VELEUČILIŠTE U RIJECI

Lea Šikić

TRAŽENJE OPTIMALNOG PUTA OD PAZINA DO RIJEKE BELLMAN-FORDOVIM ALGORITMOM

(specijalistički završni rad)

Rijeka, 2019.

VELEUČILIŠTE U RIJECI

Prometni odjel

Specijalistički diplomski stručni studij Promet

TRAŽENJE OPTIMALNOG PUTA OD PAZINA DO RIJEKE BELLMAN-FORDOVIM ALGORITMOM

(specijalistički završni rad)

MENTOR

Mr.sc Mirta Mataija, predavač

STUDENT

Lea Šikić

MBS: 2429000149/16

Rijeka, rujan, 2019.

VELEUČILIŠTE U RIJECI Prilog 1.

PROMETNI ODJEL

Rijeka, 3. travnja 2019.

ZADATAK
za specijalistički završni rad

Pristupniku: Šikić Lei **MBS: 2429000149/16**

Studentu specijalističkog diplomskog stručnog studija PROMETA izdaje se zadatak specijalističkog završnog rada – tema specijalističkog završnog rada pod nazivom:

TRAŽENJE OPTIMALNOG PUTA OD PAZINA DO RIJEKE BELLMAN-FORDOVIM ALGORITMOM

Sadržaj zadatka: Specijalistički završni rad sastoji se od teorijskog i praktičnog dijela. U teorijskom dijelu objasniti i definirati osnovne pojmove teorije grafova neophodne za rješavanje zadatka, odnosno nalaženje optimalnog puta od Pazina do Rijeke obzirom na dvije varijable: trošak i vrijeme. Posebno treba definirati i opisati transportnu mrežu Bellman-Fordov algoritam za traženje najkraćeg puta kroz mrežu od ishodišta. U praktičnom dijelu treba modelirati transportnu mrežu, objasniti metodu određivanja pondera, odrediti pondere, te naći optimalan put u zavisnosti o „težini" pojedine varijable. Posebno treba tretirati slučaj u kojem se koristi ENC uređaj, a posebno slučaj u kojem se isti ne koristi. U programskom paketu SageMath potrebno je napisati kod za traženje optimalnog puta u oba gore navedena slučaja.

Preporuka: Koristiti literaturu propisanu programom kolegija i proširenu u skladu s temom specijalističkog završnog rada, kao i vlastite izračune.

Rad obraditi sukladno odredbama Pravilnika o završnom radu Veleučilišta u Rijeci.

Zadano: 3. travnja 2019. **Predati do:** 15. rujna 2019.

Mentor:

mr. sc. Mirta Mataija

Pročelnik odjela:

dr. sc. Ivica Barišić

Zadatak primio dana: 3. travnja 2019.

Lea Šikić

Dostavlja se:
- **mentoru**
- **pristupniku**

Prilog 5.

I Z J A V A

Izjavljujem da sam specijalistički završni rad pod naslovom Traženje Optimalnog puta od Pazina do Rijeke Bellman-Fordovim algoritmom izradila samostalno pod nadzorom i uz stručnu pomoć mentora mr.sc. Mirta Mataija, predavač.

Lea Šikić

Šikić

(potpis studenta)

SAŽETAK

Transportna mreža može se prikazati pomoću težinskoga grafa. U težinskom grafu vrhovi predstavljaju gradove, raskršća, naplatne stanice, pristaništa, aerodrome, željezničke stanice i slično a bridovi, zajedno sa svojim težinama, ceste, ulice, vodne ili zračne putove, željezničke pruge i slično između danih čvorova. U ovom radu korištena je Bellman-Fordova metoda (napisan je kod u programu SageMath) kako bi se riješio problem pronalaska optimalnog puta između Pazina i Rijeke, s obzirom na dvije varijable: troškove i vrijeme. U prvom slučaju, pretpostavlja se da su troškovi važniji od vremena. U drugom slučaju vrijeme i troškovi su jednako važni. Konačno, u trećem slučaju, pretpostavlja se da je vrijeme važnije od troškova. Važnost promatranih varijabli ugrađena je u težine danih bridova. Promatrane su tri moguće rute između Pazina i Rijeke. Prva kroz tunel Učka, druga preko Učke, kroz Veprinac i treća uz more, kroz Plomin i Opatiju. Budući se kroz tunel Učka plaća cestarina u obzir su uzete dvije opcije: sa ili bez ENC uređaja. U radu je pokazano da ne postoji jedinstven optimalan put između Pazina i Rijeke. Put se mijenja ovisno o preferencijama putnika, odnosno je li putniku važnije stići brže ili uz manje troškove. Kao jedan od ciljeva rada, pokazana je i koralacija matematičkih metoda i prometne struke.

Ključne riječi: Bellman-Fordov algoritam, težinski graf, Pazin-Rijeka

SADRŽAJ

1. UVOD

1.1 Problem i predmet istraživanja

Predmet istraživanja ovog završnog rada je pronalazak optimalnog puta od Pazina do Rijeke u ovisnosti o vremenu i novcu. Naime, navedena relacija predmet je dnevnih migracija za veliki broj putnika (posao, škola, fakultet...). Za potrebe toga uzete su u obzir tri varijante, odnosno tri dionice kojima je moguće doći od Pazina do Rijeke. Kao metoda za izračun optimalnog (najkraćeg) puta izabran je Bellman-Fordov algoritam. Sukladno tome problem i predmet istraživanja ovog rada osim konkretnog izračuna predstavlja i osnova samog istraživanja, odnosno teorija grafova, te sam izbor metode za pronalazak najkraćeg puta.

Kao varijable su uzete samo vrijeme i novac, kao nekakvi osnovni parametri koje putnici i sami uzimaju u obzir. Važno je naglasiti da su ti parametri, a posebno cijena podložni promjenama. Naime cijena goriva nije fiksna već se mijenja. Isto tako u današnje vrijeme velike su razlike u potrošnji goriva kod pojedinih automobila (noviji-stariji, benzin-dizel-plin). Navedeno ne treba predstavljati ograničavajući čimbenik prilikom tumačenja dobivenih rezultata iz razloga što je koncept rada takav da je primjenjiv na različite slučajeve ukoliko su korisniku poznati ulazni podaci (cijena, vrijeme), te je lako napraviti izračune za svaki pojedini slučaj. Vrijeme isto tako može biti diskutabilan pojam, vozi li nektko sporije od ograničenja brzine ili pak brže. Tu su i vremenski uvjeti o kojima ovisi brzina vožnje.

1.2 Svrha i ciljevi istraživanja

Osnovna svrha ovog istraživanja je primjeniti Bellman-Fordov algoritam na navedeni problem pronalaska optimalnog puta od Pazina do Rijeke za tri uobičajene rute kojima se putnici najčešće služe. Da vi to bilo moguće, prije toga cilj ovog istraživanja je utvrditi usku povezanost teorije grafova i stvarne primjene u prometnoj struci. Prema tome istražiti teoriju grafova od samih početaka do metoda optimizacije. Isto tako istražiti koje su neke od metoda optimizacije, te zbog usmjerenog grafa topološkog uređenja u konačici primjeniti Bellman-Fordov algoritam za pronalazak najkraćeg puta u mreži.

Isto tako, svrha ovog istraživanja je utvrditi pri kojim vrijednostima varijabli vremena i novca se optimalni put mijenja. Uz to, cilj istraživanja je prikazati korelaciju Kvantitativnih metoda i same prometne struke kroz konkretan primjer iz prakse. U konačnici cilj je rješenje navedenog problema dobiti izradom matematičkog modela odnosno težinskog grafa, te programiranjem u programu Sage Math.

Uz to ciljevi su napraviti težinski graf za navedene dionice kako bi svaki pojedinac mogao ukoliko želi jednostavno doći do podataka koji je put optimalan u njegovom slučaju.

1.3 Struktura rada

Rad je strukturiran u sedam poglavlja s time da je prvo poglavlje uvod, a posljednje zaključak, U uvodu su naznačeni problem i predmet istraživanja, te svrha ciljevi. Drugo poglavlje odnosi se na teoriju grafova, uz dva potpoglavlja koji se odnose na Eulerove i Hamiltonover grafove. Treće poglavlje posvećeno je algoritmima za pronalazak najkrećg puta u mreži. Četvrto poglavlje, centralno je poglavlje istraživanja gdje se iznosi opis problema uz metode izračuna, te same izračune. Peto poglavlje je izlaz koda, odnosno dobiveni rezultati. Šesto poglavlje posvećeno je općoj primejni navedenog problema. Posljednje poglavlje je zaključak u kojem se sumiraju temeljna razmatranja. Na kraju rada nalazi se popis literature koja je poslužila izradi istoga, te popis slika i tabela.

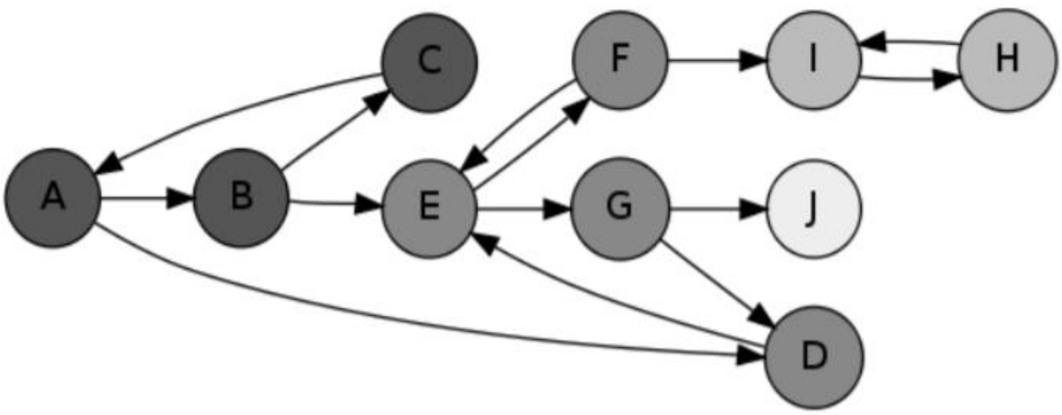
2. TEORIJA GRAFOVA

Teorija grafova jedna je od grana matematike. Ima široku primjenu prilikom rješavanja različitih problema, te isto tako ima primjenu u prometnoj struci.

Grafom možemo opisivati modele određenoga realnog sustava, kao što su gradovi povezani cestama, rafinerije i potrošači spojeni naftovodima, plinovodi, električni sklopovi i sl.
(https://www.veleri.hr/?q=system/files/nastavni_materijali/k_poduzetnistvo_s1/Kvantitativne_za_poduzetnike_Pr4_Izv.pdf)

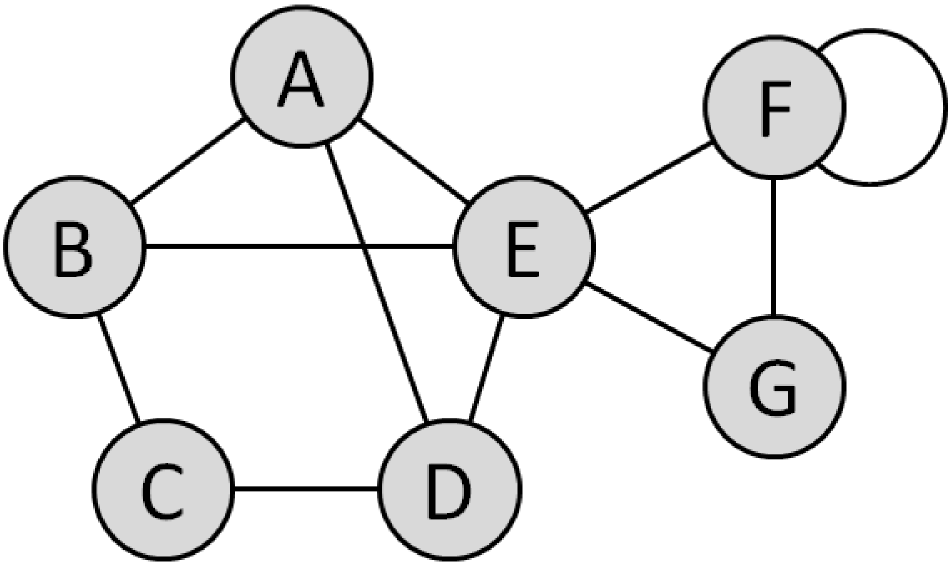
Sam pojam grafa može se definirati kao: "matematička struktura koja se koristi za opisivanje relacija između dvaju objekata iz određene kolekcije. (...) graf se odnosi na neprazni skup vrhova i kolekciju bridova koji povezuju par vrhova. Skup vrhova obično se označava s V, a skup bridova s R. Bridovi mogu biti usmjereni ili ne, ovisno o primjeru. Graf kojemu su svi bridovi usmjereni zovemo usmjerni graf (slika 1), u suprotnom, zovemo ga neusmjereni. U pravom grafu, za koji inicijalno pretpostavljamo da je neusmjeren, linija od točke do točke identificira se s linijom od točke do točke . U digrafu (skraćeno za usmjereni graf), ta dva smjera smatraju se različitim lukovima, odnosno bridovima. Ako graf nije usmjeren, tada su dva vrha pridružena jednom bridu međusobno ravnopravna. U usmjerenom grafu brid može biti usmjeren od jednog vrha prema drugome.(...) Brid koji počinje i završava u istom vrhu zove se petlja (slika 2). Brid je višestruk ako postoji drugi brid kojemu odgovaraju isti vrhovi (kao početni i krajnji vrh)."(Fošner, Kramberger)

Slika 1: Usmjereni graf



Izvor: https://www.fer.unizg.hr/_download/repository/Osnovni_pojmovi-teorija_grafova.pdf

Slika 2: Graf s petljom



Izvor: https://skolakoda.org/teorija-grafova

Graf G=(V, R) je neorijentirani ili neusmjereni graf, ako je relacija R simetrična, tj. ako vrijedi:

$$(v_i , v_j) \in R \Rightarrow (v_j , v_i) \in R$$

Graf G = (V, R) je orijentirani ili usmjereni graf ako je relacija R antisimetrična, tj. ako vrijedi:

$$(v_i ,v_j) \in R \Rightarrow (v_j ,v_i) \notin R$$

(https://www.veleri.hr/?q=system/files/nastavni_materijali/k_poduzetnistvo_s1/Teorija_grafova_skraceno.pdf)

Za grafove koji nisu ni orijentirani ni neorijentirani kaže se da su mješoviti. Kod mješovitoga grafa, između dva vrha mogu postojati kako jednosmjerne, tako i dvosmjerne veze. Jednosmjerne se veze uvijek crtaju strelicom, dok se dvosmjerne veze mogu crtati na različite načine:

- samo linijom bez strelica,
- linijom sa strelicama s obje strane,
- dvjema linijama sa suprotno orijentiranim strelicama.

(https://www.veleri.hr/?q=system/files/nastavni_materijali/k_poduzetnistvo_s1/Teorija_grafova_skraceno.pdf)

Težinski graf ili mreža je prošireni graf na kojemu se svakoj grani dodaje težina ili ponder. Ponderi su realni nenegativni brojevi pridruženi svakoj grani (cij – troškovi grane, uij – kapacitet grane, vrijeme, udaljenost.)
(https://www.weboteka.net/fpz/Osnove%20prometnog%20inženjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf)

Kao što će u narednim poglavljima ovog rada biti vidljivo, ponder ne mora biti jednostavno određen samo troškom ili vremeniom, već može biti računski dobiven vlastitim izračunima prema relevantinim parametrima kojima se žele dobiti određni izlazni podaci. Prije izračuna potrebno je iznijeti još nbeke definicije vezano za grafove.

Put je je niz grana grafa s osobinom da je kraj k-te grane u nizu, početak iduće k+1 grane. Put je niz grana koje su međusobno povezane. Počinje i završava vrhom. Duljina puta jednaka je broju lukova/grana u putu koji određuju putčine putanju, odnosno broj vrhova u putu umanjen za 1.

Ukoliko se pojedine grane pojavljuju u putu samo jedanput, put se naziva prostim (simple path). Elementarni put je put put koji najviše jednom prolazi kroz svaki vrh/čvor grafa, svi vrhovi u nizu koji opisuju put su različiti (svaki čvor u putu pojavljuje se samo jedanput).
(https://www.weboteka.net/fpz/Osnove%20prometnog%20inženjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf)

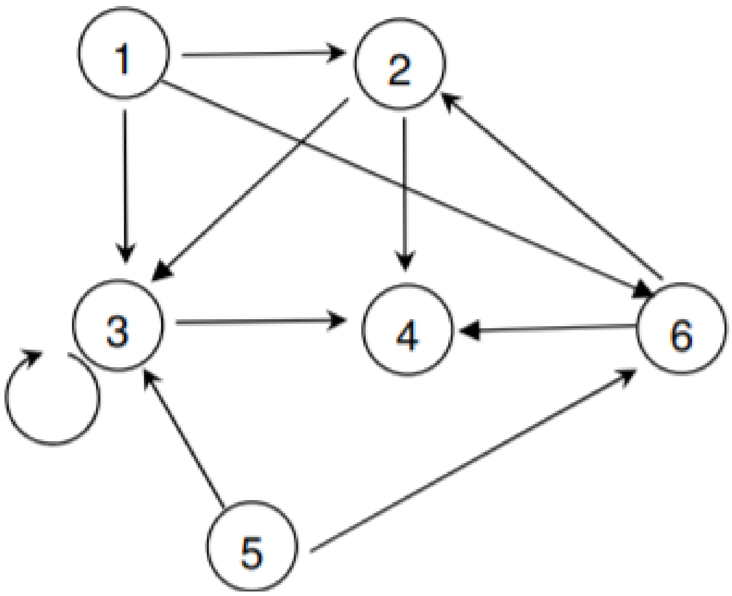
Graf se može zadati na tri načina: zadavanjem skupa vrhova i skupa bridova, crtanjem i matricom incidencije vrhova.

Primjer zadavanja grafa zadavanjem skupa vrhova i skupa bridova je:

Graf G = (V, R) zadan s V = {1, 2, 3, 4, 5, 6} i R = {(1,2), (1, 3), (1, 6), (2, 3), (2, 4), (3, 3), (3, 4), (5, 3), (5, 6), (6, 2), (6, 4)}.

Primjer zadavanja grafa crtanjem (slika 3):

Slika 3: Zadavanje grafa crtanjem



Izvor: izrada autora

Treći način zadavanja je matricom incidencije vrhova (slika 4):

Slika 4: Matrica incidencije vrhova

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Izvor: izrada autora

2.1 Eulerovi grafovi

Jedan od najpoznatijih problema vezanih za teoriju grafova svakako je zadatak o kenigsberškim mostovima.

Riječ je o gradu Kenigsbergu, odnosno današnjem Kalinjingradu. Problem se sastoji u tome da gradom teče rijega Pregel na kojoj se nalaze dva otoka, a njih sa kopnom i međusobno povezuje sedam mostova (slika 5), te je cilj pronaći rutu da se prođe svih sedam mostova, bez da se bilo kojim od njih prođe više od jedanput.

Slika 5: Prikaz kenigsberških mostova

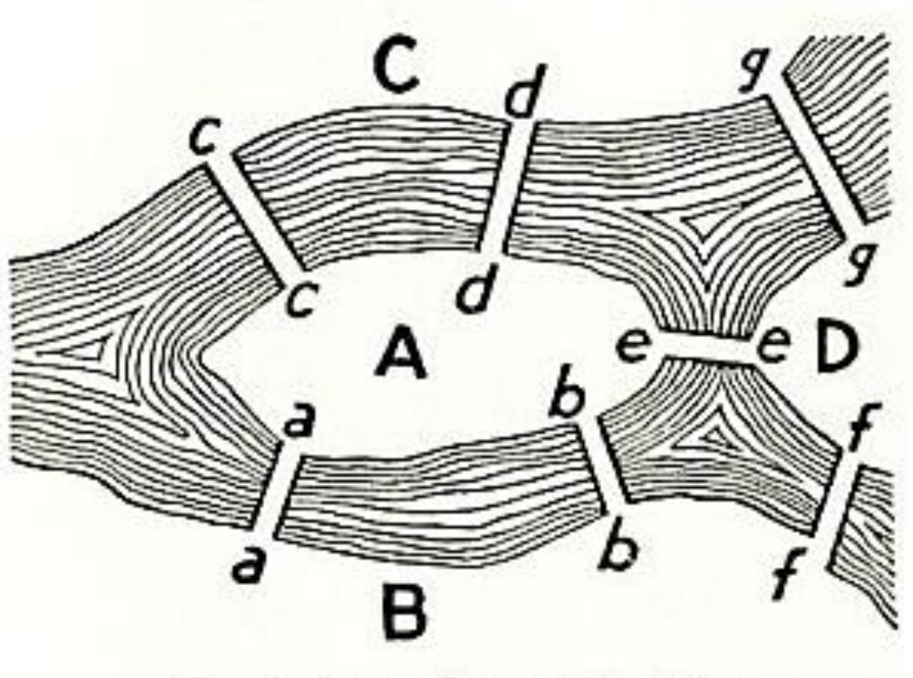


FIGURE 98. *Geographic Map: The Königsberg Bridges.*

Izvor: http://www.matheory.info/konigsberg/

Navedeno pitanje postavio je Leonhard Euler (slika 6), te isto tako pronašao i odgovor – da to nije moguće. (http://www.matheory.info/konigsberg/).

Euler je problem riješio na slijedeći način: promatrao je niz slova A, B, C, D koji predstavljaju kopno (otoke i obale) . Konstruirao je model račvanja da dokaže da ne postoji takav niz koji zadovoljava tražene uvjete, time dokazujći da problem Kenigsberških mostova nema rješenje.(Kopić, Klobučar, 2007:34)

To je zapravo dovelo do poečtka razvoja teorije grafova.
(http://mathworld.wolfram.com/KoenigsbergBridgeProblem.html)

Slika 6: Leonhard Euler

Izvor: https://www.biography.com/scientist/leonhard-euler

Prema tome, to je odrednica po kojoj je točno vidljivo kad je zasnovna navedena grana matematike. Bilo je to 1736. kad je objavljen Eulerov članak naziva:“ L. Euler, Solutio problematis ad geometriam situs pertinentis (Rješenje jednog problerna u svezi s geometrijom položaja), Comm. Acad. Sci. Imp. Petropolitanae 8 (1736), 128-140.“

Pri rješenju problema Euler je formirao tri glavna zaključka, a to su:

- Ako je bilo koje kopno (obala ili otok) povezano s nekim drugim kopnom neparnim brojem mostova, tada kružno putovanje koje prelazi svaki most točno jedanput nije moguće
- Ako je broj mostova neparan za točno dva kopna, tada je putovanje koje prelazi svaki most točno jedanput moguće samo ako putovanje počinje u jednom, a završava u drugom kopnu s neparnim brojem mostova.
- Ako nema kopna povezanog s neparnim brojem mostova putovanje može početi iz bilo kojeg kopna i završiti u tom istom kopnu.

(Kopić, Klobučar, 2007:35)

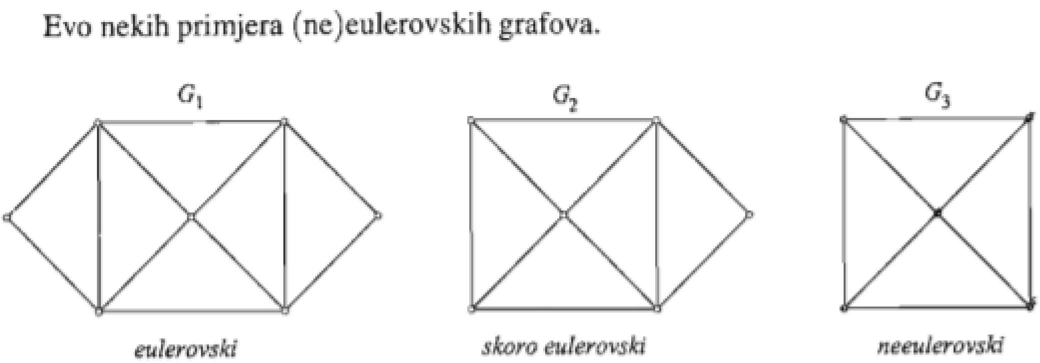
Eulerom je dakle započela teroja grafova, a važno je naglasiti da postoje i grafovi koji su naziv dobili upravo po njemu- riječ je o Eulerovim grafovima.

„Za povezani graf G kažemo da je eulerovski, ako postoji zatvorena staza koja sadrži svaki brid od G. Takvu stazu zovemo eulerovska staza. Neeulerovski graf je skoro eulerovski (semi-eulerovski), ako postoji staza koja sadrži svaki brid od G.“ (Osvin-Pavečević, 2006.)

Još jednostavnija definicija je da je: „Eulerov graf je graf koji se može nacrtati ne podižući olovku s papira.“ (https://www.weboteka.net/fpz/Osnove%20prometnog%20inženjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf)

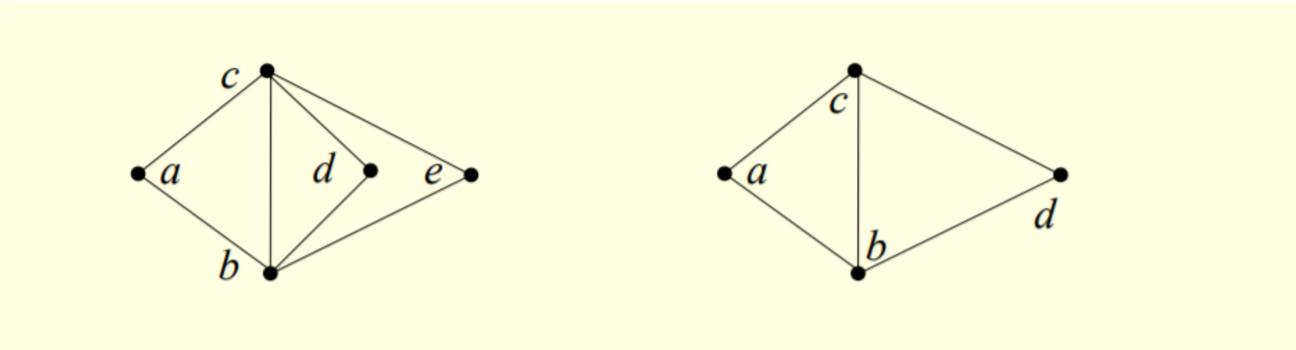
U nastavku (slika 7; slika 8) se nalaze neki od promjera eulerovskih, skoro eulerovskih i neeulerovskih grafova.

Slika 7: Primjeri



Izvor: Osvin-pavčević M, Uvod u teoriju grafova, 2006.

Slika 8: eulerov graf i neeulerov graf



Izvor: https://www.weboteka.net/fpz/Osnove%20prometnog%20inženjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf

2.2 Hamiltonovi grafovi

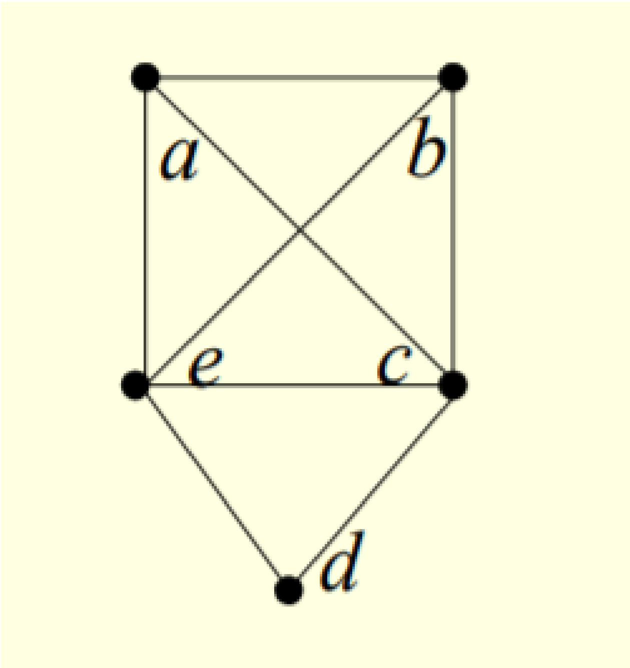
Problem obilaska svih vrhova grafa prvi je razmatrao irski matematičar William R. Hamilton. (slika 9)

Slika 9: William R. Hamilton

Izvor: https://en.wikipedia.org/wiki/William_Rowan_Hamilton

1856. godine postavio je problem trgovačkog putnika – trgovački putnik želi obići neke gradove i vratiti se na mjesto polaska tako da svakim gradom prođe samo jednom (i svakom cestom najviše jednom). Taj je problem postavio na grafu dodekaedra (pravilni poliedar s 12 strana koje su pravilni peterokuti). Graf dodekaedra je Hamiltonov graf, kao i grafovi svih ostalih pravilnih poliedara. Hamiltonov put na grafu G je put koji sadrži sve vrhove od G. Dakle, to je razapinjući put. Hamiltonov ciklus na grafu G je ciklus koji sadrži sve vrhove od G. (http://www.math.uniri.hr/~ajurasic/D-OSMO.pdf)
U nastavku se nalazi primjer Hamiltonovog grafa (slika 10).

Slika 10: Primjer Hamiltonovog grafa



Izvor:
https://www.weboteka.net/fpz/Osnove%20prometnog%20in%C5%BEenjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf

2.3 Transportna mreža

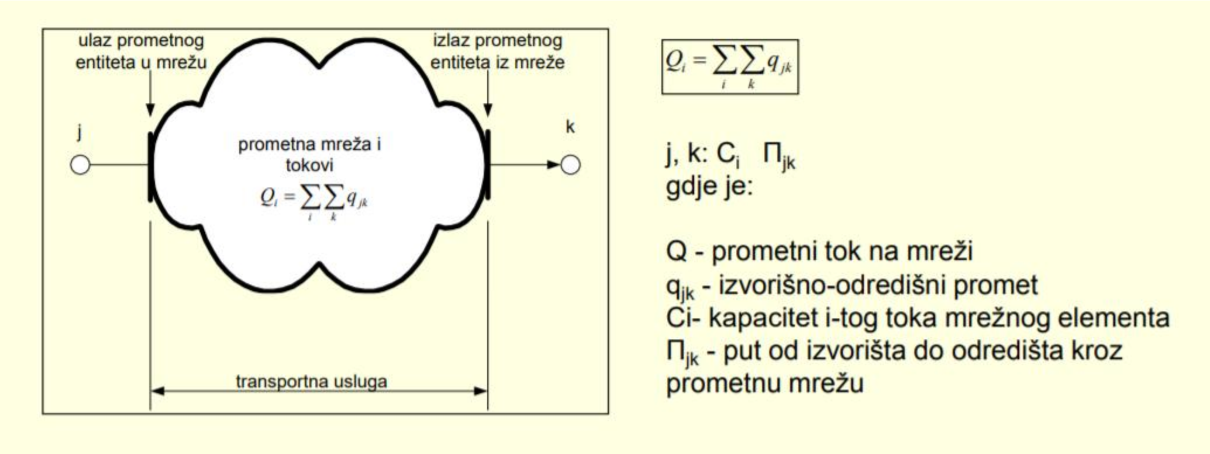
Teorija grafova ima široku primjenu u prometnij struci. Prema jednoj od definicija transportna mreža može se definirati kao: „prostorno distribuiran sustav na kojemu se odvijaju prometno - transportni procesi“ (Šimunović, 2015.).

Nadalje, temeljna funkcija mreže je omogućiti sigurno, učinkovito, ekološki i troškovno prihvatljivo premještanje ljudi, roba i informacija od izvorišta j do odredišta k.

Transportni entiteti ulaze na pristupnom dijelu mreže i izlaze na odredišnom dijelu. (slika 11)
(Šimunović, 2015.)

Na slici je viljivo da se prometna mreža sastoji od ulaza prometnog entiteta u mrežu, izlaza prometnog netitetat iz mreže, a uzlaz i izlaz zapravo povezuje prometna usluga.

Slika 11: Prometna mreža



Izvor: Šimunović, 2015.

Prema mediju putovanja prometne mreže mogu se podijeliti na:

- cestovne prometne mreže
- željezničke prometne mreže
- vodne prometne mreže
- zračne prometne mreže
- telekomunikacijske prometne mreže
- „posebne“ prometne mreže

(http://e-student.fpz.hr/Predmeti/O/Osnove_prometnog_inzenjerstva/Materijali/2011-OPI-Predavanje_009.pdf)

Svaku mrežu možemo matematički opisati preko teorije grafova i analitički preko matrica. Grafom se mogu opisivati realni sustavi, kao što su gradovi povezani cestama, rafinerije i potrošači spojeni naftovodima, plinovodi, električni sklopovi, ali i apstraktni objekti kao što su: baze podataka, tok računalnog programa, prikaz aktivnosti u projektu, socijalni odnosi, hijerarhija u radnoj organizaciji. Teorija grafova jedna je od grana

matematike koja nalazi veliku primjenu u prometnim znanostima, elektrotehnici, ekonomiji. Glavni razlog za široku primjenu teorije grafova je u jasnom geometrijskom predstavljanju mreže pomoću crteža ili grafa.
(https://www.weboteka.net/fpz/Osnove%20prometnog%20inženjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf)

Konkretno, u ovom radu jasno je vidljiva primjena prikaza stvarne prometne mreže cesta pomoću težinskog grafa gdje su vrhovi gradovi, a bridovi ceste koje ih povezuju. Za dobivanje težinskog grafa bridovima je pridružena vrijednost pondera dobivena računskim putem.

3. ALGORITMI ZA PRONALAŽENJE NAJKRAĆIH PUTEVA U MREŽI

Algoritam kao termin nije sveprisutan u svakodnevnom govoru, ali je svakako prisutan u životu.

Prema jednoj od definicija algoritam je: „je konačan slijed dobro definiranih naredbi za ostvarenje zadatka, koji će za dano početno stanje terminirati u definiranom konačnom stanju.“

(https://hr.wikipedia.org/wiki/Algoritam)

Nadalje, konkrteno u slučajevima u nastavku prilikom traženja najkraćih puteva u mreži, algoritam predstvalja skup unaprijed definiranih koraka koji vode do rješenja.

Algoritmi imaju sljedeća svojstva:

- **diskretnost** – u odvojenim koracima izvode se diskretne operacije algoritma koje vode ka konačnom cilju;
- **konačnost** – označava sposobnost algoritma da nakon konačnog broja koraka daje izlazne podatke odnosno rezultate;
- **determiniranost** – za iste ulazne podatke algoritam uvijek daje iste rezultate
- **masovnost** – algoritam je primjenjiv na veći broj ulaznih vrijednosti.

Algoritme je moguće klasificirati po raznim kriterijima:

Klasifikacija prema implementaciji

Jedan način klasifikacije algoritama je prema načinu implementacije:

- **Rekurzivni** ili **iterativni**: Rekurzivni algoritam je algoritam koji poziva samog sebe sve dok se ne postigne određen uvjet. Rekurzivni algoritmi su vrlo često usko vezani uz implementaciju pojedine matematičke funkcije na primjer Fibbonačijeve funkcije. Iterativni algoritmi su algoritmi koji ne pozivaju samog sebe već se oslanjaju na konstrukte poput petlji i dodatne strukture podataka kao što je stog ili red da bi riješili problem. Važno je napomenuti da je svaki rekurzivni algoritam moguće pretvoriti u iterativni, i da je svaki iterativni algoritam moguće pretvoriti u rekurzivni, iako ponekad pretvaranje može biti vrlo kompleksno.

- **Serijski** ili **paralelni**: Većina današnjih računala sadrži samo jedan procesor te stoga obavlja naredbe jednu po jednu, to jest serijski. Algoritmi koji su dizajnirani sa namjerom da se izvršavaju u takvom okruženju shodno tome se nazivaju serijski algoritmi. Suprotno njima su paralelni algoritmi koji sa sve većim probojem višeprocesorskih računala dobivaju sve veću važnost. Paralelni algoritmi koriste mogućnost višeprocesorskog sustava na taj način da problem rascijepe na više malih potproblema koje svaki procesor rješava zasebno te se zatim rezultati spajaju. Paralelni algoritmi uz resurse potrebne za obradu podataka također imaju i malu potrošnju resursa na komunkaciju između više procesora. Algoritmi za sortiranje su jedan od primjera algoritama koje je moguće znatno poboljšati upotrebom paralelnih procesora, dok je probleme poput problem tri tijela sasvim nemoguće riješiti paralelnim algoritmom.

- **Deterministički** ili **stohastički**: Deterministički algoritam je algoritam koji će pri svakom izvršavanju u bilo kojim uvjetima od istog unosa doći do istog izlaza sljedeći svaki put identičan niz naredbi. Stohashički algoritmi je algoritam koji barem u jednom dijelu izvršavanja donese neku odluku slučajnim odabirom.

- **Točan** ili **približan**: Iako algoritmi u principu daju točan rezultat, ponekad algoritam traži približno rješenje koje je dovoljno blizu točnom, ili je točno rješenje nemoguće naći.

Algoritmi s obzirom na metodologiju dizajna:

Brute force algoritmi - "čistom silom" računala isprobavaju sve mogućnosti i traže odgovarajuće rješenje. Najneefikasnijji algoritmi.

Podijeli i vladaj algoritmi tzv. Divide and conquer, podijeli i vladaj. Problem se dijeli na više istih, manjih problema. Podjela teče tako dugo dok se ne dođe do malog problema kojeg je jednostavno riješiti (obično rekurzijom).

Dinamički algoritmi - Metodama dinamičkog programiranja rješavaju se višefazni procesi, tj. procesi u kojima se donosi niz međusobno ovisnih odluka za pojedine godine određenog razdoblja ili za pojedine aktivnosti zadanog problema. Dinamičko programiranje poznato je i pod nazivom metoda donošenja višefaznih, odnosno višestupnjevanih odluka.

Pohlepni algoritmi tzv. greedy – Pohlepni algoritam je algoritam koji se koristi metaheuristiku za rješavanje problema, takvu da u svakom koraku bira lokalno najbolje rješenje, u nadi da će tako iznaći globalni optimum. Ovi algoritmi često ne daju najbolje rješenje već brzu aproksimaciju najboljeg rješenja.

Algoritmi za sortiranje i pobrojavanje tzv. search and enumeration - Algoritmi sortiranja služe za brzo sortiranje podataka, npr. niza brojeva. Mnogi se problemi mogu rješavati teorijom grafova.

(https://hr.wikipedia.org/wiki/Algoritam)

3.1 Bellman-Fordov algoritam

Za potrebe ovog završnog rada, obzirom da je graf topološkog uređenja izabran je Bellman-Fordov algoritam za pronalazak najkraćeg puta (u ovom slučaju optimalnog puta).

Prema Bellmanu: „Navedeni algoritam primjenjuje se za pronalazak najkraćeg, odnosno optimalnog puta u mreži koja se sastoji od N gradova (vrhova) proizvoljno numeriranih od 1 do N gdje su svaka dva povezana direktnom cestom. Vrijeme potrebno za putovanje od grada (vrha) i do grada (vrha) j nije direktno proporcionalno udaljenosti između i i j uslijed stanja ceste i prometa. U danoj matrici T=(t i,j), koja ne mora nužno biti simetrična, t i,j je vrijeme potrebno za doći od grada (vrha) i do grada (vrha) j. Potrebno je naći put za koji je potrebno najmanje vremena, odnosno najkraći put. Obzirom da je broj mogućih putova konačan, problem se svodi na izbor najkraćeg u konačnom broju putova.

Funkcionalna jednadžba aproksimacije glasi:

f_i –vrijeme potrebno za putovanje od i do N, i=1,2,…N-1, koristivši optimalan smjer

f_N =0

Upotrebnom optimalnog principa vidljivo je da f nelinearni sustav jednadžbi

f_i =Min **[**$\mathbf{t_{ij}}$ +f_i**], i=1,2…,N-1**

j<i

$\mathbf{f_N}$ =0“

(Bellman, 1958)

Isto tako, prema Fordu, Jr.: „Mreža se sastoji od točaka odnosno čvorova koji međusobno mogu biti povezani lukovima. P_i predstavljaju vrhove, dok l_{ij} predstavljaju cijenu ili udaljenost odnosno pondere.

Za pronalazak najkraćeg puta procedura računanja je sljedeća: početno se dodijeli $x_0=0$ i $x_i=\infty$, za $i\neq0$, pretražite mrežu za par P_i i P_j sa svojstvom da $x_i-x_j>l_{ji}$. Za taj par zamijenite x_i s x_j+l_{ji}. Nastavite taj proces. Na kraju kad se takvi parovi više ne mogu pronaći i x_N je minimalan i predstavlja minimalnu udaljenost od P_0 do P_N očito je da je najkraći put pronađen.

Dokaz optimalnosti: neka su P_0, $P_{i1,\ldots}P_{im,}$ P_N najkraći putevi. Na ovom putu $x_{i\ k+1}-x_{i\ k}\leq l_{i\ k}i_{k+1.}$ Sumiranjem tih relacija daje se $x_N\leq$ (duljina najkraćeg puta). S druge strane, za svaki P_j ($j\neq0$) postoji P_i s $x_i+l_{ij}=x_j$, $l_{ij}\neq0$.

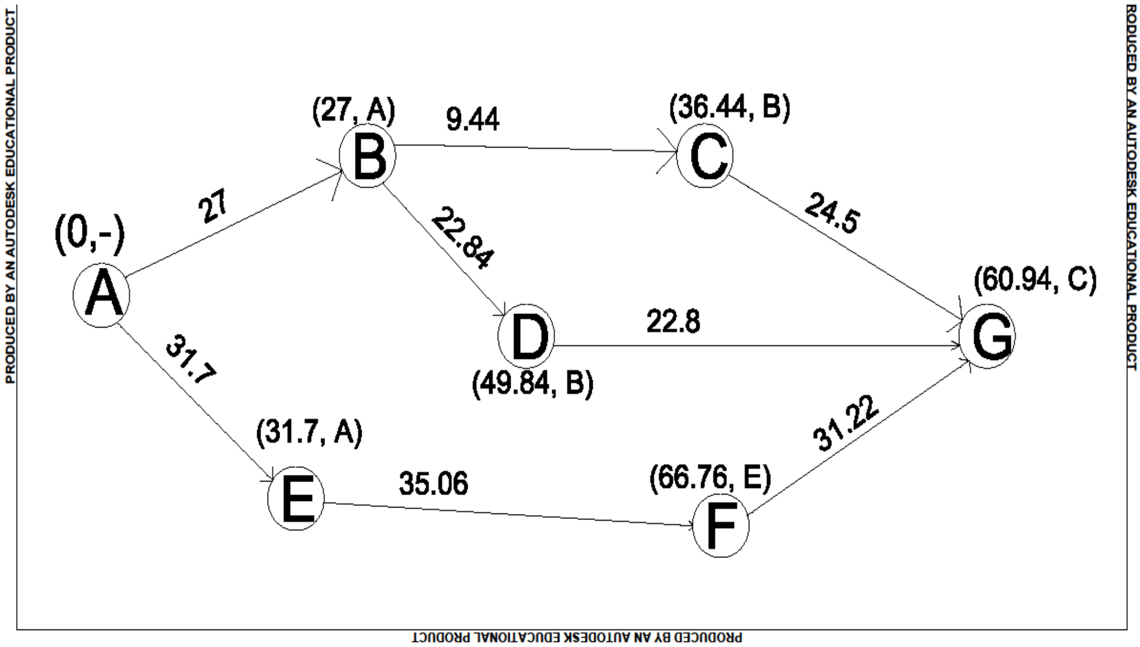
Za slučaj da je $j\neq0$, x_j je počeo s ∞ i smanjuje se monotono. Ako je x_j i dalje ∞, gotovo je, ako nije, na zadnjem smanjenjuju je bio i koji i dalje mora biti iste vrijednosti. Precrtavanjem tog lanca unatrag od P_N x_i se monotno striktno smanjuje; u konačnici je postignut početni P_0. $x_j-x_i=l_{ij}$ i zbrajanjem toga dobiva se x_N=duljina ovog lanca. Stoga to je nakraći put." (Ford 1956:9)

Kako je iz izvornika citirano, navedena metoda uspješno se riješava klasičnim putem-računskim izračunima.

Primjer klasičnog izračuna navodi se u nastavku.

Metoda je dakle, izrada težinskog grafa, te primjena Bellman-Fordovog algoritma.

Slika 12: Težinski graf



Izvor: izrada autora

Vidljivo je da je graf usmjeren, da se sastoji od sedam vrhova, te osam bridova koji ih povezuju. Cilj je pronaći najkraći put od vrha A do vrha G.

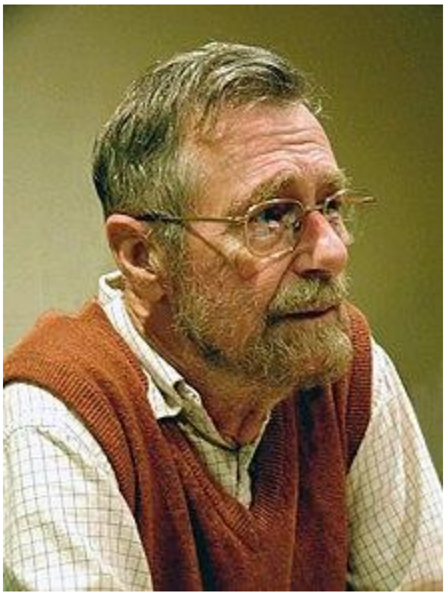
Počinje se u vrhu A iz kojeg se može u B ili u E, izabire se B zbog manje vrijednosti, zatim se iz B može u C ili u D, izabire se C zbog manje vrijednosti pondera. Pribraja se prethodna vrijednost pondera 27 novoj 9.44. Posljednji korak iz kojeg je vidljivo da se iz C može jedino u G dovodi do rješenja, ponavlja se postupak i pribraja vrijednost pondera. Jednostavnim očitavanjem sa grafa vrhova kojim se prošlo vidljivo je da je u ovom slučaju najkraći put A-B-C-G.

U današnje vrijeme problem se još lakše riješava uz pomoć računalnih programa.

3.2 Dijkstrin algoritam

Dijkstrin algoritam dobio je naziv po nizozemskom informatičaru Edsgeru Dijkstri.(slika 13)

Slika 13: Edsger Dijkstra



Izvor: https://en.wikipedia.org/wiki/Edsger_W._Dijkstra

Algoritam je razvijen 1959. i smatra se jednim od najefikasnijih i najpogodnijih algoritama za određivanje najkraćeg puta između dva zadana čvora. Neophodan uvjet za primenu ovog algoritma je da je dužina c_{ij} svake grane (i, j) ∈ L nenegativna. (http://www.math.rs/p/files/16-Operaciona_istra%C5%BEivanja_-_ve%C5%BEbe_CELOBROJNO_PROGRAMIRANJE_4.pdf)

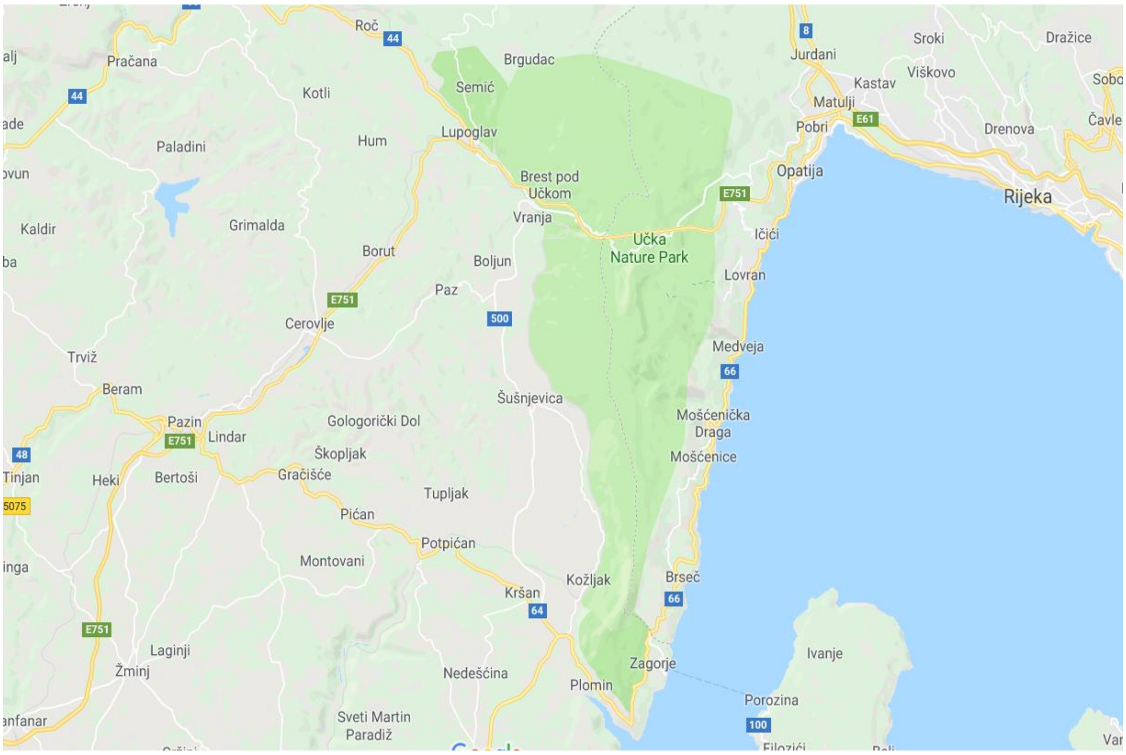
Algoritam postepeno kreira stablo od čvorova i pojedinih lukova u grafu pri čemu je polazni čvor s . Za svaki čvor u, pamti se pokazivač na prethodni (roditeljski) čvor p[u] i udaljenost (težina) koja se mjeri od polazišta (izvorišta) d[u]. Algoritam koristi red čekanja Q sa prioritetom u kojem su spremljeni neobrađeni čvorovi. Obrađeni čvorovi spremaju se u skup S.

U prvom koraku se vrši inicijalizacija – Postavljanje težine na 0 za polazni čvor s – Postavljanje težine na ∞ svim ostalim čvorovima – Postavljanje pokazivača na prethodni čvor na vrijednost null svim čvorovima – Svi čvorovi su spremljeni u redu čekanja Q, dok je skup S prazan. Čvorovi se izvlače iz reda čekanja Q po kriteriju najmanje težine i prebacuju u skup obrađenih čvorova S. Za svaki izvučeni čvor, relaksiraju se tj. ažuriraju težine susjednim povezanim čvorovima. (Carić, 2013.)

4. OPIS PROBLEMA

Iako postoji nebrojeno varijanti, odnosno puteva kojima je moguće doći od Pazina do Rijeke, za izradu ovog rada uzete su u obzir tri varijante koje se na prvi pogled čine najkraće i kojima se putnici uobičajeno služe (slika 14). Zbog mogućnosti plaćanja cestarine (tunel Učka) klasično i s ENC uređajem kojim se ostvaruje 50% niža cijena tunelarine, svi izračuni koji uključuju prolazak kroz tunel Učku napravljeni i za plaćanje s ENCom i klasično plaćanje.

Slika 14: Karta



Izvor: Google maps

Neke od tih opcija uključuju ravničarsku dionicu, neke brdsku te imaju različit izbor prometnica na kojima se odvijaju a time i izbor plaćanja ili neplaćanja cestarine što sve skupa utječe na izbor putovanja. Izbor među opcijama obuhvaća kriterije kao što su: potrošnja goriva, duljina puta te naplata korištenja cestovne infrastrukture. Sve tri opcije uzimaju u obzir da se na rutama ide osobnim automobilom na dizelski pogon, a cijena puta koja se u radu izračunava za sve tri rute uzima u obzir kao fiksnu cijenu litre Eurodisela 8,78 kn.

Prva opcija je i najčešći odabir putnika, a riječ je o putu kroz tunel Učka. Druga opcija je popularno zvana „preko Učke“, što znači kao što i sam naziv kazuje da se vozi od Pazina do Naplatne postaje Učka, a zatim zaobilazi tunel Učka, te preko Veprinca dolazi u Rijeku. Treća, i možda najmanje popularna opcija je od Pazina preko Plomina i Opatije, pa sve do Rijeke. Kroz naredna potpoglavlja biti će razjašnjeno koji je put optimalan ovisno o važnosti vremena i troškova.

4.1 Metoda izračuna

Brojčani podaci korišteni kao ulazni parametri za potrebe izračuna su: potrošnja goriva, duljina puta u kilometrima, vremenska udaljenost, te naplata korištenja cestovne infrastrukture.

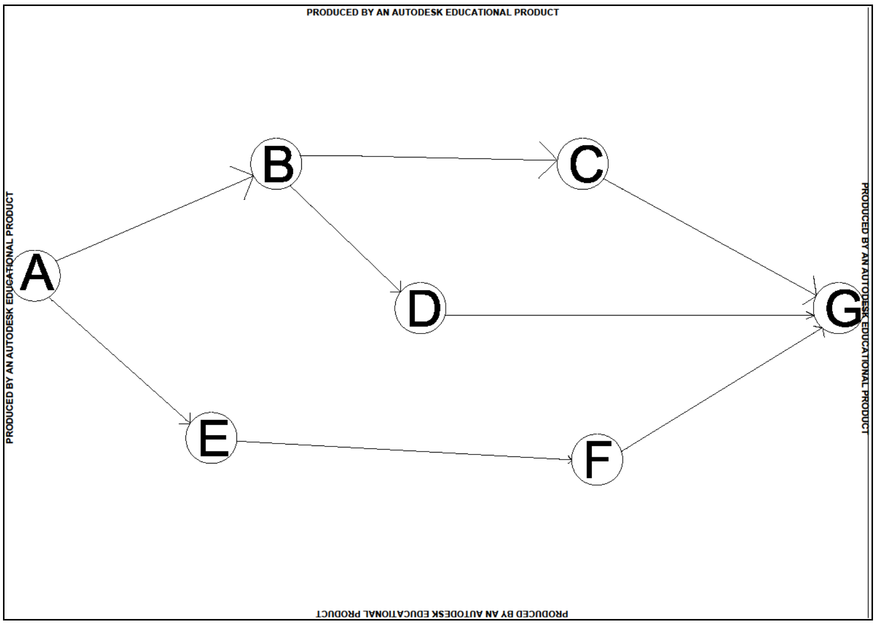
Uvažavajući definicije da je: „Težinski graf uređeni par (G, w), gdje je G = (V, E) graf i w : E → R + 0 funkcija. Nenegativan broj w(e) nazivamo težinom brida e. Težina zadanog brida može značiti visinu troška, udaljenost, vrijeme ili bilo koju drugu mjeru koja karakterizira taj brid.“ (Bosančić et al., 2012:141), te da su: „Vrhovi, odnosno čvorovi mrežni elementi u kojima se: koncentriraju, propuštaju i usmjeravaju, križaju, slijevaju ili odlijevaju prometni tokovi vozila (vlakova, zrakoplova, brodova, informacija, podatkovnih paketa), obavlja naplata karata, skladištenje, informiranje korisnika itd.

Čvorovi su gradovi, raskrižja, aerodromi, željeznički kolodvori, autobusni kolodvori, pošte, robni terminali, računala“ (Šimunović, 2015) u svrhu pronalaska optimalnog puta ovisno o indivudalnoj važnosti cijene i vremena nacrtan je težinski graf. Njegovi vrhovi su gradovi kojima se prolazi ranije navedenim dionicama i koji predstavljaju određenu prekretnicu u mreži, odnosno značajnu promjenu nadmorske visine, raskrižje i sl., dok vrijednost pondera predstavlja vremensku i prostornu udaljenost između čvorova, te se vrijednost pondera mijenja ovisno o varijablama. Naime, u ovom radu rješava se problem pronalaska optimalnog puta u ovisnosti o dvije varijable, a to su vrijeme i novac.

Uzevši u obzir činjenicu da ne postoji jedinstven optimalan put između Pazina i Rijeke, već se put se mijenja ovisno o preferencijama putnika, odnosno je li putniku važnije stići brže ili uz manje troškove s ciljem dobivanja tih podataka napisan je kod u programu Sage, pomoću kojeg je računana vrijednost pondera i dobiven optimalni put. Za izračun „najkraćeg“ puta u mreži korišten je Bellman-Fordov algoritam.

Sukladno tome u ovom radu sastavljena je mreža (slika 15) od čvorova koji predstavljaju gradove, bridova koji predstavljaju ceste koje povezuju te gradove, te pondera čija je vrijednost dobivena izlazom koda.

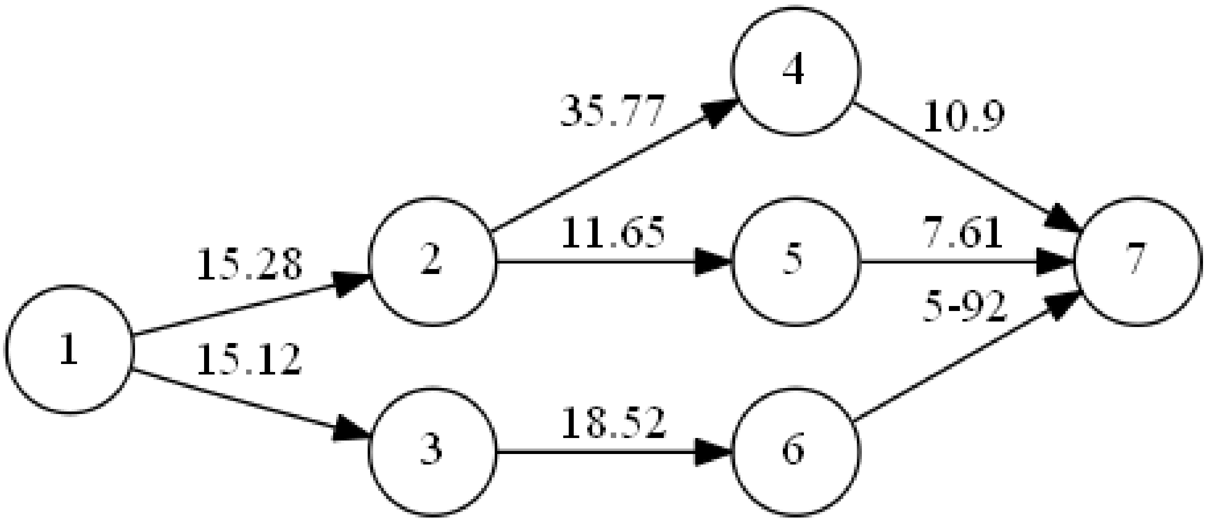
Slika 15: Mreža gdje čvorovi predstavljaju gradove, a bridovi ceste



Izvor: Izrada autora

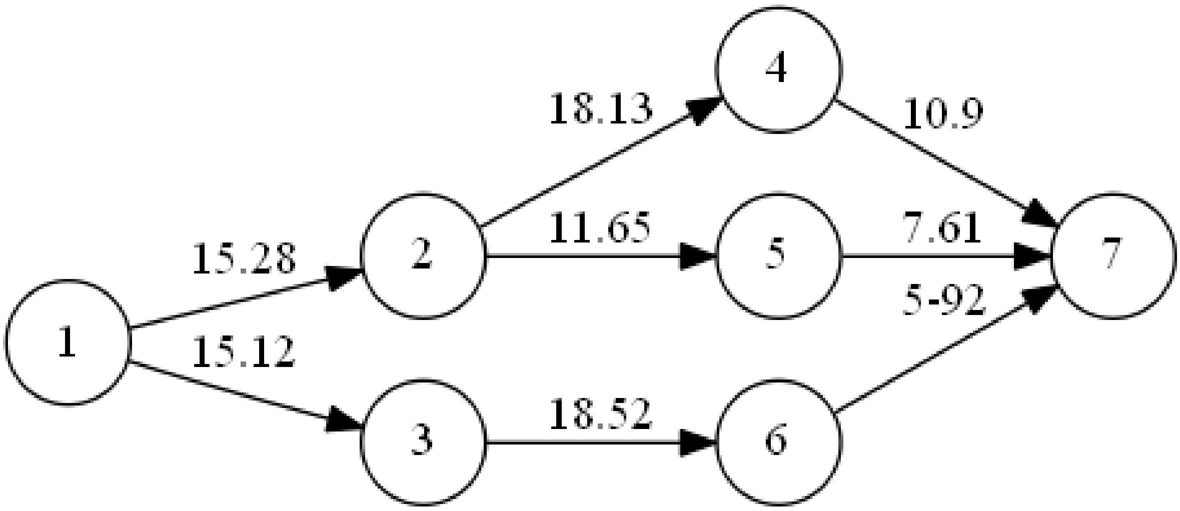
- A- Pazin
- B- Naplatna postaja Učka
- C- Tunel Učka (izlaz)
- D- Veprinac
- E- Plomin
- F- Opatija
- G- Rijeka

Slika 16: Cijena bez ENC-a



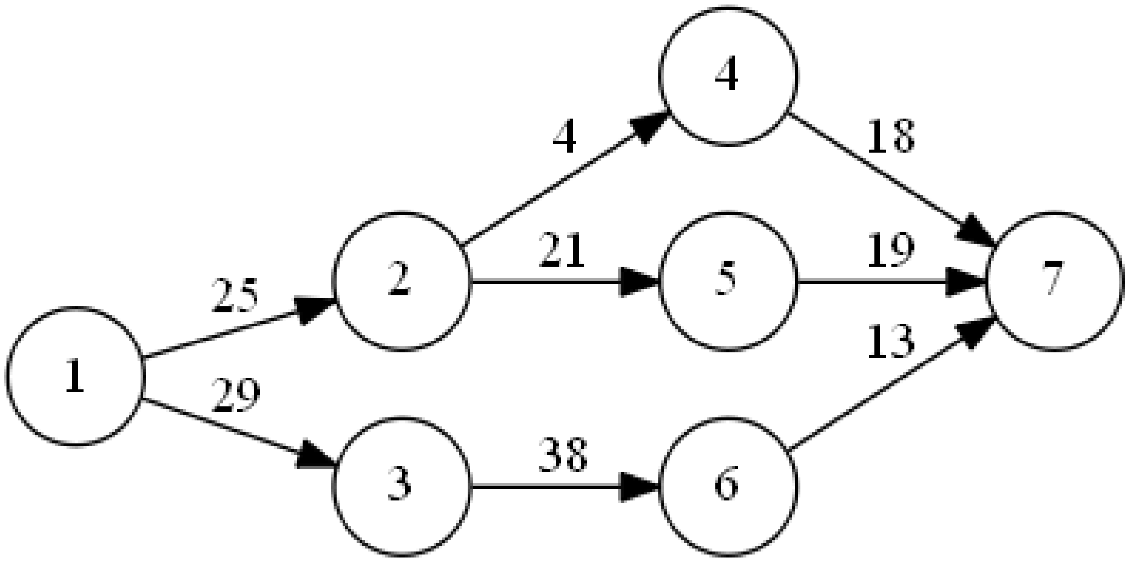
Izvor: izrada autora

Slika 17: cijena s ENC-om



Izvor: izrada autora

Slika 18: Vrijeme



Izvor: Izrada autora

Ulazne varijable koje su korištene u kodu, kao i način funkcioniranja koda biti će obrazloženi u narednim poglavljima.

Formula na kojoj se temelji vrijednost pondera je: $p = \frac{jN+kT}{100}$, $j + k = 100$, gdje j predstavlja koliko je putniku izraženo u postotku važno da put bude što jeftiniji, a varijabla k predstavlja u kojem postotku je važno da put bude vremenski što kraći.

Za vrijednost novca, odnosno cijene prilikom izračuna u obzir su uzeti podaci o prosječnoj potrošnji litre koriva na sto kilometara, povećanje potrošnje prilikom većeg uzdužnog nagiba ceste, cijena dizela prema trenutnim podacima, te naplate cestovne infrastrukture (tunelarina) na dionicama gdje se naplaćuje i udaljenost u kilometrima. Prema tome, novčana vrijednost dobivena je jednostavnim izračunom.

N-novac, odnosno cijena puta

x-udaljenost

y-prosječna potrošnja

c-cijena dizela

k-cijena cestarine

Podaci o vremenskoj i prostornoj udaljenosti koji su korišteni prilikom izračuna preuzeti su sa: https://www.google.com/maps

U nastavku su dani izračuni pojedinih dionica.

KROZ TUNEL

1l Eurodisela: 9.96 kn, cestarina: 33 kn (enc 15.36 kn),

Udaljenost: 27.9 km

Cijena goriva na djelu Pazin – naplatna postaja Učka: 0.279*5.5*9.96= 15.28

V=25 minuta.

Naplatna postaja Učka- Tunel Učka izlaz

Udaljenost: 5.062 km

0.05062*5.5*9.96=2.77

V=3.8 min

Cijena goriva na djelu Tunel Učka (izlaz) – Rijeka:

Udaljenost:24.4 km

0.244*5.5*9.96=13.37

V= 26 minuta.

PREKO UČKE

Udaljenost: 27.9 km

Cijena goriva na djelu Pazin – naplatna postaja Učka: 0.279*5.5*9.96= 15.28

V=25 minuta.

Naplatna postaja Učka-Veprinac

Daljenost: 13 km

0.13*9*9.96=11.65

V=21 min

Veprinac-Rijeka

Udaljenost: 19.4 km

0.194*5.5*9.96=10.63

V= 27 min

PLOMIN I OPATIJA

Cijena goriva na djelu Pazin – Plomin:

Udaljenost: 27.6 km

0.276*5.5*9.96=15.12

V=29 min

Cijena goriva na djelu Plomin – Opatija:

Udaljenost: 33.8 km

V=38 min

Cijena goriva na djelu Opatija – Rijeka:

Udaljenost: 13.7 km

0.137*5.5*9.96=7.5

V=23 min

4.2 SageMath

S ciljem pronalaska otimalnog puta primjenom Bellman-Fordovog algoritma napisan je kod u programu Sage Math. Sage Math je besplatan matematički softvet otvorenog koda s GPL licencom.

(http://www.sagemath.org/tour-research.html)

Kod najprije definira usmjereni graf - digraf H, zatim mu dodaje odgovarajuće vrhove - njih sedam. Zatim računa cijenu pojedine dionice pomoću ulaznih vrijednosti za udaljenost, potrošnju goriva i plaćanje cestarine. Uz to, ulazna varijabla je i vrijeme potrebno za prolazak pojedine dionice. Korišteni su podaci iz Tabele 1 i tabele 2.

Tabela 1: Podaci za izračun cijene puta s ENC-om

Dionica	Vrijeme	Udaljenost	Cijena Goriva	Cestarina	Potrošnja Goriva	Cijena Dionice
Pazin – Naplatna postaja Učka	25	27.9	15.28	0	5.5	15.28
Pazin – Plomin	29	27.6	15.12	0	5.5	15.12
Naplatna postaja Učka – Učka izlaz	4	5.062	2.77	33	5.5	35.77
Naplatna postaja Učka – Veprinac	21	13	11.65	0	9	11.65
Plomin – Opatija	38	33.8	18.52	0	5.5	18.52
Učka izlaz – Rijeka	18	19.9	10.9	0	5.5	10.9
Veprinac – Rijeka	19	13.9	7.61	0	5.5	7.61
Opatija – Rijeka	13	10.8	5.92	0	5.5	5.92

Izvor: Izrada autora

Tabela 2: Podaci za izračun cijene puta bez ENC-a

Dionica	Vrijeme	Udaljenost	Cijena Goriva	Cestarina	Potrošnja Goriva	Cijena Dionice
Pazin – Naplatna postaja Učka	25	27.9	15.28	0	5.5	15.28
Pazin – Plomin	29	27.6	15.12	0	5.5	15.12
Naplatna postaja Učka – Učka izlaz	4	5.062	2.77	15.36	5.5	18.13
Naplatna postaja Učka – Veprinac	21	13	11.65	0	9	11.65
Plomin – Opatija	38	33.8	18.52	0	5.5	18.52
Učka izlaz – Rijeka	18	19.9	10.9	0	5.5	10.9
Veprinac – Rijeka	19	13.9	7.61	0	5.5	7.61
Opatija – Rijeka	13	10.8	5.92	0	5.5	5.92

Izvor: izrada autora

Nakon što je izračunao cijenu pojedine dionice, kod računa pondere i to tako da krene od toga da je jedino važno što kraće vrijeme dolaska na cilj (j = 0, k = 100), te zatim kroz 101 prolaz kroz petlju računa vrijednosti pondera tako da u svakom pronalasku spusti za jedan parametar k, a parametar j poveća za jedan, odnosno smanjuje se važnost vremena, a povećava važnost novca - putniku je sve manje važno da čim prije dođe na cilj, a sve veću ulogu u putovanju imaju troškovi, odnosno putniku je cilj da put bude što jeftiniji. Za izračun optimalnog puta korišten je Bellman-Fordov algoritam iz razloga jer graf usmjeren i topološkog uređenja.

Obzirom da je cilj pronaći optimalni put ovisno o važnosti vremena i novca kroz povećavanje paramatra j i smanjivanje parametra k dobivaju se rezultati kako se optimalni put koji ovisi o navedenim parametrima mijenja, odnosno dobiva se postotak koji je prekretnica, kojim se optimalni put mijenja. Isto tako dobiveni su rezultati ukoliko su novac i vrijeme jednako važni. Prema tome, izlazom koda može se isčitati za svaki postotak od 0 do 100 za parametre j i k, koji je put optimalan. Izračuni su napravljeni za klsičnu naplatu cestarine i elektroničku. U nastavku se nalazi kod za izračun ukoliko se cestarina plaća klasično:

Kod za izračun ukoliko se plaća ENC-om nalazi se u nastavku:

```
sage: t=[25,29,4,21,18,19,38,13]                              #vrijeme
....: c=[5.5,5.5,5.5,9,5.5,5.5,5.5,5.5]               #potrosnja goriva
....: pt=[0,0,33,0,0,0,0,0]                        #cestarina
....: d=[27.9,27.6,5.062,13,19.9,13.9,33.8,10.8]   #udaljenost
....: pr=[]                                        #cijena dionice
....: p=[]                                          #ponder
....: H=DiGraph(weighted=True)                     #definiranje usmjerenog grafa l
....: H.add_vertices(range(6));H                              #dodavanje 7 vrhova graf
....: for n in range(8):                                          #for petlja - ra
....:        a=d[n]*c[n]*9.96/100+pt[n]
....:        pr.append(a) ....: print(pr)
....: for j in range(101):
#racunanje pondera od j=0, k=100, do j=100, k=0
....:        print(j)
....:        k=100-j
....:        p=[]
....:        for l in range(8):
....:              b=(j*pr[l]+k*t[l])/100
....:              p.append(b)
....:        H.add_edges([(0,1,p[0]),(0,4,p[1]),(1,2,p[2]),(1,3,p[3]),(2,6,p[4]),(
....:        H.shortest_path(0,6,by_weight=True,algorithm='Bellman-Ford_Boost') #r
```

```
sage: t=[25,29,4,21,18,19,38,13]                          #vrijeme
....: c=[5.5,5.5,5.5,9,5.5,5.5,5.5,5.5]           #potrosnja goriva
....: pt=[0,0,15.36,0,0,0,0,0]                           #cestarina
....: d=[27.9,27.6,5.062,13,19.9,13.9,33.8,10.8]  #udaljenost
....: pr=[]                                               #cijena dionice
....: p=[]                                                #ponder
....: H=DiGraph(weighted=True)                           #definiranje usmjerenog grafa
....: H.add_vertices(range(6));H                          #dodavanje 7 vrhova graf
....: for n in range(8):                                       #for petlja - ra
....:        a=d[n]*c[n]*9.96/100+pt[n]
....:        pr.append(a) ....: print(pr)
....: for j in range(101):
#racunanje pondera od j=0, k=100, do j=100, k=0
....:        print(j)
....:        k=100-j
....:        p=[]
....:        for l in range(8):
....:             b=(j*pr[l]+k*t[l])/100
....:             p.append(b)
....:        H.add_edges([(0,1,p[0]),(0,4,p[1]),(1,2,p[2]),(1,3,p[3]),(2,6,p[4]),(
....:        H.shortest_path(0,6,by_weight=True,algorithm='Bellman-Ford_Boost') #r
```

Ovo je izračun sa ENC-om.

Ovo je izlaz ovog koda:

```
j= 0 k= 100 [1, 2, 4, 7]
j= 1 k= 99 [1, 2, 4, 7]
j= 2 k= 98 [1, 2, 4, 7]
j= 3 k= 97 [1, 2, 4, 7]
j= 4 k= 96 [1, 2, 4, 7]
j= 5 k= 95 [1, 2, 4, 7]
j= 6 k= 94 [1, 2, 4, 7]
j= 7 k= 93 [1, 2, 4, 7]
j= 8 k= 92 [1, 2, 4, 7]
j= 9 k= 91 [1, 2, 4, 7]
```

Nameće se pitanje na koji način interpretirati postotke, odnosno na koji način pojedinac može zaključiti u kojem postotku mu je važno vrijeme, a u kojem postotku novac. U ovom izračunu točno definiranje postotka nije od presudne važnosti, naime putniku nema razlike npr. je li mu 65% ili 70% važnije da put bude što jeftiniji. Važno mu je jedino da zna koji je put optimalan ako mu je važniji novac ili vrijeme. Prema tome dobiveni rezultati pokazuju pri kojem postotku se optimalni put mijenja, te se prema tome određuje koja je ruta optimalna ako je važnije stići što brže ili je važnije da put bude što jeftiniji. Isto tako pokazuju i koja je ruta optimalna ukoliko su putniku vrijednosti vremena i novca jednako važne. Uz to vidljivo je i kako se optimalni put mijenja ovisno o načinu naplate cestarine (obzirom da se prilikom plaćanja ENC-om dobiva 50% popusta).

5. IZLAZ KODA

5.1 Novac je važniji od vremena

Ukoliko je putniku novac važniji od vremena, odnosno važno mu je da put bude što jeftiniji, prema izlazu koda vidljivo je da se optimalni put mijenja pri važnosti novca od 65% u slučaju plaćanje cestarine ENC-om. Znači ukoliko je putniku barem 65% važno da put bude što jeftiniji optimalna ruta za njega je preko Veprinca. Navedeno pokazuje da ukoliko je putniku važna cijena puta, ali pri tom ni vrijeme trajanja puta nije zanemarivo, optimalan put je zapravo kroz tunel Učku uz napomenu da se cestarina, odnosno tunelarina plaća elektroničkom naplatom cestarine (ENC). Naime, ukoliko je putniku samo malo važnije da put bude što jeftiniji, u odnosu na činjenicu da bude vremesnki što kraći za njega je optimalan put kroz tunel Učku. Izraženo u postocima npr. 60% važno da bude što jeftiniji, 40% važno vremenski što kraći.

Ako se kao orijentir uzima jednaka vrijednost obiju varijabli, zaključuje se da postotak od 60% pokazuje tek malo veću važnost. O vlastitim preferencijama putnika bespredmetno je raspravljati, izlaz koda je tu tek da pokaže pri kojem postotku se optimalni put mijenja, dok je odluka uvijek na puticima. Ipak, mora se istaknuti da određeni broj ljudi navedenom rutom prometuje svakodnevno, odnosno da je rijč o dnevnim migracijama zbog odlaska na posao, u školu, na fakultet, a u to se svakako ubrajaju i transportne tvrtke . U tom slučaju aspekt vremena nikako nije zanemariv, a s druge strane i novac igra značajnu ulogu. Za takva putovanja može se doći do zaključka da je put kroz Tunel optimalna ruta.

5.2 Novac i vrijeme su jednako važni

Ukoliko su putniku novac i vrijeme od jednake važnosti, odnosno u jednakom omjeru je važno stići što brže, a da financijski bude što isplativije najbolje je cestarinu plaćati putem ENC uređaja. U tom slučaju optimalan put je kroz tunel Učku. Ukoliko putnik cestarinu plaća klasično, u tom slučaju pri jednakim vrijednostima varijabli j i k optimalan put je preko Veprinca.

Sukladno tome može se zaključiti da je za svakodnevne migracije najisplativije ići kroz tunel Učku, a cestrainu plaćati putem ENC uređaja. Naime, svaki pojedinac ima svoje razloge i sam odabire je li mu važnije vrijeme ili novac. Ipak ukoliko je riječ o svakodnevnom putovanju može se pretpostaviti da su vrijeme i novac podjednako važni.

5.3 Vrijeme je važnije od novca

Kod započinje s varijablama j=0, k=100 gdje j označava važnost novca odnosno, činjenicu da putnik želi da put bude što jeftiniji, a k označava važnost vremena, odnosno činjenicu da putnik želi da put bude vremenski što kraći. Iako je donekle opravdana pretpostavka da putnik želi da put bude što jeftiniji na uštrp vremena, ponekad se i doslovno potvrđuje čuvena rečenica B. Frenklina: „Vrijeme je novac".

Naime, putnik kroz ove izračune saznaje koji je put optimalan ovisno o njegovim trenutnim zahtjevima. Nije zanemariv ni postotak, odnosno, postoji razlika ukoliko je putniku 1% važno da stigne što prije ili 40%. Kao što je već razjašnjeno, odabir postotka je prilično disktuabilan. Ukoliko je putniku iz nekog razloga od presudne važnosti da što prije stigne, pa pri tom potroši i više novaca u odnosu na izbor druge rute odabrati će put kroz tunel Učku, neovisno o načinu plaćanja cestraine.

Prilikom „davanja" važnosti vremenu u nekom većem postotku, a da se pri tom ne zanemarauju troškovi svakako treba obuhvatiti svakodnevne migracije na relaciji Pazin Rijeka gdje putnicima aspekt vremena nije zanemariv. Je li putniku u tom slučaju važnost vremena i novca jednako važna ovisi o njegovim osobnim stavovima, mjesečnim prihodima, razlogu putovanja, te svim ostalim razlozima koji se razlikuju od pojedinca do pojedinca. U slučaju klasočnog plaćanja cestarine bez korištenja ENC-a za vrijednost varijable j=40, što znači da je putniku 60% važno da put bude vremenski što kraći optimalan put se mijenja, odnosno više nije optimalan put kroz tunel Učku, već preko Veprinca.

6. OPĆI SLUČAJ

Formula na kojoj se temelji izračun pondera u ovom radu glasi ovako:

$$p=\frac{x*N+(100-x)*V}{100},\ 0<x<100$$

u kojoj N predstavlja novac odnosno računski dobiven iznos cijene prelaska određene dionice (u ovom slučaju od vrha do vrha), a V predstavlja vrijeme prelaska te iste dionce. Vrijednost x odabrana je ovisno o važnosti novca i vremena, odnosno o postotku važnosti istoga. Npr. ukoliko se uzme x=50, znači da je novcu i vremenu pridana jednaka važnost. Varijable uz N i V zapravo prikazuju postotak važnosti N i V. Ukoliko se uzme da je x=60 znači da je putniku 60% važan novac.

Temeljem toga formula se može primjeniti za bilo koji izračun tog tipa ukoliko se posjeduju podaci o cijeni pojedinih dionica.

U istu tu svrhu može poslužiti i prethodno napisan kod.

7. ZAKLJUČAK

Kroz ovaj rad najrpije je kroz teorijske postaveke dan prikaz teorije grafova, te nekih metoda optimizacije za traženje najkraćeg puta u mreži. Na početku dane su sve potrebne definicije kao osnova istraživanja. Nakon toga, izabran je Bellman-Fordov algoritam za pronalazak najkraćeg u ovom slučaju optimalnog puta. Iako je za ovo istraživanje podloga bila SageMath, odnosno izračin pomoću izrađenog koda, prije toga dan je prikaz i izračuna Bellman-Fordovog algoritma „klasičnim“ putem.

Kroz ovaj rad vidljiv je značaj i primjena Bellman-Fordovog algoritma na problem iz prometne struke. Izlazom napisanog koda vidljivo je na koji način se optimalni put mijenja ovisno o trenutnim preferncijama putnika je li važnije vrijeme ili novac. Putnici prilikom odabira optimalne rute mogu zanemariti neki ulazni parametar kao što je uzdužni nagib ceste, te „od oka“ donositi zaključke koji je put optimalan. Ovim radom vidljivo je za svaki pojedini postotak važnosti vremena i novca koja je ruta optimalna, te kako na promjenu iste utječe i način plaćanja cestarine (na ruti na kojoj se naplaćuje).

Kao što je u radu i navedeno, ulazni parametri podložni su promjenama i ovise o brojnim čimbenicima, ali oni se u ovom kodu vrlo jednostavno mijenjaju te kod može poslužiti za traženje bilo koje optimalne rute koja je koncipirana na ovaj način.

Ovim izračunima, odnosno ulaznim parametrima nije obuhvaćena sigurnost prometovanja cestom, koja je također vrlo bitna. Obzirom na specifičnosti ceste „preko Učke“, ali i količinu prometa na dionici istarskog ipsilona, potrebno je provesti još jedno istraživanje gdje će se na podatke dobivene u ovom radu pridodati značaj sigurnosti. Naime, specifičnost ceste od Vele drage do Veprinca su nagli usponi i strmine, te izuzetno uska širina kolnika. Cesta je za vožnju zahtjevna i opasna, a zimi tome dodatno pridonose i zimski uvjeti koji dodatno otežavaju, ponekad čak i onemogućuju prometovanje tom dionicom. Stoga se navedeno može shvatiti kao ograničavajući čimbenik u tumačenju dobivenih rezultata koji bi bili primjenjivi na svakodnevna putovanja. Unatoč tome, dobiveni rezultati se mogu smatrati točnima za dane izračune ukoliko se promatraju samo novac i vrijeme što je u konačnici bila i svrha ovog istraživanja.

U konačnici, dobiveni rezultati su pokazali da je put preko Plomina i Lovrana najduži i najneisplativiji s aspekta vremena i novca. Stoga za česta putovanja navedenom relacijom ne dolazi u obzir ako se želi uštedjeti na novcu ili na vremenu.

LITERATURA

- ČASOPISI

BELLMAN, RICHARD. “ON A ROUTING PROBLEM.” *Quarterly of Applied Mathematics*, vol. 16, no. 1, 1958, pp. 87–90. *JSTOR*, JSTOR, www.jstor.org/stable/43634538. (15.8.2018.)

Bosančić V., Golemac A., Vojković T. (2012.): Kako pomoći trgovačkom putniku, Osječki matematički list 12 , 139-149, dostupno na: file:///C:/Documents%20and%20Settings/pc1x/My%20Documents/Downloads/05_Golemac_TSP.pdf
(13.8.2018.)

Kopić M., Klobučar A.: Mostovi Kaliningrada nekad i sada, Osjecki matemati ˇ cki list ˇ 7(2007), 33–38

Fošner M., Kramberger T.:Teorija grafova i logistika, 2009., file:///C:/Users/Lea/Downloads/Fosner_vol14.pdf (10.8.2018.)

- Web
-

Šimunović Lj.: Prometna (transportna) mreža, dostupno na https://www.weboteka.net/fpz/Osnove%20prometnog%20in%C5%BEenjerstva/OPI_predavanje_5_-_Lj_Simunovic.pdf (1.9.2019.)

http://www.sagemath.org/tour-research.html (10.7.2019.)

https://www.google.com/maps (9.7.2019.)

Štambuk, Lj.: Mreže, Veleučilište u Rijeci, Rijeka, http://veleri.hr/~ljstambuk/Kvantitativne%20za%20web/Kvantitativne%202%20web.pdf
(18.7.2019.)

Osnove prometnog inženjerstva materijali: http://e-student.fpz.hr/Predmeti/O/Osnove_prometnog_inzenjerstva/Materijali/2011-OPI-Predavanje_009.pdf (10.7.2019.)

http://www.math.uniri.hr/~ajurasic/D-OSMO.pdf (10.7.2019.)

Algoritmi na grafovima, http://www.math.rs/p/files/16-Operaciona_istra%C5%BEivanja_-_ve%C5%BEbe_CELOBROJNO_PROGRAMIRANJE_4.pdf (15.7.2019.)

Carić T., Algoritmi i programiranje, Graf i Dijkstra algoritam, 2013.

https://www.weboteka.net/fpz/Algoritmi%20i%20programiranje/e-student/012_2013-04-05%20P12%20ALGPRO%20-%20Graf%20i%20Dijkstra.pdf (15.7.2019.)
https://hr.wikipedia.org/wiki/Algoritam (15.7.2019.)

- Knjige

Ford, L.R., Jr, Network flow theory, The Rand Corporation, Santa Monica, California, 1956.

http://www.dtic.mil/dtic/tr/fulltext/u2/422842.pdf

Osvin-Pavčević M, MATEMATIKA 3, Uvod u teoriju grafova, 2006., Zagreb

POPIS SLIKA

POPIS TABELA