	10	608.39	608.39	834.99	834.99	9644.72	9644.72	0.96
C108	10	10	605.59	605.59	835.01	835.01	9612.59	9612.59	1.61
C109	10	10	605.59	605.59	835.01	835.01	9605.59	9605.59	2.73
C201	3	3	441.46	441.46	591.56	591.56	9441.46	9441.46	0.43
C202	3	3	441.46	441.46	591.56	591.56	9441.46	9441.46	2.17
C203	3	3	438.92	438.92	592.89	592.89	9503.05	9503.05	4.8
C204	3	3	437.21	437.21	588.05	588.05	9437.21	9437.21	8.54
C205	3	3	439.17	439.17	588.49	588.49	9439.17	9439.17	1.15
C206	3	3	439.17	439.17	588.49	588.49	9439.17	9439.17	1.74
C207	3	3	438.54	438.54	588.32	588.32	9459.43	9459.43	2.13
C208	3	3	439.17	439.17	588.49	588.49	9439.17	9439.17	2.25
R101	15	15	1113.21	1104.57	1608.86	1585.86	2597.15	2876.84	1.98
R102	14	14	964.79	959.9	1374.44	1372.49	2407.48	2673.39	4.24
R103	11.1	11	870.54	868.72	1253.5	1251.4	2019.03	2165.54	6.51
R104	8.4	8	713.14	718.42	1010.1	1018.04	1615.18	1792.96	7.03
R105	11	11	896.51	886.13	1365.18	1355.38	1885.9	2098.52	2.34
R106	9.3	9	843.15	851.28	1227.89	1229.15	1745.8	1939.63	3.72
R107	8.7	8	734.72	763.24	1057.49	1078.47	1637.09	1769.25	5.59
R108	8	8	638.86	635.11	919.6	910.93	1506.4	1674.76	6.85
R109	9.1	9	800.52	790.81	1167.72	1147.73	1674.73	1848.08	3.99
R110	9	9	709.63	703.38	1053.81	1056.74	1634.11	1763.25	6.38
R111	9	9	697.46	695.33	1030.73	1030.25	1629.55	1804.17	8.14
R112	8	8	654.66	645.29	928.96	923.36	1710.83	1710	6.47
R201	3	3	870.6	870.46	1308.53	1307.85	2366.09	2628.98	6.04
R202	3	3	774.56	772.25	1143.69	1130.5	2436.96	2717.11	13.55
R203	2	2	719.29	717.06	1073	1074.35	1678.17	1831.77	16.94
R204	2	2	521.14	515	819.78	797.47	1620.68	1838.31	17.3
R205	2.2	2	751.08	759.95	1149.8	1139.58	1735.79	1765.63	6.42
R206	2	2	636.73	625.65	997.9	991.05	1562.87	1739.53	11.64
R207	2	2	561.32	541.68	891.47	878.82	1523.26	1707.15	16.49
R208	2	2	454.73	448.5	745.67	736.95	1458.84	1614.43	17.68
R209	2	2	645.02	634.3	999.64	982.21	1580.65	1752.6	10.99
R210	2	2	684.47	678.44	1057.65	1051.67	1677.08	1861.48	13.63
R211	2	2	525	515.07	836.01	807.97	1496.97	1630.35	17.16
RC101	12	12	1001.09	1001.09	1516.51	1516.51	2331.15	2331.15	2.61
RC102	10.1	10	1017.12	986.15	1469.26	1414.53	1934.76	2116.36	3.02
RC103	9	9	854.54	839.01	1235.73	1222.96	1729.76	1907.01	1.32
RC104	9	9	761.02	759.5	1135.9	1133.82	1656.25	1829.74	6.45
RC105	10	10	1050.54	1039.9	1491.17	1475.95	1930.87	2146.92	2.95
RC106	10	10	879.21	867.59	1330.05	1302.74	1777.1	1995.75	2.91
RC107	9	9	866.01	836.55	1270.26	1245.34	1739.32	1873.86	2.04
RC108	9	9	721.86	714.43	1077.66	1060.01	1626.92	1809.3	4.21
RC201	3	3	1012.67	1012.15	1506.91	1506.86	2343.87	2604.29	5.69
RC202	3	3	836.85	831.12	1255.31	1257.29	2346.36	2605.14	12.64
RC203	2	2	788.6	788.15	1162.33	1167.65	1699.46	1855.63	14.79
RC204	2	2	589.72	576.41	907.78	860.58	1654.92	1840.99	18.15
RC205	3	3	875.32	874.47	1352.04	1352.31	2397.8	2663.77	10.96
RC206	3	3	680.49	673	1142.94	1135.47	2061.96	2274.48	10.41
RC207	3	3	608.78	603.41	1036.97	1029.42	1968.6	2193.71	15.09
RC208	2	2	598.78	580.76	920.48	895.66	1512.24	1616.93	15.16
AVG	6.39	6.36	694.9	690.62	1019.25	1012.15	4185.43	4309.66	6.72
SUM	357.9	356	38914.28	38674.78	57078.01	56680.61	234384.24	241341.05	376.46

Table 3: TDVRPTW - A3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	569.68	569.68	828.94	828.94	9569.68	9569.68	0.2
C102	10	10	560.03	560.03	833.03	833.03	9560.03	9560.03	0.66
C103	10	10	558.41	558.41	838.04	838.04	9630.44	9630.44	2.07
C104	10	10	551.75	551.75	867.55	867.55	9616.59	9616.59	5.8
C105	10	10	554.7	554.7	834.99	834.99	9710.47	9710.47	0.37
C106	10	10	559.49	559.49	831.68	831.68	9641.86	9641.86	0.49
C107	10	10	554.7	554.7	834.99	834.99	9598.31	9598.31	0.88
C108	10	10	551.34	551.34	835.01	835.01	9560.31	9560.31	1.5
C109	10	10	547.54	547.54	892.27	892.27	9733.99	9733.99	3.06
C201	3	3	404.47	404.47	591.56	591.56	9404.47	9404.47	0.43
C202	3	3	404.47	404.47	591.56	591.56	9404.47	9404.47	2.08
C203	3	3	400.64	400.64	588.88	588.88	9400.64	9400.64	4.33
C204	3	3	398.53	398.24	590.56	587.71	8458.71	9398.24	6.37
C205	3	3	400.11	400.11	588.49	588.49	9400.11	9400.11	1.14
C206	3	3	400.11	400.11	588.49	588.49	9400.11	9400.11	1.74
C207	3	3	402.56	402.56	588.49	588.49	9419.98	9419.98	2.12
C208	3	3	400.11	400.11	588.49	588.49	9400.11	9400.11	2.24
R101	14	14	955.98	955.13	1554.65	1553.84	2430.13	2702.29	1.96
R102	13	13	856.06	852.57	1356.43	1349.29	2499.58	2505.64	4.27
R103	10	10	729.42	727.2	1168.88	1168.01	1859.05	2087.71	7.38
R104	8	8	609.63	604.29	984.73	964.58	1516.37	1659.78	7.1
R105	11	11	731.39	728.24	1313.93	1334.67	1836.8	2041.2	3.38
R106	9	9	698.02	695.14	1198.91	1185.37	1612.97	1799.9	4.41
R107	8	8	646.15	636.49	1056.17	1043.98	1528.88	1673.5	2.97
R108	8	8	553.79	549.94	937.91	946.05	1425.26	1580.11	7.39
R109	9	9	651.01	645.68	1131.08	1122.26	1545.23	1699.01	4.64
R110	8	8	664.83	656.03	1080.19	1071.6	1713.96	1702.11	2.45
R111	8	8	645.2	638	1054.54	1045.27	1539.28	1708.42	3.68
R112	8	8	549.29	546.19	919.17	927.25	1457.6	1601.52	7.42
R201	3	3	758.05	756.96	1290.78	1293.96	2357.65	2619.61	6.63
R202	3	3	693.27	689.67	1167.24	1156.34	2346.4	2594.81	14.06
R203	2	2	617.73	613.82	1078.31	1064.38	1652.39	1856.7	17.31
R204	2	2	445.1	442	815.42	801.23	1617.58	1806.25	19.86
R205	2	2	625.68	611.72	1130.99	1113.37	1708.68	1703.97	7.03
R206	2	2	533.3	520.47	1001.83	1001.82	1515.28	1652.85	13.69
R207	2	2	464.4	458.39	895.41	900.37	1508.9	1632.6	17.93
R208	2	2	379.55	372.46	748.17	736.23	1451.37	1598.02	17.9
R209	2	2	523.61	513.26	977.96	960.72	1508.58	1667.18	10.86
R210	2	2	568.74	561.62	1061.26	1054.78	1656.36	1818.9	13.5
R211	2	2	427.94	423.82	827.53	814.61	1441.75	1596.29	15.4
RC101	11	11	882.82	867.24	1555.01	1525.68	1895.99	2108.59	2.38
RC102	9.9	9	837.68	972.39	1412.83	1512.34	1789.07	1995.39	3.76
RC103	9	9	722.91	719.29	1248.39	1243.48	1803.39	1801.32	4.39
RC104	9	9	638.33	634.33	1140.62	1120.98	1543.79	1704.61	7.04
RC105	10	10	835.21	819.81	1414.98	1399.37	1832.92	2036.3	3.82
RC106	9.2	9	819.55	811.89	1370.22	1345.36	1714.23	1891.27	2.73
RC107	9	9	698.54	677.63	1230.46	1190.38	1794.54	1787.02	2.98
RC108	9	9	599.82	591.8	1093.19	1109.33	1546.98	1729.62	5.8
RC201	3	3	880.97	877.38	1486.33	1469.53	2343.87	2604.29	6.03
RC202	3	3	736.25	730.22	1244.98	1232.81	2343.91	2605.55	12.79
RC203	2	2	692.08	689.04	1147.15	1139	1680.59	1889.31	16.3
RC204	2	2	495.41	493.33	911.13	907.82	1623.13	1798.06	18.46
RC205	3	3	753.48	747.86	1347.73	1337.22	2389.34	2654.82	11.6
RC206	3	3	559.94	553.77	1140.96	1153.67	2029.64	2286.83	10.31
RC207	2	2	670.62	619.53	1177.84	1115.46	1540.83	1682.92	9.4
RC208	2	2	496.06	483.56	917.01	897.75	1452.03	1582.11	15.09
AVG	6.22	6.2	604.76	601.9	1016.13	1011.08	4106.51	4237.79	6.78
SUM	348.1	347	33866.45	33706.51	56903.34	56620.33	229964.58	237316.19	379.58

Table 4: TDVRPTW - B1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	679.05	679.05	829.7	829.7	9851.43	9851.43	0.18
C102	10	10	669.25	669.25	841.3	841.3	10078.54	10078.54	0.75
C103	10	10	657.57	657.57	864.89	864.89	10599.34	10599.34	2.47
C104	10	10	618.23	618.23	856.52	856.52	11156.25	11156.25	3.49
C105	10	10	684.17	684.17	828.94	828.94	9773.81	9773.81	0.26
C106	10	10	670.99	670.99	833.79	833.79	9850.77	9850.77	0.43
C107	10	10	674.43	674.43	835.28	835.28	9761.07	9761.07	0.95
C108	10	10	671.01	671.01	845.84	845.84	9924.35	9924.35	1.39
C109	10	10	666.89	666.89	835.26	835.26	9752.78	9752.78	2.36
C201	3	3	489.88	489.88	591.56	591.56	9500.85	9500.85	0.41
C202	3	3	489.88	489.88	591.56	591.56	9500.85	9500.85	2.08
C203	3	3	487.33	487.33	588.88	588.88	9494.55	9494.55	4.81
C204	3	3	485.32	485.32	588.72	588.72	9530.31	9530.31	7.6
C205	3	3	486.61	486.61	589.89	589.89	9497.58	9497.58	1.13
C206	3	3	485.77	485.77	591.65	591.65	9576.6	9576.6	1.72
C207	3	3	484.1	484.1	589.68	589.68	9586.44	9586.44	2.06
C208	3	3	480.8	480.8	589.72	589.72	9662.3	9662.3	2.4
R101	18	18	1365.71	1363.18	1636.11	1633.6	3017.16	3384.72	1.68
R102	17	17	1209.38	1205.79	1496.05	1485.49	2992.24	3286.31	3.16
R103	13.5	13	982.88	1016.01	1247.38	1284.76	2487.05	2720.37	6.89
R104	9	9	780.77	770.9	981.42	968.61	1768.41	1981.95	7.82
R105	13.4	13	1184.45	1195.9	1409.74	1428.4	2277.55	2534.82	2.25
R106	11.9	11	1025.91	1114.64	1253.46	1353.68	2120.35	2342.06	6.38
R107	9.9	9	866.96	955.12	1078.25	1186.29	1884.27	2126.05	8.81
R108	9	9	754.95	748.43	941.74	930.25	1711.33	1957.84	11.14
R109	11	11	981.53	976.28	1187.03	1186.19	1951.77	2171.98	4.19
R110	10	10	915.46	908.26	1121.34	1114.25	1890.45	2071.54	4.98
R111	10	10	901.93	892.96	1103.28	1091.03	1914.96	2127.44	5.43
R112	9	9	787.36	777.06	975.57	958.89	1715.49	1904.18	7.97
R201	4	4	1054.88	1054.73	1255.79	1257.3	3147.85	3497.61	4.22
R202	3	3	1005.25	1002.55	1198.59	1195.45	2449.61	2701.67	11.55
R203	3	3	747.99	746.23	942.48	939.31	2509.48	2791.87	18.37
R204	2	2	699.72	683.24	855.59	834.46	1776.65	1981.75	23.3
R205	3	3	862.14	851.7	1035.66	1033.69	2277.51	2520.05	10.02
R206	2	2	871.39	854.28	1058.32	1044.46	1752.94	1961.17	12.84
R207	2	2	729.61	716.14	899.63	881.54	1733.42	1895.15	17.61
R208	2	2	601.37	590.37	761.44	748.44	1751.86	1934.92	16.96
R209	3	3	798.57	791.88	953.09	936.26	2109.2	2384.31	14.66
R210	3	3	789.2	783.49	974.58	974.43	2434.39	2709.62	14.94
R211	2	2	735.63	729.84	889.11	877.69	1629.85	1815.23	20.21
RC101	14.7	14	1377.91	1396.64	1675.83	1697.73	2637.15	2871.37	2.09
RC102	12	12	1197.12	1176.04	1498.82	1480.46	2310.64	2586.71	3.09
RC103	10	10	1017.29	994.67	1290.58	1276.4	2000.71	2251.46	3.58
RC104	9	9	911.78	876.14	1167.47	1130.41	1839.15	2048.48	2.85
RC105	13.7	13	1279.52	1313.19	1583.78	1624.04	2530.29	2704.76	2.85
RC106	11.7	11	1135.97	1139.6	1401.86	1397.77	2186.15	2344.45	2.67
RC107	10.5	10	1010.49	1019.15	1272.83	1294.65	2020.5	2211.86	3.56
RC108	10	10	906.18	896.83	1140.96	1132.86	1954.94	2147.89	6.41
RC201	4	4	1189.45	1188.43	1393.73	1376.69	3026.01	3308.19	4.2
RC202	3	3	1177.09	1150.01	1395.53	1337.89	2370.05	2626.87	9.56
RC203	3	3	859.2	851.42	1073.01	1042	2412.07	2615.14	18.14
RC204	2	2	792.74	762.6	955.25	924.02	1709.57	1892.32	19.71
RC205	4	4	1106.96	1105.24	1303.88	1317.26	3087.49	3413.79	8.37
RC206	3	3	981.26	976.92	1147.03	1138.3	2205.89	2435.93	10.54
RC207	3	3	939.63	938.19	1084.52	1082.76	2139.7	2366.44	16.53
RC208	2	2	837.25	817.96	1031.16	1007.49	1669.31	1829.71	13.31
AVG	7.15	7.05	843.82	843.09	1035.09	1033.9	4508.95	4670.64	7.1
SUM	400.3	395	47254.16	47213.29	57965.07	57898.38	252501.23	261555.8	397.33

Table 5: TDVRPTW - B2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	607.37	607.37	829.7	829.7	9816.72	9816.72	0.18
C102	10	10	583.44	583.44	868.85	868.85	10448.58	10448.58	0.79
C103	10	10	567.28	567.28	877.96	877.96	10695.76	10695.76	3.21
C104	10	10	515.26	515.26	868.35	868.35	11355.56	11355.56	4.02
C105	10	10	613.4	613.4	831.3	831.3	9798.45	9798.45	0.25
C106	10	10	592.43	592.43	853.13	853.13	9902.15	9902.15	0.42
C107	10	10	601.94	601.94	835.28	835.28	9720.27	9720.27	0.99
C108	10	10	582.72	582.72	882.06	882.06	10062.93	10062.93	1.48
C109	10	10	581.52	581.52	848.99	848.99	9708.7	9708.7	2.25
C201	3	3	425.61	425.61	591.56	591.56	9440.38	9440.38	0.4
C202	3	3	425.61	425.61	591.56	591.56	9440.38	9440.38	1.99
C203	3	3	422.93	422.93	596.65	596.65	9747.61	9747.61	4.59
C204	3	3	420.78	420.78	593.98	593.98	9718.42	9718.42	7.92
C205	3	3	422.31	422.31	602.44	602.44	9504.56	9504.56	1.4
C206	3	3	421.37	421.37	591.65	591.65	9533.98	9533.98	1.75
C207	3	3	418.91	418.91	593.9	593.9	9650.87	9650.87	1.97
C208	3	3	417.28	417.28	589.72	589.72	9618.55	9618.55	2.48
R101	17.9	17	1154.38	1213.58	1621.85	1692.48	2977.83	3300.72	2.03
R102	16	16	977.17	974.08	1441.49	1434.08	2701.13	2978.78	3.59
R103	12	12	805.64	802.87	1222.36	1222.13	2161.59	2374.79	6.97
R104	8.5	8	654.01	670.94	987.94	1013.01	1687.04	1760.54	7.6
R105	13	13	962.62	959.98	1363.53	1363.74	2161.95	2416.72	2.16
R106	11	11	847.76	843	1242.21	1245.62	1921.31	2124.7	3.96
R107	9	9	750.74	736.78	1108.98	1092.21	1746.7	1961.81	5.38
R108	8	8	629.66	625.32	943.44	942.11	1554.2	1724.1	3.73
R109	10.1	10	856.54	848.69	1228.17	1225.61	1819.74	2007.68	3.2
R110	9	9	778.41	769.33	1128.97	1116.56	1731.6	1925.11	4.5
R111	9	9	755.81	742.5	1107.33	1093.72	1726.19	1904.17	5
R112	8.5	8	657.86	665.64	974.79	991.36	1594.75	1732.08	6.08
R201	3	3	1036.28	1018.08	1455.46	1422.87	2403.54	2672.56	4.46
R202	3	3	840.6	837.4	1188.72	1187.6	2416.33	2667.89	11.81
R203	2	2	771.81	766.37	1113.42	1109.09	1776.51	1977.92	11.98
R204	2	2	563.42	550.05	849.33	826.86	1768.49	1964.18	21.63
R205	3	3	726.75	717.54	1063.61	1068.3	2241.13	2513.62	10.62
R206	2	2	705.1	699.69	1024.86	1016.86	1730.51	1922.08	13.75
R207	2	2	611.42	598.25	931.13	902.55	1752.61	1927.61	19.6
R208	2	2	495.75	486.81	772.73	779.76	1772.81	1985.67	18.93
R209	2	2	762.75	747.54	1079.08	1050.47	1626.64	1786.67	10.97
R210	2	2	790.04	768.79	1131.19	1105.31	1716.65	1917.43	11.03
R211	2	2	608.92	598.23	879.82	884.5	1533.58	1692.73	22.24
RC101	13.9	13	1151.36	1204	1650.94	1711.49	2438.28	2734.39	2.19
RC102	11.1	11	1013.6	1009.06	1488.04	1479.61	2125.66	2346.4	3.48
RC103	9	9	882.13	870.85	1319.54	1290.9	1800.09	2014.65	2.42
RC104	9	9	734.72	718.92	1127.72	1108.99	1786.43	2005.89	5.81
RC105	12.8	12	1022.99	1072.63	1519.88	1553.75	2328.57	2497.96	3.94
RC106	11	11	939	935.38	1357.65	1365.53	2032.16	2268.34	3.24
RC107	10	10	839.91	832.42	1250.72	1224.5	1899.27	2070.55	5.23
RC108	9	9	787.66	769.02	1176.58	1150.21	1774.24	1960.9	2.82
RC201	4	4	1004.79	1004.51	1411.67	1411.27	2917.33	3198.03	4.42
RC202	3	3	978.09	970.48	1359.86	1336.98	2365.54	2613.5	12
RC203	2	2	851.08	850.95	1232.24	1232.87	1713.62	1904.54	12.53
RC204	2	2	638.81	625.67	923.11	906.48	1707.59	1896.39	19.93
RC205	3	3	1076.81	1067.55	1514.02	1514.01	2412.1	2672.26	9.68
RC206	3	3	850.76	841.83	1187.36	1158.86	2116.27	2282.94	10.32
RC207	3	3	793.28	786.38	1129.33	1147.14	2078.77	2309.19	16.48
RC208	2	2	686.08	670.5	974.17	976.56	1591.07	1755.92	17.55
AVG	6.76	6.7	725.26	723.07	1052.33	1050.05	4388.82	4534.56	6.77
SUM	378.8	375	40614.67	40491.77	58930.32	58803.03	245773.69	253935.28	379.35

Table 6: TDVRPTW - B3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	560.93	560.93	829.7	829.7	8818.02	9797.8	0.17
C102	10	10	519.59	519.59	946.71	946.71	10770.9	10770.9	0.83
C103	10	10	498.79	498.36	929.8	929.1	10059.33	11094.99	3.01
C104	10	10	440.55	440.55	869.75	869.75	11328.96	11328.96	3.93
C105	10	10	559.52	559.52	841.18	841.18	9806.2	9806.2	0.34
C106	10	10	539.19	539.19	854.88	854.88	9875.6	9875.6	0.46
C107	10	10	547.97	547.94	945.45	944.92	9954.84	9954.54	0.83
C108	10	10	524.48	524.48	883.73	883.73	10030.83	10030.83	1.2
C109	10	10	511.59	510.93	982.86	979.36	8743.38	9777.79	2.59
C201	3	3	389.2	389.2	591.56	591.56	9407.02	9407.02	0.4
C202	3	3	389.2	389.2	591.56	591.56	9407.02	9407.02	1.89
C203	3	3	383.5	383.5	601.97	601.97	9784.32	9784.32	4.8
C204	3	3	380.7	380.23	627.68	633.68	8980.24	10062.47	8.09
C205	3	3	382.92	382.92	602.44	602.44	9476.63	9476.63	1.34
C206	3	3	384.24	384.24	591.65	591.65	9507.85	9507.85	1.79
C207	3	3	380.82	380.82	593.9	593.9	9625.07	9625.07	2.02
C208	3	3	379.23	379.23	624.42	624.42	9872.32	9872.32	2.61
R101	17	17	1049.16	1048.79	1627.45	1627.77	2867.18	3197.15	1.79
R102	16	16	855.89	853.81	1437.05	1430.94	2706.9	2997.17	3.7
R103	12	12	693.5	687.28	1219.14	1209.38	2105.51	2356.98	7.94
R104	8	8	587.64	571.66	1030.86	1017.24	1571.03	1750.46	5.08
R105	12.6	12	864.37	877.09	1361.38	1365.69	2102.5	2234.14	2.56
R106	10	10	751.05	750.64	1243.94	1245.28	1801.48	1993.83	4.43
R107	8	8	669.01	652.15	1126.37	1104.45	1576.1	1754.22	3.1
R108	8	8	544.08	537.98	940.72	930.21	1546.69	1717.95	6.1
R109	10	10	737.57	726.65	1210.19	1216.2	1751.84	1922.43	3.96
R110	9	9	680.97	666.73	1107.82	1093.81	1679.26	1871.36	5.25
R111	9	9	653.62	646.91	1087.93	1090.41	1717.66	1931.69	5.38
R112	8	8	597.54	589.49	1006.12	979.56	1528.54	1686.48	2.74
R201	3	3	907.35	905.68	1425.69	1427.64	2398.16	2664.62	5.32
R202	3	3	749.8	748.95	1172.09	1169.7	2399.65	2669.81	14.8
R203	2	2	669.19	664.5	1113.99	1116.16	1762.56	1966.63	13.78
R204	2	2	489.64	476.13	855.3	833.49	1760.94	1956.9	17.47
R205	2	2	760.07	747.1	1212.11	1169.19	1640.29	1869.32	6.8
R206	2	2	625.19	619.73	1033.29	1036.75	1698.22	1898.75	13.62
R207	2	2	528.75	522.19	950.94	920.89	1762.33	1948.94	19.14
R208	2	2	429.22	414.79	789.57	786.98	1775.82	1979.46	17.72
R209	2	2	668.53	662.05	1078.9	1072.49	1604.39	1772.73	10.65
R210	2	2	680.84	663.77	1103.75	1069.54	1715.03	1913.77	13.43
R211	2	2	536.71	523.98	900.14	897.24	1507.59	1634.12	19.38
RC101	13	13	1029.46	1021.93	1628.25	1624.34	2305.51	2552.65	1.96
RC102	11	11	890.79	875.85	1458.45	1444.36	2068.22	2244.53	3.65
RC103	9	9	746.54	735.48	1285.51	1284.76	1772.29	1965.69	2.42
RC104	9	9	625.92	623.16	1121.24	1104.04	1783.43	1974.72	6.06
RC105	11	11	943.05	927.55	1551.15	1509.39	2071.21	2296.73	2.67
RC106	11	11	833.93	824.95	1347.14	1345.53	1989.93	2212.41	2.95
RC107	9.9	9	741.82	828.21	1270.21	1421.74	1840.63	1982.61	5.01
RC108	9	9	675.52	663.68	1150.93	1145.47	1736.76	1904.84	2.46
RC201	3.2	3	1035.6	1068.5	1583.81	1629.81	2704.38	2576.58	5.44
RC202	3	3	880.88	879.23	1372.04	1376.65	2355.84	2624.77	11.73
RC203	2	2	738.89	735.94	1214.77	1222.38	1696.18	1891.88	11.3
RC204	2	2	559.4	544.73	941.8	914.87	1703.97	1900.42	17.79
RC205	3	3	957.24	952.73	1513.47	1487.83	2386.59	2626.62	10.36
RC206	3	3	761.83	756.72	1242.96	1214.54	2084.7	2269.58	10.36
RC207	3	3	693.26	685.48	1156.4	1160.42	2045.68	2275.4	18.73
RC208	2	2	612.25	602.35	1008.11	995.7	1579.32	1748.08	16.21
AVG	6.6	6.57	646.94	643.85	1067.68	1064.35	4295.59	4505.66	6.6
SUM	369.7	368	36228.49	36055.37	59790.22	59603.35	240552.84	252316.73	369.54

Table 7: TDVRPTW - C1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	667.9	667.9	828.94	828.94	9757.54	9757.54	0.19
C102	10	10	667.9	667.9	828.94	828.94	9741.25	9741.25	0.63
C103	10	10	665.86	665.86	828.06	828.06	9756.36	9756.36	2.18
C104	10	10	663.21	663.21	825.11	825.11	9823.53	9823.53	3.71
C105	10	10	667.9	667.9	828.94	828.94	9757.54	9757.54	0.35
C106	10	10	667.9	667.9	828.94	828.94	9757.54	9757.54	0.47
C107	10	10	667.9	667.9	828.94	828.94	9757.54	9757.54	0.89
C108	10	10	667.9	667.9	828.94	828.94	9757.54	9757.54	1.34
C109	10	10	667.9	667.9	828.94	828.94	9757.54	9757.54	2.04
C201	3	3	499.52	499.52	591.56	591.56	9510.49	9510.49	0.43
C202	3	3	499.52	499.52	591.56	591.56	9510.49	9510.49	2.06
C203	3	3	496.58	496.58	588.49	588.49	9539.85	9539.85	4.37
C204	3	3	495.96	495.96	587.71	587.71	9500.33	9500.33	7.42
C205	3	3	496.97	496.97	588.88	588.88	9507.93	9507.93	1.19
C206	3	3	496.58	496.58	588.49	588.49	9507.55	9507.55	1.72
C207	3	3	496.57	496.57	588.32	588.32	9530.65	9530.65	2.11
C208	3	3	496.58	496.58	588.49	588.49	9507.55	9507.55	2.34
R101	18	18	1303.5	1293.54	1607.66	1587.14	3063.85	3368.52	2.01
R102	16	16	1169.74	1169.74	1442.99	1442.99	3077.68	3077.68	4.57
R103	13	13	960.7	960.66	1225.97	1226.17	2295.13	2552.61	7.85
R104	9	9	768.54	765.98	963.44	956.02	1700.1	1884.64	8.96
R105	12	12	1114.92	1101.14	1377.26	1362.87	2151.51	2374.95	2.28
R106	10.9	10	994.38	1054.45	1242.62	1299.45	1983.19	2159.92	5.87
R107	9	9	897.42	880.9	1117.61	1102.23	1777.25	1949.83	4.95
R108	8.4	8	767.31	770.16	955.32	953.55	1633.62	1783.36	6.12
R109	10	10	992.15	963.33	1218.04	1188.64	1900.38	2082.83	3.65
R110	10	10	883.61	877.85	1091.82	1075.02	1827.31	2038.77	6.44
R111	9.8	9	864.84	921.84	1068.31	1140.75	1841.92	2031.53	7.36
R112	9	9	763.03	756.79	954.97	947.28	1655.65	1847.86	9.75
R201	3	3	1111.56	1104.26	1384.43	1382.62	2466.09	2742.74	5.41
R202	3	3	912.62	910.12	1128.55	1127.37	2470.68	2741.28	11.67
R203	2.1	2	881	896.13	1102.44	1122.16	1833.75	1934.89	14.21
R204	2	2	652.45	628.83	829.45	802.08	1693.71	1842.95	18.23
R205	3	3	792.18	784.8	1008.86	995.38	2273.29	2511.17	9.94
R206	2	2	792.68	789.95	994.56	992.74	1682.76	1860.42	13.07
R207	2	2	702.25	697.44	883.86	875.27	1693.27	1867.94	17
R208	2	2	570	561.59	745.49	739.59	1607.62	1819.3	18.6
R209	2	2	891.99	873.23	1094.99	1070.57	1720.92	1888.44	9.95
R210	2	2	874.17	853.2	1083.35	1050.75	1736.11	1963.58	12.89
R211	2	2	685.7	676.46	860.76	850.36	1569.11	1743.96	18.29
RC101	13.2	13	1367.55	1339.97	1673.02	1627.97	2499.6	2732.94	2.21
RC102	11	11	1208.34	1185.43	1494.8	1466.1	2165.46	2429.2	3.07
RC103	10	10	993.63	991.7	1240.07	1247.56	1952	2167.45	6.33
RC104	9	9	927.35	916.01	1137.02	1119.83	1823.3	2028.15	4.51
RC105	13	13	1237.69	1229.88	1493.77	1490.34	2477.19	2729.54	3.63
RC106	11	11	1095.19	1090.81	1344.58	1343.2	2093.07	2288.51	3.42
RC107	10	10	1028.84	1010.99	1257.78	1242.04	1976.57	2175.7	3.77
RC108	9.5	9	916.06	932.61	1140.52	1171.43	1826.86	1992.13	5.21
RC201	3	3	1331.52	1325.99	1616.11	1608.18	2404.43	2655.77	4.79
RC202	3	3	1074.15	1068.87	1297.36	1289.32	2389.86	2649.95	11.12
RC203	3	3	790.22	782.89	995.52	988.46	2406.66	2673.93	16.53
RC204	2	2	743.61	718.28	927.13	892.29	1682.5	1910.14	17.43
RC205	3	3	1193.12	1191.95	1463.83	1466.08	2419.95	2684.88	11.2
RC206	3	3	901.43	894.71	1111.61	1101.69	2218.43	2455.62	10.22
RC207	3	3	865.36	863.16	1064.68	1062.3	2082.62	2312.41	15.51
RC208	2	2	787.39	774.16	973.54	966.26	1627.36	1791.27	14.34
AVG	6.84	6.79	835.55	832.01	1031.85	1027.56	4351.46	4495.14	6.89
SUM	382.9	380	46790.84	46592.45	57783.34	57543.3	243681.98	251727.98	385.8

Table 8: TDVRPTW - C2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	594.01	594.01	828.94	828.94	9714.92	9714.92	0.19
C102	10	10	593.03	593.03	832.5	832.5	9693.21	9693.21	0.71
C103	10	10	589.46	589.46	831.62	831.62	9701.38	9701.38	2.39
C104	10	10	587.21	587.12	831.21	828.66	8770.74	9762.07	4.37
C105	10	10	594.01	594.01	828.94	828.94	9714.92	9714.92	0.39
C106	10	10	594.01	594.01	828.94	828.94	9714.92	9714.92	0.56
C107	10	10	594.01	594.01	828.94	828.94	9714.92	9714.92	0.97
C108	10	10	594.01	594.01	828.94	828.94	9714.92	9714.92	1.51
C109	10	10	593.07	593.07	829.39	829.39	9698.63	9698.63	2.14
C201	3	3	438.46	438.46	591.56	591.56	9453.23	9453.23	0.43
C202	3	3	438.46	438.46	591.56	591.56	9453.23	9453.23	1.97
C203	3	3	434.37	434.37	588.83	588.83	9496.16	9496.16	4.21
C204	3	3	433.8	433.8	588.25	588.25	9439.7	9439.7	7.46
C205	3	3	436.29	436.29	588.49	588.49	9451.06	9451.06	1.19
C206	3	3	436.29	436.29	588.49	588.49	9451.06	9451.06	1.8
C207	3	3	436.29	436.29	588.49	588.49	9483.19	9483.19	2.09
C208	3	3	436.29	436.29	588.49	588.49	9451.06	9451.06	2.36
R101	15	15	1103.55	1102.75	1600.94	1602.01	2973.44	2970.79	2.01
R102	14	14	936.37	936.37	1368.51	1368.51	2702.85	2702.85	5.35
R103	11	11	808.7	800.33	1220.69	1203.52	2015.66	2270.65	6.39
R104	8.2	8	661.24	663.55	990.85	991.14	1572.64	1789.9	5.62
R105	11	11	923.41	914.96	1345.3	1330.11	1983.58	2192.92	2.17
R106	9.2	9	843.82	844.15	1226.59	1232.15	1769.32	1951.15	3.84
R107	8.6	8	719.99	746.08	1071.2	1089.09	1649.91	1774.52	5.61
R108	8	8	618.85	611.97	920.35	906.62	1498.27	1651.39	6.81
R109	9.5	9	815.63	842.04	1174.53	1193.05	1762.98	1980.67	5.14
R110	9	9	748.07	732.78	1083.02	1071.12	1680.78	1840.16	7.52
R111	9	9	722	717.62	1049.23	1036.59	1694.93	1886.64	7.72
R112	8	8	662.22	649.49	965.12	947.04	1549.23	1708.87	4.01
R201	3	3	882.41	880.04	1299.53	1298.79	2468.46	2742.74	5.95
R202	3	3	760.26	756.78	1112.33	1115.99	2458.37	2727.7	13.46
R203	2	2	702.95	698.38	1065.23	1056.33	1753.52	1962.94	16.57
R204	2	2	510.44	497.58	825.5	805.67	1629.04	1799.18	20.06
R205	2	2	764.45	740.46	1124.91	1079.24	1650.76	1826.47	7.53
R206	2	2	658.98	649.91	984.15	963.49	1625.7	1836.85	12.7
R207	2	2	580.65	570.87	891.74	880.38	1635.86	1744.15	17.56
R208	2	2	453.75	446.52	745.22	730.09	1531.95	1727.65	20.32
R209	2	2	696.22	675.73	1037.59	1001.76	1581.57	1740.68	12.59
R210	2	2	705.79	697.02	1045.02	1030.11	1705.47	1900	12.08
R211	2	2	557.09	547.41	850.48	824.26	1480.14	1616.87	17.94
RC101	12	12	1097.54	1077.49	1537.81	1508.43	2312.12	2563.63	2.3
RC102	10	10	1016.03	1009.11	1422.34	1399.64	2002.57	2230.43	3.51
RC103	9	9	856.35	847.7	1239.54	1214.38	1791.45	2018.67	3.4
RC104	9	9	770.75	764.6	1145.16	1145	1687.11	1866.53	6.38
RC105	11	11	1032.16	1029.49	1443.86	1448.53	2168.71	2385.95	3.47
RC106	10	10	946.94	919.51	1338.9	1312.09	1918.86	2125.01	3.23
RC107	10	10	827.02	819.63	1213.02	1210.53	1797.15	1991.2	6.6
RC108	9	9	755.61	742.01	1097.46	1061.4	1689.5	1916.45	2.56
RC201	3	3	1074.77	1071.66	1532.33	1535.93	2326.46	2610.22	5.59
RC202	3	3	914.31	910.33	1279.64	1268.74	2368.14	2670.01	12.58
RC203	2	2	792.99	787.43	1158.36	1156.44	1693.2	1902.44	11.38
RC204	2	2	575.74	566.29	883.71	873.77	1620.58	1822.19	17.16
RC205	3	3	977.4	975.73	1389.34	1386.35	2386.56	2684.88	11.05
RC206	3	3	773.71	766.15	1113.36	1099.99	2176.01	2438.93	9.4
RC207	3	3	711.17	698.58	1051.6	1034.26	1976.01	2187.35	17.48
RC208	2	2	639.94	624.28	944.69	946.67	1497.43	1635.3	16.86
AVG	6.42	6.39	703.97	699.21	1017.37	1009.65	4212.56	4366.13	6.9
SUM	359.5	358	39422.34	39155.76	56972.73	56540.24	235903.54	244503.51	386.64

Table 9: TDVRPTW - C3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	543.45	543.45	828.94	828.94	9689.38	9689.38	0.2
C102	10	10	540.6	540.6	832.5	832.5	9665.94	9665.94	0.68
C103	10	10	536.1	536.1	873.26	873.26	9678.13	9678.13	2.77
C104	10	10	533.2	533.11	854.65	846.37	8673.47	9570.54	4.72
C105	10	10	543.45	543.45	828.94	828.94	9689.38	9689.38	0.41
C106	10	10	543.45	543.45	828.94	828.94	9689.38	9689.38	0.59
C107	10	10	543.41	543.41	831.64	831.64	9682.14	9682.14	1.09
C108	10	10	542.57	542.57	862.37	862.37	9681.45	9681.45	1.75
C109	10	10	539.35	539.35	858.31	858.31	9668.6	9668.6	2.78
C201	3	3	404.26	404.26	591.56	591.56	9422.08	9422.08	0.41
C202	3	3	404.26	404.26	591.56	591.56	9422.08	9422.08	1.91
C203	3	3	398.95	398.95	591.59	591.59	9473.03	9473.03	4.11
C204	3	3	398.37	398.37	591.02	591.02	9398.37	9398.37	6.72
C205	3	3	402.34	402.34	588.49	588.49	9420.16	9420.16	1.17
C206	3	3	402.34	402.34	588.49	588.49	9420.16	9420.16	1.78
C207	3	3	402.34	402.34	588.49	588.49	9461.49	9461.49	2.06
C208	3	3	402.34	402.34	588.49	588.49	9420.16	9420.16	2.35
R101	14	14	975.58	974.86	1573.11	1569.74	2816.58	2823.06	2.17
R102	13	13	824.47	824.47	1352.77	1352.77	2509.4	2509.4	5.18
R103	10	10	672.63	671.78	1169.9	1167.27	1796.01	1994.88	6.63
R104	8	8	560.78	558.4	967.6	958.17	1472.13	1655.2	5.3
R105	10.6	10	817.74	854.66	1329.41	1371.24	1923.83	2079.59	2.66
R106	9	9	719.22	716.49	1203.95	1200.82	1648.63	1838.77	6.17
R107	8	8	631.23	623.03	1071.96	1056.95	1560.85	1718.77	4.34
R108	8	8	532.81	528.24	928.94	922.93	1435.72	1595.76	8.46
R109	9	9	716.31	704.11	1155.15	1135.55	1677.82	1852.05	4.25
R110	9	9	630.23	628.85	1060.01	1059.5	1583.94	1756.6	7.41
R111	8.1	8	671.25	670.67	1103.31	1115.17	1595.06	1758.1	3.77
R112	8	8	548.78	543.53	928.8	912.58	1462.89	1630.5	6.92
R201	3	3	779.94	776.56	1280.84	1272.47	2468.46	2742.74	6.13
R202	3	3	677.56	675.93	1117.73	1112.79	2453.73	2727.7	12.96
R203	2	2	607.77	606.64	1058.03	1058.48	1738.14	1935.66	16.93
R204	2	2	419.82	415.19	815.74	805.29	1612.71	1791.9	20.31
R205	2	2	658.25	634.64	1093.18	1078.2	1643.2	1815.71	8
R206	2	2	570.76	565.36	973.71	967.84	1595.17	1757.19	13.68
R207	2	2	509.14	499.07	897.79	871.53	1628.75	1786.29	18.66
R208	2	2	382.81	377.79	744.48	727.41	1485.15	1649.11	17.04
R209	2	2	598.93	580.04	1022.9	984.99	1545.23	1709.53	12.8
R210	2	2	609.72	602.28	1039.59	1039.66	1710.81	1915.15	12.89
R211	2	2	479.08	474.69	855.77	874.06	1427.2	1552.07	17.31
RC101	11	11	1019.03	1005.37	1576.94	1548.34	2152.96	2398.38	2.17
RC102	10	10	892.98	882.12	1398.45	1375.78	1962.23	2183.03	4.24
RC103	9	9	750.59	744.52	1251.85	1251.82	1731.09	1954.7	3.92
RC104	9	9	658.88	654.26	1123.03	1108.18	1619.41	1841.77	7.44
RC105	10.3	10	958.32	960.38	1499.64	1496.27	2067.35	2263.58	3.5
RC106	10	10	802.58	797.5	1299.89	1283.66	1877.18	2078.47	5.21
RC107	9	9	775.67	756.58	1254	1220.8	1753.61	1932.98	2.68
RC108	9	9	659.66	645.36	1117.96	1090.17	1623.39	1789.37	5.8
RC201	3	3	956.06	955.85	1519.44	1520.8	2349.2	2610.22	6.03
RC202	3	3	818.01	816.92	1274.95	1273.15	2369.73	2638.91	11.85
RC203	2	2	685.89	682.16	1133.58	1126.2	1696.86	1893.4	14.12
RC204	2	2	487.97	480.03	879.94	868.18	1614.98	1810.23	19.14
RC205	3	3	874.6	871.27	1410.42	1403.65	2388.23	2688.35	10.74
RC206	2	2	844.36	828.32	1330.96	1292.53	1683.5	1894.33	7.63
RC207	2	2	807.13	789.94	1294.55	1271.78	1628.11	1813.22	8.46
RC208	2	2	536.44	526.29	940.27	931.1	1415.21	1568.89	15.98
AVG	6.23	6.21	628.64	624.73	1024.46	1017.66	4147.85	4293	6.86
SUM	349	348	35203.76	34984.84	57369.78	56988.78	232279.85	240408.03	384.38

Table 10: TDVRPTW - D1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	$\bar{t}o\bar{t}$	tot_{best}	\bar{t}_e
C101	10	10	699.3	699.3	828.94	828.94	9699.3	9699.3	0.19
C102	10	10	699.3	699.3	828.94	828.94	9699.3	9699.3	0.67
C103	10	10	698.5	698.5	828.06	828.06	9833.85	9833.85	2.46
C104	10	10	691.53	691.53	828.07	828.07	10306.29	10306.29	3.94
C105	10	10	699.3	699.3	828.94	828.94	9699.3	9699.3	0.32
C106	10	10	699.3	699.3	828.94	828.94	9699.3	9699.3	0.4
C107	10	10	699.3	699.3	828.94	828.94	9699.3	9699.3	0.8
C108	10	10	699.26	699.26	828.94	828.94	9699.26	9699.26	1.25
C109	10	10	699.26	699.26	828.94	828.94	9699.26	9699.26	2.23
C201	3	3	493.26	493.26	591.56	591.56	9493.26	9493.26	0.43
C202	3	3	493.26	493.26	591.56	591.56	9493.26	9493.26	2.26
C203	3	3	493.02	493.02	591.17	591.17	9517.43	9517.43	5.02
C204	3	3	492.66	492.66	590.6	590.6	9492.66	9492.66	8.07
C205	3	3	490.71	490.71	588.88	588.88	9490.71	9490.71	1.11
C206	3	3	490.47	490.47	588.49	588.49	9490.47	9490.47	1.69
C207	3	3	487.63	487.63	588.29	588.29	9561.6	9561.6	2.18
C208	3	3	486.81	486.81	590.89	590.89	9791.65	9791.65	2.23
R101	18	18	1419.19	1416.75	1632.62	1629.92	3008.13	3361	1.72
R102	17	17	1268.82	1268.82	1464.78	1464.74	2890.86	3212.07	3.47
R103	13.4	13	1068.02	1083.18	1244.02	1261.34	2339.82	2533.94	5.98
R104	9	9	794.52	785.7	967.21	952.91	1789.41	1990.36	7.16
R105	14	14	1183.69	1181.6	1368.69	1365.92	2554.79	2558.14	2.8
R106	12	12	1071.84	1069.36	1247.67	1246.79	2090.25	2325.18	6.12
R107	10	10	906.24	904.39	1075.78	1085.9	1886.23	2093.62	8.7
R108	9	9	749.9	747.78	914.03	909.52	1727.06	1911.16	10.5
R109	11	11	1001.52	995.78	1186.71	1185.72	2122.8	2137.08	4.74
R110	10	10	883.77	880.88	1094.13	1090.2	1880.53	2093.96	7.18
R111	10	10	892.85	884.83	1072.83	1055.24	1865.6	2055.36	6.5
R112	9	9	772.37	764.39	956.5	953.61	1725.1	1917.39	8.41
R201	4	4	1096.59	1092.89	1247.12	1247.99	3048.21	3466.07	3.82
R202	3	3	1031.53	1025.2	1189.62	1184.75	2388.51	2728.41	11.67
R203	3	3	810.72	807.42	945.03	937.76	2409.74	2757.15	18.55
R204	2	2	693.43	685.94	842.43	836.76	1718.93	1968.43	20.43
R205	3	3	888.31	873.01	1026.65	994.43	2162.77	2490.53	11.97
R206	2	2	940.33	926.94	1112.48	1091.79	1703.37	1926.94	6.62
R207	2	2	748.96	737.53	891.79	880.72	1656.01	1914.52	16.71
R208	2	2	604.52	590.29	753.93	749.57	1719.8	1979.63	18.12
R209	3	3	781.36	773.05	912.46	909.14	2061.67	2358.02	14.31
R210	3	3	825.17	816.3	964.78	953.93	2345.45	2666.74	15.61
R211	2	2	744.16	733.12	885.58	865.49	1563.89	1779.11	17.41
RC101	14.75	14	1430.66	1435.51	1676.87	1689.54	2487.79	2737.03	1.88
RC102	12.62	12	1241.23	1275.05	1488.16	1518.89	2264.52	2476.83	3.6
RC103	10	10	1032.97	1029.09	1251.95	1243.71	1905.91	2166.46	4.55
RC104	9	9	912.97	904.12	1114.11	1105.53	1788.15	2040.63	4.6
RC105	13.88	13	1314.72	1344.89	1560.8	1603.65	2425.5	2738.01	3.5
RC106	11.75	11	1157.7	1187.71	1395.38	1440.11	2069.12	2292.68	2.56
RC107	10	10	1036.59	1023.54	1267.46	1242.5	1884.5	2113.84	3.7
RC108	10	10	922.36	914.69	1123.73	1110.42	1851.16	2121.37	7.82
RC201	4	4	1195.73	1195.73	1377.49	1377.49	2889.76	3302.59	3.84
RC202	3	3	1168.74	1113.41	1369.92	1311.85	2245.19	2640.01	9.04
RC203	3	3	911.23	900.78	1062.56	1045.99	2280.11	2687.65	15.95
RC204	2	2	789.45	765.54	960.61	932.63	1631.11	1896.86	15.8
RC205	4	4	1099.8	1099.51	1293.01	1297.28	2881.23	3424.69	7.9
RC206	3	3	981.33	974.52	1145.27	1136.85	2022.12	2343.87	10.61
RC207	3	3	901.38	897.52	1054.53	1045.45	1940.71	2247.05	16.51
RC208	2	2	857.37	835.89	1054.99	1024.77	1597.36	1838.54	9.76
AVG	7.17	7.11	863.3	859.92	1024.53	1020.73	4414.1	4601.06	6.88
SUM	401.4	398	48344.91	48155.52	57373.83	57160.95	247189.37	257659.12	385.37

Table 11: TDVRPTW - D2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	633.78	633.78	828.94	828.94	9633.78	9633.78	0.18
C102	10	10	632.19	632.19	834.64	834.64	9791.8	9791.8	0.68
C103	10	10	631.76	631.76	834.56	834.56	9946.99	9946.99	2.43
C104	10	10	622.81	622.81	833.04	833.04	10581.48	10581.48	4.01
C105	10	10	633.72	633.72	830.54	830.54	9670.91	9670.91	0.32
C106	10	10	633.42	633.42	831.31	831.31	9660	9660	0.39
C107	10	10	633.78	633.78	828.94	828.94	9633.78	9633.78	0.81
C108	10	10	633.51	633.51	828.94	828.94	9633.51	9633.51	1.25
C109	10	10	633.13	633.13	831.49	831.49	9642.51	9642.51	2.09
C201	3	3	430.11	430.11	591.56	591.56	9430.11	9430.11	0.42
C202	3	3	430.11	430.11	591.56	591.56	9430.11	9430.11	2.21
C203	3	3	429.92	429.92	591.17	591.17	9466.48	9466.48	4.81
C204	3	3	428.98	428.98	590.39	590.39	9428.98	9428.98	7.3
C205	3	3	428.32	428.32	588.88	588.88	9428.32	9428.32	1.09
C206	3	3	428.13	428.13	588.49	588.49	9428.13	9428.13	1.66
C207	3	3	426.65	426.65	588.29	588.29	9511.73	9511.73	2.12
C208	3	3	424.29	424.29	590.89	590.89	9740.86	9740.86	2.17
R101	17.9	17	1259.41	1294.48	1629.37	1689.74	2860.81	3219.92	1.96
R102	16	16	1084.01	1083.91	1417.41	1417.26	2634.94	2922.55	4.24
R103	12	12	922.5	916.77	1243.43	1232.85	2175.25	2398.17	6.04
R104	8.9	8	667.28	746.51	957.34	1055.18	1740.02	1952.38	6.76
R105	13	13	1033.82	1028.52	1358.46	1349.65	2100.88	2346.33	2.15
R106	11	11	930.59	926.55	1241.18	1228.78	1895.55	2095.7	4.4
R107	9	9	792.1	784.04	1096.68	1083.39	1732.87	1918.6	5.27
R108	8	8	643.15	634.45	921.82	908.85	1559.84	1741.29	4.57
R109	10.6	10	861.86	863.45	1183.48	1202.81	1805.09	1966.84	3.61
R110	9	9	753.71	744.57	1094.54	1090.41	1702.98	1899.43	4.84
R111	9	9	738.68	732.42	1057.25	1056.95	1916.54	1923.51	7.1
R112	8	8	677.27	662.34	970.57	947.94	1719.46	1724.76	2.71
R201	4	4	961.08	958.04	1229.15	1224.88	3083.09	3446.26	4.06
R202	3	3	879.57	876.57	1148.4	1145.33	2392.31	2691.41	11.74
R203	2	2	857.36	834.33	1146.83	1120.91	1722.42	1977.29	6.38
R204	2	2	581.36	575.9	833.86	818.34	1709.53	1921.08	18.06
R205	3	3	779.84	770.16	1024.92	1022.41	2176.7	2453.68	10.1
R206	2	2	777.95	769.22	1054.67	1037.03	1651.29	1867.24	10.22
R207	2	2	658.24	648.91	886.75	871.72	1656.96	1845.9	16.9
R208	2	2	512.07	501.46	752.23	744.88	1706.38	1960.71	17.88
R209	2.12	2	771.77	773.46	1073.24	1085.22	1651.92	1825.7	6.39
R210	2	2	838.15	823.48	1139.61	1130.64	1628.97	1835.9	9.24
R211	2	2	617.64	612.19	869.03	862.39	1546.89	1779.34	16.86
RC101	14	14	1246.95	1205.85	1656.85	1598.4	2287.91	2576.99	2.2
RC102	11.12	11	1058.67	1064.37	1438.26	1445.06	1965.68	2209.89	4.06
RC103	9.25	9	915.11	917.56	1271.76	1264.01	1777.62	1971.82	2.97
RC104	9	9	778.5	765.59	1112.87	1091.98	1746.19	2012.84	4.64
RC105	12.5	12	1146.63	1139.73	1565.32	1540.41	2160.79	2388.87	3.69
RC106	10.88	10	967.08	1059.03	1349.65	1495.89	1891.44	2170.64	3.65
RC107	10	10	861.69	854.77	1225.16	1225.21	1808.63	2081.7	5.93
RC108	9	9	820.11	798.22	1160.76	1158.62	1698.82	1917.09	1.68
RC201	4	4	1049.03	1045.72	1374.79	1379.74	2865.15	3281.07	3.84
RC202	3	3	959.29	945.7	1299.07	1287.24	2256.15	2570.13	11.94
RC203	3	3	780.96	777.68	1050.41	1050.19	2285.23	2643.85	16.9
RC204	2	2	648.98	640.02	947.87	928.32	1660.76	1908.18	17.85
RC205	3	3	1135.31	1087.71	1518.6	1456.59	2285.92	2608.91	9.16
RC206	3	3	866.7	854.44	1157.18	1140.52	2019.36	2282.95	11.68
RC207	3	3	767.54	757.05	1067.73	1036.6	1963.11	2205.76	14.81
RC208	2	2	694.2	676.33	981.74	954.75	1518.31	1741.39	15.8
AVG	6.81	6.73	757.34	754.75	1030.57	1028.3	4303.95	4470.46	6.18
SUM	381.27	377	42410.77	42266.11	57711.87	57584.72	241021.24	250345.55	346.22

Table 12: TDVRPTW - D3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	\bar{tot}	tot_{best}	\bar{t}_e
C101	10	10	592.39	592.39	828.94	828.94	9592.39	9592.39	0.19
C102	10	10	589.11	589.11	836.52	836.52	9906.75	9906.75	0.73
C103	10	10	588.58	588.58	842.28	842.28	10225.37	10225.37	2.67
C104	10	10	577.42	577.42	841.93	841.93	10706.4	10706.4	4.28
C105	10	10	591.85	591.85	830.54	830.54	9629.04	9629.04	0.33
C106	10	10	591.25	591.25	831.31	831.31	9617.83	9617.83	0.42
C107	10	10	592.39	592.39	828.94	828.94	9592.39	9592.39	0.87
C108	10	10	592.03	592.03	828.94	828.94	9592.03	9592.03	1.36
C109	10	10	590.76	590.76	831.93	831.93	9601	9601	2.25
C201	3	3	394.6	394.6	591.56	591.56	9394.6	9394.6	0.44
C202	3	3	394.6	394.6	591.56	591.56	9394.6	9394.6	2.31
C203	3	3	394.45	394.45	591.17	591.17	9437.01	9437.01	5.33
C204	3	3	393.58	393.58	590.39	590.39	9393.58	9393.58	8.93
C205	3	3	392.92	392.92	588.49	588.49	9392.92	9392.92	1.13
C206	3	3	392.92	392.92	588.49	588.49	9392.92	9392.92	1.77
C207	3	3	390.78	390.78	592.54	592.54	9575.63	9575.63	2.16
C208	3	3	388.47	388.47	590.89	590.89	9708.95	9708.95	2.25
R101	17	17	1158.76	1154.17	1610.18	1602.86	2715.29	3016.15	1.8
R102	16	16	1004.44	1004.31	1409.89	1410.03	2645.78	2940.76	4.04
R103	12	12	826.88	824.36	1207.8	1207.07	2134.95	2372.77	7.01
R104	8	8	638.49	622.95	1012.66	1001.37	1565.8	1743.84	2.57
R105	13	13	954.8	950.13	1355.42	1355.22	2067.74	2306.34	2.27
R106	10.4	10	863.54	866.1	1259.2	1263.83	1829.79	1973.46	4.99
R107	9	9	693.49	693.4	1057.37	1055.62	1690.78	1874.27	8.52
R108	8	8	560.69	558.26	914.26	909.83	1554.62	1727.06	7.27
R109	10	10	767.89	736.97	1166.62	1133.6	1700.27	1890.09	3.84
R110	9	9	654.83	646.76	1076.05	1075.48	1657.08	1845.41	5.5
R111	8.7	8	656.94	682.06	1050.84	1072.6	1651.59	1743.84	5.23
R112	8	8	572.85	561.91	942.42	947.17	1528.85	1705.78	3.51
R201	3.33	3	1015.37	1056.01	1382.46	1431.35	2502.1	2609.25	3.94
R202	3	3	818.99	814.29	1153.65	1142.73	2380.73	2675.53	11.23
R203	2	2	767.43	757.83	1116.49	1104.79	1725.73	1938.45	8.76
R204	2	2	522.45	516.76	829.42	819.66	1696.04	1963.98	19.6
R205	3	3	724.71	712.56	1023.06	1021.77	2113.31	2440.82	9.88
R206	2	2	707.92	701.03	1048.93	1034.83	1587.51	1808.07	10.45
R207	2	2	602.63	596.78	897.2	883.62	1621.21	1852.9	15.21
R208	2	2	445.62	439.53	762.08	766.08	1708.46	1981.86	20.46
R209	2	2	681.6	675.88	1045.21	1029.97	1526.28	1736.92	6.36
R210	2	2	749.84	741.05	1109.16	1094.34	1615.63	1874.43	11.74
R211	2	2	547.86	541.33	868.77	863.2	1519.28	1764.95	17.05
RC101	13	13	1094.9	1086.35	1581.12	1566.08	2076.8	2374.21	2.01
RC102	11.12	11	967.07	967.06	1428.81	1418.9	1897.28	2141.76	3.99
RC103	9	9	807.54	793.25	1265.66	1264.86	1716.46	1967.92	3.18
RC104	9	9	692.35	685.9	1119.03	1108.66	1730.46	2006.05	6.53
RC105	11.62	11	1034.84	1009.84	1550.47	1527.18	1979.78	2152.96	3.21
RC106	10.75	10	871.9	889.54	1335.49	1361.54	1824.59	1993.52	3.74
RC107	9	9	818.13	806.09	1304.6	1282.42	1688.45	1922.94	1.41
RC108	9	9	697.33	690.48	1106.86	1091.44	1665.24	1909.24	4.05
RC201	4	4	966.31	965.41	1377.69	1381.09	2855.87	3263.86	4.5
RC202	3	3	866.67	863.97	1284.26	1282.21	2226.61	2526.21	13.02
RC203	2	2	843.14	840.18	1278.76	1279.86	1654.87	1901.73	14.79
RC204	2	2	566.92	556.01	940.58	950.05	1661.09	1894.62	16.84
RC205	3	3	994.29	983.7	1458.77	1432.81	2281.91	2618.8	10
RC206	3	3	789.87	774.78	1160.04	1158.11	2032.4	2299.95	10.68
RC207	3	3	687.78	677.59	1057.57	1043.7	1919.23	2193.15	15.38
RC208	2	2	616.33	608.96	992.21	1019.97	1519.56	1788.47	15.37
AVG	6.66	6.61	691.1	687.53	1031.56	1028.97	4243.26	4408.85	6.38
SUM	372.92	370	38701.49	38501.64	57767.48	57622.32	237622.83	246895.73	357.35

C TD-VRPTW total time

Table 13: TD-VRPTW - A1 - total time - results

Inst.	K_{it}^{best}	tot_{it}^{best}	\bar{K}	K_{best}	ΔK	\overline{tot}	tot_{best}	Δtot	$\Delta_p tot$	d_{best}	\bar{t}_e
C101	10	9685.24	10	10	0	9685.24	9685.24	0	0	828.94	1.27
C102	10	9683.26	10	10	0	9683.26	9683.26	0	0	835.46	11.45
C103	10	9750.96	10	10	0	9683.26	9683.26	-67.7	-0.69	835.46	21.24
C104	10	9816.21	10	10	0	9681.82	9681.82	-134.39	-1.37	832.25	24.53
C105	10	9763.24	10	10	0	9685.24	9685.24	-78	-0.8	828.94	3.62
C106	10	9685.24	10	10	0	9685.24	9685.24	0	0	828.94	4.5
C107	10	9697.6	10	10	0	9685.24	9685.24	-12.36	-0.13	828.94	5.37
C108	10	9684.38	10	10	0	9682.83	9682.83	-1.55	-0.02	833.01	8.79
C109	10	9680.74	10	10	0	9680.74	9680.74	0	0	833.03	10.79
C201	3	9502.9	3	3	0	9502.9	9502.9	0	0	591.56	2.37
C202	3	9502.9	3	3	0	9502.9	9502.9	0	0	591.56	18.71
C203	3	9539.88	3	3	0	9502.9	9502.9	-36.98	-0.39	591.56	37.59
C204	3	9500.84	3	3	0	9500.84	9500.84	0	0	590.97	53.28
C205	3	9500.34	3	3	0	9500.34	9500.34	0	0	588.88	5.63
C206	3	9499.49	3	3	0	9499.49	9499.49	0	0	588.49	8.12
C207	3	9500.54	3	3	0	9500.54	9500.54	0	0	588.32	10.1
C208	3	9499.49	3	3	0	9499.49	9499.49	0	0	588.49	9.68
R101	18	3370.77	18	18	0	3149.7	3149.7	-221.07	-6.56	1965.99	2.8
R102	16	3059.76	16	16	0	2778.7	2778.7	-284.28	-9.29	1968.55	7.98
R103	13	2568.37	13	13	0	2268.31	2263.17	-305.2	-11.88	1472.05	14.86
R104	9	1935.73	9	9	0	1838.92	1834.24	-101.49	-5.24	982.07	22.82
R105	12	2306.89	12	12	0	2248.01	2247.94	-58.95	-2.56	1445.48	5.73
R106	10	2091.51	11	11	1	2056.42	2054.67	-36.84	-1.76	1293.17	15.94
R107	9	1960.62	9	9	0	1936.23	1936.14	-24.48	-1.25	1142.94	19.11
R108	8	1798.77	9	9	1	1778.37	1777.74	-21.03	-1.17	961.22	28.87
R109	10	2008.31	10	10	0	1997.27	1994.7	-13.61	-0.68	1198.35	12.95
R110	9	1943.07	9	9	0	1925.98	1925.98	-17.09	-0.88	1125.6	18.98
R111	9	1942.09	9	9	0	1918.45	1918.45	-23.64	-1.22	1103.95	19.86
R112	9	1842.62	9	9	0	1780.84	1780.84	-61.78	-3.35	972.18	26.4
R201	3	2661.46	3	3	0	2369.57	2369.57	-291.89	-10.97	1672.83	10.46
R202	3	2723.84	3	3	0	2240.28	2240.28	-483.56	-17.75	1534.34	30.61
R203	2	1931.76	3	3	1	2033.61	2033.61	101.85	5.27	1277.69	29.75
R204	2	1897.78	2	2	0	1733	1733	-164.78	-8.68	906.21	32.86
R205	3	2404.95	3	3	0	1999.79	1999.79	-405.16	-16.85	1254.19	25.99
R206	2	1844.31	2	2	0	1805.92	1805.92	-38.39	-2.08	1019.38	29.87
R207	2	1848.31	2	2	0	1714.4	1714.4	-133.91	-7.24	906.67	45.54
R208	2	1709.08	2	2	0	1576.44	1576.44	-132.64	-7.76	754.04	51.92
R209	2	1860.48	2	2	0	1859.75	1859.75	-0.73	-0.04	1071.79	33.9
R210	2	1901.54	2	2	0	1892.22	1892.22	-9.32	-0.49	1106.87	28.11
R211	2	1710.63	2	2	0	1687.65	1687.65	-22.98	-1.34	868.96	45.34
RC101	13	2557.91	13	13	0	2507.18	2503.1	-54.81	-2.14	1699.31	3.84
RC102	11	2368.31	11	11	0	2247.59	2241.77	-126.54	-5.34	1515.58	8.51
RC103	10	2142.61	10	10	0	2050.9	2049.79	-92.82	-4.33	1290.04	17.36
RC104	9	2005.38	9	9	0	1954.46	1945.66	-59.72	-2.98	1142.58	15.11
RC105	13	2656.73	13	13	0	2471.89	2459.25	-197.48	-7.43	1649.35	7.33
RC106	11	2247.98	11	11	0	2140.84	2136.6	-111.38	-4.95	1390.54	9.23
RC107	10	2079.54	10	10	0	2032.75	2023.79	-55.75	-2.68	1262.38	17.84
RC108	9	1969.88	9	9	0	1961.17	1931.67	-38.21	-1.94	1136.05	9.35
RC201	3	2604.29	3	3	0	2467.67	2460.12	-144.17	-5.54	1731.88	10.66
RC202	3	2609.6	3	3	0	2230.9	2230.9	-378.7	-14.51	1533.56	28.72
RC203	3	2665.36	3	3	0	2055.98	2045.28	-620.08	-23.26	1288.96	30.73
RC204	2	1915.11	2	2	0	1785.22	1779.18	-135.93	-7.1	979.66	30.29
RC205	3	2669.7	3	3	0	2347.49	2347.02	-322.68	-12.09	1661.96	23.3
RC206	3	2326.66	3	3	0	2104.03	2101.32	-225.34	-9.69	1349.79	24.83
RC207	3	2235.27	3	3	0	2008.69	2006.86	-228.41	-10.22	1294.26	41.54
RC208	2	1767.6	2	2	0	1807.57	1791.47	23.87	1.35	992.34	24.61
AVG	6.77	4457.82	6.82	6.82	0.05	4355.83	4353.35	-104.47	-4.11	1109.42	19.66
SUM	379	249637.83	382	382	3	243926.43	243787.73	-5850.1	-230.02	62127.56	1100.94

Table 14: TD-VRPTW - B1 - total time - results

Inst.	K_{it}^{best}	tot_{it}^{best}	\bar{K}	K_{best}	ΔK	\bar{tot}	tot_{best}	Δtot	$\Delta_p tot$	d_{best}	\bar{t}_e
C101	10	9851.43	10	10	0	9773.81	9773.81	-77.62	-0.79	828.94	0.86
C102	10	10078.54	10	10	0	9757.52	9757.52	-321.02	-3.19	828.94	6.58
C103	10	10599.34	10	10	0	9697.93	9697.93	-901.41	-8.5	841.35	13.87
C104	10	11156.25	10	10	0	9689.56	9689.34	-1466.91	-13.15	842.39	14.86
C105	10	9773.81	10	10	0	9773.81	9773.81	0	0	828.94	2.5
C106	10	9850.77	10	10	0	9773.81	9773.81	-76.96	-0.78	828.94	3.75
C107	10	9761.07	10	10	0	9759.24	9759.24	-1.83	-0.02	837.2	4.35
C108	10	9924.35	10	10	0	9753.4	9753.4	-170.95	-1.72	838.19	6.08
C109	10	9752.78	10	10	0	9749.72	9749.72	-3.06	-0.03	833.89	7.57
C201	3	9500.85	3	3	0	9500.85	9500.85	0	0	591.56	1.56
C202	3	9500.85	3	3	0	9500.85	9500.85	0	0	591.56	11.94
C203	3	9494.55	3	3	0	9491.91	9491.91	-2.64	-0.03	591.64	21.07
C204	3	9530.31	3	3	0	9487.76	9487.76	-42.55	-0.45	590.48	27.94
C205	3	9497.58	3	3	0	9497.58	9497.58	0	0	589.89	3.38
C206	3	9576.6	3	3	0	9497.58	9497.58	-79.02	-0.83	589.89	5.19
C207	3	9586.44	3	3	0	9500.12	9500.12	-86.32	-0.9	592.4	6.23
C208	3	9662.3	3	3	0	9497.58	9497.58	-164.72	-1.7	589.89	6.61
R101	18	3384.72	18	18	0	3226.7	3208.51	-176.21	-5.21	1741.38	5.08
R102	17	3286.31	17	17	0	2949.32	2928.11	-358.2	-10.9	2078.82	13.8
R103	13	2720.37	13	13	0	2251.43	2247.28	-473.09	-17.39	1442.93	19.71
R104	9	1981.95	9	9	0	1849.87	1843.21	-138.74	-7	1034.09	23.31
R105	13	2534.82	13	13	0	2454.04	2449.51	-85.31	-3.37	1470.77	6.55
R106	11	2342.06	12	12	1	2151.98	2150.67	-191.39	-8.17	1357.97	16.43
R107	9	2126.05	10	10	1	1925.44	1925.01	-201.04	-9.46	1126.79	23.3
R108	9	1957.84	9	9	0	1780.43	1778.48	-179.36	-9.16	954.14	27.71
R109	11	2171.98	11	11	0	2125.34	2123.92	-48.06	-2.21	1233.35	13.61
R110	10	2071.54	10	10	0	2036.57	2032.86	-38.68	-1.87	1160.56	18.68
R111	10	2127.44	10	10	0	2003.29	1998.22	-129.22	-6.07	1136.05	19.95
R112	9	1904.18	9	9	0	1845.81	1841.95	-62.23	-3.27	986.41	26.87
R201	4	3497.61	4	4	0	2870.33	2842.7	-654.91	-18.72	1833.03	11.25
R202	3	2701.67	3	3	0	2300.31	2300.31	-401.36	-14.86	1521.85	30.58
R203	3	2791.87	3	3	0	2088.63	2088.63	-703.24	-25.19	1299.89	53.13
R204	2	1981.75	2	2	0	1762.74	1762.74	-219.01	-11.05	900.62	33.53
R205	3	2520.05	3	3	0	2075.51	2075.51	-444.54	-17.64	1273.58	30.37
R206	2	1961.17	2	2	0	1887.44	1887.44	-73.73	-3.76	1075.17	22.53
R207	2	1895.15	2	2	0	1780.68	1780.68	-114.47	-6.04	936.2	33.9
R208	2	1934.92	2	2	0	1646.9	1646.9	-288.02	-14.89	777.05	37.65
R209	3	2384.31	3	3	0	2125.48	2125.48	-258.83	-10.86	1340.23	39.8
R210	3	2709.62	3	3	0	2041.96	2041.96	-667.66	-24.64	1254.48	44.38
R211	2	1815.23	2	2	0	1763.75	1763.75	-51.48	-2.84	911.86	31.64
RC101	14	2871.37	14	14	0	2804.83	2783.97	-87.4	-3.04	1775.56	5.48
RC102	12	2586.71	12	12	0	2360.1	2352	-234.71	-9.07	1615.93	11.88
RC103	10	2251.46	10	10	0	2082.1	2077.07	-174.39	-7.75	1329.54	15.56
RC104	9	2048.48	9	9	0	1952.8	1947.74	-100.74	-4.92	1176.22	9.4
RC105	13	2704.76	13.33	13	0	2658.28	2622.14	-82.62	-3.05	1752.01	11.23
RC106	11	2344.45	11	11	0	2334.88	2320.49	-23.96	-1.02	1435.04	8.68
RC107	10	2211.86	10	10	0	2157.9	2156.57	-55.29	-2.5	1327.11	13.11
RC108	10	2147.89	10	10	0	2038.16	2030.14	-117.75	-5.48	1170.76	19.63
RC201	4	3308.19	4	4	0	2883.56	2878.47	-429.72	-12.99	2011.5	11.01
RC202	3	2626.87	3	3	0	2409.51	2394.15	-232.72	-8.86	1660.42	29.86
RC203	3	2615.14	3	3	0	2064.3	2052.92	-562.22	-21.5	1262.3	47.09
RC204	2	1892.32	2	2	0	1834.67	1825.11	-67.21	-3.55	993.09	30.62
RC205	4	3413.79	4	4	0	2906.16	2827.98	-585.81	-17.16	2000.95	21.78
RC206	3	2435.93	3	3	0	2211.64	2199.28	-236.65	-9.71	1414.68	27.29
RC207	3	2366.44	3	3	0	2158.45	2154.38	-212.06	-8.96	1328.06	42.03
RC208	2	1829.71	2	2	0	1849.01	1838.12	8.41	0.46	1019.53	26.23
AVG	7.05	4670.64	7.1	7.09	0.04	4452.74	4446.56	-224.08	-6.89	1149.21	18.91
SUM	395	261555.8	397.33	397	2	249353.33	249007.17	-12548.63	-385.76	64356.01	1058.98

Table 15: TD-VRPTW - C1 - total time - results

Inst.	K_{it}^{best}	tot_{it}^{best}	\bar{K}	K_{best}	ΔK	\bar{tot}	tot_{best}	Δtot	$\Delta_p tot$	d_{best}	\bar{t}_e
C101	10	9757.54	10	10	0	9757.54	9757.54	0	0	828.94	1.34
C102	10	9741.25	10	10	0	9741.25	9741.25	0	0	828.94	10.05
C103	10	9756.36	10	10	0	9681.38	9681.38	-74.98	-0.77	828.94	18.85
C104	10	9823.53	10	10	0	9682.2	9680.99	-142.54	-1.45	831.99	21.44
C105	10	9757.54	10	10	0	9757.54	9757.54	0	0	828.94	3.64
C106	10	9757.54	10	10	0	9757.54	9757.54	0	0	828.94	4.5
C107	10	9757.54	10	10	0	9753.27	9753.27	-4.27	-0.04	831.64	5.6
C108	10	9757.54	10	10	0	9752.98	9752.98	-4.56	-0.05	837.2	8.77
C109	10	9757.54	10	10	0	9740.64	9740.64	-16.9	-0.17	840	10.67
C201	3	9510.49	3	3	0	9510.49	9510.49	0	0	591.56	2.17
C202	3	9510.49	3	3	0	9510.49	9510.49	0	0	591.56	16.18
C203	3	9539.85	3	3	0	9501.54	9501.54	-38.31	-0.4	591.64	27.97
C204	3	9500.33	3	3	0	9497.69	9497.69	-2.64	-0.03	590.48	37.54
C205	3	9507.93	3	3	0	9507.93	9507.93	0	0	588.88	4.8
C206	3	9507.55	3	3	0	9507.55	9507.55	0	0	588.49	7.47
C207	3	9530.65	3	3	0	9510.1	9510.1	-20.55	-0.22	591.17	8.53
C208	3	9507.55	3	3	0	9507.55	9507.55	0	0	588.49	9.2
R101	18	3368.52	18	18	0	3273.89	3271.24	-97.28	-2.89	1964.15	3.53
R102	16	3077.68	16	16	0	2880.55	2879.22	-198.46	-6.45	2041.6	7.48
R103	13	2552.61	13	13	0	2291.96	2290.46	-262.15	-10.27	1566.13	12.72
R104	9	1884.64	9	9	0	1793.78	1786.38	-98.26	-5.21	984.96	21.13
R105	12	2374.95	12	12	0	2355.68	2350.07	-24.88	-1.05	1448.45	5.36
R106	10	2159.92	11	11	1	2069.39	2067.99	-91.93	-4.26	1344.8	14.86
R107	9	1949.83	9	9	0	1928.69	1908.63	-41.2	-2.11	1138.87	16.3
R108	8	1783.36	8.6	8	0	1766.51	1777.33	-6.03	-0.34	951.57	22.36
R109	10	2082.83	10	10	0	2077.85	2062.86	-19.97	-0.96	1220.26	10.59
R110	10	2038.77	10	10	0	1983.76	1979.01	-59.76	-2.93	1128.42	17.93
R111	9	2031.53	10	10	1	1952.36	1945.85	-85.68	-4.22	1114.7	18.82
R112	9	1847.86	9	9	0	1810.67	1808.22	-39.64	-2.15	966.33	23.8
R201	3	2742.74	3	3	0	2557.02	2555.93	-186.81	-6.81	1895.47	11.82
R202	3	2741.28	3	3	0	2325.32	2323.9	-417.38	-15.23	1682.95	25.27
R203	2	1934.89	2.2	2	0	1946.4	1918.38	-16.51	-0.85	1142.67	25.85
R204	2	1842.95	2	2	0	1715.5	1701.99	-140.96	-7.65	889.09	42
R205	3	2511.17	3	3	0	1999.81	1965.08	-546.09	-21.75	1186.37	23.6
R206	2	1860.42	2	2	0	1828.53	1813.46	-46.96	-2.52	1016.9	25.23
R207	2	1867.94	2	2	0	1728.86	1725.01	-142.93	-7.65	916.82	43.68
R208	2	1819.3	2	2	0	1583.58	1579.47	-239.83	-13.18	760.95	49.65
R209	2	1888.44	2	2	0	1906.36	1898.44	10	0.53	1096.18	22.1
R210	2	1963.58	2	2	0	1889.74	1882.36	-81.22	-4.14	1089.74	27.74
R211	2	1743.96	2	2	0	1701.14	1693.92	-50.04	-2.87	876.43	38.39
RC101	13	2732.94	13	13	0	2697.97	2685.78	-47.16	-1.73	1766.39	3.85
RC102	11	2429.2	11	11	0	2304.87	2281	-148.2	-6.1	1558.08	8.23
RC103	10	2167.45	10	10	0	2053.28	2046.26	-121.19	-5.59	1314.39	15.82
RC104	9	2028.15	9	9	0	1959.01	1950.88	-77.27	-3.81	1175.26	15.92
RC105	13	2729.54	13	13	0	2625.7	2606.43	-123.11	-4.51	1748.68	7.49
RC106	11	2288.51	11	11	0	2271.41	2264.22	-24.29	-1.06	1368.12	7.4
RC107	10	2175.7	10	10	0	2133.6	2121.74	-53.96	-2.48	1279.71	15.75
RC108	9	1992.13	9.38	9	0	1994.95	1992.13	0	0	1171.43	13.85
RC201	3	2655.77	3	3	0	2612.44	2609.51	-46.26	-1.74	1938.18	10.08
RC202	3	2649.95	3	3	0	2514.03	2511.48	-138.47	-5.23	1857.4	24.7
RC203	3	2673.93	3	3	0	2036.33	2006.26	-667.67	-24.97	1267.89	40.15
RC204	2	1910.14	2	2	0	1772.97	1762.08	-148.06	-7.75	938.11	41.79
RC205	3	2684.88	3	3	0	2486.95	2467.92	-216.96	-8.08	1774.1	23.91
RC206	3	2455.62	3	3	0	2234.35	2219.97	-235.65	-9.6	1526.41	23.78
RC207	3	2312.41	3	3	0	2111.8	2103.38	-209.03	-9.04	1402.99	36.28
RC208	2	1791.27	2	2	0	1810.53	1786.25	-5.02	-0.28	966.49	28.45
AVG	6.79	4495.14	6.84	6.82	0.04	4404.74	4397.8	-97.34	-3.93	1137.24	18.33
SUM	380	251727.98	383.18	382	2	246665.22	246276.96	-5451.02	-220.06	63685.24	1026.38

Table 16: TD-VRPTW - D1 - total time - results

Inst.	K_{II}^{best}	tot_{II}^{best}	\bar{K}	K_{best}	ΔK	\bar{tot}	tot_{best}	Δtot	$\Delta_{p}tot$	d_{best}	\bar{t}_e
C101	10	9699.3	10	10	0	9699.3	9699.3	0	0	828.94	1.19
C102	10	9699.3	10	10	0	9699.3	9699.3	0	0	828.94	9.3
C103	10	9833.85	10	10	0	9699.3	9699.3	-134.55	-1.37	828.94	17.82
C104	10	10306.29	10	10	0	9698.81	9698.81	-607.48	-5.89	829.7	20.99
C105	10	9699.3	10	10	0	9699.3	9699.3	0	0	828.94	2.77
C106	10	9699.3	10	10	0	9699.3	9699.3	0	0	828.94	3.55
C107	10	9699.3	10	10	0	9699.3	9699.3	0	0	828.94	4.34
C108	10	9699.26	10	10	0	9699.26	9699.26	0	0	828.94	6.65
C109	10	9699.26	10	10	0	9699.26	9699.26	0	0	828.94	8.07
C201	3	9493.26	3	3	0	9493.26	9493.26	0	0	591.56	2.08
C202	3	9493.26	3	3	0	9493.26	9493.26	0	0	591.56	15.64
C203	3	9517.43	3	3	0	9493.26	9493.26	-24.17	-0.25	591.56	27.89
C204	3	9492.66	3	3	0	9492.66	9492.66	0	0	590.6	42.71
C205	3	9490.71	3	3	0	9490.71	9490.71	0	0	588.88	4.17
C206	3	9490.47	3	3	0	9490.47	9490.47	0	0	588.49	6.74
C207	3	9561.6	3	3	0	9489.3	9489.3	-72.3	-0.76	588.32	8.04
C208	3	9791.65	3	3	0	9490.47	9490.47	-301.18	-3.08	588.49	8.38
R101	18	3361	18	18	0	3106.12	3100.14	-260.86	-7.76	1876.39	3.35
R102	17	3212.07	17	17	0	2812.13	2810.04	-402.03	-12.52	1836.87	8.29
R103	13	2533.94	13.17	13	0	2294.66	2276.83	-257.11	-10.15	1412.18	12.53
R104	9	1990.36	9	9	0	1865.01	1845.91	-144.45	-7.26	987.7	16.48
R105	14	2558.14	14	14	0	2424.38	2410.9	-147.24	-5.76	1455.97	4.15
R106	12	2325.18	12	12	0	2154.97	2146.48	-178.7	-7.69	1304.55	11.58
R107	10	2093.62	10	10	0	1961.36	1955.87	-137.75	-6.58	1102.55	20.93
R108	9	1911.16	9	9	0	1792.6	1788.5	-122.66	-6.42	933.61	22.71
R109	11	2137.08	11	11	0	2067.59	2065.36	-71.72	-3.36	1182.64	9.94
R110	10	2093.96	10	10	0	1961.24	1954.55	-139.41	-6.66	1121.16	18.13
R111	10	2055.36	10	10	0	1952.83	1949.01	-106.35	-5.17	1099.44	20.34
R112	9	1917.39	9	9	0	1819.95	1815.47	-101.92	-5.32	972.09	24.51
R201	4	3466.07	4	4	0	2687.21	2670.15	-795.92	-22.96	1803.53	7.35
R202	3	2728.41	3	3	0	2227.22	2223.07	-505.34	-18.52	1389.58	25.88
R203	3	2757.15	3	3	0	2057.74	2029.73	-727.42	-26.38	1174.78	36.55
R204	2	1968.43	2	2	0	1751.8	1743.77	-224.66	-11.41	869.95	37.44
R205	3	2490.53	3	3	0	2040.79	2030.18	-460.35	-18.48	1180.51	26.62
R206	2	1926.94	2	2	0	1930.91	1923.93	-3.01	-0.16	1093.76	28.87
R207	2	1914.52	2	2	0	1780.3	1778.07	-136.45	-7.13	906.57	38.78
R208	2	1979.63	2	2	0	1651.01	1637.86	-341.77	-17.26	745.55	48.15
R209	3	2358.02	3	3	0	2029.34	2006.58	-351.44	-14.9	1127.39	30.1
R210	3	2666.74	3	3	0	2019.61	2002.03	-664.71	-24.93	1128.28	38.08
R211	2	1779.11	2	2	0	1757.72	1752.92	-26.19	-1.47	885.47	39.55
RC101	14	2737.03	14.5	14	0	2665.52	2633.41	-103.62	-3.79	1734.27	3.76
RC102	12	2476.83	12.9	12	0	2399.31	2379.08	-97.75	-3.95	1581.5	7.77
RC103	10	2166.46	10	10	0	2107.32	2085.1	-81.36	-3.76	1288.12	9.22
RC104	9	2040.63	9	9	0	1967.69	1947.94	-92.69	-4.54	1142.12	10.74
RC105	13	2738.01	13.7	13	0	2536.19	2463.86	-274.15	-10.01	1645.92	7.96
RC106	11	2292.68	11.3	11	0	2250.46	2233.61	-59.07	-2.58	1450.35	5.65
RC107	10	2113.84	10	10	0	2098.71	2080.53	-33.31	-1.58	1280.76	11.13
RC108	10	2121.37	10	10	0	1978.62	1966.19	-155.18	-7.32	1135.65	19.2
RC201	4	3302.59	4	4	0	2758.94	2717.25	-585.34	-17.72	1885.38	8.2
RC202	3	2640.01	3	3	0	2311.78	2284.33	-355.68	-13.47	1488.67	22.01
RC203	3	2687.65	3	3	0	2081.38	2063.44	-624.21	-23.23	1241.74	40.63
RC204	2	1896.86	2	2	0	1821.59	1798.04	-98.82	-5.21	947.07	40.37
RC205	4	3424.69	4	4	0	2632.29	2605.13	-819.56	-23.93	1796.18	14.35
RC206	3	2343.87	3	3	0	2145.2	2134.69	-209.18	-8.92	1306.29	21.13
RC207	3	2247.05	3	3	0	2090.78	2079.81	-167.24	-7.44	1227.88	36.89
RC208	2	1838.54	2	2	0	1853.32	1821.34	-17.2	-0.94	1008.09	29.54
AVG	7.11	4601.06	7.15	7.11	0	4412	4400.67	-200.38	-7.11	1105.91	18.02
SUM	398	257659.12	400.57	398	0	247072.11	246437.62	-11221.5	-397.99	61931.19	1009.19

D TD-EVRPTW-FR travel time

Table 17: TD-EVRPTW - A1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	913.93	913.54	1071.43	1070.4	12430.67	1070.4	8	1.714
C102	11	11	917.14	908.25	1078.32	1067.76	12098.87	1067.76	9	3.005
C103	10.67	10	893.04	913.82	1049.79	1069.67	11906.21	1069.67	11	4.215
C104	10	10	826.89	814.03	1000.96	985.36	11113.01	985.36	9	5.442
C105	11	11	944.8	938.06	1084.57	1083.57	12134.54	1083.57	11	2.057
C106	11	11	916.08	915.2	1080.53	1080.98	11939.52	1080.98	11	2.388
C107	11	11	904.1	898.71	1071.15	1053.75	12100.76	1053.75	10	2.724
C108	10.89	10	879.1	1011.91	1048.11	1144.77	11810.69	1144.77	13	3.434
C109	10	10	884.55	851.71	1043.23	1000.97	11452.94	1000.97	9	2.063
C201	4	4	528.14	528.14	636.54	636.54	10442.03	636.54	3	2.231
C202	4	4	530.07	528.14	639.3	636.54	10440.15	636.54	3	6.667
C203	4	4	528.3	528.3	641.81	641.81	10509.96	641.81	4	8.464
C204	4	4	533.11	518.4	654.71	630.52	9984.38	630.52	3	12.512
C205	4	4	527.15	527.15	632.27	632.27	10117.82	632.27	3	3.414
C206	4	4	527.15	527.15	632.27	632.27	10117.82	632.27	3	4.694
C207	4	4	521.96	521.96	635.56	635.56	10202.88	635.56	3	6.351
C208	4	4	527.15	527.15	632.27	632.27	10117.82	632.27	3	5.036
R101	16.33	16	1369.22	1390.45	1628.31	1661.32	3310.77	1661.32	28	4.124
R102	15	15	1202.06	1200.77	1467.8	1469.68	3007.74	1469.68	22	9.059
R103	13	13	1043.2	1021.12	1272.68	1250.76	2661.69	1250.76	18	5.826
R104	10.67	10	874.08	885.2	1064.78	1064.07	2215.71	1064.07	14	10.794
R105	13.67	13	1138.77	1174.74	1386.99	1425.44	2712.84	1425.44	21	6.089
R106	12	12	1076.71	1055.32	1308.15	1285.97	2451.95	1285.97	17	3.262
R107	11	11	925.94	919.27	1131.95	1128.67	2282.32	1128.67	14	10.295
R108	10	10	835.58	834.73	1020.02	1023.31	2126.95	1023.31	13	12.553
R109	11.5	11	998.94	1023.06	1211.72	1232.69	2422.76	1232.69	18	6.283
R110	10.5	10	872.96	878.47	1076.15	1069.72	2173.05	1069.72	11	7.689
R111	11	11	888.85	877.26	1099.74	1083.78	2287.08	1083.78	14	5.364
R112	10	10	831.52	829.35	1016.45	1015.27	2144.78	1015.27	13	7.325
R201	3	3	1061.21	1055.41	1251.69	1243.81	2859.09	1243.81	8	13.14
R202	3	3	887.58	885.25	1062.62	1061.84	2883.18	1061.84	3	14.533
R203	3	3	758.7	756.94	910.68	904.55	2932.15	904.55	4	17.999
R204	2	2	635.52	629.88	800.11	790.06	1987.95	790.06	3	22.985
R205	3	3	784.94	782.38	1021.01	1017.32	2619.52	1017.32	4	12.579
R206	2	2	825.62	822.79	1014.36	1017.64	1958.73	1017.64	8	18.511
R207	2	2	668.17	667.84	849.14	848.45	1892.36	848.45	2	21.931
R208	2	2	589.81	587.14	767.16	771.11	1797.39	771.11	4	23.568
R209	2	2	749.57	743.6	933.34	926.39	1984.28	926.39	7	23.122
R210	2	2	716.13	707.99	892.96	878.97	1936.04	878.97	4	22.933
R211	2	2	656.34	652.68	837.12	837.93	1795.24	837.93	3	23.956
RC101	14.25	14	1389.32	1384.15	1698.19	1679.67	3000.41	1679.67	19	4.498
RC102	13	13	1315.75	1303.27	1586.54	1560.88	2764.73	1560.88	21	5.651
RC103	12	12	1122.23	1109.67	1341.89	1321.27	2615.43	1321.27	16	8.793
RC104	10.5	10	981.37	986.11	1207.34	1201.13	2223.74	1201.13	13	8.879
RC105	13	13	1174.42	1148.08	1461.3	1444.46	2732.96	1444.46	17	5.27
RC106	12	12	1122.22	1106.02	1393.47	1374.37	2532.54	1374.37	17	5.455
RC107	11	11	994.42	988.08	1250.6	1247.51	2335.3	1247.51	15	10.435
RC108	10.67	10	959.02	982.66	1210.33	1206.33	2235.41	1206.33	14	9.328
RC201	3	3	1347.16	1337.45	1631.51	1622.08	2856.02	1622.08	9	11.464
RC202	3	3	1115.36	1113.25	1358.62	1355.38	2796.71	1355.38	4	21.529
RC203	3	3	874.15	865.35	1073.93	1068.34	2716.07	1068.34	5	23.54
RC204	3	3	710.61	698.72	918.2	899.94	2581.47	899.94	6	23.681
RC205	3	3	981.52	966.77	1252.39	1237.22	2627.25	1237.22	7	14.088
RC206	3	3	919.11	912.38	1183.17	1192.96	2514.99	1192.96	4	14.675
RC207	3	3	752.58	739.6	995.37	969.24	2451.64	969.24	5	14.469
RC208	3	3	644.22	625.44	871.02	844.8	2243.62	844.8	4	18.737
AVG	7.51	7.41	876.74	875.36	1073.06	1069.1	5082	1069.1	9.73	10.37
SUM	420.65	415	49097.54	49020.26	60091.62	59869.34	284591.93	59869.34	545	580.82

Table 18: TD-EVRPTW - A2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	835.29	835.08	1071.95	1072	12435.54	1072	8	1.631
C102	11	11	832.7	825.61	1076.29	1061.67	12186.51	1061.67	8	2.865
C103	10.11	10	835.7	818.12	1072.06	1055.29	11546.63	1055.29	9	2.812
C104	10	10	739.93	730.73	1002.51	986.35	11012.95	986.35	9	5.424
C105	11	11	842.58	838.49	1100.05	1103.87	11860.13	1103.87	10	1.93
C106	11	11	829.12	823.41	1112.81	1115.23	11717.28	1115.23	12	2.796
C107	10.89	10	819.89	883.45	1085.65	1119.17	11666.47	1119.17	10	3.025
C108	10.44	10	822.78	839.49	1098.51	1089.27	11742.11	1089.27	12	2.381
C109	10	10	798.23	770.42	1071.38	1053.72	11053.65	1053.72	8	2.092
C201	4	4	459.04	459.04	636.54	636.54	10410.21	636.54	3	2.415
C202	4	4	459.6	459.6	646.84	646.84	10482.22	646.84	5	7.438
C203	4	4	456.77	456.77	643.13	643.13	10470.97	643.13	4	8.95
C204	4	4	449.03	445.41	641.78	631.84	9944.96	631.84	3	12.466
C205	4	4	457.67	457.67	629.95	629.95	10114.71	629.95	3	3.885
C206	4	4	452.66	452.66	644.47	644.47	10257.51	644.47	4	5.671
C207	4	4	446.99	446.99	633.69	633.69	10224.16	633.69	3	6.253
C208	4	4	452.66	452.66	644.47	644.47	10224.51	644.47	4	6.23
R101	15.5	15	1206.71	1270.99	1656.3	1763.32	3133.29	1763.32	34	5.601
R102	14	14	1008.54	1006.44	1415.55	1415.32	2766.04	1415.32	21	9.302
R103	11	11	881.67	881.6	1241.69	1241.3	2360.67	1241.3	22	8.044
R104	10	10	727.31	726.09	1040.4	1033.64	2078.18	1033.64	12	12.68
R105	13	13	961.76	937.91	1374.85	1346.92	2636.09	1346.92	18	7.446
R106	12	12	867.89	866.6	1250.86	1248.54	2334.82	1248.54	17	11.57
R107	10	10	784.6	783.14	1116.81	1114.64	2127.36	1114.64	13	6.021
R108	10	10	698.77	698.12	1015.65	1011.19	2011.64	1011.19	14	14.837
R109	11	11	832.12	825.92	1187.39	1182.76	2267.88	1182.76	15	9.051
R110	10	10	726.73	718.61	1047.04	1028.2	2077.53	1028.2	11	9.941
R111	10	10	746.82	743.6	1060.08	1057.2	2121.16	1057.2	16	6.436
R112	10	10	678.73	678.53	995.2	999.14	1987.15	999.14	11	13.599
R201	3	3	922.52	914.92	1268.87	1277.08	2833.49	1277.08	9	15.92
R202	3	3	794.98	793.98	1067.45	1064.47	2898.67	1064.47	5	18.751
R203	2	2	774.44	768.07	1073.94	1070.87	1985.83	1070.87	6	19.478
R204	2	2	545.41	544.01	813.53	816.33	1930.42	816.33	3	29.343
R205	2	2	742.33	736.51	1070.72	1074.28	1929.13	1074.28	7	13.743
R206	2	2	683.91	679.01	1021.11	1008.1	1876.26	1008.1	8	21.729
R207	2	2	552.71	545.67	864.55	856.83	1782.23	856.83	3	18.657
R208	2	2	473.04	468.97	765.42	762.59	1725.71	762.59	3	30.401
R209	2	2	637.39	632.67	947.38	943.18	1932.05	943.18	8	24.245
R210	2	2	604.14	604.14	918.36	918.36	1894.45	918.36	5	19.873
R211	2	2	509.36	509.36	809.96	809.96	1746.81	809.96	5	23.137
RC101	13	13	1151.25	1139.14	1674.16	1643.9	2701.69	1643.9	22	4.245
RC102	12.5	12	1071.55	1059.87	1558.17	1536.49	2512.38	1536.49	23	7.355
RC103	11	11	952.09	944.22	1364.39	1365.18	2282.06	1365.18	16	7.216
RC104	10	10	821.84	812.91	1213.77	1189.68	2075.72	1189.68	14	9.392
RC105	12	12	952.31	940.72	1410.71	1397.93	2399.94	1397.93	13	6.656
RC106	11.75	11	928.87	1011.01	1393.42	1434.91	2409.87	1434.91	19	5.186
RC107	10.67	10	804.3	827.37	1227.35	1211.33	2171.79	1211.33	15	10.794
RC108	10	10	778.53	769.07	1164.45	1152.19	2046.74	1152.19	13	10.957
RC201	3	3	1144.39	1142.81	1585.69	1586.49	2811.08	1586.49	11	14.955
RC202	3	3	942.87	940.36	1340.12	1342.16	2743.17	1342.16	8	22.004
RC203	3	3	752.98	747.04	1089.95	1086.71	2654.33	1086.71	4	22.898
RC204	2	2	727.6	697.6	1045.92	1009.65	1908.42	1009.65	8	20.981
RC205	3	3	775.42	754.24	1208.71	1161.77	2640.85	1161.77	6	13.086
RC206	3	3	720.73	720.73	1143.48	1143.48	2478.29	1143.48	6	12.953
RC207	2.33	2	710.1	751	1051.87	1077.58	1895.09	1077.58	7	15.407
RC208	2	2	671.68	641.47	984.75	952.85	1819.89	952.85	8	24.497
AVG	7.16	7.09	754.63	754.11	1076.64	1073.29	4916.76	1073.29	10.25	11.37
SUM	401.19	397	42259.03	42230.02	60292.1	60104.02	275338.69	60104.02	574	636.65

Table 19: TD-EVRPTW - A3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{i}_e
C101	11	11	834.78	826.35	1131.84	1115.27	12274.95	1115.27	11	1.604
C102	11	11	780.78	770.7	1089.81	1112.6	11901.26	1112.6	9	2.669
C103	10	10	785.33	754.68	1071.78	1040.76	11509.75	1040.76	9	2.914
C104	10	10	680.86	675.98	998.47	1006.81	10993.97	1006.81	9	4.435
C105	11	11	785.13	777.17	1107.52	1098.28	11794.34	1098.28	11	2.119
C106	11	11	764.33	760.44	1129.43	1126.01	11641.19	1126.01	12	2.637
C107	10.89	10	763.9	831.79	1109.39	1123.25	11613.13	1123.25	10	2.884
C108	10.11	10	777.97	767.8	1123.14	1098.83	11563.25	1098.83	11	1.819
C109	10	10	726.01	702.14	1059.95	1023.55	11073.05	1023.55	8	2.203
C201	4	4	419.53	419.53	636.54	636.54	10391.79	636.54	3	2.394
C202	4	4	418.35	418.35	646.84	646.84	10449.98	646.84	5	7.853
C203	4	4	417.7	415.95	648.79	643.13	10439.14	643.13	4	9.217
C204	4	4	411.87	408.26	680.36	674.16	11102.78	674.16	6	12.42
C205	4	4	417.25	417.25	636.96	636.96	9937.53	636.96	3	3.792
C206	4	4	412.16	412.16	646.51	646.51	10234.08	646.51	4	5.79
C207	4	4	407.56	407.56	635.73	635.73	10200.4	635.73	3	6.595
C208	4	4	410.42	410.42	641.36	641.36	10227.85	641.36	3	6.148
R101	14.5	14	1048.94	1064.36	1606.19	1653.79	2896.19	1653.79	27	5.292
R102	13	13	899.91	896.2	1430.99	1409.49	2651.08	1409.49	21	9.591
R103	11	11	765.19	755.67	1230.61	1221.41	2253.83	1221.41	21	8.322
R104	10	10	652.66	648.28	1088.42	1084.1	1986.02	1084.1	13	15.763
R105	12	12	861.63	850.18	1392.92	1359.75	2469.54	1359.75	24	5.731
R106	11	11	796.68	788.53	1280.81	1280.28	2260.17	1280.28	22	10.42
R107	10	10	684.64	670.41	1135.65	1098.2	2041.37	1098.2	15	7.967
R108	9	9	649.96	633.54	1036.15	1010.72	1914.66	1010.72	15	8.119
R109	10	10	768.63	767.1	1197.37	1199.58	2145.91	1199.58	17	6.737
R110	10	10	635.89	628.55	1065.26	1048.79	1974.18	1048.79	14	12.563
R111	10	10	665.73	656.3	1093.89	1081.47	2047.52	1081.47	15	4.801
R112	9	9	625.35	608.45	998.69	975.92	1900.61	975.92	13	6.473
R201	3	3	839.29	839.29	1266.9	1266.9	2833.49	1266.9	6	13.98
R202	3	3	725.34	722.32	1076.61	1063.54	2897.27	1063.54	3	18.91
R203	2	2	699.13	694.32	1087.92	1074.04	1982.78	1074.04	7	29.388
R204	2	2	485.85	485.27	832.27	828.99	1927.44	828.99	3	22.208
R205	2	2	648.64	643.23	1085.78	1087.4	1896.07	1087.4	9	19.458
R206	2	2	585.5	579.96	1020.55	1014.29	1881.38	1014.29	10	23.029
R207	2	2	463.11	459.09	898.98	891.4	1788.97	891.4	3	25.56
R208	2	2	405.66	400.74	800.81	779.5	1675.93	779.5	3	23.362
R209	2	2	536.09	519.98	946.59	937.48	1786.79	937.48	8	21.702
R210	2	2	519.05	519.05	897.91	897.91	1931.5	897.91	9	29.07
R211	2	2	434.23	434.23	820.04	820.04	1721.4	820.04	3	18.44
RC101	12.67	12	961.82	1013.21	1622.46	1646.19	2536.06	1646.19	24	5.611
RC102	11.33	11	963.81	1002.1	1537.35	1561.9	2434.2	1561.9	26	6.063
RC103	11	11	802.57	786.75	1354.56	1336.61	2212.34	1336.61	16	10.158
RC104	10	10	700.5	699.41	1219.88	1239.82	1939.18	1239.82	14	11.354
RC105	12	12	812.52	800.29	1408.32	1400.27	2305.36	1400.27	15	5.128
RC106	11	11	796.62	784.91	1384.04	1352.84	2161.35	1352.84	17	5.832
RC107	10	10	708.9	704.14	1224.25	1213.28	2002.7	1213.28	14	8.517
RC108	10	10	660.69	654.62	1162.76	1150.93	1947.21	1150.93	14	11.961
RC201	3	3	1015.14	1005.05	1572.59	1564.4	2773.65	1564.4	11	15.027
RC202	3	3	838.44	830.79	1332.86	1332.56	2723.77	1332.56	6	18.74
RC203	3	3	674.27	670.35	1080.95	1076.96	2683.11	1076.96	5	22.598
RC204	2	2	617.44	588.99	1003.65	961.31	1875.95	961.31	9	27.83
RC205	3	3	647.99	646.87	1233.58	1236.48	2556.67	1236.48	7	13.747
RC206	3	3	628.81	621.64	1215.24	1166.96	2409.02	1166.96	6	11.449
RC207	2	2	656.45	646.81	1093.36	1075.7	1859.37	1075.7	8	24.099
RC208	2	2	550.91	545.23	947.73	935.26	1800.55	935.26	8	24.204
AVG	6.96	6.91	672.28	668.62	1083.56	1075.77	4864.88	1075.77	10.75	11.55
SUM	389.5	387	37647.91	37442.74	60679.31	60243.05	272433.03	60243.05	602	646.7

Table 20: TD-EVRPTW - B1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{i}_e
C101	12	12	798.16	798.16	1066.49	1066.49	13470.4	1066.49	8	1.799
C102	11	11	813.5	799.33	1089.6	1073.5	12823.46	1073.5	8	2.131
C103	10	10	769.24	756.08	1038.84	1027.11	11614.34	1027.11	8	3.041
C104	10	10	661.44	661.04	908.08	905.03	11262.88	905.03	6	5.244
C105	11	11	801.39	799.67	1075.04	1075.71	12568.28	1075.71	8	1.639
C106	11	11	808.23	807.45	1067.8	1072.13	12521.3	1072.13	8	2.305
C107	10.57	10	824.03	855.89	1081.57	1114.85	11746.35	1114.85	10	2.144
C108	10.71	10	795.04	851.66	1060.09	1137.4	11774.77	1137.4	10	2.906
C109	10	10	781.56	772.61	1031.41	1013.82	11832.45	1013.82	10	2.05
C201	4	4	526.44	526.44	636.5	636.5	10332.05	636.5	4	2.303
C202	4	4	523.14	523.14	634.47	634.47	10508.22	634.47	4	6.242
C203	4	4	521.86	521.4	637.76	634.63	10622.97	634.63	3	8.451
C204	4	4	521.22	520.91	632.37	631.92	10687.38	631.92	4	12.534
C205	4	4	520.13	520.13	630.12	630.12	10353.09	630.12	4	3.797
C206	4	4	520.13	520.13	630.12	630.12	10353.09	630.12	4	4.913
C207	4	4	520.13	520.13	630.12	630.12	10353.09	630.12	4	6.071
C208	4	4	520.13	520.13	630.12	630.12	10353.09	630.12	4	5.177
R101	18	18	1308.8	1291.46	1663.03	1643.12	3663.38	1643.12	22	3.684
R102	16	16	1138.4	1138.04	1459.76	1458.61	3294.45	1458.61	22	8.451
R103	12.67	12	996.2	1014.02	1268.6	1286.4	2613.13	1286.4	20	8.42
R104	10.67	10	850.42	869.23	1072.02	1087.6	2154.32	1087.6	13	10.304
R105	14	14	1111.72	1101.73	1395.42	1384.32	2882.75	1384.32	22	6.348
R106	13	13	1025.16	1019.15	1293.93	1279.26	2755.43	1279.26	19	8.582
R107	11	11	903.14	901.83	1145.18	1127.23	2347.37	1127.23	15	7.069
R108	10	10	825.55	811.14	1043.22	1023.18	2144.56	1023.18	14	8.587
R109	12	12	969.49	965.56	1225.71	1221.7	2565.72	1221.7	19	6.462
R110	11	11	870.17	868.61	1097.94	1091.52	2316.53	1091.52	15	13.446
R111	11	11	879.88	876.67	1113.9	1111.36	2370.25	1111.36	13	9.205
R112	11	11	819.53	818.14	1035.28	1034.79	2331.61	1034.79	14	14.106
R201	3	3	958.28	957.19	1212.12	1202.68	2785.6	1202.68	8	11.395
R202	3	3	802.76	802.76	1028.41	1028.41	2850.3	1028.41	4	17.912
R203	2	2	773.47	764.59	982.05	974.4	1981.83	974.4	6	25.417
R204	2	2	610.4	604.27	774.09	775.46	1987.04	775.46	2	22.518
R205	3	3	797.26	793.48	1012.72	1008.89	2753.16	1008.89	5	10.246
R206	2	2	815.92	815.74	1020.04	1018.32	1986.28	1018.32	7	15.456
R207	2	2	690.58	682.97	869.2	859.6	1975.27	859.6	2	19.864
R208	2	2	592.92	591.34	754.08	757.61	1975.04	757.61	4	22.271
R209	2	2	759.9	751.15	951.1	948.48	1973.2	948.48	6	13.68
R210	2	2	752.47	748.99	952.42	941.97	1952.38	941.97	5	17.516
R211	2	2	695.09	674.85	872.21	834.45	1949.29	834.45	3	21.798
RC101	15	15	1359.13	1353.78	1711.41	1705.17	3227.53	1705.17	24	5.423
RC102	13.25	13	1229.32	1242.18	1553.9	1567.38	2868.41	1567.38	21	6.846
RC103	12	12	1070.07	1054.98	1377.82	1365.45	2577.95	1365.45	18	8.521
RC104	10.25	10	921.33	898.82	1197.09	1166.61	2286.44	1166.61	13	6.415
RC105	13	13	1178.43	1157.94	1480.06	1463.58	2871.61	1463.58	16	4.585
RC106	13	13	1110.69	1103.03	1402	1393.84	2805.91	1393.84	16	6.525
RC107	11	11	1005.88	980.66	1273.86	1252.93	2470.9	1252.93	15	6.417
RC108	11	11	927.43	918.15	1177.98	1166.44	2410.25	1166.44	13	8.237
RC201	3	3	1337.85	1331.52	1635	1638.05	2769.82	1638.05	9	8.576
RC202	3	3	1136.5	1134.17	1397.1	1395.15	2769.17	1395.15	8	18.976
RC203	3	3	853.39	852.25	1091.52	1081.47	2757.67	1081.47	6	20.967
RC204	3	3	719.33	708.44	904	908.13	2781.04	908.13	6	23.327
RC205	3	3	1053.85	1031.26	1283.24	1235.94	2640.07	1235.94	6	13.203
RC206	3	3	980.82	967.29	1197.06	1189.23	2647.02	1189.23	6	13.878
RC207	3	3	829.85	826.44	1041.91	1032.89	2669.25	1032.89	5	16.686
RC208	3	3	737.22	707.52	900.95	874.15	2386.87	874.15	5	21.299
AVG	7.61	7.55	850.44	846.53	1077.64	1072.85	5209.39	1072.85	9.89	10.1
SUM	426.12	423	47624.37	47405.64	60347.83	60079.81	291726.01	60079.81	554	565.37

Table 21: TD-EVRPTW - B2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	677.25	674.89	1095.65	1076.06	13514.39	1076.06	8	3.139
C102	11	11	663.76	650.36	1085.26	1059.12	12645.83	1059.12	9	5.624
C103	10	10	660.2	640.32	1070.94	1042.34	11818.24	1042.34	9	5.182
C104	10	10	545.28	542.88	898.13	893.36	11624.8	893.36	8	8.74
C105	11	11	664.92	659.1	1078.64	1071.66	12405.05	1071.66	8	3.387
C106	10.9	10	682.45	768.02	1095.97	1203.22	11744.79	1203.22	11	4.89
C107	10.6	10	682.63	708.31	1097.96	1119.7	11734.95	1119.7	11	4.569
C108	10.3	10	674.85	678.7	1089.44	1092.14	11958.95	1092.14	12	3.839
C109	10	10	655.33	633.45	1049.63	1016.93	11423.17	1016.93	8	3.505
C201	4	4	458.82	458.82	641.93	641.93	10304.51	641.93	5	5.074
C202	4	4	463.26	452.29	665.99	636.75	10488.13	636.75	4	14.155
C203	4	4	449.08	449.08	639.15	639.15	10701.98	639.15	3	16.059
C204	4	4	446.95	443.15	662.26	652.83	11291.37	652.83	5	25.969
C205	4	4	447.25	447.25	634.87	634.87	10302.32	634.87	5	7.737
C206	4	4	447	447	638.15	638.15	10304.08	638.15	5	10.298
C207	4	4	447.57	447.57	636.57	636.57	10322.73	636.57	4	11.952
C208	4	4	450.05	446.97	647.88	640.62	10309.7	640.62	5	12.447
R101	17	17	1148.09	1125.09	1702.42	1670.69	3489.48	1670.69	27	5.584
R102	14.75	14	975.48	1039.26	1469.47	1536.28	2884.1	1536.28	26	7.357
R103	12.25	12	825.9	820.38	1258.74	1244.3	2523.48	1244.3	18	7.993
R104	10	10	734.25	712.15	1110.65	1069.29	2145.57	1069.29	12	6.597
R105	13.25	13	961.7	934.99	1432.53	1388.86	2704.14	1388.86	25	4.516
R106	12	12	881.44	863.71	1331.97	1308.28	2527.21	1308.28	22	5.981
R107	10	10	747.99	721.48	1141.9	1104.99	2146.18	1104.99	15	5.268
R108	10	10	682.8	658.19	1060.33	1027.21	2151.83	1027.21	14	6.618
R109	11	11	822.89	809.99	1251.91	1236.22	2390.58	1236.22	23	5.424
R110	10	10	761.43	758.11	1145.52	1153.78	2188.42	1153.78	18	5.028
R111	11	11	763.94	754.85	1161.72	1147.98	2370.49	1147.98	17	5.625
R112	10	10	720.77	688.57	1099.11	1057.37	2165.92	1057.37	17	5.17
R201	3	3	807.14	805.89	1233.2	1234.73	2780.36	1234.73	7	11.067
R202	2	2	785.51	766.45	1180.77	1148.59	1954.33	1148.59	5	5.2
R203	2	2	641.95	636.26	986.72	995.43	1974.86	995.43	5	9.878
R204	2	2	509.98	499.3	796.18	794.82	1981.91	794.82	4	11.516
R205	2	2	728.86	713.79	1092.77	1073.2	1961.69	1073.2	6	5.24
R206	2	2	677.32	666.6	1036.37	1024.02	1988.71	1024.02	6	8.575
R207	2	2	564.42	558	875.51	858.27	1981.25	858.27	3	10.593
R208	2	2	483.05	481.16	765.07	766.47	1980.49	766.47	4	11.726
R209	2	2	621.6	617.23	943.98	934.67	1963.2	934.67	5	8.43
R210	2	2	617.64	591.63	936.27	883.87	1889.16	883.87	4	8.947
R211	2	2	567.77	561.54	888.09	869.33	1913.8	869.33	4	9.913
RC101	14	14	1091.89	1079.72	1637.09	1622.86	3074.96	1622.86	24	11.71
RC102	12.25	12	1015.68	1018.38	1530.84	1537.61	2641.69	1537.61	21	17.418
RC103	11	11	908.33	852.45	1416.14	1340.24	2412.5	1340.24	15	12.793
RC104	10	10	762.12	732.97	1192.15	1165.36	2173.53	1165.36	13	18.372
RC105	12.25	12	969.09	973.24	1470.38	1470.79	2690.14	1470.79	16	9.707
RC106	12	12	926.48	915.41	1395.1	1376.76	2565.25	1376.76	16	8.356
RC107	11	11	814.79	799.56	1259.79	1243.41	2342.07	1243.41	13	19.347
RC108	10	10	777.49	762.89	1176.45	1159.16	2239.43	1159.16	13	14.892
RC201	3	3	1076.39	1069.34	1599.39	1565.94	2710.43	1565.94	8	26.446
RC202	3	3	929.89	926.99	1370.82	1356.57	2776.66	1356.57	6	39.953
RC203	3	3	722.25	714.2	1104.64	1088	2642.94	1088	6	45.771
RC204	3	3	599.41	596.86	910.78	908.97	2669.39	908.97	6	51.846
RC205	3	3	820.03	815.94	1256.45	1273.81	2692.63	1273.81	6	30.521
RC206	3	3	822.61	807.84	1222.87	1192.89	2635.43	1192.89	5	22.67
RC207	2.75	2	703.59	762.74	1065.93	1117.15	1890.08	1117.15	7	35.753
RC208	2	2	670.91	664.49	995.5	980.75	1861.75	980.75	7	46.697
AVG	7.27	7.2	717.13	711.17	1093.46	1082.58	5088.77	1082.58	10.52	13.05
SUM	407.3	403	40159.52	39825.8	61233.94	60624.32	284971.03	60624.32	589	731.06

Table 22: TD-EVRPTW - B3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	589.01	589.01	1104.31	1104.31	13301.72	1104.31	10	2.82
C102	10.5	10	622.55	647.39	1149.91	1169.41	11867.75	1169.41	12	3.075
C103	10	10	616.61	582.82	1164.55	1101.48	11746.23	1101.48	11	2.577
C104	10	10	475.39	470.06	918.69	906.27	11450.97	906.27	7	4.551
C105	11	11	582.6	574.86	1103.74	1081.74	12294.24	1081.74	10	2.527
C106	11	11	603.56	586.4	1114.13	1097.83	12450.4	1097.83	10	3.282
C107	10.67	10	599.79	637.54	1093.82	1128.96	11625.68	1128.96	11	2.745
C108	10.67	10	571.93	587.97	1088.42	1123.05	11723.91	1123.05	11	3.077
C109	10	10	603.31	580.53	1122.09	1088.64	11498.1	1088.64	10	2.013
C201	4	4	413.74	413.63	650.84	648.54	10635.55	648.54	6	4.577
C202	4	4	405.75	405.75	644.68	644.68	11133.69	644.68	5	12.441
C203	4	4	405.19	405.19	649.09	649.09	11205.82	649.09	4	14.901
C204	4	4	402.63	394.24	681.66	656.79	11141.28	656.79	4	19.204
C205	4	4	402.65	402.65	648.29	648.29	10763.89	648.29	5	8.613
C206	4	4	406.58	406.58	648.57	648.57	10294.18	648.57	6	11.529
C207	4	4	406.85	406.85	640.04	640.04	10297.26	640.04	4	12.759
C208	4	4	403.28	403.28	654.53	654.53	10866.14	654.53	5	11.804
R101	16.5	16	1029.46	1036.23	1723.68	1714.98	3298.06	1714.98	32	10.857
R102	14	14	871.43	867.28	1483.45	1473.35	2891.86	1473.35	27	25.016
R103	12	12	702.47	700.01	1238.41	1236.7	2515.61	1236.7	19	26.025
R104	10	10	620.15	620.15	1092	1092	2142.86	1092	16	20.03
R105	13	13	820.72	820.72	1380.05	1380.05	2732.08	1380.05	23	8.339
R106	12	12	737.26	737.26	1275.35	1275.35	2447.63	1275.35	19	7.933
R107	10	10	696.56	696.56	1243.03	1243.03	2137.41	1243.03	20	8.352
R108	9	9	599.05	599.05	1065.34	1065.34	1943.03	1065.34	15	11.341
R109	11	11	763.97	763.97	1307.98	1307.98	2365.95	1307.98	24	7.172
R110	10	10	704.22	704.22	1227.86	1227.86	2189.51	1227.86	21	5.813
R111	10	10	688.26	688.26	1183.62	1183.62	2210.53	1183.62	22	8.488
R112	10	10	597.5	597.5	1039.78	1039.78	2141.61	1039.78	16	7.489
R201	3	3	703.51	703.51	1239.15	1239.15	2726.15	1239.15	7	19.414
R202	2	2	654.9	654.9	1149.06	1149.06	1956.05	1149.06	5	19.961
R203	2	2	547.48	547.48	981.29	981.29	1990.93	981.29	5	28.564
R204	2	2	430.55	430.55	821.31	821.31	1983.09	821.31	4	30.558
R205	2	2	614.73	614.73	1088.94	1088.94	1935.52	1088.94	5	14.986
R206	2	2	550.53	550.53	1012.72	1012.72	1969.09	1012.72	5	33.19
R207	2	2	498.15	480.53	905.22	898.57	1981.53	898.57	4	16.793
R208	2	2	427.46	415.75	790.76	763.93	1973.29	763.93	3	19.071
R209	2	2	553.83	551.16	997.71	1004.17	1941.49	1004.17	4	7.312
R210	2	2	555.01	533.11	994.06	976.62	1930.34	976.62	4	7.859
R211	2	2	481.58	475.19	883.69	871.28	1926.33	871.28	5	10.235
RC101	13.67	13	986.07	1035.74	1671.72	1743.05	2803.3	1743.05	23	11.308
RC102	12	12	890.57	866.64	1565.09	1526.12	2566.72	1526.12	20	15.059
RC103	10.33	10	780.26	793.93	1390.82	1409.84	2223.26	1409.84	20	13.775
RC104	10	10	636.24	628.34	1165.29	1150.95	2187.78	1150.95	14	23.42
RC105	12	12	856.58	847.61	1471.66	1442.1	2607.63	1442.1	19	11.206
RC106	12	12	793.54	793.54	1387.67	1387.67	2559.6	1387.67	15	9.498
RC107	11	11	750.22	750.22	1311.19	1311.19	2401.92	1311.19	17	12.12
RC108	10	10	707.06	707.06	1264.35	1264.35	2273.13	1264.35	18	9.183
RC201	3	3	935.03	935.03	1580.1	1580.1	2724.91	1580.1	11	22.51
RC202	3	3	815.17	815.17	1413.12	1413.12	2700.08	1413.12	6	30.179
RC203	3	3	625.87	625.87	1101.06	1101.06	2754.96	1101.06	5	33.235
RC204	2	2	655.38	655.38	1093.28	1093.28	1895.57	1093.28	7	28.053
RC205	3	3	724.31	724.31	1259.26	1259.26	2631.18	1259.26	6	23.304
RC206	3	3	704.63	704.63	1266.05	1266.05	2606.54	1266.05	8	17.901
RC207	2	2	710.25	663.53	1219.94	1151.83	1890.42	1151.83	6	9.841
RC208	2	2	604.12	585.11	1040.31	1000.78	1855.49	1000.78	8	10.504
AVG	7.15	7.09	634.56	632.42	1114.33	1109.64	5076.95	1109.64	11.41	13.54
SUM	400.34	397	35535.5	35415.51	62402.73	62140.05	284309.25	62140.05	639	758.39

Table 23: TD-EVRPTW - C1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tol_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	850.75	850.75	1053.83	1053.83	12879.87	1053.83	8	2.281
C102	11	11	853.75	848.6	1065.53	1056.88	12374.46	1056.88	8	4.001
C103	10	10	835.81	816.27	1018.16	994.25	11695.39	994.25	9	2.647
C104	10	10	740.6	734.41	913.24	901.06	11167.9	901.06	6	5.243
C105	11	11	872	871.09	1074.65	1073.63	12227.48	1073.63	9	2.998
C106	11	11	854.55	854.55	1058.43	1058.43	12021.07	1058.43	9	3.828
C107	11	11	830.68	830.68	1028.64	1028.64	12106.63	1028.64	8	3.882
C108	10.67	10	843.21	882.36	1040.89	1084.89	11800.07	1084.89	12	4.717
C109	10	10	820.43	811.8	1014.49	1000.08	11537.16	1000.08	9	3.58
C201	4	4	497.6	497.6	635.92	635.92	10088.67	635.92	4	5.365
C202	4	4	496.93	496.93	635.24	635.24	10299.64	635.24	4	15.186
C203	4	4	494.41	494.41	636.42	636.42	10372.23	636.42	4	15.608
C204	4	4	491.61	491.5	631.18	628.91	10318.82	628.91	3	17.766
C205	4	4	490.89	490.89	629.95	629.95	10167.76	629.95	3	8.447
C206	4	4	490.89	490.89	629.95	629.95	10163.37	629.95	3	10.522
C207	4	4	490.89	490.89	629.95	629.95	10163.37	629.95	3	12.854
C208	4	4	490.89	490.89	629.95	629.95	10163.37	629.95	3	9.029
R101	18	18	1330.06	1330.06	1631.15	1631.15	3776.95	1631.15	25	6.773
R102	15	15	1174.88	1174.88	1474.34	1474.34	3075.76	1474.34	23	13.469
R103	12	12	990.2	990.2	1246.54	1246.54	2589.87	1246.54	18	15.52
R104	10	10	893.07	893.07	1106.93	1106.93	2220.38	1106.93	16	12.812
R105	14	14	1122.33	1122.33	1364.31	1364.31	2978.81	1364.31	22	9.133
R106	12	12	1058.13	1058.13	1308.54	1308.54	2529.49	1308.54	20	6.929
R107	11	11	948.11	948.11	1163.02	1163.02	2378.11	1163.02	14	8.31
R108	10	10	822.33	822.33	1022.65	1022.65	2192.41	1022.65	14	15.423
R109	12	12	991.33	991.33	1217.96	1217.96	2587.61	1217.96	18	8.198
R110	11	11	879.32	879.32	1094.08	1094.08	2362.05	1094.08	13	13.238
R111	11	11	888.2	888.2	1104.4	1104.4	2303.37	1104.4	15	13.243
R112	10	10	835.3	835.3	1041.05	1041.05	2168.12	1041.05	14	12.975
R201	3	3	1046.65	1046.65	1230.48	1230.48	2912.47	1230.48	8	18.543
R202	3	3	861.29	861.29	1032.94	1032.94	2865.48	1032.94	4	23.418
R203	3	3	750.47	750.47	916.03	916.03	2883.51	916.03	4	27.024
R204	2	2	630.4	630.4	779.21	779.21	1991.06	779.21	2	25.455
R205	3	3	816.11	816.11	994.52	994.52	2717.91	994.52	4	14.202
R206	2	2	840.21	835.78	1023.19	1010.33	1991.44	1010.33	6	16.36
R207	2	2	710.83	702.3	873.61	871.23	1930.31	871.23	3	14.524
R208	2	2	607.35	605.01	765.25	757.19	1877.63	757.19	3	18.399
R209	2	2	807.5	801.42	973.87	965.45	1991.69	965.45	7	7.307
R210	2	2	804.46	802.31	973.03	978.53	1987.9	978.53	8	8.032
R211	2	2	695.67	689.48	860.39	854.43	1802.94	854.43	3	8.526
RC101	15	15	1453.92	1419.91	1769.71	1731.37	3266.47	1731.37	21	9.078
RC102	13.5	13	1232.41	1243.81	1518.19	1526.12	2813.43	1526.12	17	14.813
RC103	12	12	1088.82	1084.23	1353.43	1350.01	2587.3	1350.01	14	13.765
RC104	10	10	961.99	940.78	1196.06	1172.08	2245.34	1172.08	14	12.107
RC105	13	13	1175.83	1173.9	1447.02	1444.42	2912.31	1444.42	16	11.512
RC106	12	12	1138.87	1132.57	1402.42	1400.31	2586.01	1400.31	15	10.987
RC107	11	11	1001.16	1001.16	1235.33	1235.33	2409.84	1235.33	15	12.489
RC108	11	11	994.95	994.95	1237.76	1237.76	2356.54	1237.76	13	9.408
RC201	4	4	1202.58	1202.58	1452.65	1452.65	3615.47	1452.65	4	11.768
RC202	3	3	1162.56	1162.56	1419.47	1419.47	2827.37	1419.47	8	29.139
RC203	3	3	893.44	893.44	1088.55	1088.55	2756.96	1088.55	6	31.288
RC204	3	3	698.59	698.59	900.77	900.77	2570.87	900.77	6	32.047
RC205	3	3	1083.75	1083.75	1306.6	1306.6	2778.63	1306.6	7	19.358
RC206	3	3	976.03	976.03	1173.48	1173.48	2638.63	1173.48	6	17.975
RC207	3	3	813.77	813.77	990.07	990.07	2516.07	990.07	6	21.461
RC208	3	3	719.52	680.22	885.08	838.41	2236.7	838.41	5	21.349
AVG	7.56	7.54	866.93	864.66	1069.65	1066.8	5156.79	1066.8	9.68	12.93
SUM	423.17	422	48548.28	48421.24	59900.5	59740.69	288780.47	59740.69	542	724.31

Table 24: TD-EVRPTW - C2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	tt_{best}	\bar{d}	d_{best}	tol_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	746.88	746.88	1054.63	1054.63	12838.46	1054.63	8	1.01
C102	11	11	746.71	745.41	1063.01	1064.87	12231.48	1064.87	8	1.633
C103	10	10	744.67	736.76	1039.74	1009.15	11487.17	1009.15	8	1.23
C104	10	10	671.61	665.15	943.32	942.54	11367.25	942.54	8	2.319
C105	11	11	758.36	755.98	1082.89	1071.07	12274.16	1071.07	9	1.259
C106	11	11	744.86	744.86	1060.1	1060.1	12040.27	1060.1	9	1.763
C107	11	11	734.63	734.63	1080.57	1080.57	11690.2	1080.57	9	1.68
C108	10	10	790.63	780.13	1090.99	1075.35	11667.68	1075.35	9	0.908
C109	10	10	739.17	729.91	1037.7	1028.01	11486.7	1028.01	10	0.927
C201	4	4	412.15	412.15	635.92	635.92	10022.69	635.92	4	4.28
C202	4	4	411.5	411.5	635.24	635.24	10238.89	635.24	4	12.776
C203	4	4	408.41	408.41	636.46	636.46	9983.98	636.46	3	13.478
C204	4	4	403.34	402.9	632.61	632.61	10021.21	632.61	3	17.081
C205	4	4	403.35	403.35	635.72	635.72	10073.46	635.72	3	7.572
C206	4	4	403	403	635.04	635.04	10041.16	635.04	3	9.646
C207	4	4	403	403	635.04	635.04	10041.16	635.04	3	11.804
C208	4	4	403	403	635.04	635.04	10041.16	635.04	3	9.911
R101	17	17	1143.32	1143.32	1596.38	1596.38	3602.5	1596.38	27	6.995
R102	14	14	975.02	975.02	1410.75	1410.75	2913.66	1410.75	23	14.117
R103	12	12	797.52	797.52	1212.31	1212.31	2444.49	1212.31	19	16.144
R104	10	10	697.09	697.09	1033.91	1033.91	2050.23	1033.91	14	17.924
R105	13	13	960.62	960.62	1364.95	1364.95	2716.77	1364.95	24	10.1
R106	12	12	905.69	905.69	1319.11	1319.11	2463.68	1319.11	20	16.221
R107	10	10	799.69	799.69	1140.23	1140.23	2163.96	1140.23	17	12.837
R108	10	10	671.39	671.39	1021.42	1021.42	2043.57	1021.42	14	18.533
R109	11	11	895.38	895.38	1263.71	1263.71	2409.85	1263.71	20	7.292
R110	10	10	727.5	727.5	1063.88	1063.88	2111.35	1063.88	15	14.9
R111	10	10	756.94	756.94	1093.63	1093.63	2180.15	1093.63	17	15.968
R112	10	10	689.7	689.7	1025.33	1025.33	2108.1	1025.33	18	20.857
R201	3	3	909.47	909.47	1229.17	1229.17	2891.5	1229.17	6	18.225
R202	3	3	754.64	754.64	1036.47	1036.47	2856.27	1036.47	5	22.483
R203	2	2	712.84	712.84	1008.95	1008.95	1986.94	1008.95	6	31.247
R204	2	2	564.81	564.81	820.06	820.06	1928.53	820.06	4	44.639
R205	2	2	817.47	817.47	1163.95	1163.95	1995.84	1163.95	9	13.235
R206	2	2	716.2	716.2	999.15	999.15	1972.1	999.15	5	33.965
R207	2	2	604.06	596.19	870.54	868.6	1959.65	868.6	5	15.814
R208	2	2	507.69	498.5	764.85	750.02	1774.54	750.02	4	10.224
R209	2	2	685.71	673.38	959.64	939.04	1956.49	939.04	5	6.848
R210	2	2	671.1	669.33	933.86	928.34	1929.11	928.34	5	7.523
R211	2	2	591.07	577.31	845.29	827.37	1784.8	827.37	4	8.227
RC101	14	14	1148.16	1147.71	1623.45	1621.05	3055.51	1621.05	20	12.528
RC102	12	12	1035.54	1032.73	1487.79	1481.34	2617.95	1481.34	17	14.454
RC103	11	11	915.87	915.87	1313.61	1313.61	2425.85	1313.61	13	12.598
RC104	10	10	779.52	779.52	1151.96	1151.96	2109.28	1151.96	12	13.843
RC105	12	12	1006.29	1006.29	1427.48	1427.48	2647.49	1427.48	17	9.846
RC106	12	12	955.85	955.85	1379.67	1379.67	2549.28	1379.67	15	10.585
RC107	11	11	838.32	838.32	1243.34	1243.34	2263.55	1243.34	13	14.272
RC108	10	10	806.31	806.31	1165.66	1165.66	2132.91	1165.66	13	12.726
RC201	3	3	1207.96	1207.96	1614.73	1614.73	2867.02	1614.73	12	16.126
RC202	3	3	973.41	973.41	1340.36	1340.36	2806.29	1340.36	6	29.655
RC203	3	3	745.5	745.5	1054.62	1054.62	2695.87	1054.62	5	36.723
RC204	2	2	738	738	1060.06	1060.06	1902.29	1060.06	8	35.053
RC205	3	3	843.63	843.63	1175.53	1175.53	2627.93	1175.53	9	23.077
RC206	3	3	840.83	831.75	1178.52	1162.1	2647.11	1162.1	6	17.307
RC207	3	3	688.45	683.35	1005.14	998.81	2459.1	998.81	5	22.57
RC208	2	2	702.12	676.65	990.82	949.9	1857.56	949.9	6	16.122
AVG	7.23	7.23	744.75	742.5	1070.08	1066.51	4990.29	1066.51	10.12	13.95
SUM	405	405	41705.95	41579.87	59924.3	59724.31	279456.15	59724.31	567	781.08

Table 25: TD-EVRPTW - C3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{u}	u_{best}	\bar{d}	d_{best}	tol_{best}	rec_{best}	m	\bar{i}_e
C101	11	11	707.56	705.28	1091.35	1091.35	12443.03	1091.35	11	1.987
C102	11	11	675.71	674.36	1069.21	1063.25	12251.36	1063.25	8	4.026
C103	10	10	692.3	668.12	1056.12	1002.79	11547.78	1002.79	8	3.341
C104	10	10	599.05	578.95	940.44	891.05	11109.03	891.05	7	8.197
C105	11	11	675.76	675.76	1082.16	1082.16	12252.13	1082.16	10	3.104
C106	11	11	674.69	674.1	1078.86	1081.16	12004.8	1081.16	10	4.627
C107	10.5	10	694.11	722.73	1083.15	1090.44	11733.97	1090.44	10	2.624
C108	10	10	719.06	707.58	1087.38	1077.88	11675.66	1077.88	10	1.895
C109	10	10	662.63	638.9	1016.21	980.19	11362.09	980.19	9	3.882
C201	4	4	363.51	363.51	635.92	635.92	9990.98	635.92	4	3.754
C202	4	4	362.87	362.87	635.24	635.24	10208.93	635.24	4	10.609
C203	4	4	358.87	358.87	642.69	642.69	9895.91	642.69	3	12.143
C204	4	4	350.31	349.4	634.71	632.61	9978.01	632.61	3	15.753
C205	4	4	352.7	352.7	635.72	635.72	10051.67	635.72	3	5.433
C206	4	4	352.42	352.42	635.04	635.04	9997.13	635.04	3	7.302
C207	4	4	352.42	352.42	635.04	635.04	9997.13	635.04	3	9.128
C208	4	4	352.42	352.42	635.04	635.04	9997.13	635.04	3	8.225
R101	16.5	16	1049.08	1056.81	1609.96	1617.51	3411.33	1617.51	26	7.081
R102	13.5	13	888.34	909.36	1442.88	1467.93	2747.26	1467.93	26	11.195
R103	11	11	710.7	701.96	1194.67	1173.84	2362.92	1173.84	17	14.122
R104	9.5	9	638.68	669.83	1057.65	1088.47	2021.76	1088.47	15	11.906
R105	13	13	862.28	861.07	1363.36	1363.55	2736.59	1363.55	21	11.514
R106	11.5	11	779.25	789.01	1263.97	1284.63	2290.8	1284.63	19	12.672
R107	10	10	675.63	669.74	1099.96	1098.78	2082.62	1098.78	13	16.509
R108	9	9	585.41	584.85	982.82	974.47	1901.03	974.47	13	20.322
R109	11	11	740.67	739.46	1185.58	1192.02	2330.9	1192.02	17	15.952
R110	10	10	653.43	652.95	1078.51	1088.84	2050.41	1088.84	16	17.702
R111	10	10	663.88	662.07	1086.28	1075.8	2130	1075.8	17	13.221
R112	10	10	588.33	586.27	1031.97	1043.87	1949.77	1043.87	20	26.257
R201	3	3	838.45	836.97	1231.26	1226.75	2891.5	1226.75	6	18.652
R202	3	3	691.07	685.41	1057.02	1035.94	2856.27	1035.94	5	29.368
R203	2	2	634.8	632.22	1006.35	999.61	1985.39	999.61	6	42.337
R204	2	2	496.19	495.34	813.93	822.02	1929.12	822.02	4	33.212
R205	2	2	703.11	696.77	1080.49	1070.04	1979.96	1070.04	6	13.644
R206	2	2	645.4	644.77	1007.68	996.08	1973.53	996.08	5	24.771
R207	2	2	537.18	534.03	878.83	872.97	1901.63	872.97	3	28.446
R208	2	2	440.06	440.06	756.68	756.68	1720.03	756.68	5	35.56
R209	2	2	621.01	621.01	971.28	971.28	1884.11	971.28	8	35.91
R210	2	2	610.92	604.96	956.82	944.07	1921.17	944.07	6	6.664
R211	2	2	515.02	508.55	844.93	821.1	1748.22	821.1	4	6.839
RC101	13.33	13	1022.37	1026.96	1614.5	1613.63	2858.64	1613.63	23	8.852
RC102	11.67	11	932.33	988.18	1505.66	1563.11	2492.91	1563.11	19	11.259
RC103	10.33	10	812.39	819.97	1318.85	1316.18	2211.93	1316.18	14	11.885
RC104	10	10	667.77	663.48	1144.14	1140.35	1969.77	1140.35	13	16.539
RC105	12	12	896.46	895.41	1412.69	1406.93	2603.24	1406.93	17	11.075
RC106	11.33	11	852.05	856.43	1363.36	1362.88	2391.01	1362.88	17	10.197
RC107	10	10	743.5	742.08	1218.92	1226.45	2171.93	1226.45	18	11.506
RC108	10	10	706.69	698.78	1185.78	1188.71	2022.38	1188.71	15	20.392
RC201	3	3	1121.03	1118.93	1645.91	1642.73	2867.38	1642.73	12	15.669
RC202	3	3	877.34	875.07	1349.35	1343.47	2780.53	1343.47	6	28.418
RC203	3	3	681.76	678.96	1062.41	1056.02	2695.87	1056.02	6	28.945
RC204	2	2	641.42	610.89	1034.96	989.84	1892.92	989.84	6	41.154
RC205	3	3	775.93	760.26	1214.71	1168.8	2651.69	1168.8	6	24.176
RC206	3	3	762.67	757.95	1176.75	1158.13	2647.76	1158.13	6	16.572
RC207	2	2	714.08	699.62	1134.94	1135.2	1853.02	1135.2	6	21.058
RC208	2	2	581.82	577.43	950.62	937.4	1769.29	937.4	7	31.555
AVG	7.06	6.98	666.16	665.08	1070.19	1065.78	4913.99	1065.78	10.32	15.52
SUM	395.16	391	37304.89	37244.26	59930.71	59683.65	275183.33	59683.65	578	869.14

Table 26: TD-EVRPTW - D1 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tot_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	887.06	887.06	1061.15	1061.15	13321.75	1061.15	8	2.393
C102	11	11	880.83	874.29	1062.75	1050.07	12549.25	1050.07	8	4.572
C103	10.29	10	855.26	843.58	1040.38	1026.07	11812.49	1026.07	9	4.528
C104	10	10	792.55	778.46	977.01	961.99	11798.24	961.99	8	7.762
C105	11	11	901.11	901.11	1063.64	1063.64	12136.64	1063.64	9	2.593
C106	11	11	887.56	886.8	1064.6	1065.62	12343.42	1065.62	8	3.949
C107	10.71	10	893.05	920.68	1075.35	1110.65	11650.98	1110.65	10	3.516
C108	10.57	10	874.25	906.3	1055.41	1098.51	11753.54	1098.51	12	3.521
C109	10	10	834.65	824.99	1010.24	1001.85	11795.67	1001.85	10	4.576
C201	4	4	566.63	566.63	633.98	633.98	10209.7	633.98	4	3.568
C202	4	4	565.33	565.33	636.06	636.06	10520.35	636.06	4	11.608
C203	4	4	570.33	570.09	636.6	636.06	10472.49	636.06	4	16.56
C204	4	4	576.89	567.04	655.25	630.55	10335.15	630.55	4	22.07
C205	4	4	569.09	569.09	637.84	637.84	10223.96	637.84	3	5.514
C206	4	4	569.27	569.27	631.83	631.83	10382.02	631.83	5	8.467
C207	4	4	569.27	569.27	631.83	631.83	10382.02	631.83	5	10.643
C208	4	4	567.56	567.56	635.68	635.68	10550.11	635.68	5	9.354
R101	17	17	1431.82	1430.77	1678.07	1684.01	3497.7	1684.01	26	6.672
R102	16	16	1245.56	1239.69	1465.12	1452.96	3210.75	1452.96	21	12.512
R103	13	13	1070.89	1047.02	1278.09	1251.28	2771.49	1251.28	16	13.106
R104	11	11	890.73	885.36	1067.63	1059.74	2328.4	1059.74	13	18.556
R105	14	14	1148.56	1148.56	1363.37	1363.37	2814.09	1363.37	19	10.319
R106	13	13	1067.25	1062.57	1278.18	1269.63	2705.23	1269.63	18	17.275
R107	11	11	928.38	919.8	1134.08	1126.47	2392.74	1126.47	15	17.234
R108	11	11	864.31	863.9	1057.42	1063.71	2380.5	1063.71	14	25.622
R109	12	12	984.45	976.62	1204.28	1197.93	2594.96	1197.93	15	14.38
R110	11	11	890.53	882.43	1087.54	1082.68	2329.01	1082.68	13	21.59
R111	11	11	884.57	875.85	1088.74	1080.01	2356.98	1080.01	13	15.214
R112	11	11	822.3	818.54	1038.88	1035.95	2381.28	1035.95	13	18.62
R201	3	3	1032.75	1032.71	1240.05	1240.09	2778.25	1240.09	7	21.62
R202	3	3	856.56	856.56	1048.23	1048.23	2885.63	1048.23	3	28.993
R203	3	3	722.91	722.15	899.66	900.87	2873.25	900.87	3	27.649
R204	2	2	619.86	609.35	783.24	775.64	1959.66	775.64	4	33.917
R205	3	3	826.83	819.19	1001.84	992.9	2632.44	992.9	4	15.394
R206	2	2	850.82	838.04	1035.09	1022.7	1984.29	1022.7	7	19.774
R207	2	2	693.77	692.7	854.99	850.53	1965.12	850.53	2	28.595
R208	2	2	591.05	591.05	744.35	744.35	1973.28	744.35	3	30.662
R209	2	2	762.99	762.99	944.56	944.56	1970.21	944.56	6	25.298
R210	2	2	749.75	721.76	926.04	881.53	1882.66	881.53	5	6.214
R211	2	2	683.03	667.08	845.69	839.4	1964.18	839.4	3	7.011
RC101	15	15	1415.78	1403.24	1690.84	1682.23	3178.8	1682.23	21	7.063
RC102	13	13	1322.93	1318.7	1590.69	1569.88	2810.52	1569.88	19	7.873
RC103	12	12	1126.69	1124.85	1367.66	1378.09	2704.54	1378.09	16	10.365
RC104	10.67	10	967.52	959.1	1173.74	1162.17	2248.21	1162.17	14	12.895
RC105	13	13	1211.13	1197.62	1453.66	1438.05	2793.92	1438.05	15	10.29
RC106	12	12	1166.09	1163.33	1408.18	1400.96	2568.6	1400.96	14	7.23
RC107	11	11	1012.28	1007.24	1235.41	1224.35	2479.87	1224.35	14	12.377
RC108	10.67	10	965.61	973.17	1185.89	1190.81	2281.54	1190.81	12	12.347
RC201	4	4	1194.71	1191.44	1429.51	1423.15	3553.56	1423.15	6	12.28
RC202	3	3	1154.41	1150.28	1386.23	1383.84	2733.52	1383.84	5	31.245
RC203	3	3	876.61	874.07	1060.41	1066.32	2764.25	1066.32	5	37.829
RC204	3	3	731.76	724.84	899.63	888.14	2682.33	888.14	5	31.952
RC205	3	3	1077.37	1062.59	1266.61	1242.02	2589.94	1242.02	7	21.526
RC206	3	3	1021.63	1005.1	1221.58	1196.74	2496.37	1196.74	4	20.637
RC207	3	3	833.73	826.94	1006.15	995.82	2474.76	995.82	4	22.402
RC208	3	3	697.16	692.33	855.04	853.81	2433.82	853.81	6	28.083
AVG	7.64	7.59	888.5	884.06	1067.96	1062.83	5208.29	1062.83	9.39	15.14
SUM	427.91	425	49755.77	49507.08	59805.97	59518.3	291664.43	59518.3	526	847.82

Table 27: TD-EVRPTW - D2 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{t}	t_{best}	\bar{d}	d_{best}	tol_{best}	rec_{best}	m	\bar{i}_e
C101	12	12	804.61	804.61	1059.78	1059.78	13292.38	1059.78	9	2.706
C102	11	11	789.77	785.22	1070.34	1050.03	12475.67	1050.03	9	4.795
C103	10	10	775.01	767.15	1058	1043.38	11748.77	1043.38	9	3.814
C104	10	10	696.54	690.8	952.32	940.25	11786.45	940.25	9	7.57
C105	11	11	800.5	800.5	1078.47	1078.47	12269.68	1078.47	10	3.6
C106	11	11	796.09	794.26	1058.18	1062.77	12263.69	1062.77	10	4.204
C107	10.5	10	797.9	807.72	1087.2	1118.59	11668.61	1118.59	11	2.811
C108	10.75	10	768.17	791.61	1042.11	1083.92	11669.68	1083.92	11	4.449
C109	10	10	747.71	738.34	1016.8	999.2	11683.26	999.2	10	3.94
C201	4	4	509.33	509.33	633.98	633.98	10152.4	633.98	4	3.506
C202	4	4	505.84	505.84	637.37	637.37	10492.04	637.37	4	9.592
C203	4	4	517.43	512.6	647.58	634.61	10344.94	634.61	4	12.75
C204	4	4	514.46	514.46	669.36	669.36	11779.13	669.36	4	13.632
C205	4	4	510.44	510.44	633.98	633.98	10165.73	633.98	5	6.98
C206	4	4	507.94	507.94	634.15	634.15	10323.71	634.15	6	7.893
C207	4	4	509.85	509.85	632.84	632.84	10322.6	632.84	5	9.182
C208	4	4	507.94	507.94	634.15	634.15	10323.71	634.15	6	8.825
R101	17	17	1238.18	1219.49	1636.25	1615.6	3396.65	1615.6	23	8.344
R102	15	15	1056.03	1049.54	1408.99	1410.33	3045.82	1410.33	20	7.572
R103	12	12	896.2	888.65	1208.76	1205.57	2486.62	1205.57	17	9.425
R104	10	10	785.79	777.47	1096.41	1088	2154.04	1088	15	12.446
R105	13	13	1011.37	1002.7	1381.4	1374.92	2623.53	1374.92	21	5.949
R106	12	12	906.94	901.71	1259.74	1253.12	2527.75	1253.12	20	12.339
R107	10	10	802.16	799.38	1137.51	1135.64	2162.99	1135.64	15	5.643
R108	10	10	735.74	735.74	1036.35	1036.35	2126.67	1036.35	14	19.24
R109	11	11	825.42	825.42	1176.29	1176.29	2347.67	1176.29	17	7.874
R110	11	11	736.95	736.95	1083.07	1083.07	2356.72	1083.07	16	17.431
R111	10	10	770.19	770.19	1089.87	1089.87	2165.2	1089.87	16	5.173
R112	10	10	679.63	679.63	991.55	991.55	2179.97	991.55	12	12.799
R201	3	3	878.93	878.93	1211.63	1211.63	2735.79	1211.63	6	17.059
R202	3	3	747.46	747.46	1047.21	1047.21	2846.29	1047.21	3	26.361
R203	2	2	753.35	753.35	1056.1	1056.1	1990.59	1056.1	7	35.47
R204	2	2	538.68	538.68	806.75	806.75	1985.4	806.75	3	23.814
R205	2	2	769.45	769.45	1075.63	1075.63	1925.24	1075.63	7	9.977
R206	2	2	725.9	725.9	1023.35	1023.35	1945.02	1023.35	6	23.748
R207	2	2	599.42	599.42	860.6	860.6	1891.08	860.6	3	24.096
R208	2	2	507.77	507.77	760.12	760.12	1956.59	760.12	3	25.188
R209	2	2	643.37	643.37	915.04	915.04	1947.14	915.04	7	25.692
R210	2	2	646.88	646.88	945.41	945.41	1961.25	945.41	6	22.992
R211	2	2	567.82	567.82	837.15	837.15	1864.63	837.15	2	16.31
RC101	14	14	1177.8	1171.84	1598.62	1585.35	2786.13	1585.35	19	8.874
RC102	12.5	12	1107.34	1146.43	1537.81	1615.91	2671.32	1615.91	19	11.509
RC103	11	11	979.55	978.43	1378.54	1383.18	2450.04	1383.18	16	8.13
RC104	10	10	816.68	815.98	1146.5	1140.02	2238.11	1140.02	14	12.554
RC105	12	12	1051.23	1047.65	1447.05	1449.62	2606.84	1449.62	16	7.462
RC106	12	12	970.18	965.36	1357.88	1346.69	2525.36	1346.69	16	12.656
RC107	11	11	860.71	859.56	1219.75	1233.64	2428.87	1233.64	15	17.105
RC108	10	10	836.28	820.36	1183.45	1172.86	2211.67	1172.86	14	10.978
RC201	3	3	1180.43	1159.88	1610.27	1585.36	2677.63	1585.36	9	13.222
RC202	3	3	968.48	966.52	1357.27	1356.2	2728.85	1356.2	5	21.134
RC203	3	3	752.22	748.35	1062.55	1045.02	2600.5	1045.02	5	25.534
RC204	3	3	652.03	652.03	923.32	923.32	2720.36	923.32	5	31.27
RC205	3	3	878.65	878.65	1199.59	1199.59	2573.95	1199.59	8	24.221
RC206	3	3	880.75	880.75	1233.09	1233.09	2472.11	1233.09	5	20.463
RC207	3	3	678.09	678.09	990.99	990.99	2460.75	990.99	5	26.39
RC208	2	2	674.46	674.46	969.87	969.87	1854.11	969.87	6	25.681
AVG	7.3	7.27	774.07	772.66	1068.01	1067.44	5096.28	1067.44	10.04	13.61
SUM	408.75	407	43348.04	43268.85	59808.34	59776.84	285391.7	59776.84	562	762.37

Table 28: TD-EVRPTW - D3 - travel time - results

Inst.	\bar{K}	K_{best}	\bar{u}	t_{best}	\bar{d}	d_{best}	tol_{best}	rec_{best}	m	\bar{i}_e
C101	12	12	750.92	750.92	1059.78	1059.78	13255.52	1059.78	9	1.526
C102	11	11	727.39	720.58	1068.31	1071.75	12810.3	1071.75	8	2.89
C103	10	10	712.68	703.72	1062.5	1049.78	11664.44	1049.78	8	2.751
C104	10	10	639.29	623.07	950.96	912.82	11672.81	912.82	7	5.05
C105	11	11	739.75	739.72	1079.76	1078.47	12222.6	1078.47	10	1.909
C106	10.88	10	741.61	770.94	1087.82	1180.32	11755.3	1180.32	10	2.312
C107	10.38	10	735.01	738.02	1099.4	1118.52	11613.37	1118.52	10	1.756
C108	10.12	10	728.9	716.81	1091.19	1085.91	11947.44	1085.91	12	1.56
C109	10	10	691.31	681.87	1043.41	1014.67	11590.43	1014.67	9	2.673
C201	4	4	482.12	482.12	633.98	633.98	10125.2	633.98	4	2.264
C202	4	4	477.05	477.05	637.37	637.37	10469.98	637.37	4	7.331
C203	4	4	487.59	483.89	655.49	642.12	11206.59	642.12	4	9.946
C204	4	4	477.14	477.14	678.73	678.73	11832.49	678.73	5	12.084
C205	4	4	481.53	481.53	633.98	633.98	10136.81	633.98	5	4.172
C206	4	4	478.27	478.27	634.15	634.15	10294.03	634.15	6	5.484
C207	4	4	481.09	481.09	632.84	632.84	10293.84	632.84	5	6.908
C208	4	4	478.22	478.22	638.01	638.01	10463.78	638.01	6	6.277
R101	16	16	1144.76	1143.47	1621.49	1611.89	3126.59	1611.89	26	7.572
R102	14.5	14	969.79	972.78	1414.94	1419.47	2859.98	1419.47	22	6.454
R103	11	11	820.13	818.23	1232.27	1230.21	2343.59	1230.21	19	4.198
R104	10	10	685.26	679.82	1085.02	1072.76	2156.33	1072.76	13	14.516
R105	12.5	12	912.97	922.37	1354.32	1360.82	2494.57	1360.82	24	6.637
R106	11	11	836.05	827.69	1265.29	1257.12	2337.2	1257.12	18	7.202
R107	10	10	716.65	715.48	1130.32	1143.07	2167.6	1143.07	17	9.385
R108	9.5	9	654.93	669.24	1035.17	1042.51	1958.66	1042.51	16	12.89
R109	10.5	10	750.61	768.22	1182.21	1198.58	2190.05	1198.58	19	7.416
R110	10	10	659.37	657.83	1084.4	1081.06	2145.54	1081.06	17	12.213
R111	10	10	644.79	642.51	1045.3	1033.07	2109.55	1033.07	14	13.388
R112	10	10	592.49	590.26	1025.73	1031.17	2171.81	1031.17	16	10.712
R201	3	3	804.94	801.38	1211.87	1206.15	2741.93	1206.15	7	12.511
R202	3	3	676.28	675.84	1051.37	1050.68	2774.97	1050.68	3	17.316
R203	2	2	666.85	665.12	1033.74	1029.64	1982.97	1029.64	7	25.703
R204	2	2	466.79	464.3	778.51	786.59	1989.56	786.59	3	30.138
R205	2	2	689.59	689.57	1052.07	1051.75	1844.86	1051.75	8	10.827
R206	2	2	631.03	626.01	997.67	991.37	1945.43	991.37	7	18.63
R207	2	2	536.65	535.61	885.72	887.33	1977.71	887.33	3	21.699
R208	2	2	452.66	438.3	763.79	775.03	1990.85	775.03	4	26.29
R209	2	2	572.42	572.42	945.96	945.96	1907.22	945.96	7	22.089
R210	2	2	551.25	551.25	884.04	884.04	1863.56	884.04	5	17.125
R211	2	2	500.89	500.89	859	859	1898.68	859	3	23.407
RC101	13.67	13	1105.01	1146.49	1646.95	1711.1	2695.34	1711.1	24	4.714
RC102	12	12	991.57	983.29	1522.42	1508.36	2545.92	1508.36	19	7.608
RC103	11	11	842.44	840.31	1325.91	1299.95	2359.44	1299.95	16	9.925
RC104	10	10	736.15	728.24	1137.88	1121.88	2172.07	1121.88	13	10.45
RC105	12	12	925.91	923.3	1403.63	1386.86	2522.36	1386.86	16	4.774
RC106	11	11	875.56	873.57	1361.91	1345.48	2361.17	1345.48	16	6.667
RC107	10	10	771.09	756.58	1204.37	1177.67	2203.95	1177.67	13	7.657
RC108	10	10	720.61	709.94	1153.51	1156.67	2269.77	1156.67	13	9.933
RC201	3	3	1106.94	1086.53	1648.83	1618.94	2699.36	1618.94	8	9.385
RC202	3	3	879.35	871.79	1344.01	1327.3	2624.11	1327.3	6	20.581
RC203	3	3	686.9	682.97	1074.8	1059.19	2645.72	1059.19	4	19.763
RC204	2	2	658.62	640.94	1046.98	1025.69	1904.64	1025.69	8	28.281
RC205	3	3	787.57	782.77	1177.01	1160.6	2533.72	1160.6	7	16.42
RC206	3	3	785.32	779.69	1197.58	1177.54	2435.78	1177.54	6	13.683
RC207	2	2	690.61	686.34	1096.46	1082.73	1835.69	1082.73	8	18.378
RC208	2	2	571.27	563.51	925.33	901.88	1795.08	901.88	6	20.42
AVG	7.09	7.02	703.25	701.25	1069.56	1066.33	5034.61	1066.33	10.41	11.14
SUM	397.05	393	39381.94	39269.81	59895.46	59714.31	281938.26	59714.31	583	623.85

E TD-EVRPTW-FR total time

Table 29: TD-EVRPTW-FR - A1 - total time - results

Inst.	\bar{K}	K_{best}	\bar{tot}	tot_{best}	\bar{d}	d_{best}	tt_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	11930.7	11930.7	1186.57	1186.57	1010.25	1186.57	10	4.593
C102	11	11	11718.59	11718.59	1129.32	1129.32	963.7	1129.32	10	9.037
C103	10	10	11330.53	11330.53	1128.28	1128.28	956.57	1128.28	10	25.619
C104	10	10	10883.74	10883.74	1013.98	1013.98	853.98	1013.98	8	20.364
C105	11	11	11477.08	11477.08	1155.38	1155.38	989.63	1155.38	9	7.202
C106	11	11	11414.47	11414.47	1170.23	1170.23	981.11	1170.23	10	9.318
C107	11	11	11206.17	11206.17	1141.66	1141.66	961.76	1141.66	9	10.427
C108	10	10	11403.65	11403.65	1124.37	1124.37	962.9	1124.37	10	5.36
C109	10	10	11086.9	11086.9	1076.05	1076.05	916.62	1076.05	9	2.765
C201	4	4	10038.62	10038.62	642.33	642.33	533.27	642.33	3	6.532
C202	4	4	10022.75	10022.75	662.08	662.08	547.53	662.08	4	23.636
C203	4	4	10018.65	10016.48	660.7	659.67	547.53	659.67	4	37.355
C204	4	4	9970.21	9969.38	638.93	638.66	527.85	638.66	3	38.187
C205	4	4	9981.08	9981.08	640.27	640.27	534.43	640.27	3	11.444
C206	4	4	9960.86	9960.86	644.77	644.77	539.79	644.77	3	14.592
C207	4	4	9978.83	9960.86	650.4	644.77	539.79	644.77	3	19.13
C208	4	4	9960.86	9960.86	644.77	644.77	539.79	644.77	3	16.758
R101	16	16	3217.25	3217.25	1804.78	1804.78	1498.14	1804.78	31	13.615
R102	16	16	2967.44	2967.44	1669.55	1669.55	1381.05	1669.55	28	18.756
R103	12	12	2514	2514	1365.47	1365.47	1126.12	1365.47	23	20.98
R104	11	11	2162.74	2162.74	1120.25	1120.25	926.22	1120.25	13	19.776
R105	13	13	2661.99	2661.99	1494.3	1494.3	1230.59	1494.3	27	9.761
R106	12	12	2440.44	2440.44	1318.5	1318.5	1091.01	1318.5	19	10.265
R107	11	11	2263.4	2263.4	1217.24	1217.24	987.82	1217.24	13	17.284
R108	10	10	2083.58	2083.58	1043.83	1043.83	868.53	1043.83	15	20.768
R109	12	12	2353.66	2353.66	1276.56	1276.56	1033.92	1276.56	17	20.579
R110	11	11	2135.72	2135.72	1123.56	1123.56	910.03	1123.56	15	20.85
R111	11	11	2194.26	2194.26	1147.64	1147.64	928.39	1147.64	19	18.655
R112	10	10	2076.02	2076.02	1045.14	1045.14	855.22	1045.14	13	21.14
R201	3	3	2807.51	2795.97	1602.37	1750.4	1416.75	1750.4	11	23.018
R202	3	3	2412.62	2233.42	1462.36	1346.02	1124.34	1346.02	7	27.218
R203	3	3	2068.03	2050.86	1139.85	1133.92	933.4	1133.92	7	54.065
R204	2	2	1834.67	1832.78	941.36	938.21	757.87	938.21	3	43.134
R205	3	3	2182.05	2064.77	1262.53	1149.28	951.66	1149.28	9	25.622
R206	2	2	1966.85	1947.14	1033.39	1018.2	827.75	1018.2	8	26.762
R207	2	2	1759.68	1752.58	909.8	898.83	705.94	898.83	2	37.234
R208	2	2	1650.88	1644.49	771.59	760.32	597.27	760.32	2	46.524
R209	2	2	1886.79	1877.83	959.32	949.99	777.88	949.99	8	31.874
R210	2	2	1877.55	1874.67	956.95	962.8	769.12	962.8	7	35.058
R211	2	2	1715	1709.62	853.78	850.21	672.37	850.21	4	31
RC101	14	14	2939.69	2939.69	1791.53	1791.53	1479.28	1791.53	22	8.306
RC102	13	13	2683.62	2683.62	1688.53	1688.53	1371.16	1688.53	22	18.866
RC103	12	12	2580.57	2580.57	1532.43	1532.43	1251.95	1532.43	20	14.047
RC104	10	10	2191.56	2191.56	1261.33	1261.33	1008.9	1261.33	14	19.145
RC105	13	13	2573.19	2573.19	1563.64	1563.64	1248.61	1563.64	17	13.828
RC106	12	12	2418.28	2418.28	1438.67	1438.67	1158.81	1438.67	14	14.851
RC107	11	11	2239.78	2239.78	1284.34	1284.34	1027.92	1284.34	13	18.888
RC108	10	10	2164.18	2164.18	1239.72	1239.72	975.04	1239.72	14	21.304
RC201	3	3	2790.05	2789.66	1875.89	1914.92	1563.09	1914.92	12	21.722
RC202	3	3	2689.29	2689.18	1884.51	1890.6	1524.83	1890.6	8	42.575
RC203	3	3	2234.96	2208.95	1371.11	1348.94	1101.37	1348.94	10	55.81
RC204	3	3	2013.93	1973.29	1096.43	1040.54	849.08	1040.54	7	69.484
RC205	3	3	2515.01	2509.24	1567.12	1560.6	1280.46	1560.6	7	38.101
RC206	3	3	2424.24	2412.95	1570.32	1531.73	1254.9	1531.73	9	32.592
RC207	3	3	2113.18	2092.67	1256.74	1242.41	991.3	1242.41	10	42.764
RC208	3	3	1935.49	1929.59	1045.56	1051.48	821.06	1051.48	9	51.388
AVG	7.46	7.46	4859.34	4850.25	1183.89	1179.81	967.6	1179.81	11.07	23.93
SUM	418	418	272122.84	271613.75	66298.08	66069.57	54185.65	66069.57	620	1339.93

Table 30: TD-EVRPTW-FR - B1 - total time - results

Inst.	\bar{K}	K_{best}	\bar{tot}	tot_{best}	\bar{d}	d_{best}	tt_{best}	rec_{best}	m	\bar{t}_e
C101	12	12	12000.92	11812.1	1282.87	1239.6	979.66	1239.6	10	5.02
C102	11	11	11631.35	11617.76	1222.4	1261.58	1007.78	1261.58	11	10.397
C103	10	10	11057.56	11053.37	1064.42	1060.17	819.64	1060.17	7	11.338
C104	10	10	10611.25	10611.25	995.13	995.13	765.56	995.13	7	15.33
C105	11	11	11473.53	11473.53	1210.37	1210.37	959.34	1210.37	9	7.711
C106	11	11	11435.42	11435.42	1199.35	1199.35	941.61	1199.35	10	11.601
C107	11	11	11188.5	11188.5	1120.63	1120.63	895.46	1120.63	7	10.028
C108	10	10	11485.22	11485.22	1126.12	1126.12	876.04	1126.12	10	6.435
C109	10	10	10985.5	10985.5	1059.86	1059.86	833.3	1059.86	9	9.214
C201	4	4	10068.2	10068.2	642.33	642.33	538.74	642.33	3	6.861
C202	4	4	10060.58	10060.58	662.08	662.08	561.25	662.08	4	23.608
C203	4	4	10032.67	10032.67	648.11	648.11	546.3	648.11	3	37.773
C204	4	4	9988.99	9988.99	641.17	641.17	538.56	641.17	3	38.901
C205	4	4	10008.68	10008.68	640.69	640.69	537.82	640.69	3	11.88
C206	4	4	9994.56	9994.56	639.76	639.76	530.84	639.76	3	15.048
C207	4	4	9999.94	9999.94	648.19	648.19	547.79	648.19	3	18.536
C208	4	4	9987.81	9987.81	647.95	647.95	541.24	647.95	3	17.928
R101	17	17	3391.71	3391.71	1889.17	1889.17	1537.6	1889.17	32	9.271
R102	15	15	2919.91	2919.91	1623.1	1623.1	1317.7	1623.1	29	20.989
R103	13	13	2497.19	2497.19	1364.47	1364.47	1108.3	1364.47	18	28.423
R104	10	10	2115.73	2115.73	1086.94	1086.94	877.3	1086.94	14	27.643
R105	14	14	2894.48	2894.48	1574.98	1574.98	1265.14	1574.98	28	10.577
R106	13	13	2515.68	2515.68	1411.96	1411.96	1152.98	1411.96	19	16.009
R107	11	11	2199.71	2199.71	1176.84	1176.84	949.51	1176.84	13	16.439
R108	10	10	2072.07	2072.07	1037.97	1037.97	830.45	1037.97	13	23.765
R109	12	12	2395.99	2395.99	1316.34	1316.34	1062.92	1316.34	22	20.695
R110	11	11	2231.03	2231.03	1171.81	1171.81	951.7	1171.81	18	24.346
R111	11	11	2278.4	2278.4	1192.79	1192.79	968.93	1192.79	18	15.103
R112	10	10	2122.09	2122.09	1082	1082	869.28	1082	15	17.972
R201	3	3	2736.82	2736.82	1686.12	1755.25	1442.55	1755.25	9	26.632
R202	3	3	2082.46	2078.11	1221.53	1228.08	972.6	1228.08	5	37.862
R203	2	2	1922.87	1921.33	1018.93	1018.47	814.64	1018.47	5	50.34
R204	2	2	1781.28	1780.8	888.46	892.56	728.5	892.56	3	56.885
R205	3	3	2220.7	2114.51	1316.86	1241.31	972.48	1241.31	7	22.404
R206	2	2	1960.02	1956.93	1045.37	1043.11	838.51	1043.11	8	25.289
R207	2	2	1775.34	1773.66	882.14	881.85	734.01	881.85	2	46.226
R208	2	2	1674	1669.51	760.97	755.65	623.19	755.65	4	53.336
R209	2	2	1895.32	1894.4	962.87	967.21	782.23	967.21	8	35.552
R210	2	2	1895.59	1881.82	963.43	942.41	770.88	942.41	5	41.042
R211	2	2	1760.13	1760.13	843.36	843.07	706.36	843.07	4	41.835
RC101	15	15	3073.38	3073.38	1915.55	1915.55	1558.02	1915.55	27	11.83
RC102	14	14	2728.99	2728.99	1676.8	1676.8	1358.18	1676.8	21	18.036
RC103	12	12	2500.32	2500.32	1490.13	1490.13	1205.79	1490.13	18	12.852
RC104	10	10	2188.18	2188.18	1222.95	1222.95	971.51	1222.95	14	12.778
RC105	13	13	2702.57	2702.57	1591.14	1591.14	1299.8	1591.14	20	14.142
RC106	12	12	2539.39	2539.39	1460.61	1460.61	1191.64	1460.61	16	11.326
RC107	11	11	2310.44	2310.44	1313.88	1313.88	1056.04	1313.88	15	16.195
RC108	11	11	2227.88	2227.88	1239.75	1239.75	1004.93	1239.75	15	21.074
RC201	3	3	2719.68	2706.29	1802.04	1789.54	1472.47	1789.54	13	17.271
RC202	3	3	2493.24	2481.52	1658.1	1638.11	1333.46	1638.11	7	37.798
RC203	3	3	2197.75	2183.5	1345.61	1330.23	1066.59	1330.23	5	55.795
RC204	3	3	1994.28	1973.27	1071.41	1051.19	857.25	1051.19	6	67.972
RC205	3	3	2490.31	2439.3	1575.42	1545.58	1251.19	1545.58	8	28.199
RC206	3	3	2418.7	2386.36	1533.11	1514.74	1228.25	1514.74	11	29.485
RC207	3	3	2126	2109.82	1264.36	1249.5	996.1	1249.5	7	42.188
RC208	3	3	1929.31	1915.43	1011.33	992.11	794.96	992.11	6	48.108
AVG	7.54	7.54	4856.96	4847.73	1181.11	1177.9	952.59	1177.9	10.95	24.49
SUM	422	422	271989.62	271472.73	66142.03	65962.24	53344.87	65962.24	613	1371.29

Table 31: TD-EVRPTW-FR - C1 - total time - results

Inst.	\bar{K}	K_{best}	\bar{tot}	tot_{best}	\bar{d}	d_{best}	t_{best}	rec_{best}	m	\bar{i}_e
C101	12	12	11935.22	11890.4	1262.03	1255.28	1016.06	1255.28	10	4.927
C102	11	11	11643.46	11585.67	1175.99	1137.17	915.47	1137.17	10	8.82
C103	10	10	11112.28	11109.7	1066.95	1066.8	862.62	1066.8	8	11.97
C104	10	10	10746.21	10686.23	993.83	957.85	779.33	957.85	6	13.902
C105	11	11	11518.5	11518.5	1208.21	1208.21	995.88	1208.21	9	6.246
C106	11	11	11414.52	11414.52	1205.37	1205.37	971.18	1205.37	9	8.796
C107	11	11	11182.6	11182.6	1111.88	1111.88	893.45	1111.88	8	10.079
C108	10	10	11346.18	11346.18	1097.44	1097.44	908.91	1097.44	9	10.973
C109	10	10	10956.79	10956.79	1064.5	1064.5	872.77	1064.5	9	9.365
C201	4	4	10030.12	10030.12	642.33	642.33	500.66	642.33	3	6.899
C202	4	4	10014	10014	662.08	662.08	514.66	662.08	4	23.956
C203	4	4	9990.46	9990.46	648.11	648.11	504.08	648.11	3	38.36
C204	4	4	9951.6	9951.6	648.34	648.34	502.47	648.34	3	38.796
C205	4	4	9965.61	9965.61	640.69	640.69	494.74	640.69	3	11.982
C206	4	4	9944.61	9944.61	645.2	645.2	499.33	645.2	3	15.185
C207	4	4	9953.35	9953.35	648.19	648.19	501.2	648.19	3	18.277
C208	4	4	9944.61	9944.61	645.2	645.2	499.33	645.2	3	17.479
R101	17	17	3536.49	3536.49	1858.72	1858.72	1512.66	1858.72	32	12.05
R102	15	15	2957.79	2957.79	1649.02	1649.02	1312.91	1649.02	26	18.305
R103	13	13	2483.25	2483.25	1415.09	1415.09	1109.86	1415.09	22	25.955
R104	10	10	2092.32	2092.32	1068.51	1068.51	859.74	1068.51	13	23.307
R105	13	13	2756.94	2756.94	1493.57	1493.57	1203.49	1493.57	30	9.16
R106	13	13	2541.33	2541.33	1415.31	1415.31	1135.91	1415.31	19	16.83
R107	11	11	2250.5	2250.5	1195.54	1195.54	973.63	1195.54	14	14.595
R108	10	10	2054.72	2054.72	1035.02	1035.02	825.44	1035.02	14	23.391
R109	12	12	2420.4	2420.4	1302.01	1302.01	1040.53	1302.01	23	20.479
R110	11	11	2191.62	2191.62	1178.86	1178.86	931.95	1178.86	15	22.933
R111	11	11	2238.69	2238.69	1170.16	1170.16	929.65	1170.16	18	16.864
R112	10	10	2108.37	2108.37	1060.45	1060.45	844.95	1060.45	17	23.932
R201	3	3	2849.53	2826.55	1854.83	1791.53	1449.71	1791.53	13	22.93
R202	3	3	2739.8	2739.8	1634.36	1858.08	1478.11	1858.08	6	27.018
R203	2.5	2	2030.94	1983.56	1119.3	1058.64	867.9	1058.64	7	37.648
R204	2	2	1837.91	1833.52	942.41	938.71	772.85	938.71	3	34.873
R205	3	3	2524.48	2423.78	1659.38	1559.94	1203.56	1559.94	10	17.358
R206	2	2	1978.71	1966.59	1045.14	1031.27	848	1031.27	7	15.699
R207	2	2	1792.32	1788.6	913.55	905.19	745.19	905.19	4	27.86
R208	2	2	1667.45	1666.72	769.46	770.58	615.65	770.58	2	34.115
R209	2	2	1929.52	1916.91	983.86	970.83	799.67	970.83	6	26.785
R210	2	2	1923.94	1912.11	983.37	973.73	795	973.73	6	25.755
R211	2	2	1758.82	1747.61	881.56	867.67	697.64	867.67	3	22.73
RC101	15	15	3091.84	3091.84	1933.29	1933.29	1541.14	1933.29	27	11.14
RC102	14	14	2733.08	2733.08	1739.14	1739.14	1378.55	1739.14	20	20.053
RC103	12	12	2427.88	2427.88	1451.35	1451.35	1157.81	1451.35	14	17.27
RC104	10	10	2183.07	2183.07	1218.19	1218.19	983.78	1218.19	13	18.514
RC105	13	13	2669.88	2669.88	1607.97	1607.97	1290.22	1607.97	19	15.684
RC106	12	12	2515.73	2515.73	1455.87	1455.87	1167.17	1455.87	16	12.84
RC107	11	11	2287.53	2287.53	1278.86	1278.86	1025.41	1278.86	14	18.158
RC108	11	11	2211.47	2211.47	1234.58	1234.58	985.03	1234.58	13	18.897
RC201	3.6	3	3259.92	2861.18	2301.8	1958.42	1606.53	1958.42	14	14.816
RC202	3	3	2704.89	2704.75	1889.67	1894.3	1535.85	1894.3	6	30.276
RC203	3	3	2339.7	2298.29	1482.68	1443.06	1157.72	1443.06	10	42.44
RC204	3	3	2020.93	1993.41	1123.3	1083.96	864.78	1083.96	6	51.19
RC205	3	3	2575.39	2572.47	1655.52	1607.22	1289.84	1607.22	8	27.257
RC206	3	3	2542.88	2539.9	1678.26	1684.07	1339.07	1684.07	8	24.057
RC207	3	3	2149.6	2137.25	1324.68	1308.96	1020.12	1308.96	9	32.996
RC208	3	3	1985.13	1969.27	1116.28	1098.31	837.66	1098.31	9	38.823
AVG	7.54	7.52	4893.12	4877.14	1222.99	1211.62	970.84	1211.62	11.05	20.91
SUM	422.1	421	274014.88	273120.12	68487.26	67850.62	54366.82	67850.62	619	1171

Table 32: TD-EVRPTW-FR - D1 - total time - results

Inst.	\bar{K}	K_{best}	\bar{tot}	tot_{best}	\bar{d}	d_{best}	tt_{best}	rec_{best}	m	\bar{i}_e
C101	12	12	11770.61	11770.61	1212.56	1212.56	1025.18	1212.56	9	4.608
C102	11	11	11685.47	11685.47	1174.16	1174.16	985.08	1174.16	10	10.739
C103	10	10	11648.38	11648.38	1195.28	1195.28	983.46	1195.28	13	7.376
C104	10	10	10883.42	10883.42	1031.98	1031.98	854.96	1031.98	8	13.093
C105	11	11	11487.24	11487.24	1193.43	1193.43	999.39	1193.43	10	7.235
C106	11	11	11371.11	11371.11	1217.16	1217.16	1028.05	1217.16	9	10.039
C107	10	10	11628.11	11628.11	1160.6	1160.6	951.12	1160.6	10	3.854
C108	10	10	11392.08	11392.08	1123.98	1123.98	931.21	1123.98	11	6.252
C109	10	10	11036.9	11036.9	1093.21	1093.21	912.85	1093.21	7	8.705
C201	4	4	10084.8	10084.8	642.33	642.33	579.46	642.33	3	6.727
C202	4	4	10071.25	10071.25	661.68	661.68	596.02	661.68	4	24.115
C203	4	4	10069.17	10069.17	659.91	659.91	594.54	659.91	4	37.607
C204	4	4	10015.82	10015.82	638.71	638.71	576.09	638.71	3	38.101
C205	4	4	10027.16	10027.16	640.03	640.03	581.05	640.03	3	10.547
C206	4	4	10006.55	10006.55	644.77	644.77	585.48	644.77	3	14.372
C207	4	4	10016.41	10016.41	647.76	647.76	588.47	647.76	3	17.714
C208	4	4	10006.55	10006.55	644.77	644.77	585.48	644.77	3	16.185
R101	17	17	3271.71	3271.71	1783.78	1783.78	1530.73	1783.78	26	12.744
R102	16	16	2950.9	2950.9	1666.99	1666.99	1439.23	1666.99	26	17.229
R103	13	13	2529.08	2529.08	1373.53	1373.53	1163.6	1373.53	20	24.573
R104	11	11	2145.49	2145.49	1116.62	1116.62	944.62	1116.62	10	25.642
R105	14	14	2688.91	2688.91	1510.41	1510.41	1284.35	1510.41	24	15.639
R106	13	13	2465.6	2465.6	1353.58	1353.58	1153.64	1353.58	18	16.968
R107	11	11	2263.82	2263.82	1220.29	1220.29	1003.96	1220.29	14	16.604
R108	10	10	2109.74	2109.74	1058.74	1058.74	870.06	1058.74	14	21.151
R109	12	12	2358.56	2358.56	1256.18	1256.18	1068.53	1256.18	17	17.33
R110	11	11	2180.08	2180.08	1144.56	1144.56	963.11	1144.56	14	24.21
R111	11	11	2205.25	2205.25	1147.82	1147.82	963.27	1147.82	14	16.49
R112	10	10	2140.12	2140.12	1093.55	1093.55	906.9	1093.55	15	10.839
R201	3	3	2700.7	2683.92	1697.9	1672.97	1392.97	1672.97	14	15.363
R202	3	3	2133.39	2120.24	1236.04	1217.55	1023.11	1217.55	6	23.869
R203	3	3	2100.63	2019.02	1165.57	1091.98	909.46	1091.98	5	42.016
R204	2	2	1770.33	1747.9	870.1	837.59	696.79	837.59	3	22.457
R205	3	3	2152.12	2004.93	1192.27	1084.54	902.84	1084.54	8	19.31
R206	2	2	1968.05	1965.74	1024.7	1017.72	845.34	1017.72	6	11.986
R207	2	2	1780.9	1769.34	886.03	872.73	728.17	872.73	3	19.506
R208	2	2	1694.56	1681.59	766.17	752.72	634.11	752.72	3	24.203
R209	2	2	1882.04	1873.28	943.51	940.35	777.52	940.35	8	16.114
R210	2	2	1865.23	1854.5	935.61	927.65	760.6	927.65	6	17.9
R211	2	2	1759.81	1751.63	850.25	842.2	713.16	842.2	2	17.025
RC101	14	14	2988.48	2988.48	1822.11	1822.11	1528.7	1822.11	24	6.455
RC102	14	14	2720.11	2720.11	1632.91	1632.91	1397.85	1632.91	20	15.751
RC103	12	12	2492.2	2492.2	1443.18	1443.18	1219.68	1443.18	17	16.016
RC104	11	11	2253.58	2253.58	1251.51	1251.51	1056.35	1251.51	13	21.953
RC105	13	13	2636.45	2636.45	1550.96	1550.96	1306.09	1550.96	17	10.581
RC106	12	12	2512.74	2512.74	1419.97	1419.97	1188.67	1419.97	16	10.29
RC107	11	11	2284.43	2284.43	1261.33	1261.33	1057.43	1261.33	15	16.187
RC108	10	10	2206.27	2206.27	1202.44	1202.44	993.26	1202.44	11	14.144
RC201	4	4	2859.67	2755.22	1915.02	1852.24	1546.18	1852.24	11	15.16
RC202	3	3	2465.82	2457.33	1612.78	1599.52	1352.01	1599.52	8	25.298
RC203	3	3	2213.24	2196.7	1301.31	1283.97	1080.37	1283.97	6	35.033
RC204	3	3	2027.85	2008.51	1062.39	1040.25	884.3	1040.25	7	43.463
RC205	3	3	2483.94	2409	1523.05	1450.95	1250.81	1450.95	8	20.171
RC206	3	3	2337.15	2297.61	1412.31	1378.94	1187.03	1378.94	8	18.55
RC207	3	3	2115.5	2055.22	1187.88	1140	961.97	1140	7	27.428
RC208	3	3	1965.59	1950.83	990.64	969.09	845.85	969.09	6	32.374
AVG	7.57	7.57	4872.88	4860.84	1169.04	1158.32	980.19	1158.32	10.41	18.13
SUM	424	424	272881.07	272207.06	65466.31	64865.74	54890.51	64865.74	583	1015.29

F TD-EVRPTWDCS-FR recharging cost

Table 33: TD-EVRPTWDCS-FR - A1 - recharging cost - results

Inst.	\bar{K}	K_{best}	\bar{rec}	rec_{best}	\bar{d}	d_{best}	tt_{best}	tol_{best}	m	\bar{t}_e
C101	11	11	1124.27	1124.27	1106.05	1106.05	969.96	12523.23	12	2.162
C102	10	10	1045.54	1043	1021.79	1012.9	901.8	11757.76	9	2.221
C103	10	10	988.3	977.75	962.81	944.7	833.35	11339.45	8	3.903
C104	10	10	895.14	893.37	875.73	874.01	766.19	10857.91	7	4.155
C105	10	10	1082.15	1077.88	1046.06	1043.55	926.13	11397.36	9	1.678
C106	10	10	1069	1067.55	1038.63	1036.46	908.35	11170.56	10	1.938
C107	10	10	1055.67	1048.94	1028.48	1020.77	906.69	11122.6	9	2.145
C108	10	10	1035.07	1029.99	1006.54	999.7	861.66	11216.14	9	2.529
C109	10	10	941.65	939.71	921.36	919.05	806.77	11112.01	8	2.863
C201	4	4	633.98	633.98	633.98	633.98	528.82	10177.11	4	5.91
C202	4	4	633.84	633.3	633.84	633.3	530.69	10401.19	4	14.268
C203	3	3	942.98	901.23	880.98	841.8	706.74	10074.27	7	6.021
C204	3	3	737.04	719.21	705.67	688.8	575.73	10102.37	5	7.407
C205	4	4	629.95	629.95	629.95	629.95	527.77	10176.14	3	6.55
C206	4	4	629.95	629.95	629.95	629.95	527.77	10176.14	3	7.434
C207	4	4	629.95	629.95	629.95	629.95	527.77	10176.14	3	8.904
C208	4	4	629.95	629.95	629.95	629.95	527.77	10176.14	3	6.943
R101	15.2	15	1658.09	1650.49	1610.79	1601.69	1354.46	3162.08	27	4.763
R102	14	14	1473.36	1462.21	1431.78	1414.03	1186.65	2890.8	24	7.033
R103	10.8	10	1274.54	1369.75	1217.64	1281.44	1056.32	2200.45	21	8.885
R104	9	9	1083.37	1053.86	1032.01	1006	843.74	1973.43	13	9.905
R105	11.1	11	1452.55	1418.54	1368.55	1347.69	1126.44	2374.04	20	5.048
R106	10.7	10	1324.11	1402.86	1258.46	1308.02	1089.6	2225.36	19	6.484
R107	9.3	9	1162.97	1148.31	1096.33	1069.98	896.2	1999.7	13	7.415
R108	9	9	1011.44	991.13	973.13	958	805.07	1958.58	14	9.333
R109	10	10	1202.75	1178.26	1152.84	1131.97	945.27	2179.79	15	6.727
R110	9	9	1117.84	1077.38	1058.14	1020.25	855.74	1982.66	13	7.44
R111	9	9	1128.94	1095.91	1068.03	1041.2	867.8	2007.13	15	6.037
R112	9	9	993.93	979.31	956.76	939.52	793.07	1990.15	12	7.57
R201	3	3	1248.61	1240.69	1248.61	1240.69	1073.95	2862.33	7	58.435
R202	3	3	1039.46	1035.33	1039.46	1035.33	899.42	2913.81	4	46.488
R203	2	2	1100.66	1070.79	1077.63	1062.65	889.44	1995.69	9	20.404
R204	2	2	787.07	778.55	787.07	778.55	645.81	1953.12	2	13.703
R205	2	2	1126.57	1104.02	1091.1	1079.32	893.54	1979.43	7	11.58
R206	2	2	1000.49	989.31	1000.49	989.31	817.43	1976.06	6	14.385
R207	2	2	848.84	830.37	848.84	830.37	686.74	1978.42	2	11.658
R208	2	2	745.53	735.6	745.53	735.6	608.96	1881.47	3	12.376
R209	2	2	935.77	919.18	935.05	919.18	763.41	1966.5	5	13.01
R210	2	2	915.78	883.37	915.78	883.37	735.43	1932.38	6	13.68
R211	2	2	824.2	809.32	824.2	809.32	646.12	1789.62	3	13.391
RC101	13	13	1669.58	1653.77	1617.57	1613.18	1344.86	2797.06	19	4.977
RC102	11.9	11	1536.77	1624.37	1485.28	1542.65	1291.87	2477.42	17	6.575
RC103	10.7	10	1378.35	1443.83	1336.25	1379.8	1171.02	2291.49	15	6.85
RC104	9	9	1252.08	1184.43	1183.74	1132.24	941.96	2039.57	12	4
RC105	11.1	11	1491.93	1469.97	1432.51	1415.95	1185.74	2460.96	17	4.585
RC106	10.9	10	1399.94	1475.25	1355.77	1391.91	1148.26	2281.97	17	4.996
RC107	10	10	1211.95	1200.27	1176.61	1163.97	963.22	2252.26	15	7.536
RC108	9	9	1167.96	1120.51	1107.78	1067.45	885.35	1999.99	12	3.649
RC201	3	3	1633.74	1618.91	1631.45	1614.32	1353.28	2849.26	9	64.195
RC202	3	3	1348.11	1345.66	1348.11	1345.66	1125.26	2806.15	4	50.018
RC203	3	3	1059.67	1051.08	1058.93	1051.08	869.45	2706.3	5	40.676
RC204	2	2	1087.16	1064.54	1030.64	1020.53	836.22	1917.55	8	17.534
RC205	3	3	1262.98	1250.86	1262.98	1250.86	1009.68	2663.03	8	36.83
RC206	3	3	1159.37	1151.86	1159.37	1151.86	940.14	2632.59	5	29.765
RC207	3	3	983.86	975.11	983.86	975.11	770.08	2507.74	5	24.342
RC208	2	2	990.13	972.68	955.05	942.35	768.65	1883.11	8	18.302
AVG	6.76	6.68	1085.62	1079.24	1057.96	1050.31	882.49	4869.57	9.8	12.99
SUM	378.7	374	60794.88	60437.62	59245.84	58817.27	49419.61	272695.97	549	727.61

Table 34: TD-EVRPTWDCS-FR - B1 - recharging cost - results

Inst.	\bar{K}	K_{best}	\bar{rec}	rec_{best}	\bar{d}	d_{best}	tt_{best}	tot_{best}	m	\bar{t}_e
C101	11	11	1127.69	1127.69	1106.05	1106.05	872.47	12458.48	12	2.083
C102	10	10	1046.64	1046.46	1020.43	1025.5	784.91	11655.44	10	1.836
C103	10	10	970.35	969.86	954.1	954.68	722.68	11455.03	8	3.424
C104	10	10	880.77	873.97	871.44	866.32	642.22	11349.77	7	3.509
C105	10	10	1073.87	1071.85	1041.99	1039.33	804.1	11365.56	9	1.642
C106	10	10	1069.18	1069.18	1038.44	1038.44	825.01	11259.45	11	1.791
C107	10	10	1049.38	1045.54	1020.56	1013.22	801.71	10962.26	9	1.979
C108	10	10	1035.52	1033.88	1007.75	1003.07	805.47	11146.07	9	2.576
C109	10	10	945.48	944.93	926.02	921.22	703.19	10919.93	8	2.632
C201	4	4	633.98	633.98	633.98	633.98	526.63	10209.16	4	5.378
C202	4	4	633.98	633.98	633.98	633.98	526.63	10209.16	4	14.971
C203	3	3	950.78	915.58	891.87	862.15	718.9	10103.95	8	6.338
C204	3	3	723.89	723.89	696.27	696.27	569.1	10106.23	6	7.3
C205	4	4	629.95	629.95	629.95	629.95	520.34	10192.82	3	6.21
C206	4	4	629.95	629.95	629.95	629.95	520.34	10192.82	3	7.132
C207	4	4	629.95	629.95	629.95	629.95	520.34	10192.82	3	8.374
C208	4	4	629.95	629.95	629.95	629.95	520.34	10192.82	3	6.78
R101	17	17	1649.23	1649.23	1627.77	1627.77	1324.07	3460.51	25	4.621
R102	15	15	1481.2	1481.2	1452.89	1452.89	1182.12	3012.36	23	8.938
R103	12	12	1214.48	1214.48	1184.28	1184.28	957.12	2513.06	15	9.352
R104	9	9	1087.46	1087.46	1024.39	1024.39	828.36	1991.98	14	12.219
R105	13	13	1368.25	1368.25	1340.21	1340.21	1102.1	2688.19	19	5.202
R106	11	11	1393.65	1393.65	1325.97	1325.97	1085.3	2337.9	20	7.714
R107	9	9	1145.96	1145.96	1075.62	1075.62	872.22	1968.76	15	5.959
R108	9	9	1038.96	1038.96	999.32	999.32	814.38	1962.03	12	10.507
R109	10	10	1234.9	1234.9	1172	1172	947.72	2173.81	15	7.2
R110	10	10	1101.19	1101.19	1067.02	1067.02	866.14	2140.24	15	7.635
R111	10	10	1106.97	1106.97	1068.27	1068.27	866.9	2156.88	14	9.219
R112	9	9	1019.49	1019.49	972.07	972.07	784.7	1952.5	12	10.93
R201	3	3	1211.26	1201.97	1211.26	1201.97	960.36	2785.6	7	51.582
R202	3	3	1023.37	1019.94	1023.37	1019.94	810.36	2840.43	4	34.839
R203	2	2	974.4	970.53	974.4	970.53	764.69	1981.39	5	13.535
R204	2	2	768.16	757.97	768.16	757.97	610.34	1919.27	2	11.843
R205	2	2	1139.96	1110.78	1097.91	1077.22	864.96	1959.34	6	10.207
R206	2	2	1011.81	1001.67	1008.34	1001.67	821.18	1972.11	4	12.809
R207	2	2	864.38	855.67	864.38	855.67	706.54	1917.33	2	12.641
R208	2	2	747.32	735.08	747.32	735.08	605.56	1865.52	2	12.306
R209	2	2	947.54	920.36	944.76	917.55	754.68	1916.13	6	11.023
R210	2	2	913.71	894.82	913.71	894.82	730.51	1938.56	6	12.682
R211	2	2	844.18	827.63	842.77	820.56	683.8	1852.09	6	13.535
RC101	14	14	1695.47	1687.7	1667.29	1664.76	1360.28	2998.99	20	4.264
RC102	12.8	12	1552.62	1597.3	1524.15	1548.49	1254.32	2586.41	18	6.326
RC103	11	11	1355.79	1331.45	1321.16	1296.94	1024.6	2359.39	16	6.634
RC104	9	9	1201.81	1183.06	1152.92	1142.07	909.7	2026.12	11	4.46
RC105	12	12	1504.42	1487.14	1463.88	1453.23	1191.51	2642.13	18	4.215
RC106	11.6	11	1452.1	1514.89	1414.81	1458.08	1170.97	2470.71	19	4.169
RC107	10	10	1237.12	1218.93	1188.47	1179.85	953.82	2218.91	13	6.19
RC108	10	10	1159.49	1152.69	1132.13	1123.19	910.25	2224.59	13	6.392
RC201	3	3	1661.82	1618.18	1651.96	1603.15	1319.57	2757.9	12	55.427
RC202	3	3	1381.14	1363.76	1381.14	1363.76	1131.87	2719.4	5	46.346
RC203	3	3	1047.89	1042.54	1047.89	1042.54	864.29	2710.01	5	39.598
RC204	2	2	1133.34	1091.48	1057.73	1035.36	840.77	1908.45	9	29.129
RC205	3	3	1273.77	1258.07	1273.77	1258.07	1063.48	2662.03	9	36.886
RC206	3	3	1206.53	1192.99	1206.53	1192.99	1021.32	2534.38	5	28.427
RC207	3	3	1004.78	996.12	1004.78	996.12	856.96	2535.6	5	25.539
RC208	2	2	1014.94	990.83	973.91	963.77	777.81	1893.53	8	12.17
AVG	6.99	6.96	1086.29	1079.93	1063.06	1057.13	856.79	4902.26	9.86	12.44
SUM	391.4	390	60832.17	60475.88	59531.39	59199.17	47980.01	274526.31	552	696.62

Table 35: TD-EVRPTWDCS-FR - C1 - recharging cost - results

Inst.	\bar{K}	K_{best}	\overline{rec}	rec_{best}	\bar{d}	d_{best}	t_{best}	tot_{best}	m	\bar{t}_e
C101	11	11	1124.27	1124.27	1106.05	1106.05	896.21	12534.79	12	2
C102	10	10	1046.73	1046.73	1017.89	1017.89	833.31	11686	9	1.988
C103	10	10	969.19	965.78	951.06	949.37	778.87	11611.94	8	3.681
C104	10	10	876.31	876.31	866.07	866.07	703.9	11345.2	7	3.358
C105	10	10	1078.06	1075.87	1041.23	1038.01	863.24	11438.66	9	1.707
C106	10	10	1069.18	1069.18	1038.44	1038.44	854.99	11317.97	11	1.852
C107	10	10	1052.7	1052.7	1021.57	1021.57	841.32	11023.25	10	2.067
C108	10	10	1030.91	1028.68	1000.84	995.15	809.75	11355.61	10	2.638
C109	10	10	942.24	940.54	922.65	922.08	743.2	10919.45	7	2.879
C201	4	4	633.98	633.98	633.98	633.98	499.87	10191.74	4	5.962
C202	4	4	633.98	633.98	633.98	633.98	499.87	10191.74	4	14.366
C203	3	3	933.8	910.42	878.33	856.29	698.83	10049.87	7	6.172
C204	3	3	743.39	719.21	711.9	688.8	560.88	10098.82	5	7.624
C205	4	4	629.95	629.95	629.95	629.95	490.89	10167.76	3	6.559
C206	4	4	629.95	629.95	629.95	629.95	490.89	10163.37	3	7.191
C207	4	4	629.95	629.95	629.95	629.95	490.89	10163.37	3	8.52
C208	4	4	629.95	629.95	629.95	629.95	490.89	10163.37	3	6.95
R101	16	16	1651.45	1651.45	1613.11	1613.11	1329.16	3388.28	24	4.575
R102	14	14	1483.24	1483.24	1444.47	1444.47	1192.34	2988.83	23	7.297
R103	11	11	1232.16	1232.16	1197.55	1197.55	963.5	2369.3	18	9.741
R104	9	9	1026.25	1026.25	986.93	986.93	795.86	1976.9	12	10.596
R105	12	12	1393.93	1393.93	1332.92	1332.92	1110.67	2590.81	21	5.891
R106	11	11	1311.05	1311.05	1257.52	1257.52	1027.45	2344.84	19	8
R107	9	9	1178.06	1178.06	1091.64	1091.64	896.1	2013.86	18	10.104
R108	9	9	1000.95	1000.95	971.77	971.77	800.68	1988.01	14	10.934
R109	10	10	1272.94	1272.94	1219.47	1219.47	997.18	2250.46	21	5.032
R110	9	9	1143.79	1143.79	1085.01	1085.01	884.59	2032.19	14	8.287
R111	10	10	1064.81	1064.81	1030.01	1030.01	847.87	2148.75	14	8.505
R112	9	9	1001.83	1001.83	961.3	961.3	794	2006.37	11	9.445
R201	3	3	1241.69	1241.69	1241.69	1241.69	1054.23	2874.05	7	38.926
R202	3	3	1035.85	1035.85	1035.85	1035.85	872.5	2916.07	4	28.266
R203	2	2	1030.25	1030.25	1030.25	1030.25	857.86	1994.67	6	13.211
R204	2	2	778.04	778.04	778.04	778.04	641.64	1981.98	2	11.656
R205	3	3	991.8	991.8	988.13	988.13	830.86	2759.52	7	15.194
R206	2	2	1017.94	1017.94	1017.94	1017.94	846.34	1993.22	5	15.862
R207	2	2	846.18	846.18	846.18	846.18	699.92	1928.73	2	11.498
R208	2	2	736.62	736.62	736.62	736.62	595.29	1885.72	3	12.607
R209	2	2	970.13	970.13	966.31	966.31	808.88	1995.32	6	11.451
R210	2	2	890.79	890.79	890.79	890.79	754.51	1981.57	5	12.677
R211	2	2	837.54	821.53	837.54	821.53	686.26	1895.12	4	11.938
RC101	13	13	1714.97	1714.97	1627.73	1627.73	1370.45	2929.82	19	4.128
RC102	12	12	1498.21	1498.21	1449.05	1449.05	1195.77	2662.84	16	7.215
RC103	10	10	1384.69	1384.69	1318.6	1318.6	1070.2	2268.38	16	8.303
RC104	9	9	1345.33	1345.33	1279.52	1279.52	1040.4	2132.36	13	5.322
RC105	12	12	1454.96	1454.96	1436.84	1436.84	1192.13	2728.04	20	5.184
RC106	11	11	1455.6	1455.6	1401.82	1401.82	1168.06	2516.76	18	4.93
RC107	10	10	1240.62	1240.62	1197.7	1197.7	989.11	2255.95	14	7.141
RC108	10	10	1152.96	1152.96	1127.06	1127.06	920.54	2165.43	12	7.155
RC201	3	3	1736.4	1736.4	1719.78	1719.78	1455.63	2874.08	14	47.031
RC202	3	3	1371.03	1371.03	1371.03	1371.03	1149.31	2788.79	5	34.731
RC203	3	3	1050.99	1050.99	1050.99	1050.99	864.21	2691.26	5	34.753
RC204	2	2	1096.97	1096.97	1041.68	1041.68	850.9	1915.41	8	24.906
RC205	3	3	1304.2	1304.2	1304.2	1304.2	1087.24	2778.63	6	29.33
RC206	3	3	1173.32	1173.32	1173.32	1173.32	989.94	2661.2	3	26.433
RC207	3	3	1010.22	1010.22	1010.22	1010.22	837.5	2587.91	4	23.855
RC208	2	2	1090.06	1090.06	1027.41	1027.41	844.51	1904.33	9	2.308
AVG	6.86	6.86	1087.01	1085.7	1061.46	1060.17	872.52	4939.08	9.95	11.43
SUM	384	384	60872.36	60799.26	59441.78	59369.46	48861.39	276588.67	557	639.93

Table 36: TD-EVRPTWDCS-FR - D1 - recharging cost - results

Inst.	\bar{K}	K_{best}	\bar{rec}	rec_{best}	\bar{d}	d_{best}	t_{best}	tot_{best}	m	\bar{t}_e
C101	11	11	1127.69	1127.69	1106.05	1106.05	941.24	12445.81	12	2.308
C102	10	10	1045.58	1045.58	1023.38	1023.38	859.15	11642.15	9	2.383
C103	10	10	996.31	987.82	969.47	963.09	796.23	11441.02	9	4.369
C104	10	10	896.62	893.84	878.23	874.47	735.76	10871.88	7	4.365
C105	10	10	1077.23	1071.85	1045.85	1039.33	880	11362.68	9	1.934
C106	10	10	1068.1	1067.55	1037.12	1036.46	876.18	11111.53	10	2.073
C107	10	10	1060.02	1053.25	1031.52	1024.76	863.86	11046.46	9	2.255
C108	10	10	1037.44	1031.96	1008.97	1001.94	847.05	11206.54	9	3.097
C109	10	10	945.92	945.92	922.12	922.12	779.62	10882.3	8	3.475
C201	4	4	633.98	633.98	633.98	633.98	566.63	10209.7	4	6.052
C202	4	4	633.98	633.98	633.98	633.98	566.63	10209.7	4	16.646
C203	3	3	918.77	907.85	855.86	850.01	730.47	10088.51	8	5.13
C204	3	3	769.8	754.2	738.87	728.32	612.42	10120.58	5	7.719
C205	4	4	629.95	629.95	629.95	629.95	570.15	10218.52	3	6.039
C206	4	4	629.95	629.95	629.95	629.95	570.15	10218.52	3	7.018
C207	4	4	629.95	629.95	629.95	629.95	570.15	10218.52	3	8.578
C208	4	4	629.95	629.95	629.95	629.95	570.15	10218.52	3	6.877
R101	16	16	1629.68	1629.68	1597.63	1597.63	1363.94	3255.52	23	6.185
R102	15	15	1453.31	1453.31	1421.04	1421.04	1216.77	2955.74	22	10.337
R103	12	12	1210.48	1210.48	1174.49	1174.49	993.22	2501.27	14	10.634
R104	9	9	1109.12	1109.12	1049.41	1049.41	868.13	1990.06	14	13.361
R105	12	12	1414.96	1414.96	1336.93	1336.93	1119.33	2457.84	20	7.787
R106	11	11	1290.81	1290.81	1227.36	1227.36	1034.09	2260.44	16	10.612
R107	10	10	1086.95	1086.95	1053.42	1053.42	878.06	2155.87	14	9.947
R108	9	9	1016.39	1016.39	973.9	973.9	818.79	1940.67	13	10.957
R109	10	10	1210.95	1210.95	1161.69	1161.69	969	2172.51	18	6.3
R110	9	9	1113.5	1113.5	1052.01	1052.01	872.49	1963.21	14	7.522
R111	9	9	1167.69	1167.69	1086.23	1086.23	896.19	2016.7	18	5.695
R112	9	9	1058.85	1058.85	1006.79	1006.79	834.22	1965.92	14	6.428
R201	3	3	1245.86	1233.47	1244.27	1233.47	1034.98	2777.28	6	80.04
R202	3	3	1039.33	1033.67	1039.33	1033.67	865.41	2871.62	4	46.44
R203	2	2	1060.71	1043.63	1046.65	1036.45	855.99	1981.18	6	13.188
R204	2	2	779.65	764.16	779.65	764.16	625	1941.62	2	5.591
R205	2	2	1118.02	1084.15	1078.45	1063.97	879.95	1964.89	7	8.149
R206	2	2	1032	1004.19	1027.34	1001.13	827.75	1977.28	8	6.797
R207	2	2	859.14	842.92	859.14	842.92	702.75	1951.72	4	5.434
R208	2	2	752.12	747.7	752.12	747.7	618.6	1933.68	3	6.147
R209	2	2	938.72	927.62	937.24	927.62	760.13	1928.86	6	5.962
R210	2	2	916.82	902.2	916.82	902.2	738.7	1937.04	4	5.091
R211	2	2	838.96	825.08	838.96	825.08	689.79	1840.12	3	6.13
RC101	14	14	1703.64	1703.64	1668.39	1668.39	1403.92	2949.5	21	4.134
RC102	12	12	1583.06	1583.06	1525.87	1525.87	1278.92	2596.93	18	6.707
RC103	11	11	1377.18	1377.18	1336.18	1336.18	1111.63	2450.43	16	10.514
RC104	10	10	1140.44	1140.44	1124.86	1124.86	936.51	2214.16	14	8.881
RC105	12	12	1431.01	1431.01	1378.06	1378.06	1154.59	2501.98	14	7.785
RC106	11	11	1385.92	1385.92	1332.27	1332.27	1118.05	2379.36	15	7.881
RC107	10	10	1236.49	1236.49	1203.02	1203.02	994.73	2255.74	14	6.054
RC108	9	9	1171.04	1171.04	1106.47	1106.47	909.11	2022.78	11	5.948
RC201	3	3	1714.12	1648.28	1681.55	1637.37	1382.38	2751.09	11	53.846
RC202	3	3	1383.83	1377.15	1383.83	1377.15	1156.27	2752.48	4	55.253
RC203	3	3	1071.96	1044.48	1071.96	1044.48	882.39	2667.18	5	44.068
RC204	2.67	2	973.57	1112.33	950.28	1042.47	864.65	1907.76	10	25.724
RC205	3	3	1331.05	1278.57	1326.98	1278.57	1095.98	2589.94	7	41.704
RC206	3	3	1216.65	1189.38	1216.65	1189.38	1000.65	2525.58	3	33.634
RC207	3	3	1018.23	1002.61	1018.23	1002.61	842.34	2465.18	5	22.23
RC208	2	2	1003.01	979.53	957.56	941.1	767.22	1853.57	7	7.168
AVG	6.92	6.91	1086.01	1080.5	1059.79	1054.74	887.47	4877.46	9.7	12.97
SUM	387.67	387	60816.46	60507.86	59348.23	59065.21	49698.46	273137.64	543	726.58

Biography

Tomislav Erdelić was born on the 23rd of December 1990 in Zagreb, where he attended and finished elementary and high school. In 2009, he continued his education at the Faculty of Electrical Engineering and Computing, University of Zagreb, profile Control engineering and automation, where he received his M.Sc. degree with honor (CUM LAUDE). He graduated on the topic „Methods for dynamic path planning in multi-vehicle laboratory setup“ (mentor: Professor Stjepan Bogdan, Ph.D.). In November 2014, he started to work as a young researcher on the research project "System for route optimization in dynamic transport environment RC.2.2.08-0022" funded by the European Regional Development Fund. In February 2015, he enrolled in Ph.D. study „Technological Systems in Traffic and Transport“ at the Faculty of Transport and Traffic Sciences and for the study advisor he chose professor Tonči Carić, Ph.D. In January 2016, he was selected as an assistant in the engineering sciences, field technology of traffic and transport, at the Faculty of Transport and Traffic Sciences, University of Zagreb.

He currently works at Faculty of Transport and Traffic Sciences at the Department of Intelligent Transport Systems, and participates in teaching of the following courses: Algorithms and programming, Database Systems, Advanced Database System, Transport Geoinformation Systems and Transport Optimization. He was also a mentor on a dozen of undergraduate theses and immediate supervisor on several graduate theses. He has a C1.1 certificate in English language and a B1 certificate in German language. His skills include programming in several languages C++, C#, Python, and Java, and a knowledge of working in MATLAB, Android Studio, Linux, ASP.NET, ROS, ILOG, Microsoft Office, Microsoft SQL, MVC, SIMATIC STEP 7 (TIA Portal), etc.

His research interests include optimization procedures, data-mining, and machine learning procedures in traffic and transportation. In the current research process, he published two journal papers indexed in the Web of Science Core Collection (one in the Current Contents database) and eight conference papers.

List of published works

Papers in journals

1. Erdelić, T., Carić, T., Erdelić, M., Tišljarić, L., Turković, A., Jelušić, N., “Estimating congestion zones and travel time indexes based on the floating car data”, *Computers, Environment and Urban Systems*, Vol. 87, 2021, pp. 101604
2. Erdelić, T., Carić, T., “A survey on the electric vehicle routing problem: Variants and solution approaches”, *Journal of Advanced Transportation*, Vol. 2019, may 2019, pp. 1–48.

Papers at international scientific conferences

1. Tišljarić, L., Majstorović, Ž., Erdelić, T., Carić, T., “Measure for traffic anomaly detection on the urban roads using speed transition matrices”, in 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), 2020, pp. 252-259.
2. Tišljarić, L., Carić, T., Erdelić, T., Erdelić, M., “Traffic state estimation using speed profiles and convolutional neural networks”, in 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), 2020, pp. 2147-2152
3. Erdelić, T., Carić, T., Erdelić, M., Tišljarić, L., “Electric vehicle routing problem with single or multiple recharges”, *Transportation Research Procedia*, Vol. 40, 2019, pp. 217 - 224, TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport
4. Tišljarić, L., Erdelić, T., Carić, T., “Analysis of intersection queue lengths and level of service using gps data”, in 2018 International Symposium ELMAR, 2018, pp. 43-46.
5. Erdelić, T., Carić, T., Ravlić, M., “Predicting waiting times at intersections”, in 2017 International Symposium ELMAR, 2017, pp. 31-34.
6. Ravlić, M., Erdelić, T., Carić, T., “Optimizing charging station locations for fleet of electric vehicles using multi-source weber problem”, in 2016 International Symposium ELMAR, 2016, pp. 111-114.
7. Erdelić, T., Ravlić, M., Carić, T., “Travel time prediction using speed profiles for road network of Croatia”, in 2016 International Symposium ELMAR, Sept 2016, pp. 97-100.
8. Erdelić, T., Vrbančić, S., Rožić, L., “A model of speed profiles for urban road networks using g-means clustering”, in 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015, pp. 1081-1086.

Životopis

Tomislav Erdelić rođen je 23. prosinca 1990. godine u Zagrebu gdje je završio osnovnu školu i Opću gimnaziju u Sesvetama. Godine 2009. upisao je redoviti studij na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu smjer automatika, gdje je 16. srpnja 2014. godine stekao zvanje magistra inženjera elektrotehnike i informacijske tehnologije s pohvalom (CUM LAUDE). Diplomirao je na temu "Metode za dinamičko određivanje staza na laboratorijskom sustavu s više vozila" (mentor: prof. dr.sc Stjepan Bogdan) . U studenom 2014. godine zaposlio se na Fakultetu prometnih znanosti u Zagrebu kao mladi istraživač na istraživačkom projektu "Sustav za Optimizaciju Ruta u DINamičkom Transportnom Okruženju RC.2.2.08-0022" (SORDITO) financiran iz Europskog fonda za regionalni razvoj. U veljači 2015. godine upisuje poslijediplomski doktorski studij "Tehnološki sustavi u prometu i transportu" na Fakultetu prometnih znanosti te za studijskog savjetnika izabire prof. dr. sc. Tonči Carić. U suradničko zvanje asistenta iz područja tehničkih znanosti, polje tehnologija prometa i transport, izabran je u siječnju 2016. godine na Fakultetu prometnih znanosti, Sveučilište u Zagrebu.

Trenutno je zaposlen na Fakultetu prometnih znanosti u Zavodu za inteligentne transportne sustave, te je nastavno angažiran na kolegijima Algoritmi i programiranje, Baze podataka, Napredne baze podataka, Prometni geoinformacijski sustavi i Optimizacija prometnih procesa. Bio je mentor na desetak završnih radova i sudjelovao je kao neposredni voditelj na dva diplomska rada. Iz engleskog jezika ima položen certifikat C1.1, dok iz njemačkog jezika ima položen certifikat B1. U njegove vještine ubrajaju se programiranje u nekoliko programskih jezika C++, C#, Python i Java, te znanje rada u sljedećim programskim alatima: MATLAB, Android Studio, Linux, ASP.NET, ROS, ILOG, Microsoft Office, Microsoft SQL, MVC, SIMATIC STEP 7 (TIA Portal) 7 i dr.

Njegovo područje interesa uključuje optimizacijske tehnike, postupke rudarenja podataka i strojnog učenja u području prometa i transporta. U dosadašnjem istraživanju objavio je dva rada u časopisima indeksiranim u Web of Science Core Collection (jedan u Current Contents bazi podataka) i osam radova na međunarodnim znanstvenim konferencijama.

