

Izbor energetske optimalne rute električnog vozila s obzirom na konfiguraciju terena

Cvetek, Dominik

Undergraduate thesis / Završni rad

2016

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:119:817913>

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

Download date / Datum preuzimanja: **2024-04-19**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Dominik Cvetek

Izbor energetski optimalne rute električnog vozila s obzirom na
konfiguraciju terena

Završni rad

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

ZAVRŠNI RAD

Izbor energetske optimalne rute električnog vozila s obzirom na
konfiguraciju terena

Selection of energy optimal routes electric vehicles with regard to
the configuration of the terrain

Mentor: izv. prof. dr.sc. Tonči Carić

Student: Dominik Cvetek

JMBAG: 1191225501

Zagreb, lipanj, 2016.

IZBOR ENERGETSKI OPTIMALNE RUTE ELEKTRIČNOG VOZILA S OBZIROM NA KONFIGURACIJU TERENA

SAŽETAK

U ovom radu prikazan je način modifikacije Dijkstrinog algoritma u svrhu izračuna energetske optimalne rute. Istaknut je značaj poznavanja parametra stanja napunjenosti akumulatora te kako se energija akumulatora troši u odnosu na konfiguraciju terena po kojem vozi mobilni robot. Dan je osvrt na postojeće algoritme pronalaženja najkraćeg puta na grafu. Opisan je način na koji je moguća implementacija razvijenog algoritma u vozilima kao ITS usluga rutnih i navigacijskih vodiča.

KLJUČNE RIJEČI: Pioneer 3-AT mobilni robot, električno vozilo, konfiguracija terena, algoritmi najkraćeg puta na grafu, Dijkstrin algoritam, energetske optimalne rute, rutni vodiči i navigacija

SUMMARY

In this work has been shown way of modification Dijkstra's algorithm for the calculation of energy optimal path. It was pointed out on the knowledge of battery state of charge and state of discharge in regards to lie of the land on which mobile robot drive. In this work has been given review on algorithm for finding the shortest path in the graph. In this work has been described the way in which implementation in vehicles within the service of Intelligent transport system is possible.

KEYWORDS: Pioneer 3 - AT mobile robot, electric vehicle, the configuration of the terrain, algorithms shortest path in the graph, Dijkstra's algorithm, the optimal energy pathways, guidance and navigation

Sadržaj

1	Uvod.....	1
2	Rutni vodiči i navigacija kao usluga ITS-a	3
3	Električna i autonomna vozila.....	7
3.1	Električna vozila	7
3.2	Autonomna vozila.....	7
3.3	Osvrt na dosadašnja istraživanja.....	8
3.4	Pioneer 3 AT model električnog autonomnog vozila	9
4	Optimizacijski problemi u prometu i transportu	12
4.1	Teorija grafova.....	13
4.2	Najkraći put na težinskom grafu	16
5	Potrošnja energije električnog vozila na terenu s nagibom.....	25
5.1	Sile na kosini.....	25
5.2	Energija koju vozilo troši s obzirom na konfiguraciju terena.....	27
5.3	Definiranje parametara za težinu cestovnih segmenata	28
6	Primjena Dijkstrinog algoritam za izračun energetske optimalne rute.....	30
7	Zaključak.....	35
	Literatura	37
	Popis slike.....	40
	Popis tablica	40

1 Uvod

Električni pogon dobiva sve značajniju ulogu u autoindustriji, a poboljšanja autonomije električnih vozila doprinijeti će sve većoj upotrebi električnog pogona, ne samo u autoindustriji, nego i u globalnoj ekonomskoj slici. U zaštitu okoliša ulažu se veliki naponi i novac, a električni automobili ne proizvode štetne plinove i sigurni su za okoliš. Sve je veća potražnja javnosti za smanjenjem emisije štetnih plinova i zaštitom okoliša. Glavna prednost električnog automobila pred konvencionalnim je nulta emisija štetnih plinova. Ovisnost o naftnim derivatima u prometu potaknula je korištenje alternativnih izvora energije, gdje izrazito veliku primjenu nalaze električna vozila. Onečišćenja mogu nastati jedino pri proizvodnji električne energije [1] što je još jedan od razloga ulaganja u korištenje obnovljivih izvora energije. Vrijedi napomenuti da su onečišćenja pri proizvodnji električne energije uglavnom izvan gradova i područja gušće naseljenosti, što implicira na očuvanje kvalitete zraka u gradovima. Elektromotori imaju znatno veću iskoristivost od benzinskih i dizelskih motora, vozila su znatno jednostavnije konstrukcije što drastično smanjuje troškove održavanja vozila. Na električnom vozilu nema potrebe za promjenom motornog ulja koje predstavlja veliku potencijalnu opasnost za onečišćenje okoliša [2]. Električna vozila su tiha, stoga se u današnje vrijeme uglavnom primjenjuju u prostorijama koje zahtijevaju nisku razinu buke, primjerice u bolnicama, skladišnim prostorima i sl. [3]. Problem buke u gradovima moguće je reducirati korištenjem električnih vozila koja proizvode samo buku uzrokovanu kotrljanjem pneumatika po podlozi [1]. Zbog navedenih razloga broj električnih vozila raste, time postoji potreba za optimiziranjem potrošnje energije. Budući da su električna vozila značajno ograničena autonomnošću koju mogu postići. Jedna od usluga ITS-a je usluga rutnog vođača i navigacije i odnosi se na pred putno i putno informiranje o optimalnoj ruti ili putanji do specificirane destinacije.

Problem preusmjeravanja vozila na energetski optimalnu rutu predstavlja problem kombinatorne optimizacije. Metode rješavanja optimizacijskih problema mogu biti egzaktna, heurističke i metaheurističke metode. Egzaktna metoda su u stanju naći optimalno rješenje ali se zbog dugotrajnog izvršavanja mogu primijeniti samo na manjim problemima. Heurističke i metaheurističke metode ne garantiraju pronalazak optimalnog rješenja, ali je vrijeme izvršavanja algoritma manje i daju zadovoljavajuće rezultate. Ukoliko je potrebno pronaći rješenje za jedno vozilo nema potrebe za heurističkim metodama jer je i egzaktna metoda u zadovoljavajućim granicama izvršavanja. Promatramo li jedno ili više vozila u postavljenom razmatranju konfiguracija terena ima značajnu ulogu kod izbora rute vozila.

Hipoteza ovog rada usmjerena je pronalasku parametara za optimizacijski algoritam. Optimizacijski algoritam odabrao bi energetski optimalnu rutu u svrhu smanjenja potrošnje energije vozila. Smanjenjem potrošnje energije vozila posredno se povećava njegova autonomija. Proširenje problema koje je bitno u praksi bilo bi da se istovremeno prati stanje napunjenosti akumulatora. Problem se sagledava na način da vozilo ne smije ostati bez energije akumulatora, na nekom cestovnom segmentu gdje nema mogućnost ponovnog punjenja.

Opći cilj istraživanja prikaz je mogućnosti optimizacije potrošnje energije električnog vozila radi povećanja njegove autonomije. Specifični cilj istraživanja je matematički modelirati potrošnju energije na nagibu i potrošnju na terenu bez nagiba. Stvarne podatke potrošnje energije električnog mobilnog robota Pioneer 3 AT je potrebno modificirati na način da se

određenom metodom za pronalazak najkraćeg puta pronađe energetske optimalna ruta. Predložena je optimizacija potrošnje električne energije primjenom Dijkstrinog algoritma s osvrtima na druge algoritme za pronalaženje minimalnog puta na asimetričnom težinskom grafu. Navedeno bi se postiglo preusmjeravanjem električnog vozila na energetske optimalnu rutu, uzimajući u obzir ograničenje kapaciteta akumulatora. Rad je podijeljen u 7 cjelina:

1. Uvod
2. Rutni vodiči i navigacija
3. Električna i autonomna vozila
4. Optimizacijski problemi u prometu i transportu
5. Potrošnja energije električnog vozila na terenu s nagibom
6. Primjena Dijkstrinog algoritma za izračun energetske optimalne rute
7. Zaključak

U prvom dijelu rada definiran je problem i predmet istraživanja te opisana struktura rada.

U drugom dijelu rada opisani su rutni vodiči kao usluga ITS-a s naglaskom na multimodalne vodiče i kooperativno vođenje te informiranje korisnika usluge u prometu sustavu.

U trećem dijelu rada opisana su električna vozila s njihovim glavnim negativnim karakteristikama s naglaskom na nužan napor više znanstvenih područja kako bi se riješio navedeni problem. Opisana su i autonomna vozila s naglaskom na njene potencijalne prednosti u odnosu na konvencionalna vozila. Također pobliže je objašnjen pojam autonomnih vozila te je dan osvrt na dosadašnja istraživanja koja su usko vezana za rad. Opisane su karakteristike malog električnog autonomnog vozila. Mobilnog robota Pioneer 3-AT na kojem su vršena eksperimentiranja i dobiveni podaci koji su naknadno primijenjeni na optimizacijskom algoritmu.

U četvrtom dijelu rada izloženi su neki optimizacijski problemi koji se odnose na promet i logistiku. Definirani su pojmovi iz matematičkog područja teorija grafova. Navedeni su određeni problemi koji nastupaju želimo li optimizirati put na grafu. Detaljnije je opisana posebna metoda traženja najkraćeg puta na grafu kao podskup optimizacijskih problema na grafu. Opisane su neke metode za rješavanje najkraćeg puta na težinskom grafu. Najčešće područje primjene određene metode i kada je pogodna za korištenje. Opisane metode za pronalazak najkraćeg puta su Dijkstra, Bellman-Fordov, A zvjezdica i D zvjezdica algoritmi.

U petom djelu rada analizirana je potrošnja energije vozila na terenu s nagibom i terenu bez nagiba. Također definirani su parametri za težinski asimetričan graf. Korišteni su izmjereni podaci potrebne energije pri različitoj konfiguraciji terena Pioneer 3 –AT mobilnog robota.

U šestom dijelu rada opisna je implementacija Dijkstrinog algoritma te je modificirana osnovna verzija istog algoritma kako bi se računala energetske optimalna ruta. Dan je kritički osvrt na rezultate mjerenja i mogućnosti primjene modificiranog Dijkstrinog algoritma u vozilima.

U posljednjem djelu rada dan je zaključak rada te mogućnosti nekih budućih istraživanja u ovome području.

2 Rutni vodiči i navigacija kao usluga ITS-a

Usluge rutnog vodiča mogu se odnositi na pred putno i putno informiranje o optimalnoj ruti do specificirane destinacije [4]. Praćenje i usmjeravanje vozila i putnika ima veliku ulogu u smanjenju prometnih zagušenja, smanjenju vremena putovanja i uštedi energije što olakšava dostupnost za to namijenjenih sustava. Navigacijski sustavi u vozilima mogu biti zemaljski

(npr. GSM) i satelitski (npr. GPS) navigacijski sustavi, koji omogućuju pokrivenost na onim područjima koja zemaljski sustavi ne pokrivaju.

Primjenom računske navigacije i GPS (engl. Global Positioning System) tehnologije moguće je dinamički izračun optimalne rute puta što nije slučaj za primjenu isključivo vizualne ili instrumentalne navigacije. Putem instrumentalne navigacije nije moguće postavljanje uputa vizualno i auditivno. Usluge sustava rutnog vodiča i navigacije podržavaju navedenu tehnologiju te omogućuju jednostavnije i efikasnije putovanje do ciljanog odredišta. Također uz korištenje digitalne mape sa statičkim informacijama moguće je kombinirati stvarno vremenske informacije radi izbjegavanja ruta koje su neprohodne ili teško prohodne radi raznih čimbenika. Na Fakultetu prometnih znanosti u sklopu europskog projekta (Sustav za optimizaciju ruta u dinamičkom transportnom okruženju –SORDITO) nastojalo se pomoću GPS tragova¹ analizirati vrijeme putovanja u najvećim hrvatskim gradovima s obzirom na prometna zagušenja. Na temelju dobivenih GPS podataka izračunata je brzina kretanja vozila u određeno doba dana. Predviđajući vrijeme putovanja moguće je dostavna vozila čije je vrijeme dostave od krucijalne važnosti permutirati na način da se izbjegnu ili smanje posljedice prometnog zagušenja.

Općenito, autonomni rutni vodič (engl. Autonomous Route Guidance) izračunava optimalne rute na računalnoj opremi vozila uz korištenje digitalne mape koja se nalazi na računalu. Dakle, vozač unese konačno odredište puta te računalno odredi najbolju rutu na osnovi trenutne lokacije vozila i korisnikovih prioriteta. Primjerice vozač može odabrati vremenski najpovoljniju rutu ili energetski najpovoljniju rutu ili pak neku ne navedenu. Ako na nekoj dionici puta vozač pogriješi u navigacijskim smjernicama, navigacijska oprema prepozna pogrešku te daje alternativni plan puta.

Centralizirani dinamički rutni vodič (engl. Centralised Dynamic Route Guidance) rade na principu obrade zahtjeva u središnjem računalu prometnog informacijskog centra. Centar raspolaže dinamičkim podacima o stanju prometa te na temelju podataka i dobivenog zahtjeva vozača izračunava optimalnu rutu koju mu zatim prosljeđuje. Komunicira sa vozilima i time usmjerava i harmonizira promet. Za ovakav način komunikacije i obrade nužan je prometni centar koji nadzire pojedino područje.

Dualni mod rutnog vodiča bila bi kombinacija autonomnog i centraliziranog rutnog vodiča. Omogućuje obradu stvarno vremenskih podataka o prometu u čitavoj prometnoj mreži koju prometni centar nadzire

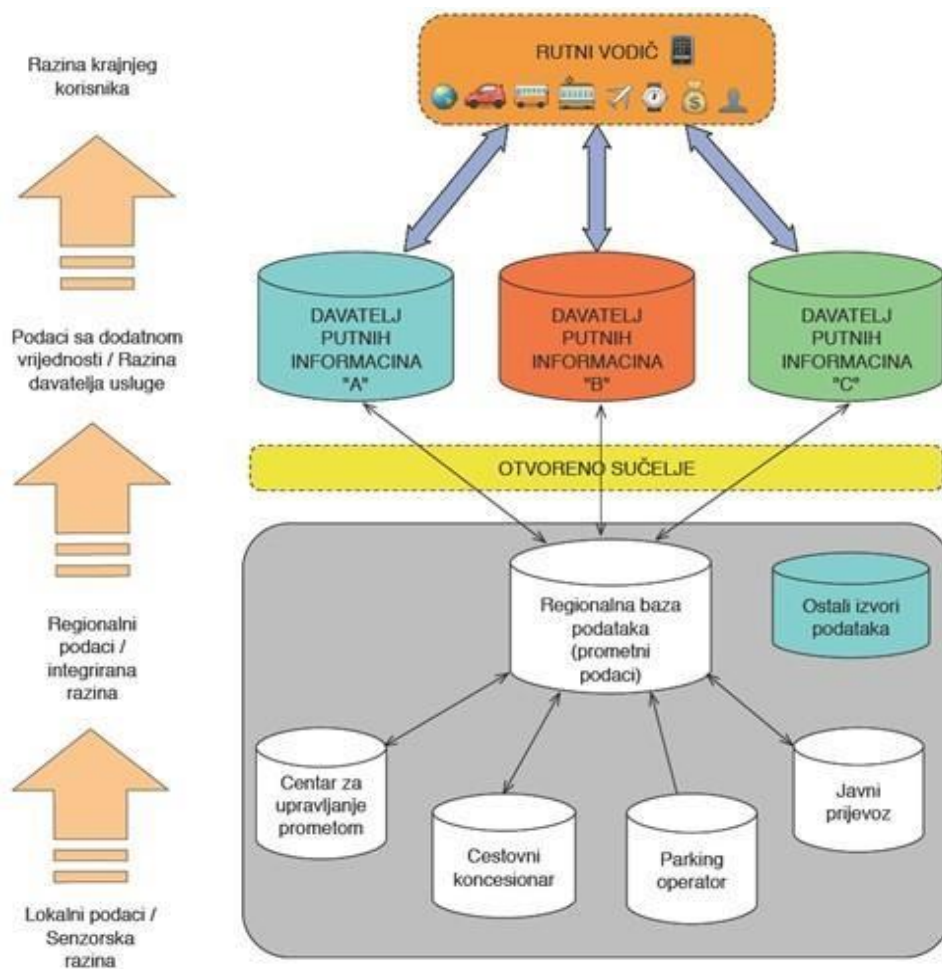
Slično drugim ITS uslugama zahtjeve korisnika potrebno je specificirati i usuglasiti kako bi se mogle provesti odgovarajuće funkcijske specifikacije. Bitne koristi imati će individualni korisnici i davatelji usluga. Neki od posebnih zahtjeva korisnika i interesi davatelja usluga

¹ Povjesni GPS podaci dostavnih vozila u vremenu i prostoru.

vezani su za zaštitu privatnosti, zajedničko financiranje i integraciju s drugim lokacijskim sustavima.

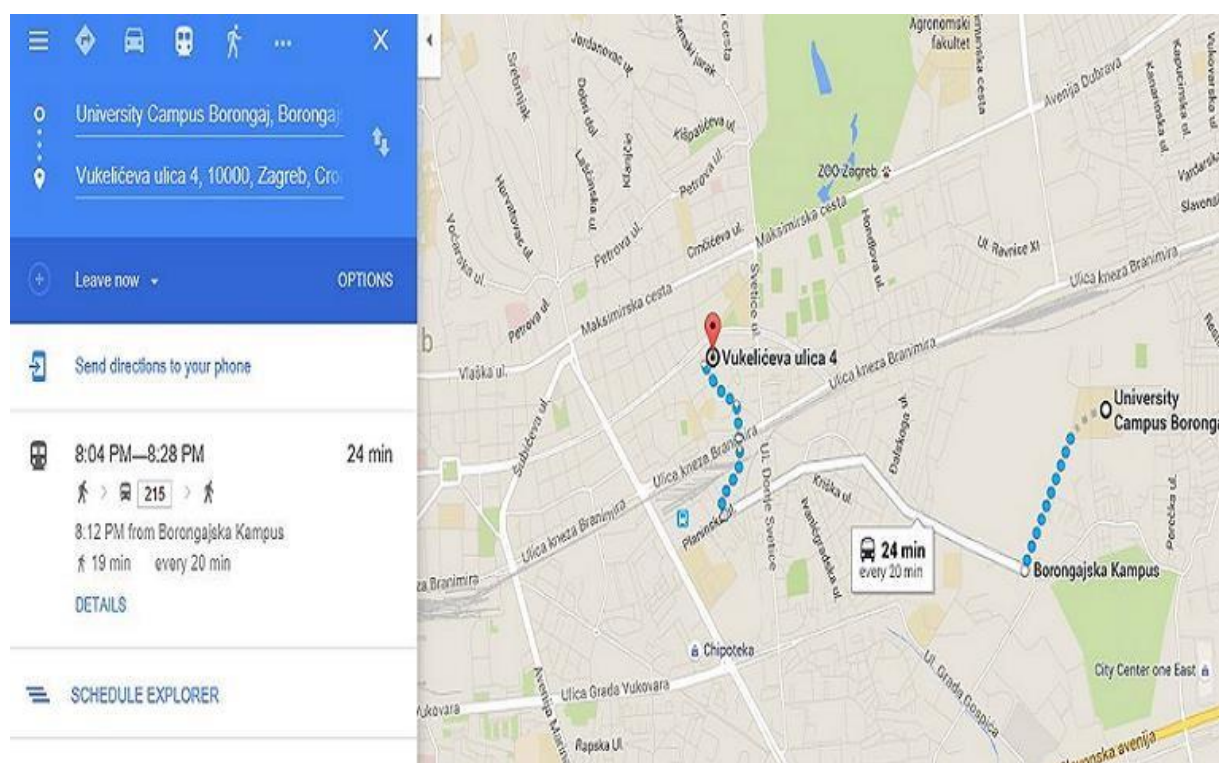
Multimodalni rutni vodiči (engl. Multimodal Route Planners) su usluge inteligentnih transportnih sustava temeljene na kooperativnoj komunikaciji u prometnom sustavu. Temelj kooperativnih multimodalnih vodiča je optimalno korištenje putnih i prometnih podataka poput informacija o polascima i odlascima svih dostupnih prijevoznih sredstava kao što su primjerice tramvaji, vlakovi, autobusi i sl.. Mogućnosti kupnje karata, stanju na cestama, vremenu potrebnog putovanja ovisno o modu prijevoza, incidentnim situacijama i dr.. Kooperativni multimodalni putni vodiči prikazan na Slici 1. pružaju spektar informacija jer združuju više kooperativnih usluga u jedinstvenu cjelinu kao što su: trenutno stanje u prometu, lokacijama parkirališta i trenutno dostupni broj parkirnih mjesta, informiranje o javnom prijevozu, planiranje multimodalnog putovanja. Ova ITS usluga prije svega korisnicima pruža odgovor na pitanje "Koje su mogućnosti krećemo li se od mjesta A u mjesto B u određenom vremenu polaska ili dolaska?", uzimajući u obzir troškove pojedine rute, optimalne rute s obzirom na želje korisnika ali i meteorološke uvjete na prometnicama. Korisnici dobivaju detaljne upute kako bi od trenutne pozicije došli do odredišta [5].

Najveći izazov pri realizaciji multimodalnih vodiča su organizacijski problemi presudnu ulogu ima Europska komisija kako bi se uspostavio pravni okvir i uspostavila standardizacija. Postoji svijest o organizacijskim i tehničkim ograničenjima što se tiče stvarno vremenskih podataka te je načelo otvorenog pristupa podacima podržano od strane većine interesnih skupina iako u različitim opsezima. Glavni zahtjev dionika je pouzdanost podataka koji se pružaju krajnjim korisnicima te potvrda kredibiliteta izvora podataka [6]. S tehničkog gledišta, cilj multimodalnih rutnih vodiča je razmjena informacija između različitih ITS aplikacija na način da se ITS stranice nalaze u domenama koje su jasno razgraničene, upravljive te sigurne za razmjenu podataka. Tok podataka osnovnog sustava može se prikazati kao niz uzastopnih operacija koje sačinjava prikupljanje podataka, validacija i integracija podataka, obrada podataka, uključivanje podataka u uslugu sustava, distribucija podataka, korištenje usluga sustava od strane korisnika. Kvaliteta izvršavanja svake operacije utječe na konačnu kvalitetu multimodalnih vodiča. Kako bi se očuvala kvaliteta razmjene podataka za kooperativne vodiče potrebno je uspostaviti koncept podatkovne i informacijske transparentnosti.



Slika 1. Logički prikaz arhitekture multimodalnih rutnog vodiča, [7]

Jedan od primjera realiziranog sustava multimodalnog rutnog vodiča je Google Transit. Google Transit je rutni vodič koji je namijenjen planiranju putovanja na temelju najnovijih podataka agencija i Google Maps aplikacije. Google Maps čini skup podataka o prometnicama sa pripadajućim svojstvima i oznakama, stajalištima javnog gradskog prijevoza, voznim redovima, rutama te uz korištenje Google Traffic usluge koja sad predviđa i vrijeme putovanja ovisno o modu koji koristimo. Primjer korištenja Google Transit usluge prikazan je na Slici 2.



Slika 2. Usluga Google Transit

3 Električna i autonomna vozila

U poglavlju električna i autonomna vozila ukratko se opisuju značajke navedenih vozila. Električna vozila su sve više u uporabi i predviđa se njihova sve veća zastupljenost. Električna vozila su znatno ograničena autonomijom. Nastoji im se postići sve veća autonomija, također autonomna vozila su predmetom istraživanja već određeni period stoga je dan kratak informativni pogled na navedena vozila. Pogled je koristan za lakše razumijevanje problematike ovog rada, s obzirom da se algoritam testirao s malim električnom autonomnim vozilom kojeg predstavlja Pioneer 3-AT mobilni robot.

3.1 Električna vozila

Električna vozila za pogon koriste električnu energiju koja je spremjena u akumulatoru. Energija potrebna za domet električnih vozila uvelike ovisi o kapacitetu ugrađenih akumulatora.

Primjerice tvrtka Tesla Motors proizvela je električni automobil Model S s akumulatorom kapaciteta 40 , 60 Ah i 85 Ah (autonomija do 450 km). Hibridna vozila imaju kapacitet akumulatora približno od 3 do 5 Ah, s električnom autonomijom od 60 do 70 km [3].

Akumulator predstavlja još uvijek najveći nedostatak ove vrste vozila zbog ograničenog kapaciteta i vremena potrebnog za punjenje što potvrđuje istraživanje [8].

Vozila s unutarnjim izgaranjem mogu biti napunjena gorivom vrlo brzo i punjenje je moguće na široko rasprostranjenoj mreži punionica. Električna vozila imaju manji maksimalni domet po punjenju u odnosu na automobile koja pokreću fosilna goriva, a punjenje traje znatno duže. Mnogi proizvođači označili su električna vozila na tržištu kao „dnevno vozilo“, pogodno za gradske izlete i druga kratka putovanja. Potreban je napor više znanstvenih područja kako bi se doskočilo njihovim problemima u prometovanju.

3.2 Autonomna vozila

Autonomna vozila predmet su istraživanja već duži period. Vojska u mnogim državama ulaže sredstva u razvoj autonomne transportne tehnologije koja bi donijela mnoge prednosti na bojnopolju. Kako bi potaknuli razvoj, organiziraju se utrke autonomnih električnih vozila popraćene novčanim nagradama, u kojima u najvećem dijelu sudjeluju sveučilišta u suradnji s autoindustrijskim kompanijama [9]. Mnoga istraživanja provode se s naglaskom na

upravljačkim funkcijama i algoritmima koja će olakšati upravljanje ili ga preuzeti u potpunosti, omogućavajući time putniku da nakon unosa željene destinacije neometano uživa u prijevozu. Primjer automobila s takvim performansama je Nissan IDS inteligentni sustav upravljanja (engl. Intelligent Driving System Concept) koji je predstavljen na sajmu automobila u Tokiju

2015. Uzimajući u obzir čovjeka kao uzročnika nastanka prometne nesreće, zbog vožnje u alkoholiziranom stanju, umora, nedovoljne pažnje na cestama i slično jasno je da se razvojem autonomnih vozila povećava i sigurnost prometa, osobito cestovnog [10].

U Sjedinjenim Američkim Državama 2013. godine NHTSA (engl. National Highway Traffic Safety Administration) klasificirala je razine autonomije cestovnih motornih vozila. Danas se ona primjenjuje u cijelom svijetu i postala je svojevrsan referentni faktor za kategorizaciju razine autonomije pojedinih cestovnih vozila. Razina autonomije svakog vozila može se

podijeliti u pet kategorija. NHTSA definira pet stupnjeva autonomije, počevši od razine 0 do razine 4. Nulta razina autonomije je ne autonomna razina, vozač u potpunosti kontrolira upravljanje vozilom. Vozačeva kontrola podrazumijeva upravljanje kočnicama, upravljačem, papučicom pogonske snage u svakom trenutku. Prva razina autonomije uključuje jednu ili više specifičnih upravljačkih funkcija. Takvi primjeri uključuju elektronsku kontrolu stabilnosti ESP (engl. Electronic Stability Control), sustave protiv blokiranja kotača ABS (engl. Anti-lock

Braking System), sustave za poboljšanje stražnje preglednosti (engl. Blind Spot Information System). Druga razina uključuje autonomiju za barem dvije primarne kontrolne funkcije vozila, koje rade kooperativno kako bi vozača oslobodile od upravljanja tim funkcijama. Primjer je sustav za adaptivnu prilagodbu brzine (engl. Adaptive Control Speed) u kombinaciji sa sustavom praćenja (ne prelaska) bijele kolničke trake LWDS (engl. Lane Departure Warning System). Vozila koja posjeduju treću razinu autonomije su sposobna od vozača preuzeti potpunu kontrolu nad svim sigurnosnim funkcijama vozila u određenim kritičnim trenucima, u takvim situacijama od vozača se očekuje gotovo potpuno oslanjanje na autonomno vozilo. Zbog toga vozač mora biti u mogućnosti za povremenu kontrolu, ali s dovoljno vremena za siguran prijelaz na njegovo upravljanje. Na primjer, vozilo će upravljati vožnjom, no ako sustav nema dostatnu podršku senzora koji mu omogućuju upravljanje preuzeti će je vozač. Google automobil primjer je ove razine, „polu-autonomnog” vozila. Četvrta razina podrazumijeva potpuno autonomno vozilo: vozilo je dizajnirano za provedbu svih kritičnih operacija i prati uvjete na prometnici tijekom čitave vožnje [11]. Ovakav oblik vožnje zahtijeva od vozača unos željene rute ili destinacije, ali ne zahtijeva nikakve ostale sposobnosti za upravljanje vozilom niti u jednom trenutku vožnje. Ovakva vozila sposobna su prometovati i bez prisutnosti ljudi.

3.3 Osvrt na dosadašnja istraživanja

Istraživanja, [8], [12], [13], [14] i [15] bave se tematikom optimiziranja potrošnje energije električnih i hibridnih vozila. Istraživanje, [4] predlaže optimizaciju izradom profila brzina koji bi smanjili potrošnju električne energije. U [8] prikazuju se energetske optimalne rute s dodatnim uvjetom da vozilo neće stići na odredište, već da mora stati na punionicu. Istraživanja [5], [6] i [7] predlažu primjenu Dijkstrinog algoritma za rješavanje problema energetske optimalne rute, no navode kako se problem optimizacije potrošnje energije ne može riješiti klasičnim

Dijkstrinim algoritmom. Razlog tomu je što za funkcioniranje Dijkstrinog algoritma energetske težine pridodane grafu ne smiju biti negativne vrijednosti, koje bi se pojavile zbog regenerativnog kočenja. U ovome radu optimizirala se potrošnja mobilnog robota koji nema opciju regenerativnog kočenja pa se to ograničenje nije uzelo u obzir. Također u istraživanju [5] se u proračun potrošnje energije uzelo u obzir i zagušenje prometne mreže, energetske se optimizirala ruta i po pitanju koliko vozilo troši zbog stani-kreni vožnje. U istraživanju [6] takvo ograničenje zanemaruje se i algoritam se primjenjuje na težinskom grafu koji posjeduje samo zračne udaljenosti. Na takav način postavljen je i graf u ovome radu.

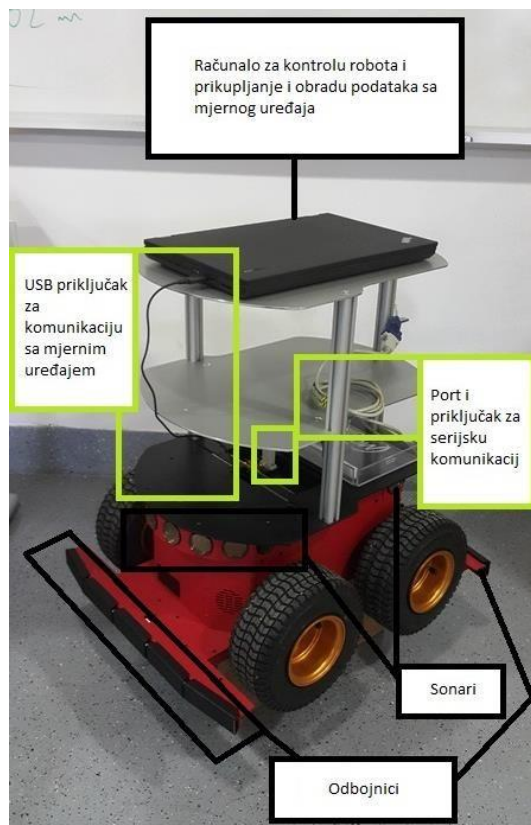
3.4 Pioneer 3 AT model električnog autonomnog vozila

U ovom radu umanjeni prikaz autonomnog električnog vozila mobilni je robot Pioneer 3-AT. Mobilni robot Pioneer 3-AT², ili skraćeno P3-AT, potječe iz porodice malih i inteligentnih robota, koji imaju pogon na dva ili četiri kotača. Arhitekturu Pioneer skupine robota razvio je dr.sc. Kurt Konolige sa sveučilišta Stanford. Pioneer istraživački roboti nalaze se među svjetski najpopularnijim inteligentnim robotima za edukaciju i istraživanje. Pioneer roboti prilagodljivi su i nadograđivi, pogodni za rad u laboratoriju i na otvorenom [16].

3.4.1 Fizičke karakteristike

Glavni dijelovi P3-AT mobilnog robota su: kućište, nos, konzola, paluba, pogonski kotači, mreža sonara, razni priključci te kontrolne lampice. Konzola i paluba čine dio robota, u koji se mogu ugraditi različiti uređaji ili osjetila robota. Unutar aluminijskog kućišta nalaze se akumulator, pogonski motori, elektronika i druge standardne komponente poput prednjih i stražnjih sonara i odbojnika [17]. Na Slici 3. prikazane su osnovne fizičke komponente. Za kontrolu mobilnog robota zadužen je 32-bitni mikrokontroler tvrtke *Renesas SH2 – 7144 RISC* s ARCOS [18] (engl. Advanced Robot Control and Operations Software) programskim paketom (engl. firmware) kao i Pioneer SDK (engl. Software Development Kit) napredni razvojni programski paket. Akumulatori/baterije mogu pogoniti robota 2 do 4 h bez dodatnih pribora i osjetila. 8 prednjih i 8 stražnjih sonara ugrađenih na robota moguće je koristiti kao cjelinu ili pojedinačno te s njima detektirati objekte koji su udaljeni od 15 cm do 7 m. Maksimalna brzina robota iznosi 0.75 m/s, a nosivosti je do 40 kg ovisno o terenu na kojem se nalazi [16]. Za mjerenje brzine razvijene na kotačima robot koristi dva enkodera, jedan za lijevi i jedan za desni kotač. Kako bi se kompenzirale pogreške u mjerenju uzrokovane proklizavanjem kotača. Moguće je korigirati enkodere s obzirom na krajnju točku.

² Korišteni mobilni robot Pioneer 3-AT nabavljen je iz tri projekta Sveučilišta u Zagrebu: Optimizacija ruta malih električnih vozila uz kriterij najmanje potrošnje energije, Optimalna navigacija malih električnih vozila u unutarnjim prostorima, Optimiranje ruta flote vozila primjenom više-agentskih optimizacijskih algoritama i stvarno-vremenskih prometnih podataka, 2013-ZUID-2



Slika 3. Fizičke komponente Pioneer 3-AT mobilnog robota

Osjetila robota proširuju se daleko više od uobičajenog korištenjem lidara, ugrađenim ispravljačem za pogreške pri proklizavanju, GPS-om, odbojnicima, računalnim vidom, robotskom rukom itd. [16]. Za pokretanje koristi četiri elektromotora koji zupčastim remenom prenose snagu na kotače. S obzirom na os simetrije robota dva elektromotora pogone lijevu stranu robota, a dva desnu. Od dva elektromotora jedan je glavni i na sebi ima enkoder, a drugi je sporedni. Tablica 1. prikazuje ostale specifikacije Pioneer platforme.

Prijenosno računalo koje se koristilo za upravljanje mobilnim robotom i procesiranje podataka je Lenovo ThinkPad T540 sa 8 GB RAM memorije, 64-bitnim i7 procesorom četvrte generacije, grafičkom karticom GeForce GT730M te Windows 7 operativnim sustavom³.

³ Korišteno prijenosno računalo nabavljeno je u projektu SORDITO - Sustav za optimizaciju ruta u dinamičkom transportnom okruženju RC.2.2.08-0022, koji je sufinanciran sredstvima Europske unije iz Europskog fonda za regionalni razvoj

Tablica 1. Specifikacije mobilnog robota Pioneer 3-AT [18]

Težina (bez akumulatora i dodatne opreme):	14 kg
Težina (s aluminijskim stalkom i prijenosnim računalom):	33 kg
Nosivost (laboratorij):	40 kg
Maksimalna brzina:	0,75 m/s
Maksimalni iznos nagiba terena:	40%
Vrijeme trajanja rada:	2-3 h
Vrijeme punjenja baterija:	6 h
Broj baterija:	3
Kapacitet svake baterije:	9 Ah
Nazivni napon baterije:	12 V
Vrsta baterije:	Olovne baterije

3.4.2 Programska podrška

Programska podrška sastoji se od niza programa koji omogućuju komunikaciju s robotom putem napisane biblioteke naredbi. Predstavlja svojevrsan jezik koji omogućuje davanje naredbi kojima se upravlja robotom te dolazi s kompletnom programskom podrškom prilagođenom za njega. MobileRobots programska platforma uključuje Pioneer SDK, kompletni set programa i datoteka koji omogućavaju razvoj željenih projekata.

3.4.3 Osnovna podrška

ARIA (engl. Advanced Robot Interface for Applications) pruža programsku podršku za kontrolu, komunikaciju s robotom i primanje podataka. Uključuje otvorenu infrastrukturu u

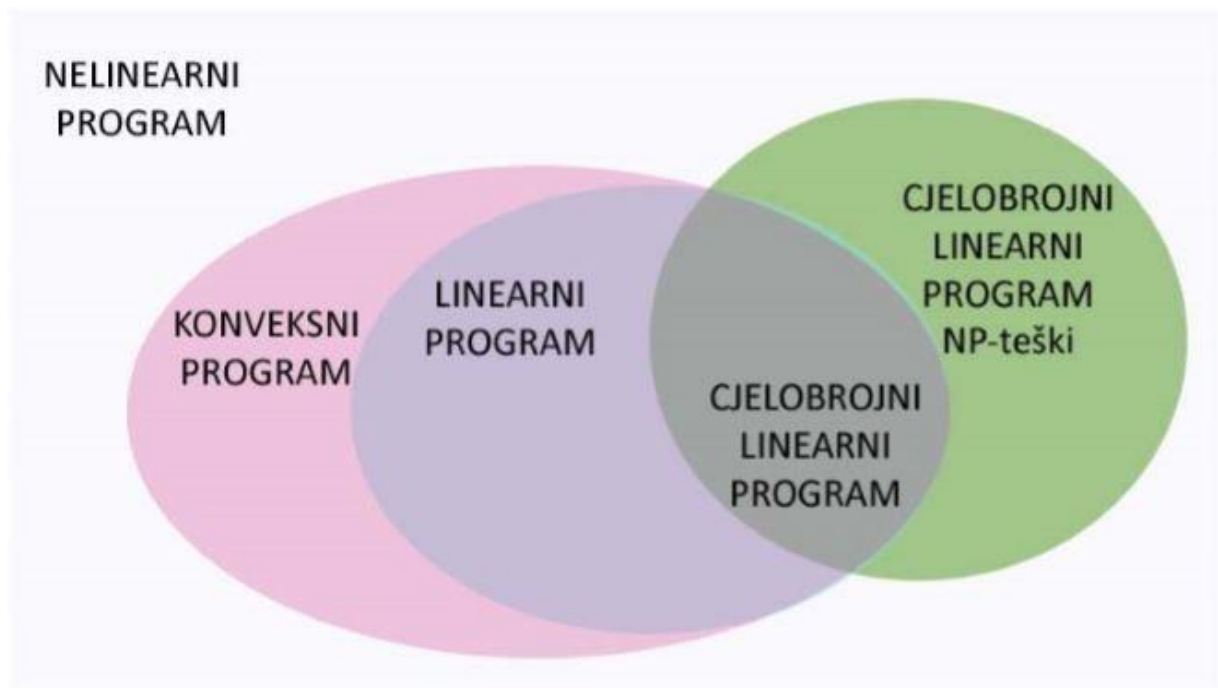
obliku biblioteka klasa pisanih u C++, Python i Java programskim jezicima. MobileSim otvoreni je simulator koji omogućuje otklanjanje pogrešaka i eksperimentiranje s ARIA ili drugim programima koje podržava MobileRobots platforma. MobileEyes grafičko korisničko sučelje za daljinsko upravljanje i nadgledanje robota. Mapper 3-Basic alat za kreiranje i montažu karte za potrebe ARIA, MobileSima i navigacijskih programa. SONARNL pruža aproksimaciju željenih parametara na temelju sonara za lokalizaciju i navigaciju robota [19].

4 Optimizacijski problemi u prometu i transportu

Veliki broj problema u prometu može se matematički modelirati. Mnogo prometnih i logističkih problema moguće je optimizirati kako bi unaprijedili prometni ili logistički sustav. Na primjer: dostava pošte, skupljanje otpada, čišćenje snijega, posipanje ulica solju zimi, čišćenje ulica, održavanje puteva, određivanje ruta školskih autobusa i mnogi drugi. Održavanje putova zimi zahtijeva mnogo složenog strateškog i operativnog planiranja. Optimiranje čišćenja snijega i posipanja ulica solju treba izvesti pazeći na dva aspekta sigurnost i ekonomičnost. Sigurnosni aspekti zahtijevaju da najosjetljivije točke mreže (primjerice ona mjesta koja se lede prva) budu očišćena prva. Ekonomičnost zahtijeva da ruta čišćenja bude najjeftinija. Glavni logistički problemi uključuju pozicioniranje skladišta, raspoređivanje radnog osoblja, određivanje ruta dostavnih vozila, određivanje voznog reda autobusa i vlakova. Svi ovi problemi mogu biti riješeni koristimo li metode optimizacije koji uključuju algoritme teorije grafova [20].

Općenito optimizacija ili matematičko programiranje je grana matematike koja proučava probleme traženja točaka u nekom vektorskom prostoru koja zadovoljavaju neka ograničenja., a u kojoj funkcija (funkcija cilja) doseže ekstremne vrijednosti. Područja matematičkog programiranja ili optimizacije se mogu podijeliti na linearno, nelinearno, diskretno i stohastičko programiranje i teoriju igara [21].

Promatraju se problemi nad konačno dimenzionalnim prostorima. Svaki optimizacijski problem može se najopćenitije opisati kao problem nelinearnog programiranja [22]. Podskup nelinearnog programa je konveksni program. Podskup konveksnog programa je linearni program, kojem ako damo dodatno ograničenje da sve komponente vektora x moraju biti cijeli brojevi, dolazimo do skupa cjelobrojnih linearnih programa. Pošto se ne mogu svi cjelobrojni programi riješiti simplex metodom karakterističnom za linearno programiranje. Takvi problemi cjelobrojnog linearnog programa spadaju u klasu NP-teških problema za koje nisu poznati efikasni algoritmi koji se izvršavaju u polinomijalnom vremenu. Na Slici 4. je prikazana formulacija optimizacijskih problema.



Slika 4. Optimizacijski problemi prikazani pomoću skupova, [22]

Kombinatorna optimizacija je matematička disciplina koja proučava probleme nalaženja ekstremnih vrijednosti funkcije definirane na konačnom skupu. Najpoznatiji optimizacijski problemi su problem kineskog poštara, problem trgovačkog putnika, potraga za najkraćim putem [22]. Neke glavne instance optimizacijskih problema su problem cjelobrojnog programiranja i optimizacijski problemi na grafovima. Problemi kombinatorne optimizacije na grafovima se mogu prikazati kao problemi cjelobrojnog programiranja. Ukoliko za dani problem ne postoji efikasna egzaktna metoda pristupa se približnom rješavanju pomoću odgovarajućih heurističkih metoda

Problem traženja najkraćeg puta u grafu vrlo je čest praktičan problem. Mreže prometnica mogu se predstaviti kao graf, stoga problem traženja najkraćeg puta ovdje može naći svoju primjenu. Mnoga vozila koriste navigacijsku tehnologiju te uz pomoć računala određenim metodama za pronalaženje najboljeg puta. Sugeriraju vozaču kojim bi putanjom trebao voziti od jedne do druge točke na zemljopisnoj karti.

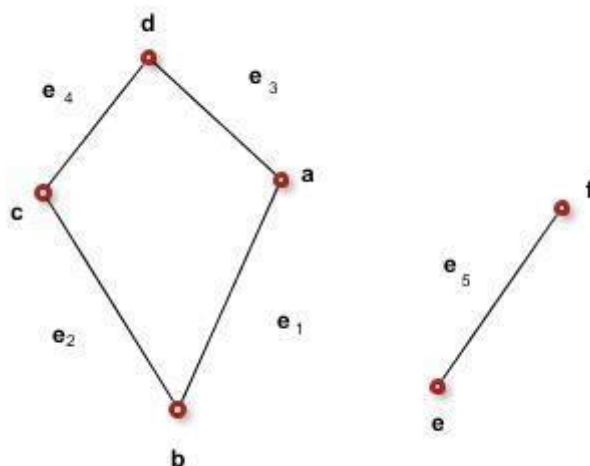
4.1 Teorija grafova

Grafovi su jedna od osnovnih i najčešće primjenjivanih matematičkih struktura. Grafovima se kvalitetno mogu prikazati i modelirati neke pojave i situacije. Graf je definiran s dva skupa: vrhovi V (engl. Vertex) i bridovi E (engl. Edges). Odnosno graf je uređeni par $G = (V, E)$ gdje je $\emptyset = V = V(G)$ skup vrhova na grafu, a $E = E(G)$ je skup bridova na grafu disjunktan sa skupom V , gdje svaki brid $e \in E$ spaja dva vrha $v \in V$. Bridom [23]. Primjer nepotpunog grafa dan je na Slici 5. Mreža prometnica može se opisati grafom [24]. Vrhovi grafa (čvorovi) mogu predstavljati raskrižja, lokacije korisnika, lokacija skladišta ili lokacije punionica, odnosno polazišne ili odredišne točke, a bridovi predstavljaju cestovne segmente prometne mreže.

Svaki graf se po broju bridova koji izlaze iz vrhova svrstavaju u Pseudografove. Pseudografovi su multigrafovi koji ne dozvoljavaju petlje iz vrhova. Multigraf je podgraf pseudografa koji ne sadrži petlje, ali sadrži višestruke bridove. Jednostavni graf ne sadrži ni petlje, ni višestruke bridove [22].

Put na grafu duljine k je svaki niz grana koji ima slijedeća svojstva. Prvo svojstvo je da grana e_1 polazi iz proizvoljnog vrha, dok je drugo svojstvo da grana e_i ($i = 2, \dots, k$) počinje u onom vrhu u kojemu završava grana e_{i-1} . Lanac duljine k na grafu definiramo kao naizmjenični niz vrhova i grana $v_1, e_1, v_2, e_2, \dots, v_k, e_k, v_{k+1}$ takav da za svaki $i = 1, 2, \dots, k$ vrhovi x_i, x_{i+1} međusobno su različiti i predstavljaju krajnje točke grane e_i [25]. Za dva vrha neorijentiranog grafa kažemo da su susjedni vrhovi ako su spojeni granom. Stupanj vrha je broj susjednih vrhova. Stupanj vrha mjerimo na način da se oko vrha nacrtaju kružnica. Tada stupanj vrha predstavlja broj točaka u kojima kružnica sječe bridove koji izlaze iz vrha. Stupanj vrhova bitan je za znanje o načinu na koji se moguće gibati po grafu [22].

Graf može potpun i nepotpun. Neorijentirani graf je povezan ako se svaka dva njegova vrha mogu povezati putem (ili lancem). Neorijentiran graf je regularan (pravilan) stupnja r , ukoliko su stupnjevi svih vrhova jednaki r . Konačno, regularan graf sa n vrhova stupnja $n - 1$ zove se potpun graf. Kod njega su svi čvorovi međusobno povezani [26].



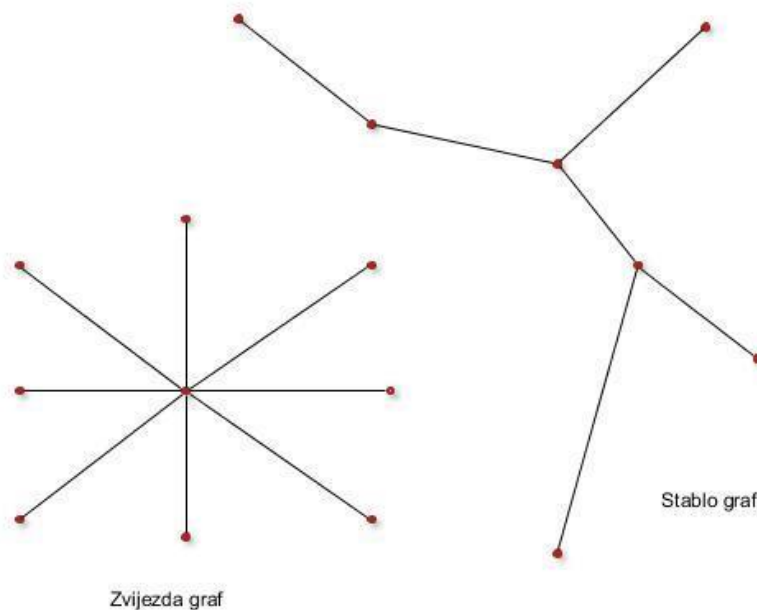
$$G = (V, E);$$

$$V = \{a, b, c, d, e, f\};$$

$$E = \{e_1, e_2, e_3, e_4, e_5\}; e_1 = \{a, b\}, e_2 = \{b, c\}, e_3 = \{c, d\}, e_4 = \{d, a\}, e_5 = \{e, f\}$$

Slika 5. Primjer nepovezanog grafa

Neke posebne vrste povezanih graf koji ne sadrži petlju zovemo stablo (engl. Tree) gdje je broj grana stabla $n-1$. Zvijezda (engl. Star) je stablo sa $n + 1$ vrhova koje ima točno jedan vrh stupnja n i sve ostale stupnja 1. Na Slici 6. skiciran je primjer navedenih grafova.



Slika 6. Primjer stabla i zvijezda grafa

S obzirom na povezanost vrhova povezanog neorijetiranog grafa G , metriku grafa možemo definirati:

- $d(v_i, v_i) = 0, \forall v_i \in V(G)$
- $d(v_i, v_j) = w, \forall v_i, v_j \in V(G) \wedge v_i \neq v_j$

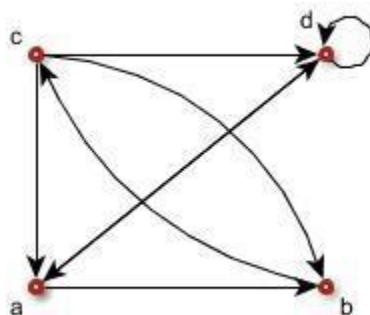
Ovako definirana metrika zadovoljava aksiome metrike, što je trivijalno za dokazati i dokaz se ispušta:

- $d(v_i, v_j) = 0 \Leftrightarrow i = j$ (jednoznačnost)
- $d(v_i, v_j) = d(v_j, v_i)$ (simetričnost)
- $d(v_i, v_k) + d(v_k, v_j) \geq d(v_i, v_j)$ (nejednakost trokuta)

Slijedno asimetričnost definiramo $d(v_i, v_j) \neq d(v_i, v_j)$. Proširivanjem definicije metrike i za nepovezane grafove. Za vrhove v_i, v_j koji nisu međusobno povezani, definiramo $d(v_i, v_j) = +\infty$ [26].

Bridovi grafa mogu biti usmjereni ili neusmjereni. Ako su svi bridovi u grafu usmjereni, tada takav graf zovemo usmjereni graf, u suprotnom, zovemo ga neusmjereni [22]. Na primjer usmjerene bridove bi mogle predstavljati jednosmjerne prometnice, dok bi neusmjereni bridovi mogli biti prometnice u kojima smjer kretanja u oba smjera. Kod neusmjerenih grafova bridovi

su simetrični. Za prikaz grafa na kojima ima simetričnih i asimetričnih bridova koristi se usmjereni graf. Primjer pseudografa sa simetričnim i asimetričnim bridovima vidljiv je na Slici 7.



Slika 7. Usmjeren graf

U praksi se često javlja potreba da svakom bridu e grafa G pridružimo realan broj $w(e)$ koji zovemo težina brida e . Takav graf se naziva težinski graf. Svakom bridu pridružena je njemu svojstvena težina $w(e) \in (v_i, v_j): V \times V \rightarrow \mathbb{R}$ (npr. cijena prijevoza, vrijeme potrebno za prolazak, potrošnja energije, troškovi cestarina, udaljenost između gradova itd.). Ukoliko su težine pozitivni realni brojevi, a graf je bez petlji, možemo definirati udaljenost kao metriku gdje duljina puta je suma svih težina na putu, udaljenost vrhova je duljina minimalnog puta između dva vrha, udaljenost vrha od samog sebe je 0. Ako imamo cikluse ili negativne težine, onda ovako definirana udaljenost nije metrika [26].

Graf je moguće reprezentirati pomoću matrice susjedstva. Svakom grafu s $V(G)$ vrhova možemo pridružiti kvadratnu matricu M dimenzija $V(G) \times V(G)$ u čijem se i -tom retku i j -tom stupcu nalazi 12 ako postoji brid koji spaja i -ti i j -ti vrh, odnosno 0 ako takav brid ne postoji [27]. Pojam matrice susjedstva možemo jednostavno proširiti i na slučaj težinskog grafa. Tada će elementi matrice biti težine odgovarajućih bridova, dakle realni brojevi $w(e)$, a ukoliko ne postoji brid koji spaja neki par vrhova, tada na odgovarajuće mjesto u težinskoj matrici susjedstva stavljamo beskonačno.

4.2 Najkraći put na težinskom grafu

Problem najkraćeg puta u grafu ustvari predstavlja problem traženja puta u težinskom grafu koji spaja dva zadana vrha (v_p, v_z) s najmanjim mogućim zbrojem težina bridova na tom putu, gdje v_p predstavlja početni, a v_z završni vrh. Problem najkraćeg puta može biti definiran za bilo koje usmjerene, neusmjerene ili kombinirane grafove. Najčešće primjene su u određivanju najkraće rute između dva grada (gdje se može tražiti i put sa najmanjom potrošnjom energije ili slično). Optimalni ili najkraći put $s = [v_1, v_2, \dots, v_k]$ predstavlja vektor čiji je zbroj težina najmanji, gdje su $v_1, v_2, \dots, v_k$ vrhovi grafa kroz koje vozilo treba proći [12]. U ovom radu težina će predstavljati utrošenu energiju akumulatora za svladavanje brida tj. cestovnog segmenta.

Problem najkraćeg puta također se naziva problem najkraćeg puta jednog para, kako bi ga razlikovali od ostalih varijacija problema. Problem najkraćeg puta sa jednim izvorom gdje se

pronalaži najkraći put od izvornog vrha v_p do svih ostalih vrhova u grafu. Problem najkraćeg puta sa jednim odredištem, gdje se pronalazi najkraći put od svih vrhova u usmjerenom grafu do jednog odredišnog vrha v_z . Navedeni problem može biti sveden na problem najkraćeg puta sa jednim izvorom tako što se okrenu strelice u usmjerenom grafu. Problem najkraćih puteva svih parova, gdje se pronalazi najkraći put između svih parova vrhova v_p, v_z u grafu.

Ove generalizacije imaju značajno efikasnije algoritme od korištenja klasičnog Dijkstrinog algoritma za problem najkraćeg puta jednog para na svakom paru vrhova u grafu.

Problem najkraćeg puta je specifičan problem u teoriji grafova. Postoje metode za rješavanje tog problema koje se razlikuju s obzirom na tip problema najkraćeg puta. Primjerice Dijkstrin algoritam nije moguće primijeniti u slučajevima, ako su pojedini bridovi negativni. Zato se u nekim slučajevima koriste druge metode kao što su Bellman- Fordov algoritam, A * algoritam, D* algoritam koje će se detaljnije opisati u nastavku.

4.2.1 Dijkstrin algoritam

Dijkstrin algoritam za pronalaženje najkraćega puta između dva vrha u mreži uspješan je, kako u orijentiranim, tako i u neorijentiranim grafovima. Problem pronalaska energetski optimalne rute unutar prometne cestovne mreže može se riješiti primjenom neke od metoda za pronalazak najkraćeg puta na grafu [36]. Da bi se riješio problem pronalaska optimalnog (najkraćeg) puta između dva grada, nizozemski matematičar Edsger Dijkstra, dizajnirao je algoritam koji je nazvan po njemu. Algoritam je dizajniran tako da pronalazi najkraće putove od zadanog vrha do svih ostalih vrhova i za rezultat ispisuje najkraću rutu [28]. Pseudo kod algoritma prikazan je Slikom 8. [26], a modificirao se tako da se težinama na bridove grafa uz duljine cestovnog segmenta pridodaje i utrošak energije.

```

Dijkstra_najkraći_put(Graf g, Vrh  $v_p$ )
početak
    -- početno stanje
    za svaki  $v_i$  iz Vrhovi(g) radi
        početak                                 $d[v_i] := \infty$ 
         $p[v_i] := \emptyset$                         kraj

     $d[v_p] := 0$  -- duljina puta od početnog vrha  $v_p$  do vrha  $v_p$  je nula
     $S := \emptyset$  -- inicijalno nismo odradili niti jedan vrh
     $Q := \text{Vrh}(g)$  -- ostali su svi vrhovi za odraditi

    -- glavna petlja
    sve dok  $Q \neq \emptyset$  radi
        početak
            izaberi  $v$  iz  $Q$  takav da je -- nađimo pivot vrh
             $d[v] = \min \{d[v_i]\}$  za svaki  $v_i$  iz  $Q$                                  $S$ 
             $:= S \cup \{v\}$  -- pivot čvor proglasimo odradjeni
             $Q := Q \setminus \{v\}$  -- pivot čvor izbacimo iz popisa neodrađenih

```

```

    za svaki  $v_i$  iz  $Q$  radi -- svim preostalim čvorovima ažuriraj
    udaljenosti
        ako je  $d[v_i] > d[v] + w(v, v_i)$  onda
    početak
         $p[v_i] := v$  -- promijenio se prethodni čvor
     $d[v_i] := d[v] + w(v, v_i)$  -- promijenila se udaljenost
    kraj
kraj kraj

```

Slika 8. Pseudo kod Dijkstrinog algoritma

Izvor: [26]

Proceduri predajemo graf g i vrh v_p od kojeg želimo naći sve najkraće putove. Kao rezultat algoritma u polju d dobivamo duljine najkraćeg puta do pojedinog, a u polju p prethodnika pojedinog vrha u najkraćim putovima. U polju S čuvamo riješene, a u polju Q neriješene vrhove. U varijabli v imamo trenutni vrh od kojega gledamo udaljenosti (predstavlja pivot vrh).

U inicijalizaciji se postavlja da je udaljenost do svakog vrha beskonačna (tako da prvi put do tog vrha bude kraći od početne vrijednosti i da nema prethodnika), osim do početnog koja je trivijalno nula. Skup riješenih čvorova početno je prazan ($S := \emptyset$), a skup neriješenih svi preostali vrhovi ($Q := Vrhovi(g)$). U glavnoj petlji za sljedeći pivot vrh, izaberemo vrh sa minimalnom evidentiranom udaljenošću ($d[v] = \min \{d[v_i]\}$ za svaki v_i iz Q), dodajemo ga u skup riješenih ($S := S \cup \{v\}$) i izbacujemo iz skupa neriješenih ($Q := Q \setminus \{v\}$). Svakom vrhu v_i iz Q koji je bliži vrhu nego što je za dati vrh evidentirano (ako je $d[v_i] > d[v] + w(v, v_i)$ onda), pridružimo vrh v kao prethodnika ($p[v_i] := v$) i udaljenost od početnog vrha preko vrha v ($d[v_i] := d[v] + w(v, v_i)$).

Ukoliko želimo da Dijkstrin algoritam radi i za nepovezane grafove, onda trebamo dodati kontrolu u traženju sljedećeg pivot vrha, da ukoliko su sve preostale evidentirane udaljenosti beskonačne, da se izađe iz petlje.

Da bismo ispisali pronađeni najkraći put od početnog vrha v_p do završnog vrha v_z , koristiti će se informacije koje smo skupili o prethodnicima svakog vrha na najkraćem putu i sljedećom procedurom na Slici 9. opisan je Ispis Dijkstrinog algoritma.

```

--Počni proceduru
Ispis(Graf g, Vrh  $v_p$ , Vrh  $v_z$ ).
Ako je  $v_p = v_z$ , ispiši  $v_p$  --inače
radi sljedeće:
    Ako je  $p[v_i] = \emptyset$  ispiši "Ne postoji put od "  $v_p$  "do"  $v_z$ 
    inače Ispis(G,  $v_p$ ,  $p[v_i]$ ), ispiši  $v_i$ .

```

Slika 9. Ispis Dijkstrinog algoritma

Izvor: [26]

Procedura rekurzivno prolazi putem od v_z do v_p pomoću sačuvanih prethodnika p i na povratku ispisuje svaki vrh na putu od v_p do v_z .

Složenost Dijkstrinog algoritama može se lako pronaći. Vanjska glavna petlja je reda $O(n)$, jer se svaki vrh točno jednom uzima za pivota. Za svaki pivot vrh gledamo susjedne vrhove i određujemo nove udaljenosti (kroz cijelu glavnu petlju radimo sve skupa m puta), tako da je trenutna složenost $O(n + m)$. Problem je u određivanju sljedećeg pivot vrha i složenost ovisi o strukturi podataka koju koristimo za polje udaljenosti. Ako je polje jednostavna linearna struktura, onda je traženje pivot vrha složenosti $O(n)$ i ukupna složenost Dijkstrinog algoritma je $O(n^2)$ [26].

Ovim postupkom Dijkstrin će algoritam dati za rezultat energetski optimalnu rutu. Dodatno, pomoću Dijkstrinog algoritma na grafu kojem težine predstavljaju udaljenosti između vrhova izračunat će se najkraća ruta u prijedenoj udaljenosti. Ako pak težinama pridodamo vremensku komponentu izračunati će se vremenski najmanje zahtjevna ruta i slično.

4.2.2 Bellman-Fordov algoritam

Primjenjuje se u slučaju traženja najkraćeg puta sa jednim izvorom ako težine bridova mogu biti negativne. Ovaj algoritam je interesantan pod uvjetom ako vozilo ima mogućnost regenerativnog kočenja. Pod regenerativnim kočenjem smatra se svako kočenje kod kojeg se dio kinetičke energije vozila pretvara u neki oblik energije, umjesto da se topline rasprši u okolinu. Dio pohranjene energije može se iskoristiti kako bi se efikasnije iskoristila energija i doseg vozila koju ima na raspolaganju [29]. Postoji regenerativno kočenje koje se temelji na principu zamašnjaka ili koje se temelji na principu generatora i druge tehnologije. Primjena ove tehnologije sve je obuhvatnija u vozilima koje koriste električni pogon. Pseudo kod algoritma prikazan je na Slici 10.

```

Bellman_Ford_najkraći_put(Graf g, Vrh vp)
    početak          -- početno
    stanje           za svaki vi iz
    Vrhovi(g) radi
        početak      d[vi] := ∞      p[vi] := ∅      kraj
    d[vp] := 0 -- duljina puta od vrha vp do vrha vp je nula
    Q := Vrhovi(g) -- ostali su svi vrhovi za odraditi
    -- glavna petlja

    za i od 1 do Broj_čvorova(g) radi
    početak

        za svaki ei(u,v) iz Bridovi(g) radi
        ako je d[v] > d[u]+w(u,v) onda          -- duljina dosadašnjeg vrha se
        uspoređuje novom početak
            p[v] := u -- promijenio se prethodni čvor
        d[v] := d[u]+w(u,v) -- promijenila se udaljenost
    kraj      kraj

```



```

    za svaki  $e_i(u,r)$  iz Bridovi( $g$ ) radi          ako je
     $d[r] > d[u] + w(u,r)$  onda          vrati krivo --
    postoji ciklus sa negativnom težinom    kraj

```

Slika 10. Pseudokod Bellman-Fordovog algoritma

Izvor: [26]

Složenost Bellman-Fordovog algoritma je $O(m \cdot n)$ (vanjska petlja se vrti n puta, a unutarnja m puta), dok za najgori slučaj potpunog grafa iznosi $O(n^3)$ [26].

4.2.3 A * algoritam

Ovaj heuristički algoritam koristi se u slučajevima gdje imamo jako velik broj vrhova, ali u grafu postoji struktura koja nam daje sugestije u kojem je smjeru najvjerojatniji najkraći put (tipično u planarnim⁴ grafovima). A * (čitamo : A zvjezdica) algoritam je tip informiranog pretraživanja. Snaga algoritma leži u korištenju znanja specifičnog za problem, koje nije sastavni dio definicije problema. To je iskustveno znanje koje nudi heurističko rješenje. Ukoliko A* algoritam nikad ne pretpostavlja veću od stvarne udaljenosti od ciljnog vrha, onda on svojim heurističkim rješenjem daje ujedno i optimalni put. Dijelovi heuristike su procjena troška od trenutnog vrha do ciljanog vrha h , trošak dolaska od početnog do trenutnog vrha g te procjena troška od početnog do ciljnog vrha jednaka je zbroju $f = g + h$, uz uvjet $g_{i+1} = f(g_i)$. Pseudokod je prikazan na Slici 11.

```

A_zvijezda_najkraći_put(Graf  $g$ , Vrh  $v_p$ , Vrh  $v_x$ )

početak          -
- početno stanje

 $S := \emptyset$  -- inicijalno nismo odradili niti jedan vrh
 $Q := \{v_p\}$  -- krećemo od prvog vrha
 $f[v_p] := 0$ 

-- glavna petlja

sve dok  $Q \neq \emptyset$  radi

    početak
    izaberi  $u$  iz  $Q$  takav da je -- nađimo sljedeći pivot
vrh     $f[u] = \min \{f[v_i]\}$  za svaki  $v_i$  iz  $Q$ 
     $Q := Q \setminus \{u\}$  -- pivot vrh izbacimo iz popisa neodrađenih
     $S := S \cup \{u\}$  -- ubacimo pivot vrh u popis odrađenih

```

⁴ Graf je planaran ukoliko je smjestiv u ravnini, ako se može nacrtati u ravnini tako da mu se bridovi sijeku samo u vrhovima

```

    ako je  $u = v_x$  onda
završi      za svaki  $v_i$ 
povezan sa u radi
početak

     $h\_t[v_i] := \text{Heuristički}(v_i)$  -- procjena udaljenosti do ciljnog
vrha       $g\_t[v_i] := g[u] + w(u, v_i)$        $f\_t[v_i] :=$ 
 $g\_t[v_i] + g\_t[v_i]$ 

    ako je  $v_i$  iz Q i  $f\_t[v_i] < f[v_i]$ 
onda      ignoriraj

    ako je  $v_i$  iz S i  $f\_t[v_i] < f[v_i]$ 
onda      ignoriraj
inače
     $Q := Q \setminus \{v_i\}$ 
     $S := S \setminus \{v_i\}$ 
     $h[v_i] := h\_t[v_i]$  -- procjena udaljenosti do ciljnog
vrha       $g[v_i] := g\_t[v_i]$        $f[v_i] :=$ 
 $f\_t[v_i]$        $Q := Q \cup \{v_i\}$        $p[v_i] := u$ 
kraj kraj

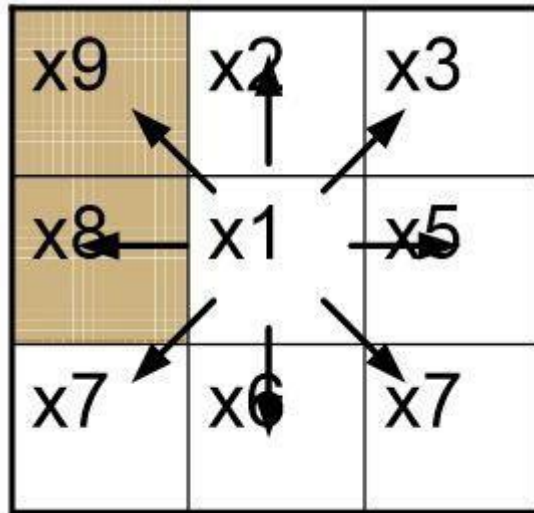
```

Slika 11. Pseudo kod algoritma A*

Izvor: [26]

4.2.4 D* algoritam

Algoritam (čitamo: D zvjezdica) je razvio Anthony Stentz 1994. Algoritam je razvijen iz A* algoritma koji nije uspješan ukoliko mu nešto nije poznato. D* je nadogradnja A* algoritma u slučajevima ako se izmjene određeni parametri u mreži, no D* algoritam ne spada u heurističku metodu. Služi za dinamičko planiranje ruta jer ima mogućnost ponovnog planiranja (stigavši do prepreke ponovno preračunao rutu u odnosu na svoju poziciju). Za inicijalni plan koristi Dijkstrin algoritam i omogućuje napredno korištenje input podataka za brže ponovno planiranje ruta. Optimalan je i kompletan u kompleksnijim situacijama mnogo efektivniji od A* algoritma. Lokalne promjene u okruženju ne utječu na izbor puta. Većina troškova do cilja ostaju iste. Koristi se u mobilnoj robotici i kod autonomnih vozila. Bazira se na koordinaciju u Kartezijevom koordinatnom sustavu. Primjer kretanje robota prikazan je na Slici 12.



Slika 12. Kretnje robota u 8 smjerova [30]

Trošak kretnje u prazno polje je malen u odnosu na kretnju robota na polje u kojem se nalazi prepreka. Primjer je prikazan u Tablici 2:

Tablica 2. Primjer težina u poljima
Izvor: [30]

Horizontalno ili Vertikalno kretanje	Dijagonalno kretanje
$C(x1,x5)=1,4$	$C((x1,x3)=1$
$C(x1,x8)=1000$	$C(x1,x9)=1000$

Glavna procedura algoritma dana je na slikama Slika 13., Slika 14., Slika 15., Slika 16, Slika 17, Slika 18 dane su procedure D zvjezdica algoritma.

```
--X, Y→ pozicije robota u koordinatnom sustavu
--b(X) = Y→ backpointer pozicije X iduću poziciju Y
--c(X,Y)→ težina puta od Y do X
--r(X,Y)→ težina puta od Y do X dobivena sa senzora
--t(X)→oznaka (npr. NEW,OPEN, i CLOSED) u odnosu na poziciju X
--h(X)→path cost težina puta
--k(X)→ najmanja vrijednost vrijednosti h(x) od kada je X stavljen u OPEN
List
INPUT: Lista svih pozicija L
OUTPUT: Ako je moguće doći do cilja i Lista L je update tako da b(x) lista
pokazuje put od početka do cilja onda idi. Ako cilj nije dohvatljiv vrati
NULL

PROCESS_STATE( 0,L)
  Za svaki  $X \in L$  radi
    T(X)= NEW
  završi
h(G)=0;
```

```

INSERT( O, G ,h(G))
 $X_c = S$  --početna koordinata
 $P = \text{INIT\_PLAN}(O, L, X_c, G)$  --definiranje inicijalnog
puta Ako je  $P = \text{null}$  onda vrati null završi
sve dok  $X_c \neq G$  --petlja sve dok je koordinata različita od cilja
PREPARE_REPAIR( $O, L, X_c$ )
 $P = \text{REPAIR\_REPLAN}((O, L, X_c, G)$ 
Ako je  $P = \text{null}$  vrati null
završi
 $X_c =$  drugi element od  $P$  {pomakni se do drugog elementa u  $P$ 
Završi
Vrati  $X_c$ 

```

Slika 13. Glavna procedura D* algoritma

Izvor: [30]

```

INSERT( $O, X, h_{new}$ )
Ako je  $t(x) = \text{NEW}$  onda
 $k(X) = h_{new}$ 
Inače ako je  $t(x) = \text{OPEN}$  onda
 $k(X) = \min(k(X), h_{new})$ 
Inače ako je  $t(X) = \text{CLOSED}$  onda
 $(X) = \min(h(X), h_{new})$ 
Zatvori
 $h(X) = h_{new}$ 
 $t(x) = \text{OPEN}$ 
Sortiraj  $O$  na osnovu povećane  $k$  vrijednosti

```

Slika 14. Prikaz procedure za umetanje elementa u listu INSERT

Izvor: [30]

```

INIT_PLAN( $\cdot O, L, X_c, G$ )
┌
└ Ponavljaј
   $k_{min} = \text{PROCESS\_STATE}(\cdot O, L)$ 
  Sve dok  $k_{min} = -1$  ili  $t(X_c) = \text{CLOSED}$ 
   $P = \text{GET\_BACKPOINTER\_LIST}(L, X_c, G)$ 
  Vрати  $P$ 

```

Slika 15. Procedura inicijalnog plana INIT_PLAN

Izvor: [30]

```

PREPARE_REPAIR( $O, L, X_c$ )
Za svaku poziciju  $X \in L$  unutar senzorskog dosega  $X_c$  i  $X_c$  radi
┌ Za svaki susjedni  $Y$  od  $X$  radi
┌ Ako je  $r(Y, X) \neq c(X, Y)$  onda
┌ ┌ ┌ MODIFY_COST( $O, Y, X, r(Y, X)$ )
┌ ┌ └ završi
┌ └ završi
┌ Za svaki susjedni  $Y$  od  $X$  radi
┌ Ako je  $r(X, Y) \neq c(X, Y)$ 
┌ ┌ ┌ MODIFY_COST( $O, X, Y, r(X, Y)$ )
┌ ┌ └ završi
┌ └ završi
└ završi

```

Slika 16. Procedura za kontrolu sa senzora odabranog puta PREPARE_REPAIR Izvor:

[30]

```

MODIFY_COST( $O, X, Y, c_{težina}$ )
┌  $c(X, Y) = c_{težina}$ 
└ Ako je  $t(x) = \text{CLOSED}$  onda
┌ INSERT( $O, X, h(x)$ )
└ završi
Vрати uzetu  $k_{min}(O)$ 

```

Slika 17. Procedura za modifikaciju težina MODIFY_COST

Izvor: [30]

```

REPAIR_REPLAN( $O, L, X_c, G$ )
┌ ponavljaј
┌  $k_{min} = \text{PROCESS\_STATE}(O, L)$ 
┌ Sve dok  $(k_{min} \geq h(X_c))$  ili  $k_{min} = -1$ 
┌  $P = \text{GET\_BACKPOINTER\_LIST}(L, X_c, G)$ 
└ Vрати  $P$ 

```

Slika 18. Procedura za izmjenu plana REPAIR_REPLAN Izvor: [30]

5 Potrošnja energije električnog vozila na terenu s nagibom

Poznavanjem ponašanja potrošnje energije akumulatora električnog vozila može se predvidjeti kojim će intenzitetom vozilo trošiti energiju koju ima na raspolaganju. Vozilo treba svladati nekoliko otpora, no u ovome radu, proučavamo otpor koji vozilo svladava utjecajem terena pod nagibom. Potrošnja energije vozeći se terenom bez nagiba i s nagibom nije jednaka. Pretpostavka u radu jest da je potrebno značajno više energije za svladavanje terena s nagibom.

Energija je sposobnost tijela ili sustava da obavlja rad; ako tijelo obavlja rad energija se smanjuje. U suprotnom, ako okolina obavlja rad na tijelu energija se povećava. Energija i rad imaju iste mjerne jedinice, no postoji potreba za dvjema fizičkim veličinama jer se energija definira kao stanje sustava dok se rad definira kao promjena stanja sustava [31]. Rad se definira sljedećim izrazom gdje je:

$$W = \int_{s_1}^{s_2} F \cdot ds, \quad (1)$$

$$t_1, s_1 \rightarrow E_1 \Rightarrow W_{1-2} \Rightarrow E_2 \rightarrow t_2, s_2 [J]$$

$W [J]$ – rad

$F [N]$ – sila

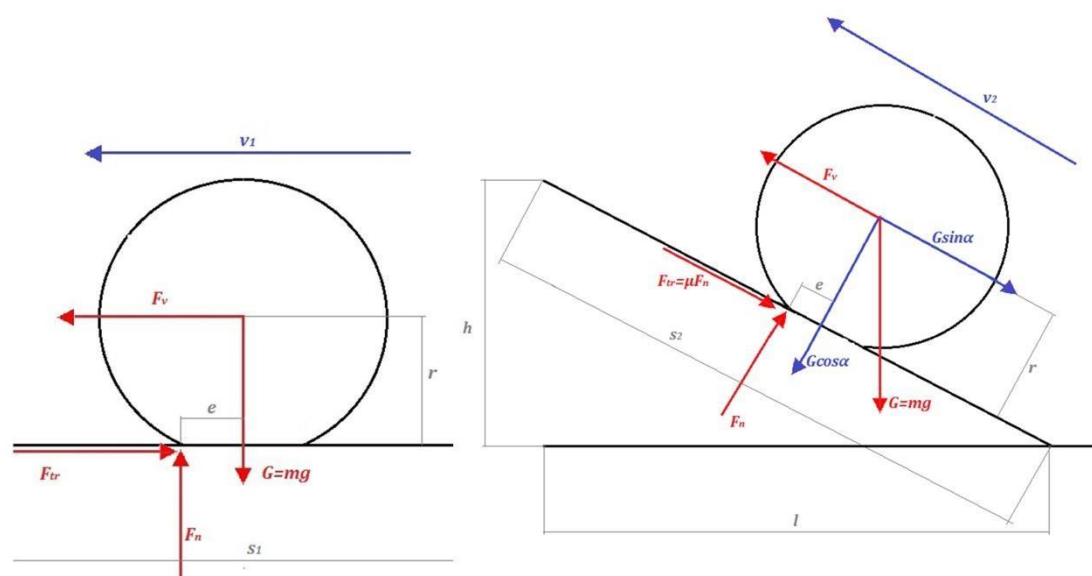
$s [m]$ – put

5.1 Sile na kosini

Kako bi se mogla mjeriti potrošnja energije vozila potrebno je poznavati djelovanje sila na vozilo. Poznavanje sila kada se vozilo kreće uz nagib terena i kada se kreće po terenu bez nagiba, nužno je zbog saznanja gdje se uložena energija gubi [32]. Slika 19. prikazuje sile koje se javljaju na terenu s nagibom i na terenu bez nagiba.

Slučaj 1: Kretanje vozila po terenu
bez nagiba

Slučaj 2: Kretanje vozila po terenu
sa nagibom



Slika 19. Sile na terenu s i bez nagiba

Da bi se vozilo kretalo mora postojati neka pogonska sila F_v kojoj se opiru sile trenja F_{tr} , sila otpora zraka te komponenta gravitacijske sile $G \sin \alpha$ koja djeluje na kosini. Iz momentnog pravila, ako se postavi oko središta kotača dobiju se izrazi za silu trenja kotrljanja na terenu s nagibom [32]:

$$F_{tr_{kot1}} = \frac{e}{r} \cdot m \cdot g \quad (2)$$

i silu trenja kotrljanja na kosini:

$$F_{tr_{kot2}} = \frac{e}{r} \cdot m \cdot g \cdot \cos \alpha \quad (3)$$

gdje je:

F_{tr1}, F_{tr2} [N] - sila trenja kotrljanja
 e [mm] - koeficijent trenja kotrljanja
 r [mm] - polumjer kotača
 m [kg] - masa vozila
 α [°] - kut nagiba terena

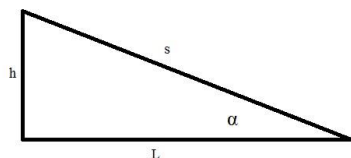
$g [\frac{m}{s^2}]$ - ubrzanje sile teže

Postotak nagiba terena je omjer duljine L pravokutnog trokuta i visine h [33] kao što je prikazano na Slici 20. te odgovara sljedećem izrazu:

$$tg\alpha = p \quad (4)$$

gdje je:

$p[\%]$ - postotak nagiba



Slika 20. Trokut potreban za izvod postotka nagiba terena

5.2 Energija koju vozilo troši s obzirom na konfiguraciju terena

Rad i energija koji vozilo ulaže za kretanje po terenu s nagibom i bez nagiba izvodi se iz zakona kinetičke energije [12]:

$$E_{k2} - E_{k1} = W_{r1} - W_{trkot1} - W_{otp1} \quad (5)$$

rad koji robot ulaže za kretanje po terenu s nagibom:

$$E_{k2} - E_{k1} = W_{r2} - W_{trkot2} - W_{otp2} - W_p \quad (6)$$

gdje je:

W_{r1}, W_{r2} [J] - rad koji ulaže robot

W_p [J] - rad potencijalne energije (rad koji se ulaže za svladavanje visine)

W_{otp1}, W_{otp2} [J] - rad koji se ulaže zbog djelovanja otpora zraka

W_{trkot1}, W_{trkot2} [J] - rad koji se ulaže zbog djelovanja trenja kotrljanja

E_{k1} [J] - kinetička energija s početnom brzinom

E_{k2} [J] - kinetička energija s krajnjom brzinom

5.3 Definiranje parametara za težinu cestovnih segmenata

Težina brida na grafu definirana je kao utrošena energija za savladavanje određenog puta. Snaga koja je potrebna mobilnom robotu za kretanje između dva vrha na grafu računa se prema izrazu (17). Mobilni robot kretao se konstantnom brzinom po ruti i prešao zadani put u vremenu t . Energija se definira izrazom:

$$E = P \cdot t \quad (7)$$

Sređivanjem izraza (19) uzimajući u obzir da je brzina konstantna te da je udaljenost jedina veličina koja se učitava, dobije se sljedeća funkcija:

$$E(s) = \frac{P \cdot s}{v} \quad (8)$$

gdje je:

$E(s)$ [J] - funkcija težine koja se pridodaje grafu (energija)

s [m] - prijeđeni put (duljina rute ili brida)

v [m/s] - brzina mobilnog robota

Udaljenosti se mogu učitati pomoću pomagala koja služe za navigaciju i pozicioniranje [34] npr. globalni pozicijski sustavi [35]. Poznate veličine U, I, v, t dobivene su mjerenjem na testnom poligonu što je prikazano u prethodnim poglavljima. Energetski optimalna brzina je ona brzina

na kojoj mobilni robot troši najmanje energije. Brzina iznosi 0,7 m/s, a ujedno radi o maksimalnoj brzini kojom su se izvodila mjerenja. Ukoliko se u izraz (19) uvrste vrijednosti iz

Tablice 3 i promatra isti prijedeni put (u ovom slučaju računato je sa 100 m) za svaku brzinu, utoliko se dobije optimalna brzina mobilnog robota. Tablica 5 dokazuje tu tvrdnju.

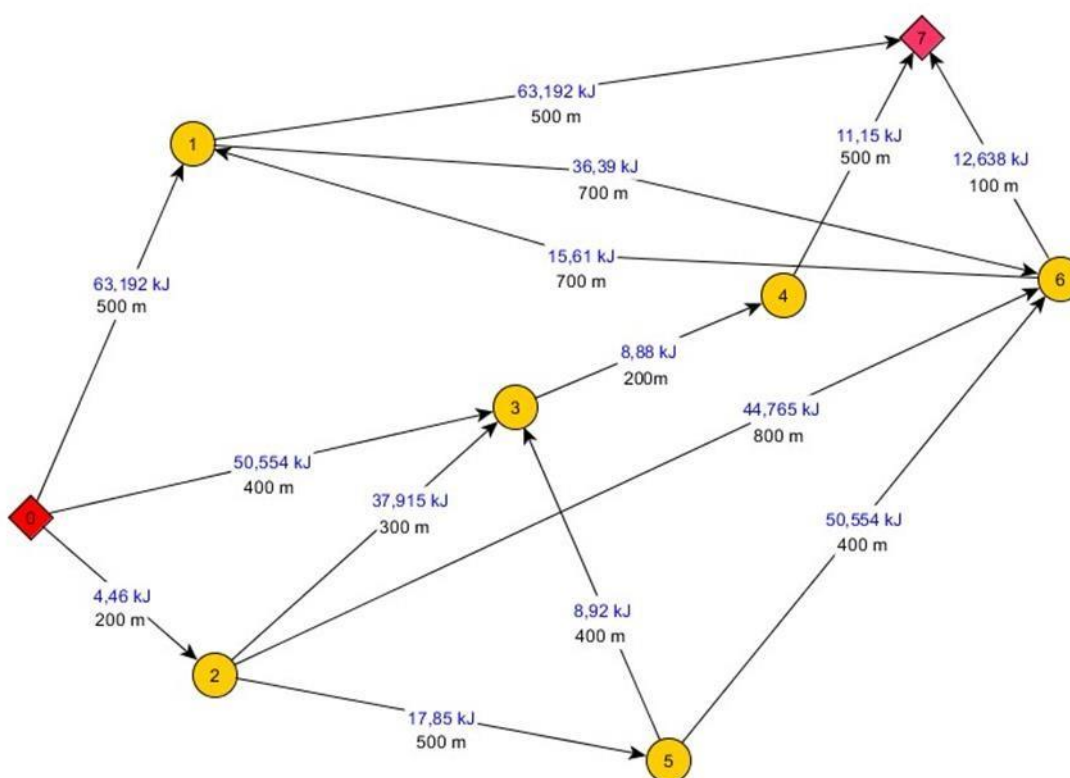
Tablica 3. Potrošnja energije s obzirom na brzinu kretanja mobilnog robota na putu od 100 m

Nagib terena:	Brzina [m/s]:	Snaga [W]:	Energija [kJ]
10 %	0,1	$17,29 \pm 0,3\%$	17,290
	0,4	$31,86 \pm 0,6\%$	7,565
	0,7	$46,65 \pm 0,3\%$	6,664
20 %	0,1	$35,41 \pm 1,2\%$	35,41
	0,4	$60,94 \pm 0,9\%$	15,235
	0,7	$88,47 \pm 1,3\%$	12,638

6 Primjena Dijkstrinog algoritam za izračun energetske optimalne rute

Dijkstrin algoritam implementiran je na zamišljenom primjeru nasumično generirane prometne cestovne mreže koji je ilustriran Slikom 21. Na slici prikazana je cestovna mreža s prometnicama koje su određene svojom dužinom, potrošnjom energije i nagibom terena.

Parametri dužine prometnog cestovnog segmenta, postotak nagiba terena, snaga potrebna za savladavanje takvog puta, postotak napunjenosti akumulatora potreban za savladavanje pojedine rute te dobivene težine prikazani su Tablicom 4.



Slika 21. Prometna mreža prikazana kao nepotpuni težinski usmjereni graf

Tablica 4. Parametri prometne mreže iz primjera prikazanog Slikom 21.

Bridovi:	Duljina prometnog segmenta:	Postotak nagiba [%]:	Snaga [W]:	Potreban SOC za savladavanje puta [%]:	Težina E(s)[kJ]:
0-1	500	20	88,47	5,29	63,19
0-2	200	0	15,61	0,36	4,46
0-3	400	20	88,47	4,23	50,54
1-6	700	7	36,39	2,91	36,39
1-7	500	20	88,47	5,29	63,19
2-3	300	20	88,47	3,17	37,92,
2-5	500	3	24,99	1,43	17,85
2-6	800	8	39,17	3,63	44,77
3-4	200	6	31,08	0,73	8,88
4-7	500	0	15,61	0,91	11,15
5-3	400	0	15,61	0,72	8,92
5-6	400	20	88,47	4,23	50,55,
6-1	700	0	15,61	1,28	15,61
6-7	100	20	88,47	1,06	12,64

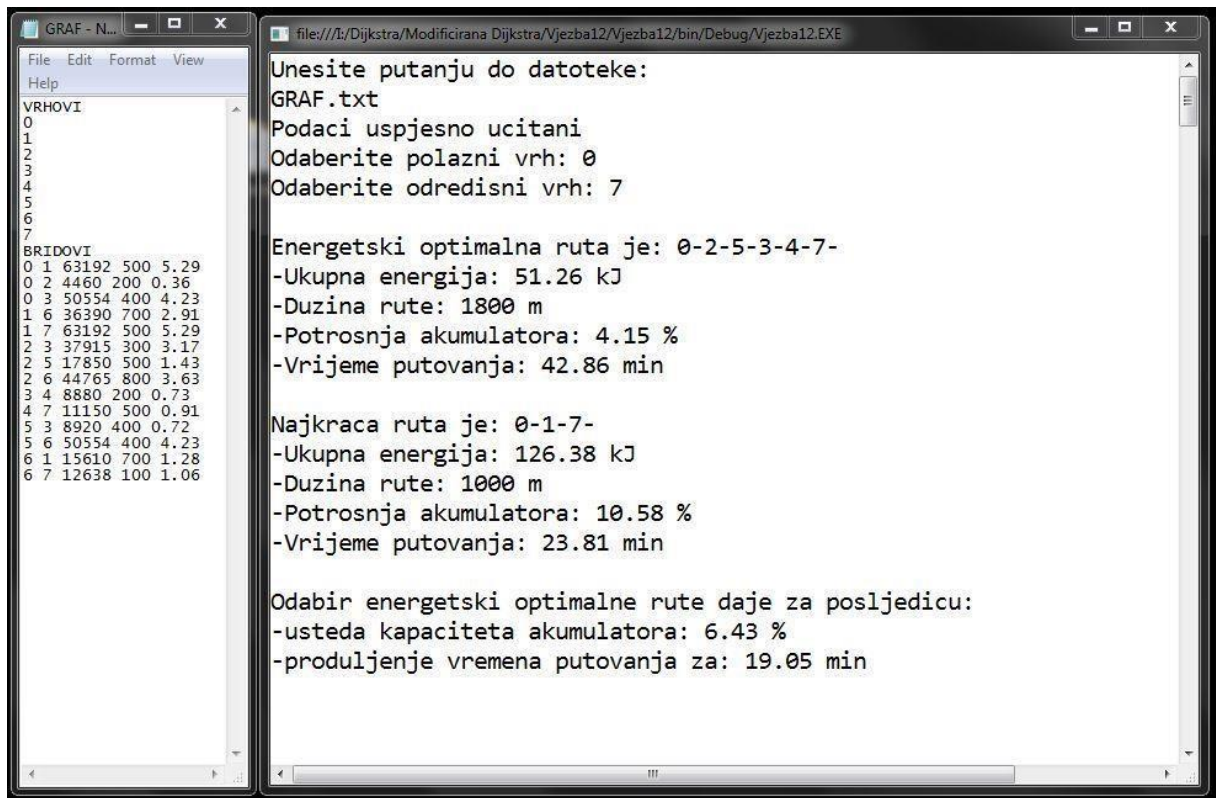
Dijkstrin algoritam korišten za implementaciju napisan je u programskom jeziku C# od strane asistenata Prometnog fakulteta u Zagrebu [36]. Originalni algoritam računao je samo najkraću rutu na cestovnoj mreži koja se definira upisom podataka u tekstualnu datoteku. Algoritam je izveden sa tri klase. Brid, Vrh i Dijkstra. Klasa vrh definira broj vrha, prethodni vrh te trenutnu težinu kada algoritam dođe do toga vrha. Klasa Brid definira brid sa dvije točke, početnim i završnim vrhom te težine svakog brida. U klasi Dijkstra definirano je funkcioniranje istoimenog algoritma te ispis rezultata na konzolu. Takav algoritam modificirao se na način da se uz najkraću rutu računa, energetska optimalna ruta i postotak akumulatora koji se troši zbog

vožnje po određenoj ruti. Modificiranje se izvršilo metodom pod nazivom Ukupno, prikazanom Slikom 22., koja nakon što je Dijkstrinim algoritmom izračunata energetski optimalna ruta, računa duljinu i SOC potrebne za savladavanje te rute. Taj postupak se ponavlja i za računanje najkraće rute, jedina je razlika što se Dijkstrinim algoritmom računa najkraća ruta, a izrađenom metodom računa se ukupna energija i SOC potrebni za savladavanje najkraće rute.

```
{
    foreach (Brid brid in Bridovi)
    {
        if (izbor == "energija")
            vrstaRacuna = brid.Energija;
        else if (izbor == "soc")
            vrstaRacuna = brid.Soc;
        else
            vrstaRacuna = brid.Duljina;
        if (brid.Od == Lista[i] && brid.Do == Lista[i + 1])
        {
            tezina = tezina + vrstaRacuna;
        }
    }
}
return tezina;
}
```

Slika 22. Metoda Ukupno

Metoda prima dva parametra, listu tipa *Vrh* i niz znakova koji određuje funkciju metode. U listu su nakon izvršavanja Dijkstrinog algoritma, upisani brojevi vrhova koji predstavljaju rutu (najkraća ili energetski optimalna). Nakon učitane liste, u ovisnosti o parametru *izbor* metoda će izvršiti jednu od definiranih funkcija, izračun potrošnje energije najkraće rute, duljinu energetski optimalne rute ili SOC potreban za savladavanje te rute. Nakon izvršavanja metode, najkraća i energetski optimalna ruta uspoređuju se po potrošnji kapaciteta akumulatora koji je potreban za savladavanje pojedine rute te vremenu putovanja. Rad ovako modificiranog algoritma prikazan je Slikom 23.



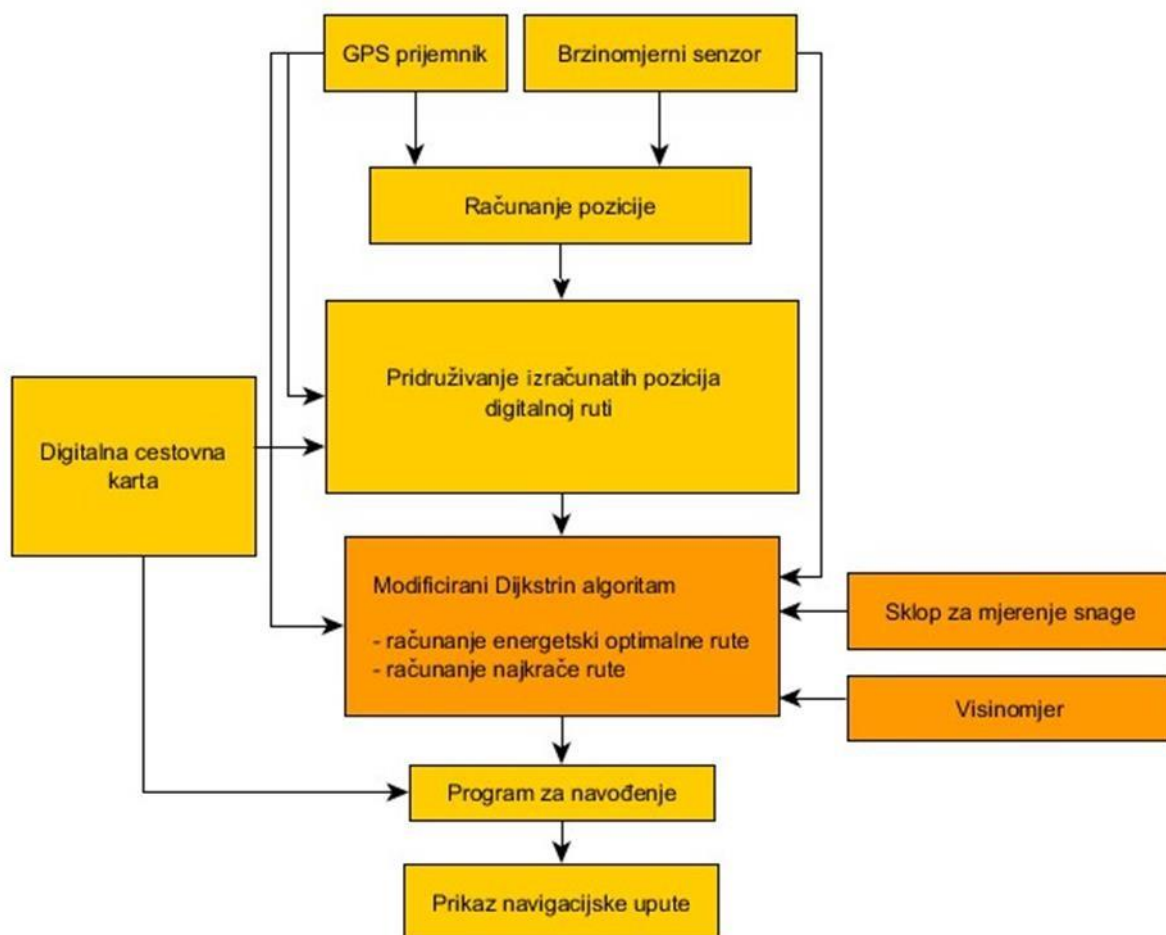
Slika 23. Prikaz rada modificiranog Dijkstrinog algoritma

Za ovaj primjer energetski optimalna ruta je $s_{0,2,5,3,4,7}$ ukupna udaljenost rute je 1800 m. Najkraća ruta je $s_{0,1,7}$ čija je udaljenost 1000 m. Energetski optimalnom rutom potrebno je približno 43 min, dok je za najkraći put potrebno 24 min. Ovaj primjer zorno prikazuje da je vozilu energetski optimalnije voziti rutom koja ima manji nagib terena, iako je u ovom slučaju duža. Najkraći put ne mora biti i energetski optimalan, iako i to može biti slučaj. Najkraću rutu može se izabirati ako se radi o kraćim relacijama koje neće primorati vozilo na punjenje. Radi li se o duljim relacijama pogodnije je birati energetski optimalne rute i izbjeći punjenje ili savladati rutu uz što manje punjenja. Punjenje elektrokemijskih akumulatora vremenski je zahtjevno te vrijeme koje se „zaradi“ vožnjom najkraćom rutom višestruko se izgubi potrebnim vremenom punjenja [15]. Uzmemo li u obzir vrijeme punjenja mobilnog robota do 6 h očita je navedena tvrdnja. Osvrnemo li se na doseg današnje tehnologije u tom aspektu. Primjerice super punjači tvrtke Tesla snage 20 kW pune akumulator 40 min do 80 % kapaciteta. Dok je 75 min potrebno je za punjenje do maksimalnog kapaciteta akumulatora [37].

Implementacija modificiranog Dijkstrinog algoritma omogućuje da mobilni robot u stvarnom vremenu optimalno iskoristi kapacitet akumulatora koji mu je na raspolaganju. S obzirom na problem ne dovoljnog kapaciteta akumulatora kod električnih vozila, razvoj ovakvih i sličnih algoritama [38] doprinosi optimalnoj potrošnji kapaciteta akumulatora električnih vozila.

Ovakav algoritam moguće je primijeniti na navigacijskom sustavu [32] koji bi vozaču mogao predložiti odabir više ruta, odnosno predložiti energetski isplativiji pravac kretanja,

ako vozilo nema dovoljan kapacitet akumulatora za završetak rute. Skica takvog sustava prikazana je Slikom 24.



Slika 24. Model navigacijskog sustava uz dodatak modificiranog Dijkstrinog algoritma

Kod potpuno autonomnih vozila algoritmi ovog tipa pronalaze značajniju primjenu jer vozač ne odlučuje o ruti kojom će se vozilo kretati.

7 Zaključak

Smanjenje potrošnje energije i sprječavanje energetske gubitaka imaju sve veću važnost za Europsku Uniju (EU). Čelnici EU-a postavili su 2007. cilj smanjenja godišnje potrošnje energije u Uniji za 20 % do 2020. Provode se mjere za energetske učinkovitost sa ciljem postizanja održive opskrbe energijom, smanjenje emisije stakleničkih plinova, poboljšanje sigurnosti opskrbe i smanjenje troškova uvoza, ali i za promicanje konkurentnosti europskih gospodarstava.

Rutni vodiči i navigacija usko su povezani s pojmovima putno i pred putno informiranje. Glavne značajke za navedene sustave je da povećavaju informiranost korisnika usluge.

Posredno se povećava sigurnost, primjerice informiranost o radovima na cesti ili incidentnoj situaciji mogu smanjiti prometna zagušenja te smanjiti vjerojatnost sekundarnih incidenata. Preusmjeravanjem vozila na prohodnije skraćuje vrijeme putovanja ali i štedi energija odabere li se vodič za energetske optimalnu rutu koji je demonstriran u ovom radu. Naglasak je na kooperativnim multimodalnim rutni vodičima kako bi se integriralo više sustava, a time i efektivnije iskoristio postojeći prometni sustav.

Mobilni robot P3-AT posjeduje senzore za snalaženje u prostoru i pozicioniranje. Pogodan je mikro prikaz potpunog autonomnog električnog vozila. Problematika koja se javlja pri upotrebi mobilnog robota također je uvelike prisutna i kod svih potpuno autonomnih vozila.

Spremnici energije predstavljaju problem u razvoju tehnologije električnih vozila. Što je veća potreba za energijom time se povećava masa spremnika energije. Ako je cilj povećati doseg električnog automobila, mogu mu se dodati spremnici energije, no to utječe na masu vozila i time se ne čini značajni pomak. Predloženo u ovome radu jest korištenje određenih algoritama kako bi se izabrala energetske optimalna ruta, s ciljem smanjenja potrošnje energije.

Opisani su algoritmi za pronalazak najkraće rute na težinskom grafu koji se mogu iskoristiti za pronalazak energetske optimalne rute. Također D zvjezdica algoritmom nastojalo pobliže uči u problematiku autonomnih vozila. Kojima je uz problem odabira energetske optimalne rute, problem i samo snalaženja u prostoru, pogotovo izmjeni li se nešto na isplaniranoj putanji.

U radu je razrađena ideja o optimalnom iskorištavanju dostupne energije, odnosno autonomno vozilo s obzirom na željene destinacije može odredi njemu energetske optimalni put.

U sklopu istraživanja prikupljeni su podaci o potrošnji mobilnog električnog robota Pioneer 3-AT koji su se koristili za implementaciju algoritma za odabir optimalne rute, a koji se mogu koristiti i u druge istraživačke svrhe. Korištenjem izmjerenih podataka o potrošnji energije pokazao se način modifikacije Dijkstrinog algoritma kako bi računao energetske optimalnu rutu te mogućnosti implementacije istog u navigacijski sustav vozila. Izmjereni podaci o potrošnji energije malog autonomnog električnog vozila korišteni su u izračunu težina bridova na grafu korištenih kao ulazni podatak u Dijkstrin algoritam.

Daljnja istraživanja usmjerena su na implementaciju algoritma za snalaženje potpuno autonomnog vozila u prostoru. Pritom će se koristiti prethodno navedeni Dijkstrin algoritam kako bi se smanjila potrošnja energije kao inicijalno rješenje i jedan od algoritama za

izbjegavanje prepreka primjerice D zvjezdica algoritam. Mobilni robot će se opremiti računalom, lidarom i kamerom, iskoristiti fuziju podataka sa senzora za bolje snalaženje u prostoru.

Osim toga vodilo bi se računa o obnovljivim izvorima energije te bi se tako na nosač od P3AT mogle ugraditi solarne ploče koje bi obnavljale energiju akumulatora. Razvijenim algoritmom bi se gubitak energije nastojao bi se minimizirati, te bi se primjenom solarnih ploča povećala autonomija vozila. Provelo bi se istraživanje vezanom za potpuno autonomnu vožnju i minimiziranje grešaka dodavanjem podataka senzora i implementacijom metoda za ispravljanje grešaka. Moguća su istraživanja usmjerena k povezivanju autonomnih letjelica s malim električnim vozilom, s ciljem skeniranja prostora pomoću letjelice te navođenje autonomnog vozila.

Literatura

- [1] Golubić, J.: Promet i okoliš, Fakultet prometnih znanosti Sveučilišta u Zagrebu, Zagreb, 1999.
- [2] V. Cvitanović, T. Ćurković, M. Robić, S. Tirić i L. Vidić, O električnim automobilima i preradi klasičnog automobila u električni, Zagreb: Škola za cestovni promet , 2014.
- [3] H. Glavaš, M. Antunović i T. Keser, » Cestovna vozila na električni pogon,« u *Dvadesetšesti skup o prometnim sustavima s međunarodnim sudjelovanjem Automatizacija u prometu* , Zagreb, 2006.
- [4] I. Bošnjak, Inteligentni transportni sustavi, Zagreb: Fakultet prometnih znanosti Sveučilište u Zagrebu, 2006.
- [5] A. Mežnarek, Arhitektura naprednih sustava informiranja putnika i vozača, Završni rad, Zagreb: Fakultet prometnih znanosti Sveučilišta u zagrebu, 2015.
- [6] »Algoé - Rapp Trans Grouping for DG Mobility and Transport: ITS ACTION PLAN, Framework Service Contract TREN/G4/FV-2008/475/01, Study “Towards a European Multi-Modal Journey Planner”, Lyon, 2011«.
- [7] Škorput, P., Manduka, S, Vujić, M.: Razvoj kooperativnih multimodalnih putnih vodiča , TELFOR, 2006..
- [8] » Carlos, F., Vicente, M., Joshue, P., David, G., Fawzi, N.: Optimal energy consumption algorithm based on speed reference generation for urban electric vehicles, IEEE Intelligent Vehicles Symposium, Seoul, 2016.«.
- [9] Opačić, A.: Autonomna vozila - automobili budućnosti, Ekscentar, svez. 10, pp. 114-115, 2007.
- [10] http://www.mup.hr/UserDocsImages/PU_PG/AKCIJE/aim.pdf, pristupljeno: 20.2.2016..
- [11] N.H.T.S. Administration: Preliminary Statement of Policy Concerning Automated Vehicles, Wasington, 2013..
- [12] T. Jurik, A. Cela, R. Hamouche, R. Natowicz, A. Reama, S.-I. Niculescu i J. Julien, »Energy Optimal Real-Time Navigation System,« *IEEE Intelligent transport system magazine*, pp. 66-79, 2014.

- [13] » Sachenbacher, M., Leucker, M., i Artmeier, A.: Efficient Energy-Optimal Routing for Electric Vehicles, Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, 2011.«.
- [14] » Eisner, J., Funke, S., Storandt, S.: Optimal Route Planning for Electric Vehicles in Large Networks, Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, 2011.«.
- [15] T. Sweda i D. Kalbjan, »Finding minimum-cost paths for electric vehicles,« u *Electric Vehicle Conference (IEVC), 2012 IEEE International, Greenville, , SC*, 2012.
- [16] <http://www.mobilerobots.com/ResearchRobots/P3AT.aspx>, »pristupljeno, 28.12.2015.«.
- [17] T. Erdelić, »Lokalizacija mobilnog robota Pioneer 3-DX pomiću kinect senzora,« Završni rad, Fakultet elektrotehnike i računarstva, Zagreb, 2010.
- [18] »Operativni priručnik za Pioneer robote,« Mobile Robots Inc., Amherst, 2006.
- [19] <http://www.mobilerobots.com/Software.aspx>.
- [20] Fošner, M., Kramberger, T., hrvatski matematički elektronički časopis, Teorija grafova i logostika, vol. 27, 2012..
- [21] Bronstein, I., N., Semendjajev, K., A., Musiol, G., Muhling, H., Matematički priručnik, Golden marketing- Tehnička knjiga, Zagreb, 2004..
- [22] Carić, T., Optimizacija prometnih procesa (nastavni tekst), Fakultet prometnih znanosti Sveučilišta u Zagrebu, Zagreb, 2014..
- [23] Majstorović, S. ,Grafovi (nastavni materijal), Odjel za matematiku Sveučilišta u Osijeku, Osijek, 2012/2013..
- [24] Goldemac, A., Mimica, A., Vučićić, T.: Od koenigsberskih mostova do kineskog poštara, Math.e, svez. 21, 2004..
- [25] Stanimirović, Z., Ivanović, M., Matematičko programiranje i optimizacija, Matematički fakultet Univerziteta u Beogradu, Beograd, 2010..
- [26] URL:
<http://www.selimbegovic.com/Dokumenti/Metricki%20problemi%20u%20teoriji%20grafova.pdf>, pristupljeno 16.06.2016.

- [27] Nikšić, F., Traženje najkraćeg puta, playMath časopis, br. 3, 2003..
- [28] Dijkstra, E.W.: A note on two problems in connexion with graphs, Numerische mathematik, svez. 1, pp. 269-271, 1959..
- [29] URL: http://www.sup.hr/index.php?option=com_content&view=article&id=582:volvo-potvrdio-datehnologija-zamanjaka-tedi-do-25-posto-goriva&catid=40:novosti&Itemid=61, pristupljeno lipanj, 2016..
- [30] URL : Robotic Motion Planning: A* and D* Search, Robotics Institute, AyorkorMills-Tetty, Vincent Lee-ShueJr. Prasad NarendraAtkar, Kevin Tantisevi, pristupljeno: lipanj, 2016..
- [31] Cindro, N.: Fizika 1, Školska knjiga, Zagreb, 1975..
- [32] Jurum-Kipke, J.: Mehanika u prometnom inženjerstvu, Fakultet prometnih znanosti Sveučilišta u Zagrebu, Zagreb, 2001..
- [33] K. Pisačić, »Osnove inženjerskog proračuna,« Skripta, Sveučilište Sjever, Koprivnica, 2014.
- [34] Drane C., Rizos, C.: Positioning Systems in Intelligent Transportation Systems, Norwood, Artech House, 1998..
- [35] Zhao, Y.: Vehicle Location and Navigation Systems, Norwood, Artech House, 1997..
- [36] Galić, A., Mario, B., Fosić, J.: Laboratorijske vježbe, Algoritmi i programiranje, Fakultet prometnih znanosti Sveučilišta u Zagrebu, Zagreb, 2010..
- [37] URL: <https://www.teslamotors.com/models-charging#/onthego>, pristupljeno: 20.2.2016..
- [38] Sachenbacher, M., Leucker, M., i Artmeier, A.: Efficient Energy-Optimal Routing for Electric Vehicles, Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, 2011.
- [39] J. Jurum-Kipke, Mehanika u prometnom inženjerstvu, Zagreb: Fakultet prometnih znanosti, 2001.
- [40] Stojkov, M., Gašparović, D., Pelin, D., Glavaš, H.: Električni automobil - povijest razvoja i sastavni dijelovi, Plin 2014., 12. skup o prirodnom plinu, toplini i vodi, Osijek, 2014..

Popis slike

Slika 1. Logički prikaz arhitekture multimodalnih rutnog vodiča.....	5
Slika 2. Usluga Google Transit.....	6
Slika 3. Fizičke komponente Pioneer 3-AT mobilnog robota	10
Slika 4. Optimizacijski problemi prikazani pomoću skupova	13
Slika 5. Primjer nepovezanog grafa	14
Slika 6. Primjer stabla i zvijezda grafa	15
Slika 7. Usmjeren graf	16
Slika 8. Pseudo kod Dijkstrinog algoritma	18
Slika 9. Ispis Dijkstrinog algoritma	18
Slika 10. Pseudokod Bellman-Fordovog algoritma	20
Slika 11. Pseudo kod algoritma A*	21
Slika 12. Kretanje robota u 8 smjerova	22
Slika 13. Glavna procedura D* algoritma	23
Slika 14. Prikaz procedure za umetanje elementa u listu INSERT	23
Slika 15. Procedura inicijalnog plana INIT_PLAN	24
Slika 16. Procedura za kontrolu sa senzora odabranog puta PREPARE_REPAIR.....	24
Slika 17. Procedura za modifikaciju težina MODIFY_COST	24
Slika 18. Procedura za izmjenu plana REPAIR_REPLAN	24
Slika 19. Sile na terenu s i bez nagiba	26
Slika 20. Trokut potreban za izvod postotka nagiba terena	27
Slika 21. Prometna mreža prikazana kao nepotpuni težinski usmjereni graf	30
Slika 22. Metoda Ukupno	32
Slika 23. Prikaz rada modificiranog Dijkstrinog algoritma	33
Slika 24. Model navigacijskog sustava uz dodatak modificiranog Dijkstrinog algoritma	34

Popis tablica

Tablica 1. Specifikacije mobilnog robota Pioneer 3-AT	11
Tablica 2. Primjer težina u poljima.....	22
Tablica 3. Potrošnja energije s obzirom na brzinu kretanja mobilnog robota na putu od 100 m	29
Tablica 4. Parametri prometne mreže iz primjera prikazanog Slikom 21.	31