

# Definiranje parametara algoritma za dodjeljivanje prioriteta vlakovima u postupku automatskog postavljanja putova vožnje

---

**Belobrajdić, Ivan**

**Master's thesis / Diplomski rad**

**2020**

*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://um.nsk.hr/um:nbn:hr:119:778960>

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

*Download date / Datum preuzimanja:* **2025-03-11**



*Repository / Repozitorij:*

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



**SVEUČILIŠTE U ZAGREBU  
FAKULTET PROMETNIH ZNANOSTI**

**Ivan Belobrajdić**

**DEFINIRANJE PARAMETARA ALGORITMA ZA DODJELJIVANJE  
PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG  
POSTAVLJANJA PUTOVA VOŽNJE**

**DIPLOMSKI RAD**

**Zagreb, 2020.**

Sveučilište u Zagrebu  
Fakultet prometnih znanosti

## **DIPLOMSKI RAD**

**DEFINIRANJE PARAMETARA ALGORITMA ZA DODJELJIVANJE  
PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG  
POSTAVLJANJA PUTOVA VOŽNJE**

**DEFINING OF ALGORITHM PARAMETERS FOR TRAIN PRIORITY  
ASSIGNING WITHIN THE PROCESS OF AUTOMATIC ROUTE  
SETTING**

Mentor: izv. prof. dr. sc. Hrvoje Haramina

Student: Ivan Belobrajdić  
JMBAG:0195026572

Zagreb, 2020.

# DEFINIRANJE PARAMETARA ALGORITMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG POSTAVLJANJA PUTOVA VOŽNJE

## SAŽETAK

U radu se obrađuje problem dodjeljivanja prioriteta vlakovima s ciljem podrške dispečeru u procesu regulacije željezničkog prometa, a u svrhu unapređenja procesa automatskog postavljanja putova vožnje. U tu svrhu razvijen je novi algoritam dodjeljivanja prioriteta vlakovima koji su definirani različiti ulazni parametri koji utječu na proces donošenja odluke pri dodjeljivanju prioriteta vlakovima. Program je izrađen u programskom jeziku C Sharp, te je izrađena aplikacija za računalo koja služi za njegoo testiranje. Program je testiran na tri različita primjera vezana za kolodvore Beli Manastir i Zagreb Klara. Rezultati testiranja potvrđuju da bi primjena ovog algoritma mogla unaprijediti proces regulacije željezničkog prometa.

**KLJUČNE RIJEČI:** algoritam za dodjeljivanje prioriteta vlakovima, podrška u odlučivanju, C Sharp, željeznički promet

## ABSTRACT

The paper deals with the problem of assigning priorities to trains in order to support the dispatcher in the process of regulating railway traffic, and in order to improve the process of automatic setting of tracks. To this end, a new train prioritization algorithm has been developed that defines various input parameters that influence the decision-making process when prioritizing trains. The program was created in the C Sharp programming language, and a computer application was created for testing. The program was tested on three different examples related to Beli Manastir and Zagreb Klara stations. The test results confirm that the application of this algorithm could improve the process of railway traffic regulation.

**KEY WORDS:** train priority assigning, algorithm, decision suport, C Sharp, railway traffic

## SADRŽAJ

1. UVOD.....	1
2. DEFIRANJE PARAMETARA ALGORITMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA PRI REGULACIJI ŽELJEZNIČKOG PROMETA U KOLODVORSKOM PROMETU.....	2
2.1 Rang vlaka .....	2
2.2 Popunjenost vlaka .....	2
2.3 Integrirani prijevoz putnika i veze sa drugim vlakovima i brodovima .....	3
2.4 Prolazna vožnja teretnog vlaka kroz kolodvorsko područje .....	4
2.5 Vrijeme kašnjenja.....	5
2.6 Međunarodni promet.....	5
2.7 Približavanje važnijim kolodvorima .....	5
2.8 Predviđeno vrijeme razrješavanja puta vožnje.....	5
3. IZRADA PROGRAMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG POSTAVLJANJA PUTOVA VOŽNJE U KOLODVORSKOM PODRUČJU .....	6
3.1 Programski jezik C Sharp.....	9
3.2 Rad programa .....	10
3.3 Definiranje ulaznih parametara .....	10
3.4 Vrijednosti varijabli .....	13
3.5 Izrada aplikacije.....	16
4. TESTIRANJE PROGRAMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG POSTAVLJANJA PUTOVA VOŽNJE U KOLODVORSKOM PODRUČJU I ANALIZA NJIHOVOG REZULTATA.....	19
4.1 Prvi primjer analize novog programa.....	20
4.2 Drugi primjer analize novog programa .....	23
4.3 Treći primjer analize novog programa.....	28
5. ZAKLJUČAK .....	31
6. LITERATURA .....	32
POPIS SLIKA.....	33
POPIS TABLICA.....	34

# 1. UVOD

Za uspješno upravljanje željezničkim prometom potrebno je smanjiti ili ukloniti svako kašnjenje, jer ako jedan vlak zakasni moguće je da će zbog tog kašnjenja doći do širenja posljedičnog kašnjenja vlakova u mreži

Cilj ovog diplomskog rada je definirati parametre za dodjeljivanje prioriteta vlakovima u postupku automatskog postavljanja puta vožnje, te izraditi algoritam i napraviti program. Ovaj algoritam bi trebao služiti za podršku u odlučivanju u procesu regulacije željezničkog prometa. U tu svrhu potrebno je definirati ulazne kriterije koji mogu utjecati na dodjelu prioriteta određenom vlaku.

Ovaj algoritam služio bi za pomoć pri donošenju odluka dispečeru koji regulira željeznički promet u slučaju kada se dogodi da dva ili više vlakova potencijalno pretendira na iste elemente infrastrukture u kolodvorskom području u svrhu osiguranja njihovih putova vožnje, a to se događa kada postoji kašnjenje vlakova u odnosu na planirani vozni red. Rad je podijeljen u pet cjelina:

1. Uvod
2. Definiranje parametara algoritma za dodjeljivanje prioriteta vlakovima pri regulaciji željezničkog prometa u kolodvorskom području
3. Izrada programa za dodjeljivanje prioriteta vlakovima u postupku automatskog postavljanja putova vožnje u kolodvorskom području
4. Testiranje programa za dodjeljivanje prioriteta vlakovima u postupku automatskog postavljanja putova vožnje u kolodvorskom području i analiza dobivenih rezultata
5. Zaključak

U drugom poglavlju opisani su ulazni parametri koji su odabrani za rad algoritma.

U trećem poglavlju iznesene su neke osnove o programskom jeziku C#, te je opisan postupak izrade algoritma. Prikazan je njegov dijagram toka i opisana je izrada programa u programskom jeziku C#, te su objašnjene neke osnove vezane z dizajniranje računalne aplikacije za testiranje programa.

U četvrtom poglavlju je program testiran na tri različita primjera vezana za dva različita kolodvora

## 2. DEFIRANJE PARAMETARA ALGORITMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA PRI REGULACIJI ŽELJEZNIČKOG PROMETA U KOLODVORSKOM PROMETU

Dodjeljivanje prioriteta vlakovima pri regulaciji željezničkog prometa u kolodvorskom prometu je potrebno samo kada dođe do kašnjenja nekog vlaka. Taj problem zatim rješava prometni dispečer pomoću lokalno kriterija za donošenje takvih odluka. Dva najčešća pravila su first-in-first-out(FIFO) i first-out-first-in(FOFI). U ovom diplomskom radu će se uz uobičajenu podjelu po rangu vlaka koja je za Hrvatsku definirana u Prometnom pravilniku dodjeljivati i dodatna vrijednost za svaki vlak. [1]

Svakom vlaku će se dodati određen broj bodova prema njegovom rangu, te će se uz to dodavati i dodatni bodovi za popunjenost kapaciteta vlaka, ako je taj vlak veza za neki drugi vlak koji ga čeka ili za neko prijevozno sredstvo iz drugog oblika prometa (npr. Autobus, trajekt, itd.).

### 2.1 Rang vlaka

Kod rješavanja ovog problema vrlo je bitno koja vrsta vlaka prometuje da bi se znalo odrediti koji vlak ima prioritet za dodjelu puta vožnje. U prometnom pravilniku Republike Hrvatske su već rangirani vlakovi prema važnosti.

### 2.2. Popunjenost vlaka

Kod donošenja odluke koji vlak ima prednost prolaska kroz kolodvor treba uzeti i u obzir postotak popunjenosti kapaciteta pojedinog vlaka te prema tome podatku dodati dodatne bodove pri računanju ukupne vrijednosti koju svaki vlak ima. Naravno to vrijedi samo kod vlakova za prijevoz putnika.

U ovom radu popunjenost svakog vlaka za prijevoz putnika će biti podijeljen na 4 dijela, a to su: popunjenost od 0%, od 0 – 50%, od 50 – 75%, te na vlakove popunjene

više od 75 %, pa će se prema tom kriteriju svakom vlaku za prijevoz putnika dodati određeni broj bodova.



Slika 1. Primjer popunjenosti preko 75% kapaciteta

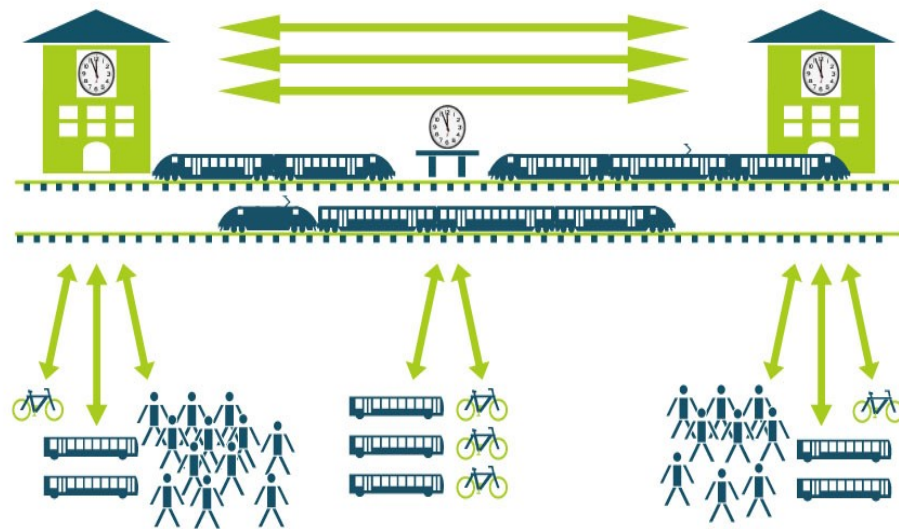
Izvor: <https://www.vecernji.hr/zagreb/samo-se-najratoborniji-uguraju-u-vlak-a-onda-se-voze-ko-sardine-1307869/galerija-352888?page=8>

### 2.3. Integrirani prijevoz putnika i veze sa drugim vlakovima i brodovima

Integrirani javni prijevoz putnika je suvremeni sustav javnog prijevoza putnika u kojem su vozila različitih vrsta javnog prijevoza (vlakovi, autobusi, tramvaji, brodovi ...) objedinjena u zajednički sustav prijevoza putnika. Za sve linije javnog prijevoza vrijede jedinstvene (zajedničke) karte na području određene regije. Sustav uglavnom koristi taktne vozne redove, odnosno polasci sa svakog stajališta su u pravilnim vremenskim razmacima (npr. svakih 10, 20, 30, 60 minuta itd.). Vozni redovi u sustavu su usklađeni



te putnicima omogućuju brz nastavak putovanja nakon presjedanja. Kralježnicu sustava čine tračnički sustavi zbog svojih ekoloških, energetskih i ostalih i infrastrukturnih prednosti [2]. Upravo zbog tih razloga vrlo je bitno da vlakovi ne kasne i ako kasne da kašnjenje bude što manje i zbog toga će vlakovi uključeni u integrirani sustav prijevoza u okviru izrade algoritma dobiti dodatne bodove za dobivanje prioriteta u vožnji.



Slika 2. Shema Integriranog prijevoza putnika

Izvor: <https://szz.hr/projekti/ipp>

Vlakovi koji su veza nekim drugim vlakovima (npr. InterCity, EuroCity) ili su veza za neki brod ili avion također dobivaju dodatan broj bodova jer ti vlakovi ne bi smjeli kasniti.

#### 2.4. Prolazna vožnja teretnog vlaka kroz kolodvorsko područje

Teretni vlakovi zbog svoje težine veliku količinu energije troše pri kretanju i zbog toga bi vlakovi koji se prema voznom redu ne zaustavljaju već tranzitiraju kroz kolodvor trebali s obzirom na svoju masu dobiti određen broj dodatnih bodova. S obzirom na to ovaj parametar će imati utjecaj na rad algoritma za dodjelu prioriteta vlakovima.

## 2.5. Vrijeme kašnjenja

Vrijeme kašnjenja utječe na određivanje njegovog prioriteta. U okviru izrade algoritma primjenjena je logika da kod vlakova za prijevoz putnika vlak koji kasni više treba dobiti veći prioritet sa svrhom da se to kašnjenje ne poveća odnosno eventualno smanji. Kod vlakova za prijevoz tereta onaj vlak koji manje kasni trebao bi dobiti veći prioritet. Bitno je da se teži vlakovi propuštaju jer je njih teže zaustaviti i pokrenuti, a također i veća je ušteda ako se teški vlakovi ne trebaju zaustavljati. Iz tog razloga ponekad će takav teretni vlak dobiti prednost čak i ispred nekog putničkog vlaka. To je povezano s time da teški i dugački teretni vlakovi nakon zaustavljanje ne mogu odmah krenuti.

## 2.6. Međunarodni promet

Vlakovima iz međunarodnog prometa se dodjeljuje prednost. Ovaj kriterij vrijedi za vlakove za prijevoz putnika i tereta.

## 2.7. Približavanje važnijim kolodvorima

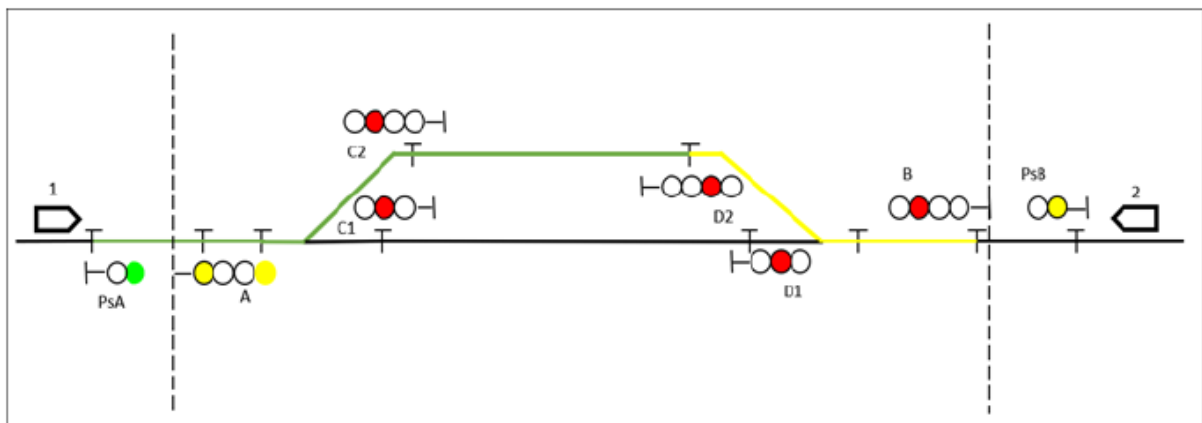
Kada se vlak približava velikom kolodvoru trebalo bi mu dati veći prioritet jer takvi vlakovi ne bi trebali kasniti. Kada vlak koji ide prema značajnijem kolodvoru kasni onda zbog toga može doći do većeg utjecaja tog kašnjenja na uredno odvijanje prometa drugih vlakova u mreži.

## 2.8. Predviđeno vrijeme razrješavanja puta vožnje

Vlak za koji se predviđa razrješavanje puta vožnje može dobiti određenu prednost u odnosu na konkretne vlakove iz razloga kako bi se optimiralo vrijeme vožnje vlakova i pokušalo anulirati nastalo kašnjenje.

### 3. IZRADA PROGRAMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG POSTAVLJANJA PUTOVA VOŽNJE U KOLODVORSKOM PODRUČJU

U ovom radu predstavljen je novi algoritam za pomoć dispečeru kod odnošenja odluke o tome koji vlak ima prioritet prolaska kroz kolodvor. Ovaj program bi trebao brže donositi odluke o dodjeljivanju prioriteta vlakovima na temelju definiranih ulaznih parametara, koje dispečer nakon toga može, ali i ne mora, prihvatiti. Krajnji cilj ovog sustava je da se u potpunosti rastereti dispečera te da sustav sam donosi odluke bez krajnjeg odobrenja dispečera. Do te situacije, kada treba odlučiti koji vlak ima prednost prolaska, dolazi tek kada jedan od vlakova počne kasniti, što može dovesti do toga da se putevi vožnje za različite vlakove ne mogu postaviti istovremeno kako se ne bi preklapali, dodirivali i sjekli, kao što je vidljivo na slici 7.

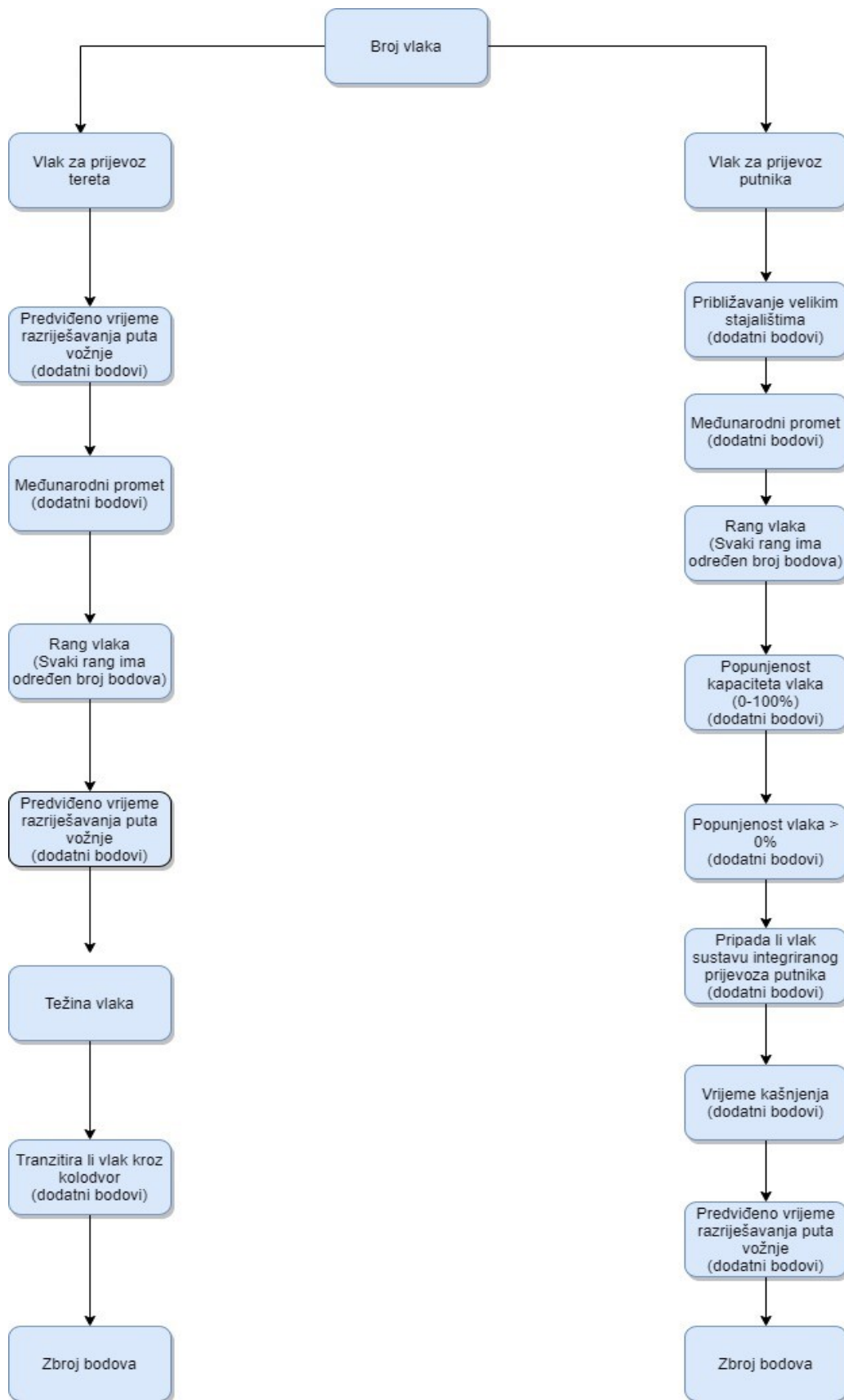


Slika 3. Suprotstavljene putevi vožnje

Izvor. Haramina, H., Sumpor, D., Širol M.: Modelling of Fuzzy Logic Based Dispatcher Support System for Railway Traffic Control, Book of Proceedings of the 7<sup>th</sup> International Ergonomics Conference, Croatian Ergonomics Society, Zagreb, 2018.

Ovaj algoritam će biti kodiran u programskom jeziku C# prema definiranom dijagramu toka (Slika 8.). Dijagram toka je napravljen da izračuna vrijednost jednoga vlaka, a program će računati vrijednosti za dva ili više vlakova te uspoređivati rezultate i donijeti odluku koji vlak ima prioritet da mu se postavi put vožnje. U algoritmu se

definira 9 ulaznih parametara od čega su tri zajednička za teretne i vlakove za prijevoz putnika. Vlakovi za prijevoz putnika koriste 6 ulaznih parametara a vlakovi za prijevoz tereta koriste također 6 ulaznih parametara.

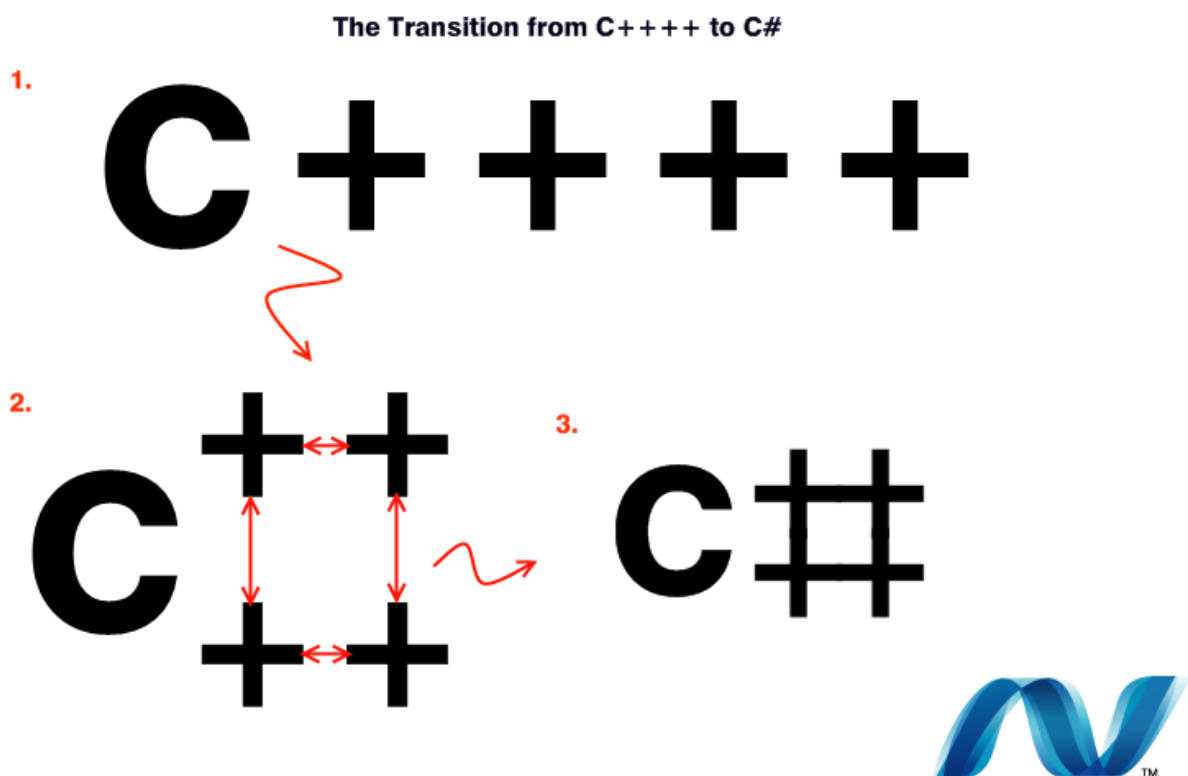


Slika 4. Dijagram toka za izračunavanje broja bodova za svaki vlak

### 3.1. Programski jezik C Sharp

Programski jezik C Sharp (C#) je proizvod kompanije Microsoft i nastao je kao odgovor na nedostatke postojećih jezika kao što su C, C++ i Visual Basic, u isto vrijeme kombinirajući njihove dobre strane. C# je programski jezik koji omogućava izradu aplikacija za web sa svom potrebnom funkcionalnošću. Osim što pruža programski okvir za razvoj softvera, omogućava i podršku za pristupanje brojnim bazama podataka. Kako predstavlja evoluciju jezika C i C++, kao takav je kreiran uz korištenje svih prednosti ostalih jezika i otklanjanje njihovih mana. Sam jezik nema ograničenja u pogledu toga kakve se aplikacije mogu napraviti u njemu, on koristi okruženja i samim time nema ograničenja u vezi s mogućim aplikacijama[3].

C# je korišten za izradu ovog programa prema dijagramu toka (slika 8) zato što su uklonjene mane iz prijašnjih programskih jezika C i C++, i jer ga je lakše koristiti od ta dva programa.



Slika 5. način na koji je nastalo logo C#

Izvor: <https://www.donnfelker.com/how-did-c-get-its-name/>

## 3.2. Rad programa

Nakon pokretanja programa unosi se određen broj vlakova. Za svaki pojedini vlak dodjeljuje se njegovu oznaku te bira da li je on za prijevoz putnika ili prijevoz tereta. Ako je odabrana opcija da je vlak za prijevoz putnika onda se bira njegov rang za što on može dobiti određen broj bodova. Nakon toga postoji mogućnost unosa još 6 dodatnih parametara i svaki od njih vlaku povećava ili smanjuje određeni broj bodova. A ako smo odabrali da je vlak za prijevoz tereta također onda biramo njegov rang i još dodatnih 6 parametara. Dodatni bodovi su objašnjeni u poglavlju 3.4. Nakon što smo unijeli sve potrebne podatke program izračuna za svaki vlak koliko je skupio bodova tj. koja je njegova vrijednost. Nakon čega program uspoređuje sve unesene vlakove i poreda ih po broju sakupljenih bodova od većega prema manjem i tako prikaže koji vlak ima prioritet da mu se postavi put vožnje.

## 3.3. Definiranje ulaznih parametara

Na samom početku programa se definiraju ulazne varijable u programskom jeziku C#. Te varijable služe kao parametri pomoću kojih ćemo kasnije uspoređivati vrijednosti svakog vlaka te na kraju donosimo odluku na temelju tih vrijednosti koji vlak ima prioritet u postupku postavljanja automatskog puta vožnje.

Te varijable su kombinacija pravila za određivanje prioriteta iz nekoliko različitih država[4]. One su pomno proučene i odabrane kako bi dobili što precizniji i pošteniji rezultat. Te varijable nije moguće mijenjati nakon što se program pokrene.

Na slici 10. vidljiv je prikaz koda di su definirane ulazne varijable. Za te varijable uzete su one koje bi dispečer trebao uzimati u obzir. U ovom programu definirani su sljedeći ulazni parametri:

- Rang vlaka
- Popunjenost vlaka
- Integrirani prijevoz putnika
- Tranzicija teretnog vlaka
- Vrijeme kašnjenja
- Težina teretnog vlaka
- Međunarodni promet
- Približavanje velikim stajalištima
- Predviđeno vrijeme razrješavanja puta vožnje

Pomoću tih ulaznih parametara dolazimo do broja bodova koji nosi svaki vlak. Na kraju pomoću tih parametara, odnosno vrijednosti vlaka dolazimo do izlazne varijable. Tu izlaznu varijablu dobijemo uspoređivanjem dvaju ili više vlakova, te onaj vlak koji nosi na kraju najveću vrijednost ima prioritet kod automatskog dodjeljivanja putova vožnje.



```

4
5 public class Train
6 {
7     2 references
8     public string Mark { get; set; }
9
10    5 references
11    public TrainType TrainType { get; set; }
12
13    3 references
14    public TrainSubType TrainSubType { get; set; }
15
16    2 references
17    public bool IsTransitTrain { get; set; }
18
19    2 references
20    public bool IsConnection { get; set; }
21
22    9 references
23    public decimal PassengersPercentage { get; set; }
24
25    5 references
26    public decimal TrainWeight { get; set; }
27
28    13 references
29    public decimal TrainDelay { get; set; }
30
31    4 references
32    public bool IsInternational { get; set; }
33
34    4 references
35    public bool IsApproachingHUB { get; set; }
36
37    3 references
38    public decimal TimeForRouteSetting { get; set; }
39
40    2 references
41    public int CoeffForRouteSetting { get; set; }
42
43    2 references
44    public int CoeffTrainDelayFreight { get; set; }

```

Slika 6. Definiranje ulaznih parametara u programskom jeziku C#

Nakon definiranja ulaznih varijabli potrebno je svakoj od njih dodijeliti određen broj bodova, što će se vidjeti u daljnjem tekstu. Neki od ulaznih parametara mogu dobiti različiti broj bodova, što ovisi o tome kolika je vrijednost te varijable.

### 3.4. Vrijednosti varijabli

Kao što je već navedeno svakoj varijabli je dodijeljen određen broj bodova (njezina vrijednost). Pomoću tih vrijednosti i uspoređivanja krajnjih rezultata dobivamo vlak s prioritetom za postavljanje puta vožnje.

Prva od varijabli je rang vlaka a ona je podijeljena na 10 vlakova za prijevoz putnika te na 10 vlakova za prijevoz tereta i svaki od njih ima svoju vrijednost, kao što je prikazano u tablici 1.

Tablica 1. Vrijednosti za rang vlaka

<b>Vlakovi za prijevoz putnika</b>	EuroCity (EC)	155	<b>Vlakovi za prijevoz tereta</b>	Ekspresni teretni vlak	75
	InterCity (IC)	145		Brzi teretni vlak	70
	Ekspresni vlak	120		Vojni vlak	65
	Brzi vlak	110		Maršrutni vlak	63
	Ubrzani vlak	100		Izravni teretni vlak	62
	Putnički vlak	90		Dionički teretni vlak	55
	Lokalni vlak	85		Vlak za posebne namjene	50
	Pogranični vlak	80		Sabirni vlak	45
	Prigradski vlak	80		Kružni vlak	40
	Vlak za prijevoz željezničkih radnika	67		Industrijski vlak	30

U tablici 1. su prikazane vrijednosti prema rangu vlaka, a uz to svaki vlak može dobiti još dodatne bodove prema idućim pravilima:

- Ako je vlak za prijevoz putnika dio integriranog prijevoza putnika ili je on veza za neki drugi vlak (ili neko drugo prijevozno sredstvo) njemu se dodjeljuje dodatnih 30 bodova
- Ako vlak za prijevoz putnika kasni:
  - 1-2 minute dodijelit će mu se dodatnih 6 bodova
  - 2-5 minuta dodijelit će mu se dodatnih 11 bodova
  - Više od 5 ali manje od 10 minuta dodijelit će mu se dodatnih 16 bodova
  - Između 10 i 15 minuta izgubiti će 6 bodova i
  - Ako kasni više od 15 minuta izgubiti će 11 bodova
- Ako vlak za prije putnika dosegne popunjenost kapaciteta, dodijelit će mi se idući broj bodova:
  - Između 35 i 50 % dodjeljuje mu se 5 bodova
  - Između 51 i 70 % dodjeljuje mu se 8 bodova
  - Između 71 i 85 % dodjeljuje mu se 12 bodova
  - Iznad 85 % dodjeljuje mu se 20 bodova
- Vlakovima za prijevoz tereta prema njihovoj masi dodijelit će se idući broj bodova:
  - Između 1500 i 2000 tona dodijelit će mu se 20 bodova
  - Između 2000 i 2500 tona dodijelit će mu se 30 bodova
  - Više od 2500 tona dodijelit će mu se 50 bodova
- Teretnom vlaku koji tranzitira kroz kolodvor dodijelit će se dodatna 3 boda
- Vlaku kojem će se brže razriješiti put vožnje dodjeljuje se dodatna 2 boda
- Vlaku iz međunarodnog prometa dodjeljuje se dodatnih 15 bodova
- Ako se vlak približava nekom velikom čvorištu (npr. Zg Glavni kolodvor) dodijelit će mu se dodatnih 25 bodova

Definiranje tih dodatnih bodova u programskom jeziku C# je vidljivo na slici 11.

```

else
{
    if (PassengersPercentage >= 35 && PassengersPercentage <= 50)
        totalCoeff += 5;
    else if (PassengersPercentage > 50 && PassengersPercentage <= 70)
        totalCoeff += 8;
    else if (PassengersPercentage > 70 && PassengersPercentage <= 85)
        totalCoeff += 12;
    else if (PassengersPercentage > 85 )
        totalCoeff += 20;

    if (TrainDelay >= 1 && TrainDelay <= 2)
        totalCoeff += 6;
    else if (TrainDelay > 2 && TrainDelay <= 5)
        totalCoeff += 11;
    else if (TrainDelay > 5 && TrainDelay <= 10)
        totalCoeff += 16;
    else if (TrainDelay > 10 && TrainDelay <= 15)
        totalCoeff -= 6;
    else if (TrainDelay > 15)
        totalCoeff -= 11;

    if (IsConnection)
        totalCoeff += 30;

    if (IsApproachingHUB)
        totalCoeff += 25;

    if (IsInternational)
        totalCoeff += 15;
}

totalCoeff += CoeffForRouteSetting;

```

*Slika 7. Definiranje bodova u programskom jeziku C#*

### 3.5. Izrada aplikacije

Aplikacija je napravljena u Windows Presentation Foundation (WPF), a on je podsistem Microsoft Visual Studio u kojem je i napravljen ovaj program. WPF aplikacije mogu se koristiti kao samostalni desktop programi ili mogu biti postavljeni na željenim web lokacijama.

Windows Presentation Foundation platforma podržava širok skup značajki za razvoj aplikacija uključujući: aplikacijski model, resurse, kontrole, grafiku, izgled, povezivanje podataka i dokumenata[5].

Kod izrade aplikacije mora se unijeti margina za svaku kućicu u aplikaciji, kao što je to vidljivo na slici 12. Ako želimo da nam nešto piše u aplikaciji to se radi uz naredbu „Content =“ tako da se iza te naredbe unutar navodnih znaka unese tekst koji želimo da se ispiše u aplikaciji, za koji se također moraju unijeti margine da bi se znalo na kojem mjestu se taj tekst ispisuje. Svaka kućica za unos podataka ili za stavljanje kvačice se mora povezati sa ulaznim parametrom za koji se taj podatak unosi, a to se radi sa naredbom „Binding“.

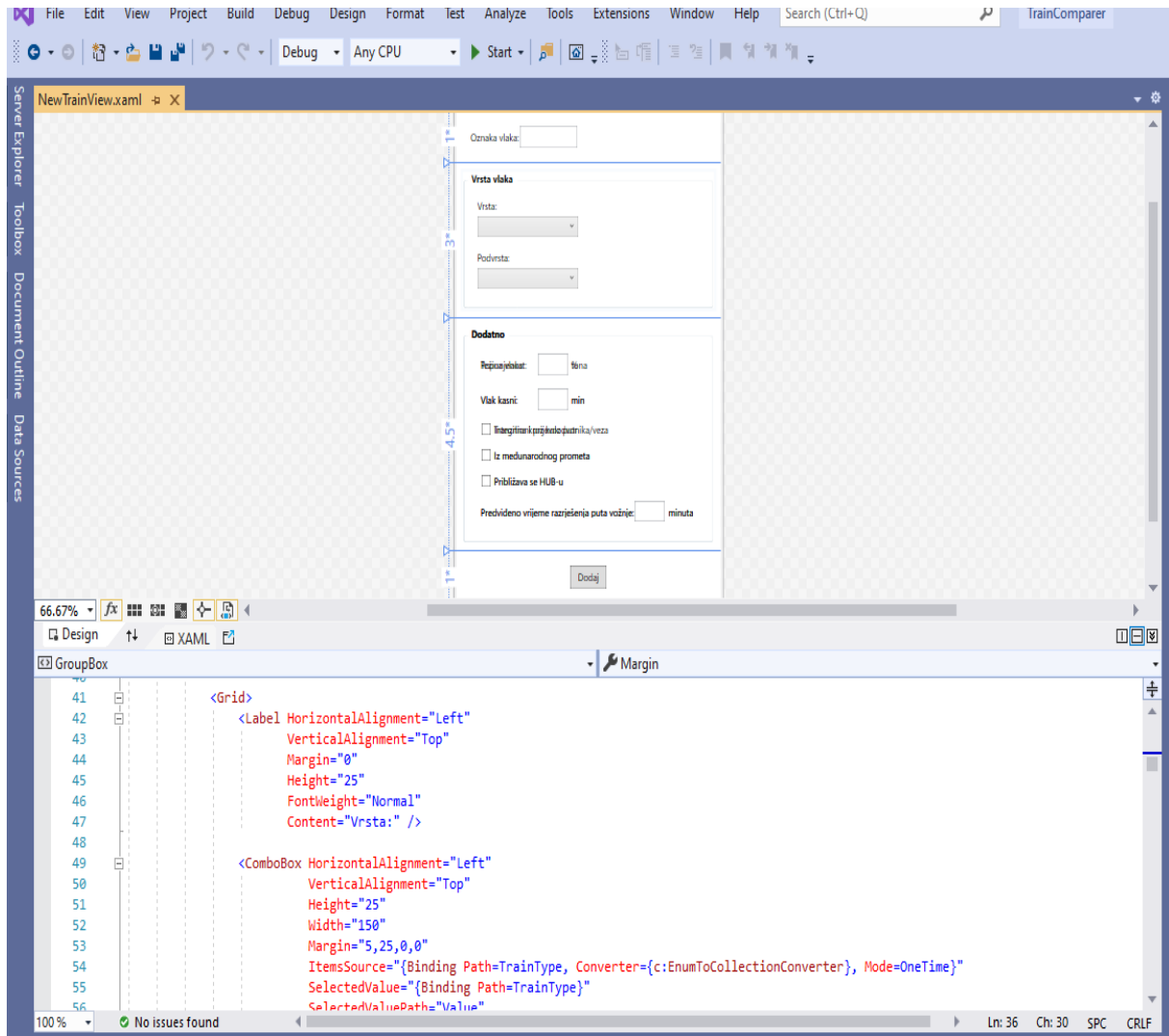
```

41 | <Grid>
42 |   <Label HorizontalAlignment="Left"
43 |     VerticalAlignment="Top"
44 |     Margin="0"
45 |     Height="25"
46 |     FontWeight="Normal"
47 |     Content="Vrsta:" />
48 |
49 |   <ComboBox HorizontalAlignment="Left"
50 |     VerticalAlignment="Top"
51 |     Height="25"
52 |     Width="150"
53 |     Margin="5,25,0,0"
54 |     ItemsSource="{Binding Path=TrainType, Converter={c:EnumToCollectionConverter}, Mode=OneTime}"
55 |     SelectedValue="{Binding Path=TrainType}"
56 |     SelectedValuePath="Value"
57 |     DisplayMemberPath="Description"/>
58 |
59 |   <Label HorizontalAlignment="Left"
60 |     VerticalAlignment="Top"
61 |     Margin="0,60,0,0"
62 |     Height="25"
63 |     FontWeight="Normal"
64 |     Content="Podvrsta:" />
65 |
66 |   <ComboBox HorizontalAlignment="Left"
67 |     VerticalAlignment="Top"
68 |     DisplayMemberPath="Name"
69 |     ItemsSource="{Binding TrainSubTypes}"
70 |     SelectedItem="{Binding SelectedTrainSubType, Mode=TwoWay}"
71 |     Height="25"
72 |     Width="150"

```

Slika 8. Izrada aplikacije u C#

Za vrijeme izrade aplikacije tj. njenog izgleda, u mogućnosti je stalno pratiti njen izgled. S time je olakšano unošenje margina za svaku kućicu jer odmah vidimo ti se ona nalazi u aplikaciji te dali se preklapa sa nekim drugim elementima te aplikacije. Također je u svakom trenutku moću pokrenuti cijeli program da se vidi dali je sve dobro napravljeno i povezano sa programom, kao što je vidljivo na slici 13.



Slika 9. Praćenje izgleda aplikacije

#### 4. TESTIRANJE PROGRAMA ZA DODJELJIVANJE PRIORITETA VLAKOVIMA U POSTUPKU AUTOMATSKOG POSTAVLJANJA PUTOVA VOŽNJE U KOLODVORSKOM PODRUČJU I ANALIZA NJIHOVOG REZULTATA

Nakon izrade algoritma i programa za dodjeljivanje prioriteta vlakovima u postupku automatskog postavljanja putova vožnje u programskom jeziku C#, potrebno je taj program i testirati prije potrebe za njegovu upotrebu.


Analiziranje rada novog algoritma i programa vršit će se na temelju stvarnog voznog reda 2019/2020 koji je propisao HŽ infrastruktura. Uzet će se dva ili više vlakova te će se napraviti takav scenarij da dolazi do suprotstavljanja putova vožnje kod tih vlakova. Odabrat će se vlakovi različitih karakteristika i na temelju toga će se testirati novi program.

Za testiranje programa koristit će se različita željeznička infrastruktura. Podaci za izradu modela željezničke infrastrukture je preuzeta iz stvarne mreže željezničke infrastrukture u Republici Hrvatskoj, a ona je modelirana u programu OpenTrack.



## 4.1. Prvi primjer analize novog programa

Za prvi primjer uzet je kolodvor Zagreb Klara (slika 17.) te su izabrani prigradski i putnički vlak. Putnički vlak oznake 2001 koji prometuje iz smjera Velike Gorice prema Zagreb Glavnom kolodvoru, za svoj rang dobiva 90 bodova, popunjenost njegovog kapaciteta iznosi 60% te za to on dobiva dodatnih 8 bodova. Vlak kasni 7 minuta zbog čega dobiva dodatnih 16 bodova, te on nije dio integriranog prijevoza putnika. Vlak putuje prema velikom kolodvoru te iz tog razloga dobiva dodatnih 25 bodova, te njegovo predviđeno vrijeme razrješenja puta vožnje iznosi 4 minute. Svi ti podaci su vidljivi na slici 14.

 Novi vlak ×

Oznaka vlaka:

**Vrsta vlaka**

Vrsta:

Podvrsta:

**Dodatno**

Popunjenost:  %

Vlak kasni:  min

Integrirani prijevoz putnika/veza

Iz međunarodnog prometa

Približava se HUB-u

Predviđeno vrijeme razrješenja puta vožnje:  minuta

Slika 10. Unošenje podataka za vlak 2001

Prigradski vlak oznake 8252 putuje iz smjera Zagreb glavni kolodvor prema Sisku, te on za svoj rang dobiva 80 bodova. popunjenost njegovog ukupnog kapaciteta iznosi 40%. Vlak kasni 2 minute i za to dobiva dodatnih 6 bodova. Njegovo predviđeno vrijeme razrješenja puta vožnje iznosi 2 minute i on iz tog razloga jer mu treba manje da se razriješi njegov put vožnje dobiva dodatna 2 boda. Svi ti podaci su vidljivi na slici 15.

Novi vlak

Oznaka vlaka:

**Vrsta vlaka**

Vrsta:

Podvrsta:

**Dodatno**

Popunjenost:  %

Vlak kasni:  min

Integrirani prijevoz putnika/veza

Iz međunarodnog prometa

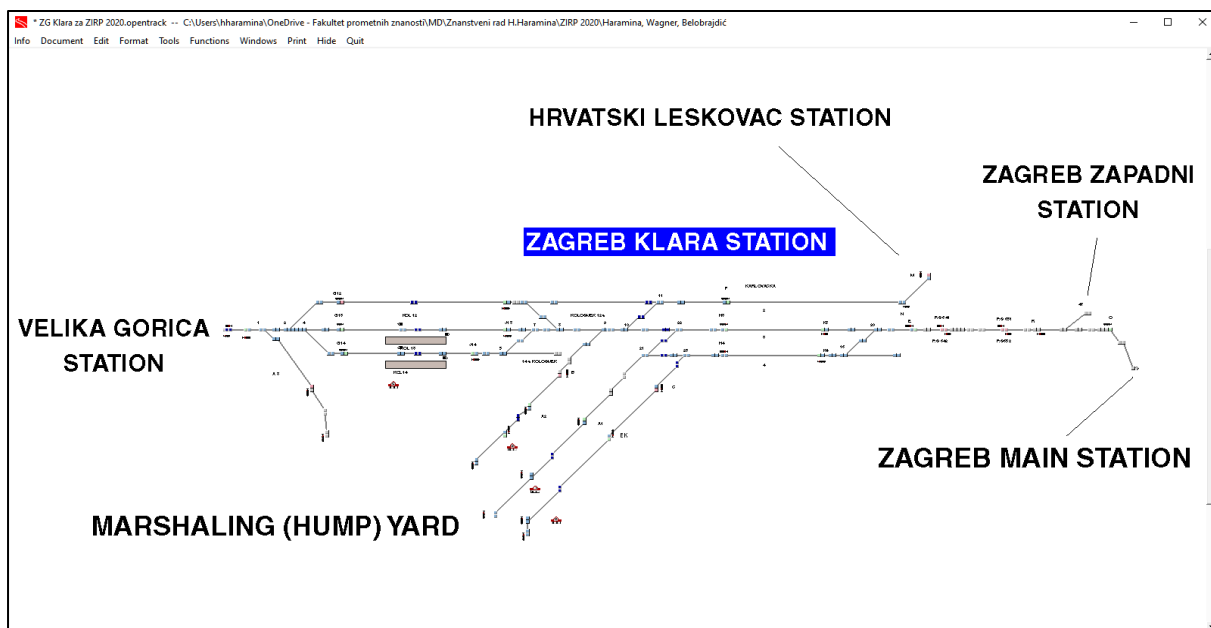
Približava se HUB-u

Predviđeno vrijeme razrješenja puta vožnje:  minuta

Slika 11. Unošenje podataka za vlak 8252

Putnički vlak 2001 na kraju je sakupio 139 bodova, a prigradski vlak 8252 je skupio 88 bodova. Kada usporedimo vrijednosti ostvarenih bodova vidljivo je da je





Slika 13. Kolodvor Zagreb Klara u programu OpenTrack

Izvor: Haramina H., Wagner A., Belobrajdić I.: A New Model an Expert System for Train Priority Assigning in Railway Traffic Control Process

## 4.2. Drugi primjer analize novog programa

Za drugi primjer je uzet kolodvor Beli Manastir (slika 21) te putnički i brzi teretni vlak. Putnički vlak oznake 2056 putuje iz smjera Osijeka prema Mađarskoj granici te je taj vlak dio međunarodnog prometa za što dobiva dodatnih 15 bodova a za njegov rang dobiva 90 bodova. Vlak kasni 20 minuta i on zbog toga kašnjenja gubi 11 bodova, te je popunjenost njegovog kapaciteta 45% i on za to dobiva 5 bodova. Njegovo predviđeno vrijeme razrješenja puta vožnje iznosi 2 minute i iz razloga što mu treba manje vremena da se razriješi put vožnje dobiva 2 dodatna boda. Svi ti podaci su vidljivi na slici 18.

Novi vlak ×

Oznaka vlaka:

**Vrsta vlaka**

Vrsta:

Podvrsta:

**Dodatno**

Popunjenost:  %

Vlak kasni:  min

Integrirani prijevoz putnika/veza

Iz međunarodnog prometa

Približava se HUB-u

Predviđeno vrijeme razrješenja puta vožnje:  minuta

Slika 14. Unošenje podataka za vlak 2056

Brzi teretni vlak (70 bodova) oznake 45652, putuje iz smjera Mađarske granice prema Osijeku. Vlak je prijevozi robu ukupne težine 2782 tone i za to dobiva dodatnih 50 bodova, te vlak ne kasni. Vlak se ne treba zaustavljati u kolodvoru nego on tranzitira kroz njega te zbog toga dobiva dodatna 3 boda, te je on dio međunarodnog prometa te za to dobiva dodatnih 15 bodova. Vlak putuje prema Osijeku koji je veliki kolodvor te iz toga razloga dobiva dodatnih 25 bodova. Predviđeno vrijeme za razrješenja puta vožnje iznosi 6 minuta. Svi ti podaci su vidljivi na slici 19.

Novi vlak ×

Oznaka vlaka:

**Vrsta vlaka**

Vrsta:

Podvrsta:

**Dodatno**

Težina vlaka:  tona

Vlak kasni:  min

Transitira kroz kolodvor

Iz međunarodnog prometa

Približava se HUB-u

Predviđeno vrijeme razrješenja puta vožnje:  minuta

Slika 15. Unošenje podataka za vlak 45652

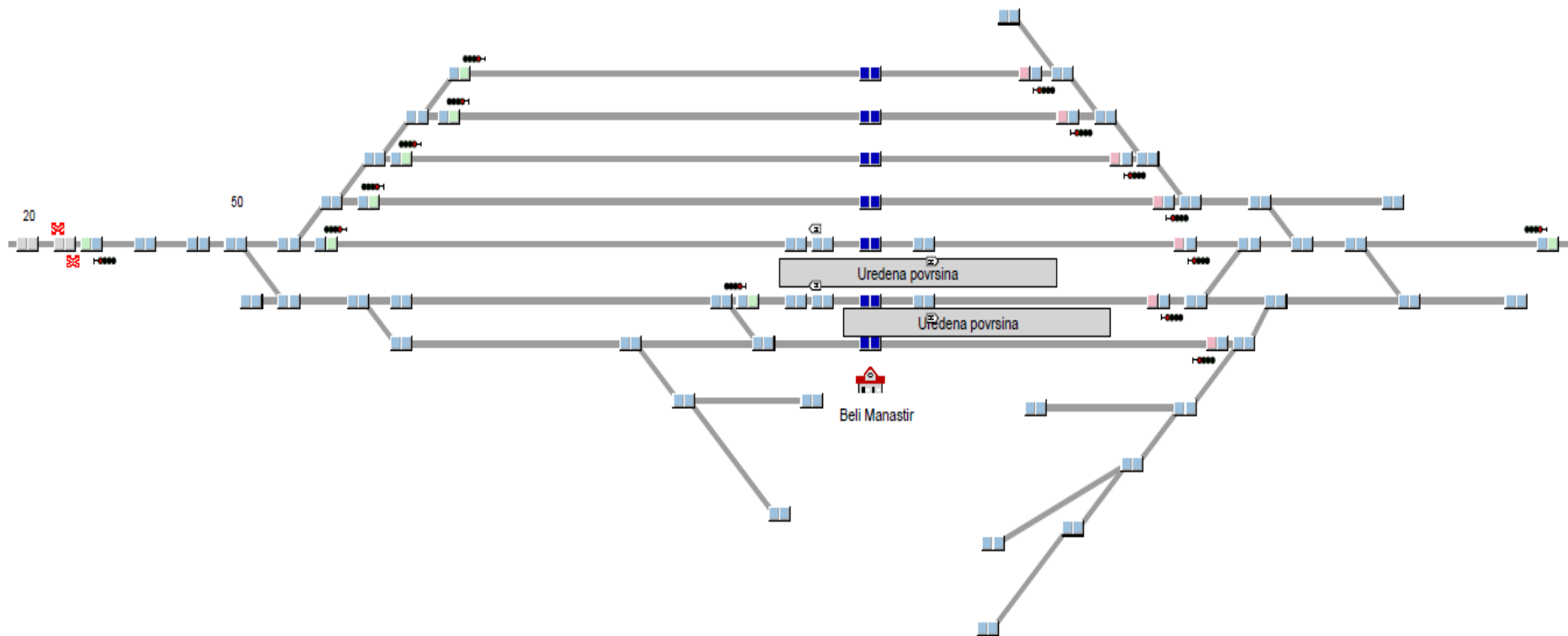
Brzi teretni vlak na kraju je sakupio 163 bodova, a putnički vlak 2056 je skupio 101 bod. Kada usporedimo ta dva broja vidljivo je da je brzi teretni vlak sakupio više bodova te on ima prednost (slika 20.).

## Redoslijed vlakova

Oznaka vlaka
45652
2056

Slika 16. Redoslijed vlakova za drugi primjer

Kao što je vidljivo prednost je dobio brzi teretni vlak 45652, što inače nije normalno da teretni vlak dobije prednost ispred putničkog. Ovakvu odluku dispečer možda ne bi donio jer možda ne bi uzeo u obzir sve ulazne parametre koje ima opisani algoritam.



Slika 17. Kolodvor Beli Manastir a u programu OpenTrack



### 4.3. Treći primjer analize novog programa

Za treći primjer je uzet kolodvor Zagreb Klara (slika 17), te dva teretna vlaka. Prvi je dionički teretni vlak (55 bodova), oznake 53123, koji putuje iz smjera Velika Gorica prema kolodvoru Zagreb Ranžirni. Vlak teži 1943 tone te kasni 2 minute. Vlak samo tranzitira kroz ovaj kolodvor, te predviđeno vrijeme za razrješenja puta vožnje iznosi 2 minute. Svi navedeni podaci vidljivi su u prikazu iz programa na slici 22.

Novi vlak

Oznaka vlaka:

**Vrsta vlaka**

Vrsta:

Podvrsta:

**Dodatno**

Težina vlaka:  tona

Vlak kasni:  min

Tranzitira kroz kolodvor

Iz međunarodnog prometa


Približava se HUB-u

Predviđeno vrijeme razrješenja puta vožnje:  minuta

Slika 18. Unošenje podataka za vlak 53123

Brzi teretni vlak (70 bodova), oznake 51252, putuje iz smjera velika gorica prema ranžirnom kolodvoru. Vlak teži 2320 tona, te kasni 4 minute. On također tranzitira kroz

kolodvor, te predviđeno vrijeme za razrješenja puta vožnje iznosi 7 minuta. Svi ti podaci su vidljivi na slici 23.

 Novi vlak ×

Oznaka vlaka:

**Vrsta vlaka**

Vrsta:

Podvrsta:

**Dodatno**

Težina vlaka:  tona

Vlak kasni:  min

Transitira kroz kolodvor

Iz međunarodnog prometa

Približava se HUB-u

Predviđeno vrijeme razrješenja puta vožnje:  minuta

Slika 19. Unošenje podataka za vlak 51252

Veći zbroj bodova je sakupio vlak 51252, što mu daje prioritet kod postavljanja puta vožnje. Rezultat je vidljiv na slici 24.

## Redoslijed vlakova

Oznaka vlaka
51252
53123

*Slika 20. Redoslijed vlakova za treći primjer*

## 5. ZAKLJUČAK

U ovom radu su definirani ulazni parametri za algoritam za dodjeljivanje prioriteta Vlakovima, te je opisana aplikacija koja je napravljena na bazi tog algoritma. Program je izrađen u programskom jeziku C#. Ovaj algoritam je napravljen da bi olakšao posao dispečeru te ubrzao njegov rad

U radu je prikazan novi model koji se do sada još nikada nije bio korišten. Aplikacija je jednostavnog izgleda i lagano se njome upravlja. Iako se za testiranje testiranja aplikacije u okviru ovog rada podaci unose ručno, primjena ove aplikacije u praksi podrazumijeva automatsko prikupljanje i obradu podataka.

Analizom rezultata provedenih testiranja vidljivo je da ovaj algoritam daje zadovoljavajuće rezultate.

## 6. LITERATURA

[1] Nikola Stojadinović, Branislav Bošković: Train path allocation on congested railway infrastructure – is a service priority criteria limiting development od competition? 2016: 209-212

[2] Savez za željeznicu. Preuzeto sa: <https://szz.hr/projekti/ipp> [pristupljeno: svibanj 2020.]

[3] <https://www.talentlyft.com/hr/resources/programski-jezik-c-c-sharp-tehnoloski-rijecnik> [pristupljeno svibanj 2020.]

[4] RailNetEurope, Overview of priority rules in operation, RNE Operation & After-sales WG, 4th December 2019.

[5] Microsoft (WPF). Preuzeto sa: <https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf?view=vs-2019> [pristupljeno: svibanj 2020.]

## POPIS SLIKA

Slika 1. Primjer popunjenosti preko 75% kapaciteta .....	3
Slika 2. Shema Integriranog prijevoza putnika.....	4
Slika 3. Suprotstavljeni putevi vožnje .....	6
Slika 4. Dijagram toka za izračunavanje broja bodova za svaki vlak .....	8
Slika 5. način na koji je nastalo logo C#.....	9
Slika 6. Definiranje ulaznih parametara u programskom jeziku C#.....	12
Slika 7. Definiranje bodova u programskom jeziku C#.....	15
Slika 8. Izrada aplikacije u C#.....	17
Slika 9. Praćenje izgleda aplikacije.....	18
Slika 10. Unošenje podataka za vlak 2001 .....	20
Slika 11. Unošenje podataka za vlak 8252 .....	21
Slika 12. Redoslijed vlakova za prvi primjer.....	22
Slika 13. Kolodvor Zagreb Klara u programu OpenTrack .....	23
Slika 14. Unošenje podataka za vlak 2056 .....	24
Slika 15. Unošenje podataka za vlak 45652 .....	25
Slika 16. Redoslijed vlakova za drugi primjer.....	26
Slika 17. Kolodvor Beli Manastir a u programu OpenTrack .....	27
Slika 18. Unošenje podataka za vlak 53123 .....	28
Slika 19. Unošenje podataka za vlak 51252 .....	29
Slika 20. Redoslijed vlakova za treći primjer.....	30

## POPIS TABLICA

Tablica 1. Vrijednosti za rang vlaka .....	13
--	----



Sveučilište u Zagrebu  
Fakultet prometnih znanosti  
10000 Zagreb  
Vukelićeva 4

## IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj diplomski rad  
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na  
objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz  
necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj  
visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu diplomskog rada  
pod naslovom **Definiranje parametara algoritma za dodjeljivanje prioriteta vlakovima  
u postupku automatskog postavljanja putova vožnje**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom  
repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 5/30/2020

\_\_\_\_\_  
(potpis)