Dodjeljivanje frekvencije komunikacijskom kanalu pomoću optimizacije kolonijom mrava

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Tin Vujasinović

Dodjeljivanje frekvencije komunikacijskom kanalu pomoću optimizacije kolonijom mrava

ZAVRŠNI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Tin Vujasinović

Matični broj: 43503/14–R

Studij: Informacijski sustavi

Dodjeljivanje frekvencije komunikacijskom kanalu pomoću optimizacije kolonijom mrava

ZAVRŠNI RAD

Mentor/Mentorica:

Doc. dr. sc. Nikola Ivković

Varaždin, listopad 2018.

Tin Vujasinović

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

____

Sažetak

Glavna tema ovog rada je bežična komunikacija, odnosno dodjeljivanje frekvencije komunikacijskom kanalu kako bi se izbjegle smetnje i omogućila bolja komunikacija. Problem dodjeljivanja frekvencije pojavljuje se svugdje gdje postoji bežična komunikacija, kao npr. mobilna telefonija, odašiljanju radio i TV signala, satelitskoj komunikaciji i vojnim primjenama. S obzirom na primjenu bežične komunikacije, postoje različite varijante problema koje su prilagođene za tu specifičnu primjenu. Za rješavanje nekog od ovih problema, koriste se razni algoritmi, a u ovom radu specifično koristi se algoritam optimizacije kolonijom mrava. Optimizacija kolonijom mrava je algoritam koji pripada klasi inteligencije roja (*Swarm Intelligence).* „Algoritmi te klase imaju svojstva da su fleksibilni (brzo se prilagođavaju promjenama), robusni (male promjene u sustavu ne znače prestanak rada) te imaju sposobnost samoorganizacije (sustavu nije potrebna vanjska kontrola, rad sustava organizira sam sustav)“ [1]. U ovom radu, optimizacija kolonijom mrava je uspješno primijenjena na problem vezan uz dodjeljivanje frekvencije komunikacijskom kanalu u mobilnoj telefoniji te su dobiveni rezultati prikazani u obliku dijagrama.

Ključne riječi: frekvencija, komunikacijski kanal, optimizacija kolonijom mrava, bežična komunikacija, inteligencija roja, problem dodjeljivanja frekvencije

Sadržaj

1. Uvod

Komunikacija je vrlo važan aspekt ljudskog života. Kroz povijest, ljudi su komunicirali na daljinu na razne neobične načine, kao što su dimni signali kojima se nije mogla prenijeti točna poruka ili pismima što je bilo dosta sporo. Istraživanje bežične komunikacije pomoću radiovalova započeo je Guglielmo Marconi, na kraju 19. stoljeća. Nakon nekoliko godina istraživanja, uspio je uspostaviti vezu s brodom na pučini, na udaljenosti od 29 kilometara [2]. Ubrzo su brodovi počeli koristiti bežičnu komunikaciju temeljenu na radiovalovima za komunikaciju s drugim brodovima i obalnim stanicama. Nakon Prvog svjetskog rata, dostupnost tehnologije dovela je do pojavljivanja amaterskih radio stanica, a kasnije se pojavilo i profesionalno radio emitiranje. S eksperimentalnim odašiljanjem TV signala započelo se 1930-tih godina, a do kraja 1940-tih godina, televizija je postala dostupna širokim masama ljudi [2].

Današnja bežična komunikacija uvelike je napredovala, otkrićem radiovalova postalo je moguće prenositi poruke na daljinu u realnom vremenu. Radiovalovi su vrsta elektromagnetskih valova koji nastaju kada kroz vodič protječe izmjenična električna struja [2]. Propuštanjem izmjenične električne struje kroz uređaje za odašiljanje, antene, radio valovi se šalju kroz medij. Signal koji dolazi na antenu najprije je potrebno modulirati te ga nakon toga poslati. S druge strane, druga antena prima radiovalove koje je poslala prva antena i demodulira ih te su se tako bežično poslali podaci.

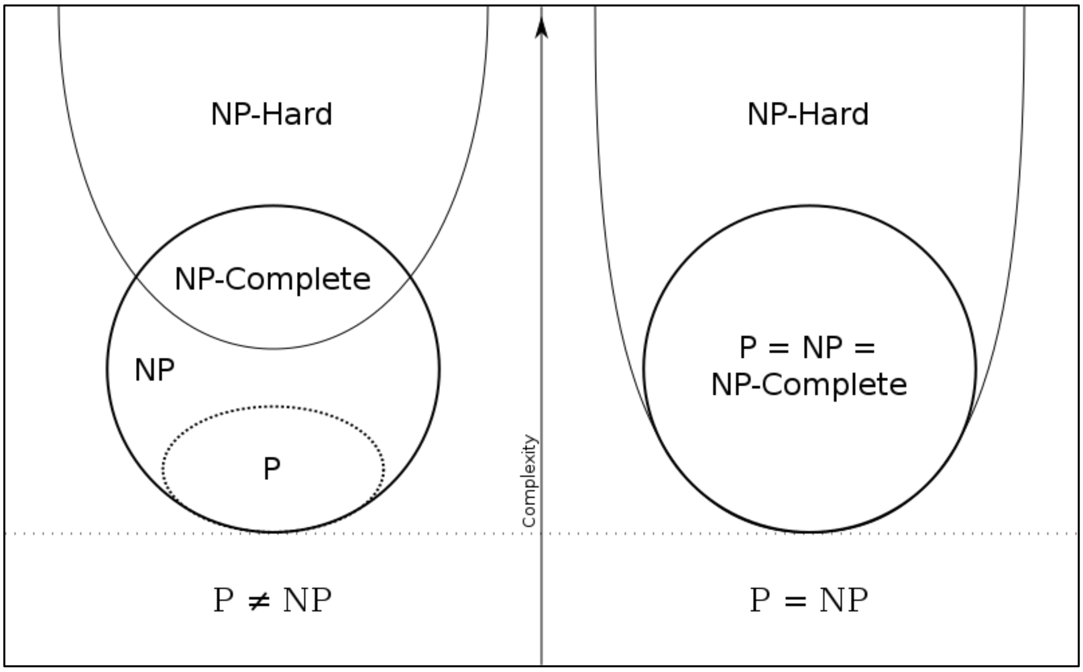
Radiovalovi dijele prema valnoj duljini na *valna područja*, no danas je češća podjela na frekvencijske pojase [3]. Frekvencijski raspon radio valova koji se aktivno koristi je između 3Hz i 300GHz. Taj raspon podijeljen je na nekoliko frekvencijskih pojasa od kojih svaki ima određenu primjenu. Radiovalovi svoju su primjenu našli u odašiljanu radio i TV signala, satelitskoj komunikaciji, vojnim primjenama te se koriste za komunikaciju u mobilnim mrežama. Daljnjim istraživanjem radiovalova i frekvencijskog spektra, određeni su pojasevi te se za svaki od tih pojasa odredila primjena, kako bi se izbjegle smetnje i smanjili troškovi. U sljedećoj tablici prikazani su frekvencijski pojasevi te njihova uobičajena primjena.

Tablica 1. Prikaz raspodjele frekvencijskih pojasa

Kratica	**Naziv**	**Frekvencija**	**Primjena**
ELF (*Extremly Low Frequency*)	-	3 Hz – 30Hz	komunikacija s podmornicama
SLF (*Super Low Frequency*)	-	30 Hz – 300 Hz	-
ULF (*Super Low Frequency*)	-	300 Hz – 3 kHz	-
VLF (*Very Low Frequency*)	mirijametarski valovi	3 kHz – 30 kHz	komunikacija s podmornicama
LF (*Low Frequency*)	dugi val	30 kHz – 300 kHz	radio, radijski satovi, radionavigacija
MF (*Medium Frequency*)	srednji val	300 kHz – 3 MHz	radio
HF (*High Frequency*)	kratki val	3MHz – 30 MHz	Radio
VHF (*Very High Frequency*)	ultra kratki val	30 MHz – 300 MHz	FM radio, radar
UHF (*Ultra High Frequency*)	mikrovalovi	300 MHz – 3 GHz	televizija, GSM, mikrovalne pećnice, Wi-Fi
SHF (*Super High Frequency*)	centimetarski val	3 GHz – 30 GHz	radar, usmjerene veze, satelitska televizija
EHF (*Extremly High Frequency*)	milimetarski val	30 GHz – 300 GHz	usmjerene veze

(Izvor: https://hr.wikipedia.org/wiki/Radiovalovi)

Razvojem tehnologije broj uređaja koji komuniciraju putem radiovalova počeo je rasti te se pojavio problem kako efikasno dodijeliti frekvencije uređajima kako bi oni komunicirali bez smetnji i šuma. Problem dodjeljivanja frekvencije komunikacijskom kanalu danas se javlja kod mobilne telefonije, satelitskoj komunikaciji, odašiljanju radio i TV signala te u vojnim primjenama. Kako postoji više primjena bežične komunikacije gdje se javlja ovaj problem, postoje i različite varijante problema. Proučavanjem ovog problema od strane raznih znanstvenika, došlo se do zaključka da ovaj problem pripada klasi NP-kompletnih problema. NP-kompletni problemi najteži su problemi iz klase NP problema [2]. Ova klasa predstavlja probleme kojima je vrijeme potrebno za rješavanje nekim algoritmom drastično raste kako se problem povećava.



Slika 1. Euler-ov dijagram za P,NP-kompletne i NP-teške probleme [7]

Slika iznad prikazuje klasifikaciju NP problema, gdje lijeva strana vrijedi ako je $P \neq NP$, dok desna strana vrijedi ako je $P = NP$. Prema lijevom dijelu slike, vidimo da su NP-kompletni problemi, kojima pripada problem dodjeljivanja frekvencije, podskup NP-teških i NP problema te su oni najteži problemi iz klase NP problema [7].

Za rješavanje problema dodjeljivanja frekvencije komunikacijskom kanalu postoje razne tehnike i metode. U ovome radu naglasak je na algoritmu ACO (*Ant Colony Optimization*). Ovaj algoritam inspiriran je ponašanjem mrava u prirodi kod traženja hrane, a pripada klasi algoritama inteligencije roja (*Swarm Intelligence*). Algoritmi ove klase simuliraju ponašanje roja insekata ili drugih životinja u prirodi kako bi doveli do optimalnog rješenja nekog problema.

2. Problemi dodjeljivanja frekvencije komunikacijskom kanalu

Problem dodjeljivanja frekvencije komunikacijskom kanalu javlja se u raznim primjenama, od mobilne telefonije do raznih vojnih primjena. Korijen problema je u tome što je frekvencijski pojas ograničen te povećanjem broja uređaja koji bežično komuniciraju dolazi do zagušenja pa uređaji međusobno ometaju pravilan rad. Prema [2], u osnovi, problem dodjeljivanja frekvencije sastoji se od dva dijela:

a) skupu veza za bežičnu komunikaciju moraju biti dodijeljene frekvencije tako da se omogući komunikacija između odašiljača i prijemnika za svaku vezu iz skupa, gdje moramo uzeti u obzir da je promet između odašiljača i prijemnika dvosmjeran pa tako moramo dodijeliti dvije frekvencije svakoj vezi, po jednu za svaki smjer komunikacije
b) ako dvama vezama dodijelimo frekvencije, one mogu uzrokovati ometanje što bi dovelo do gubljenja kvalitete signala. Da bi došlo do ometanja između dva signala, moraju biti zadovoljena dva uvjeta:
    i) frekvencije koje smo dodijelili dvama vezama moraju biti blizu jedna druge, u smislu elektromagnetsko pojasa koji se koristi za bežičnu komunikaciju
    ii) veze moraju biti geografski smještene jedna blizu druge, a ometanje će se dogoditi na mjestu gdje je energija odašiljanja otprilike jednaka za oba signala

Frekvencijski pojas, $[f_{min}, f_{max}]$, koji je dostupan nekom pružatelju usluga bežične komunikacije podijeljen je na skup kanala, gdje je razlika između frekvencije jednaka nekom Δ [3]. Prema tome je frekvencijski pojas podijeljen na kanale koji su obično pobrojani od 1 do N, gdje je N ukupan broj kanala dostupan tom pružatelju usluga bežične komunikacije, a izračunava se prema $N = (f_{max} - f_{min})/\Delta$. Ovom formulom prvo izračunamo raspon dostupnih frekvencija te podijelimo razlikom između kanala kako bi dobili broj kanala koji možemo smjestiti u taj frekvencijski raspon. Skup kanala obično se označava s $D = \{1, \ldots, N\}$. Iz raznih razloga, za određenu komunikaciju nisu svi kanali uvijek dostupni, na primjer, u pograničnim zonama korištenjem nekih kanala može dovesti do ometanja signala iz druge države pa zbog nekih zakonskih regulativa nije dozvoljeno koristiti određene frekvencije u tim područjima. Iz tog razloga, skup kanala koji su dostupni za uspostavu neke veze v na tom području je podskup skupa svih kanala, tj. $D_v \subseteq D$. Na svakom kanalu moguća je komunikacija između odašiljača i prijemnika u jednom smjeru. Za dvosmjernu komunikaciju potrebna su dva kanala. U većini primjena, koriste se dva frekvencijska pojasa, $[f_{min}^1, f_{max}^1]$ i $[f_{min}^2, f_{max}^2]$ koji se svaki sastoji od N kanala. Prvi frekvencijski pojas sastoji se od kanala označenih s $\{1, \ldots, N\}$, a drugi od kanala označenih s $\{1 + s, \ldots, N + s\}$. Jedan skup kanala koristi se za komunikaciju u jednom smjeru, a drugi skup za komunikaciju u drugom smjeru.

Nadalje, smetnje kod bežične komunikacije mjere se omjerom signala i šuma (**SNR** ili **S/N** – *signal-noise ratio*) na strani prijemnika. Ukoliko nije došlo do smetnji, signal koji je primio prijemnik može se razumjeti. Ukoliko postoji više signala koji se odašilju na istoj ili bliskoj frekvenciji, dolazi do ometanja i šuma te se signal izobličuje te na stranu prijemnika ne dolazi isti signal koji je poslan. Snaga signala izražava se u decibelima (dB). Snaga signala koji šaljemo izračunava se kao $P_{signal,dB} = 10 \log_{10}(P_{signal})$ dok se snaga šuma izračunava pomoću formule $P_{noise,dB} = 10 \log_{10}(P_{noise})$. Omjer signala i šuma također se izražava u decibelima. Možemo izračunati omjer signala i šuma prema formuli $SNR_{dB} = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)$. Najčešće se u bežičnoj komunikaciji uzima prag od 12 dB ili 15 dB kao zadovoljavajući omjer signala i šuma [2].

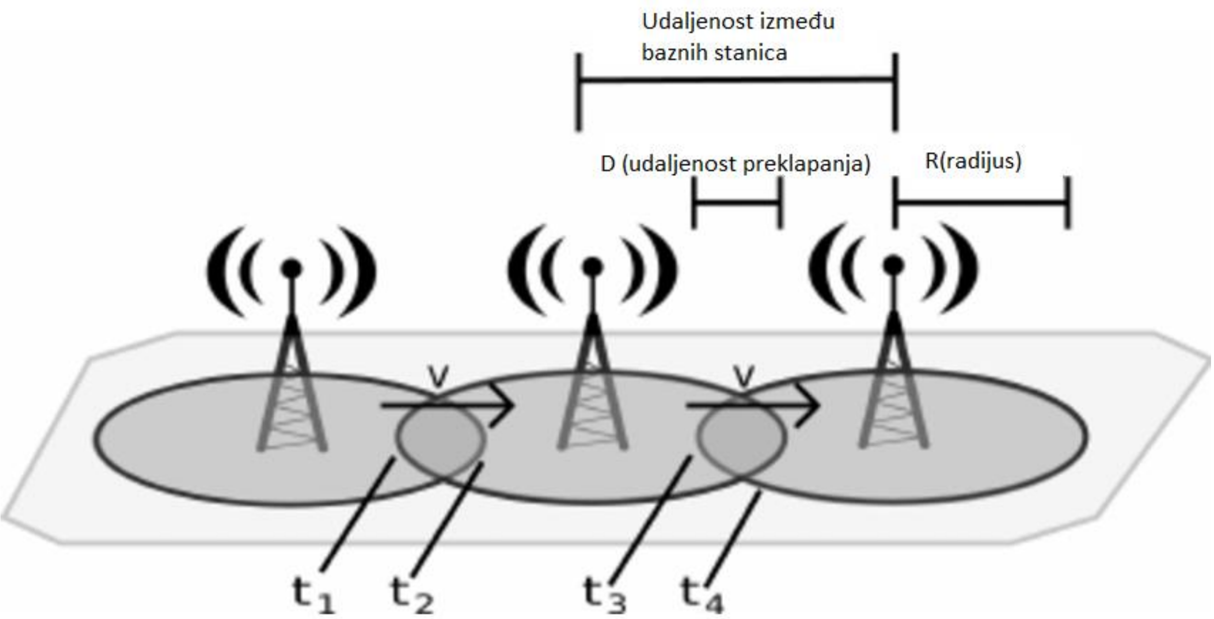
2.1. Primjena u mobilnoj telefoniji

Jedna od najraširenijih primjena bežične komunikacije jesu mobilne mreže. Kako se kod mobilnih mreža koristi bežična komunikacija, i kod mobilnih mreža se javlja problem dodjeljivanja frekvencija. Mobilne mreže sastoje se od velikog broja mobilnih telefona koji se povezuju na bazne stanice te preko njih uspostavljaju vezu. Bazne stanice raspoređene su geografski kako bi se pokrilo određeno područje te omogućilo uspostavljanje veze, a ta područja nazivaju se čelije (*cell)*. Svaka bazna stanica koristi određenu frekvenciju. Kako mobilni telefoni nisu fiksni, njihova lokacija može se promijeniti pa tako mogu prelaziti iz područja jedne bazne stanice u područje druge bez da korisnik primijeti da je bio prebačen na drugu baznu stanicu. Do ometanja signala dolazi kada se signali odašilju u sličnom smjeru, sličnom snagom i za to koristi slična frekvencija.

Kod mobilne telefonije, za svaku vezu potreban je poseban signal koji se šalje između dva uređaja, što dodatno otežava problem dodjeljivanja frekvencije. Na primjer, kod radio i TV odašiljanja, isti signal (program) se odašilje prema korisnicima koji ga zatraže, no više o tome u sljedećem poglavlju. U mobilnoj telefoniji, razgovori koji se odvijaju preko mobilnih telefona su različiti te se moduliraju u različite signale te se taj signal izmjenjuje između dva korisnika, odnosno odašiljača i prijemnika koji sudjeluju u razgovoru. Prema tome, koliko je razgovora između dva korisnika, toliko se jedinstvenih signala odašilje. Danas se za posluživanje tako velikog broja korisnika koriste tehnologije *Frequency Division Multiple Access* (*FDMA*)*,* odnosno dijeljenje frekvencije za pristup više korisnika istovremeno, *Time Division Multiple Access* (*TDMA*), odnosno dijeljenje kanala na vremenske odsječke za svakog korisnika i *Code Division Multiple Access* (*CDMA*), gdje se signal za svaki odašiljač posebno kodira te se na taj način isti kanal može iskoristiti za više korisnika.

2.1.1. Opis problema u mobilnoj telefoniji

U ovome radu naglasak je na problemu dodjeljivanja frekvencije komunikacijskim kanalima u mobilnim mrežama. Mobilna mreža sastoji se od mreže baznih stanica koje međusobno komuniciraju prema potrebi. One su raspoređene geografski kako bi se neko područje pokrilo signalom te se omogućila bežična komunikacija. Slika ispod prikazuje primjer smještanja baznih stanica u prostoru. Cilj je što veću površinu pokriti signalom na način da svaka bazna stanica pokriva određeni radijus prostora oko sebe signalom te da je svaka bazna stanica u dometu drugih baznih stanica kako bi mogle međusobno komunicirati. Na slici su prikazane tri bazne stanice koje su raspoređene u prostoru tako da određeno područje pokrivaju signalom i mogu međusobno komunicirati jer postoji preklapanje signala između njih.



Slika 2. Razmještaj baznih stanica u prostoru [9]

Radijus koji svaka bazna stanica pokriva na slici je označen sa R, dok je udaljenost preklapanja označena sa D. Optimalno je rasporediti bazne stanice kako bi se pokrila najveća moguća površina signalom i da pri tome bazne stanice budu jedna drugoj dovoljno blizu da mogu komunicirati međusobno, odnosno da postoji preklapanje D, no da je ono što manje. Kada se dva mobilna uređaja žele povezati kako bi komunicirali, svaki od njih se spaja na njemu najbližu baznu stanicu. Zatim se signal prenosi između mobilnih uređaja preko baznih stanica i omogućuje se komunikacija. Svaki mobilni uređaj za svoju vezu s baznom stanicom zahtjeva određenu frekvenciju, te bazne stanice koriste određene frekvencije za komunikaciju jedna s drugom. Problem dodjeljivanja frekvencije javlja se kada se na području jedne bazne stanice nalazi više mobilnih uređaja koji žele uspostaviti vezu. Svaki od njih zahtjeva kanal, odnosno frekvenciju, za svoju komunikaciju i tu dolazi do mogućih smetnji ukoliko su dodijeljene frekvencije preblizu jedna drugom prema frekvencijskom pojasu. Frekvencijski pojas koji je dostupan za mobilnu komunikaciju može biti ograničen raznim zakonskim regulativama, npr. zbog velikog broja pružatelja usluga mobilne mreže, i zbog toga je vrlo važno frekvencije koje su dostupne dobro rasporediti kako bi se u potpunosti ili što više izbjegle smetnje u komunikaciji. U ovome radu rješava se problem kada se više mobilnih uređaja spaja na jednu baznu stanicu koja ima ograničen broj frekvencija koje može dodijeliti. Cilj je svakoj poveznici, odnosno mobilnom uređaju, dodijeliti frekvenciju kako bi se što više izbjeglo ometanje signala kod komunikacije mobilnih uređaja s baznom stanicom.

2.2. Primjena u radio i TV odašiljanju

Kao što je ranije spomenuto, dodjeljivanje frekvencije ima svoju primjenu i kod odašiljanja radio i TV signala. Prije uvođenja kabelske i satelitske televizije, radio i TV signal odašiljao se putem zraka. Slično kao kod mobilne telefonije, kako bi se signal odašiljao korisnicima, kroz neko geografsko područje moraju biti raspoređene antene koje će taj signal odašiljati. Kod radio signala, antene odašilju signal koji može biti amplitudno moduliran (AM) ili frekvencijski moduliran (FM). Za amplitudno moduliran signal, koristi se frekvencijski raspon između 540 kHz i 1600 kHz, dok se za frekvencijski moduliran signal koristi raspon od 87 MHz do 108 MHz [2]. Signal koji odašilje jedna antena ne smije ometati signal drugih antena u blizini koje odašilju radio ili TV signal. U slučaju kada promatramo dvije antene koje odašilju isti signal istovremeno, te antene ne smiju ometati jedna drugu. Kako bi korisnici primili radio ili TV signal, svoj prijemnik moraju postaviti na frekvenciju željenog kanala.

2.3. Primjena u satelitskoj komunikaciji

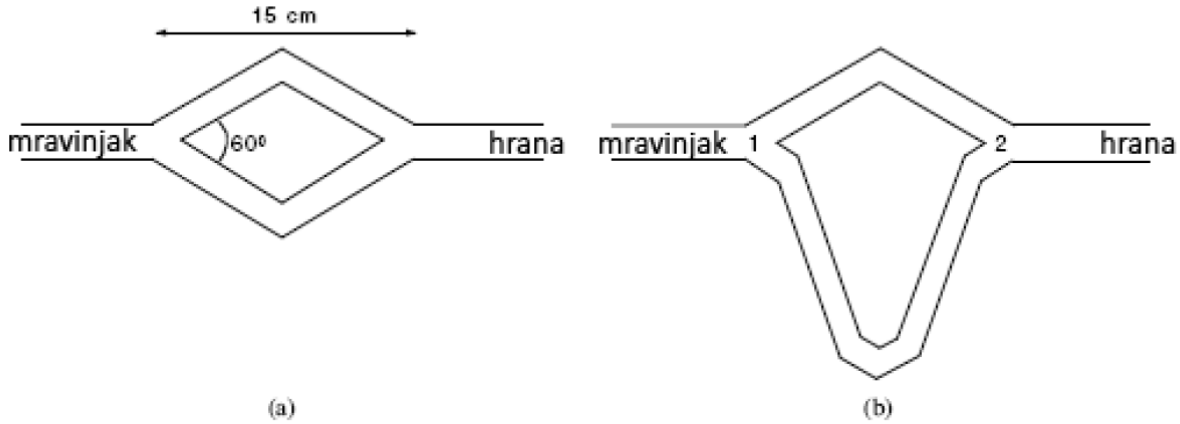
Uz zemaljske mobilne mreže, razvile su se i mreže koje komuniciraju preko satelita. Takve mreže koriste se u područjima gdje pokrivenost signalom zemaljskih mobilnih mreža nije zadovoljavajuća ili uopće ne postoji. Satelitske mreže omogućile su komunikaciju sa bilo kojim dijelom svijeta. Satelitske mreže sastoje se od satelita smještenih u niskoj putanji oko Zemlje, na visini od 780 km [3]. Zbog kratke udaljenosti do Zemlje, moguća je komunikacija satelita s prijenosnim uređajem za komunikaciju. Uređaj se povezuje s njemu najbližim satelitom te mu prenosi signal, dok su sateliti tako raspoređeni da između njih postoji optička veza (nema prepreka između satelita koje bi ometale signal) te signal putuje od satelita do satelita dok ne dođe do satelita koji je najbliži prijemniku signala te ga usmjerava prema njemu. Prva takva mreža implementirana je od strane *Iridium Inc.*, grupacije vlada i korporacija iz cijeloga svijeta. Sastoji se od 66 satelita koji su povezani i čine međusobno povezanu mrežu u Zemljinoj atmosferi [2]. Kao i zemaljske mobilne mreže, sateliti koriste određene frekvencije za komunikaciju. Te frekvencije moraju biti odabrane tako da bi se izbjegle smetnje. Za ovu vrstu komunikacije potrebne su frekvencije pomoću kojih će sateliti komunicirati sa zemaljskim uređajima i frekvencije pomoću kojih će sateliti komunicirati jedan s drugim kako bi se signal poslao. Problem dodjeljivanja frekvencije ovdje je vrlo izražen, mora se pronaći optimalan raspored frekvencija kako bi se izbjegle smetnje između satelita i zemaljskih uređaja.

2.4. Vojne primjene

Dodjeljivanje frekvencije u vojnim primjenama javlja se kod korištenja terenskih telefona za komunikaciju. Njihovo korištenje dovodi do dinamičkog problema dodjeljivanja frekvencije gdje se lokacija i vrijeme mijenjaju. Problem se javlja kada imamo dva terenska telefona koji žele uspostaviti komunikaciju. Za jednu vezu potrebno je dodijeliti dvije frekvencije na fiksnoj udaljenosti, po jednu za svaki smjer komunikacije. Zbog toga, sve frekvencije su dane u parovima s fiksnom udaljenošću između njih. Proširenje ovog problema dodjeljivanja frekvencije dolazi s uvođenjem ograničenja polarizacije. Za svaku vezu, potrebno je odrediti polarizaciju signala (vertikalna ili horizontalna). Smetnje kod neke uspostavljene veze ne ovise samo o frekvencijama koje su dodijeljene, već i o odabiru polarizacije.

3. Algoritam optimizacije kolonijom mrava

Algoritam optimizacije kolonijom mrava inspiriran je ponašanjem mrava u prirodi, odnosno načinom na koji mravi traže hranu. Mravi u prirodi ostavljaju tragove feromona kada se kreću kroz okoliš, što im omogućuje da komuniciraju jedni s drugima. Kada mravi napuštaju gnijezdo u potrazi za hranom, imaju dvije mogućnosti: slijediti trag feromona koji su ostavili drugi mravi ili odabrati neki svoj nasumični put. Ovaj izbor ovisi o jačini traga feromona koji su ostavili mravi, što je trag intenzivniji, veća je vjerojatnost da će ga mrav slijediti. Eksperiment s dvostrukim mostom koji su proveli Goss, Aron, Deneubourg i Pasteels 1989. godine pokazao je kako točno feromonski trag omogućava da mravi odabiru kraći put do hrane. Eksperiment se sastoji od dva puta, jedan duži i jedan kraći. Na početku eksperimenta, ta dva puta nisu označena feromonima. Kada mravi krenu, nasumično će odabrati jedan od ta dva puta pa će ih otprilike 50% odabrati jedan put, a 50% drugi. Kako je jedan put kraći, mravima koji su odabrali taj put trebati će manje vremena da se vrate do gnijezda s hranom pa će feromonski trag na tom putu biti sve intenzivniji. Kako će taj trag jačati, mravi koji se vrate s hranom s dužeg puta, za sljedeći put će izabrati onaj na kojem je feromonski trag jači, a taj put je kraći. „Ta povratna veza uzrokuje autokatalitički proces jer onda veći broj mrava ide kraćim putem pa intenzitet feromona postaje još veći i privlači sve više mrava.“ [4, str. 16].



Slika 3. Postavke eksperimenta dvostrukog mosta (a) s jednakim granama (b) s različitim granama [5]

Slika iznad prikazuje postavke eksperimenta dvostrukog mosta gdje su u jednom slučaju duljine grana iste, dok u drugom slučaju imamo kraću i dulju granu. U prvom slučaju, nije bitno koju će granu mravi odabrati jer je put koji im je potreban do hrane jednak. Drugi slučaj opisan je već ranije, gdje postoje dvije grane različitih duljina te mravi s vremenom zbog jačanja traga feromona odabiru kraću granu kao put do hrane.

3.1. Općenito o algoritmima ACO

Kada promatramo implementaciju algoritma, feromonski trag predstavlja informaciju o poželjnosti nekog rješenja. Algoritam optimizacije kolonijom mrava osmislio je Marco Dorigo, u svom doktorskom radu predstavio je algoritam nazvan sustav mrava (*Ant System*) iz kojeg su se kasnije razvili razni oblici algoritma optimizacije kolonijom mrava za rješavanje teških optimizacijskih problema. Na visokoj razini apstrakcije, algoritmi optimizacije kolonijom mrava mogu se prikazati pseudokodom prikazanim na sljedećoj slici.

```
Inicijalizacija();
PONAVLJAJ_U_PETLJI {
 Mravi_konstruiraju_rješenja();
 Demonske_akcije(); //opcionalno
 Ažuriranje_feromonskih_tragova();
} DOK_NIJE_ZADOVOLJEN_UVJET_ZAUSTAVLJANJA;
```

Slika 4. Metaheuristika optimizacije kolonijom mrava [4]

Funkcija `Inicijalizacija()` služi za postavljanje osnovnih vrijednosti algoritma kao što su vrijednosti feromonskih tragova, parametri algoritma te se pripremaju strukture podataka koje su potrebne za izvršavanje algoritma. Zatim u petlji koja se ponavlja sve dok nije zadovoljen uvjet zaustavljanja, koji može biti broj iteracija, vrijeme izvođenja ili slično, mravi konstruiraju vlastita rješenja, neovisno o drugim mravima uzimajući u obzir feromonske tragove koje su ostavili ostali mravi. Kada su konstruirali rješenje, mogu se izvoditi Demonske_akcije() koje su opcionalne, te se na kraju ažuriraju feromonski tragovi na temelju dobivenih rješenja.

U funkciji `Mravi_konstruiraju_rješenja()` svaki od *m* mrava (*m* je broj mrava) konstruira svoje rješenje, što dovodi do *m* mogućih rješenja. Svaki mrav konstruira svoje rješenje od početka, uzimajući u obzir feromonske tragove. Mravi dodaju u parcijalno rješenje S^P komponentu po komponentu dok ne konstruiraju gotovo rješenje s [4].

U funkciji `Ažuriranje_feromonskih_tragova()` mravi ažuriraju vrijednosti tragova prema tome koje su se komponente koristile u rješenjima. Mravi u kasnijim iteracijama za konstruiranje rješenja koriste ažurirani feromonski trag.

U funkciji `Demonske_akcije()` mogu se izvoditi ostali postupci koji se dodaju osnovnom algoritmu kako bi se poboljšao rad algoritma, na primjer može se dodati postupak lokalne optimizacije rješenja. Ovisno o tome na koji način mravi konstruiraju rješenja i ažuriraju feromonski trag, postoje razne verzije algoritma optimizacije kolonijom mrava kao što su sustav mrava (*Ant System*, AS), elitistički sustav mrava (EAS), rangirani sustav mrava (*Rank-Based Ant System,* AS_{rank}), sustav kolonije mrava (*Ant Colony System*, ACS), MAKS-MIN sustav mrava (*MAX-MIN Ant System*, MMAS).

3.1.1. Sustav mrava

Sustav mrava (*Ant System*, AS) prva je inačica mravljeg algoritma koji je nastao u okviru doktorskog rada Marca Doriga 1992. godine. Iz ovog algoritma razvile su se druge verzije ACO algoritama. Za odabir komponente c_l u parcijalno rješenje s^P koristi se vjerojatnost koja se određuje prema formuli:

$$p(c_l|s^P) = \frac{\tau_{c(l)}^{\alpha} \cdot \eta_{c(l)}^{\beta}}{\sum_{c\in C^P}\left(\tau_c^{\alpha} \cdot \eta_c^{\beta}\right)},$$

gdje je $\tau_{c(l)}$ vrijednost feromonskog traga za komponentu c_l, a $\eta_{c(l)}$ označava heurističku informaciju za to komponentu. Uz pomoć heurističke informacije η može se za svaku komponentu pridružiti poželjnost odabira te komponente. Vjerojatnost odabira neke komponente ovisi o vrijednosti feromonskih tragova, parametrima α i β i vrijednosti heurističke informacije. Komponente koje imaju veću vrijednost feromonskog traga imaju veću vjerojatnost da budu dodane u parcijalno rješenje. Nakon konstrukcije parcijalnih rješenja, slijedi ažuriranje feromonskih tragova. Najprije se provodi isparavanje feromonskih tragova, a zatim pojačavanje. Isparavanje feromonskih tragova izvodi se prema formuli:

$$\tau_c = (1-\rho) \cdot \tau_c, \forall c \in C.$$

Parametar ρ je faktor isparavanja feromona koji može poprimiti vrijednosti između 0 i 1, odnosno $\rho \in [0, 1\rangle$, a vrijednost $(1-\rho)$ je perzistencija feromonskog traga. Nakon isparavanja, slijedi pojačavanje feromonskih tragova. Za svaku komponentu koja se nalazi u nekom rješenju koje je konstruirao jedan od mrava u zadnjoj iteraciji algoritma, povećava se vrijednost feromonskog traga ovisno o kvaliteti rješenja prema formuli:

$$\tau_c = \tau_c + \sum_{k=1}^{m} \Delta\tau_c^{(k)},$$

gdje je

$$\Delta\tau_c^{(k)} = \begin{cases} \dfrac{Q}{f(s^k)}, & ako\ c \in s^k \\ 0, & ako\ c \notin s^k \end{cases}.$$

Parametar Q obično je postavljen na 1, rješenje koje je u zadnjoj iteraciji konstruirao mrav s rednim brojem k označeno je s s^k, a dobrota tog rješenja je $f(s^k)$. Isparavanje i pojačavanje feromonskog traga može se provesti u istom koraku, kombinacijom prethodnih izraza za isparavanje i pojačavanje traga.

3.1.2. Elitistički sustav mrava

Performanse algoritma sustav mrava nisu bile najbolje, mogao se koristiti samo za instance problema manjih dimenzija pa je tako razvijen elitistički sustav mrava. Kako bi algoritam brže konvergirao prema dobrim rješenjima, uvedeno je pravilo da se na komponente najboljeg rješenja od početka algoritma s^{gb} stavi dodatne količina feromonskih tragova. U ovoj verziji algoritma, za pojačavanje feromonskih tragova koristi se formula:

$$\tau_c = \tau_c + \sum_{k=1}^{m} \Delta\tau_c^{(k)} + e \cdot \Delta\tau_c^{(gb)},$$

pri čemu je

$$\Delta\tau_c^{(gb)} = \begin{cases} \frac{1}{f(s^{gb})}, ako\ c \in s^{gb} \\ 0, \quad\quad ako\ c \notin\ s^{gb} \end{cases}.$$

Parametar e je novi parametar algoritma. Konstrukcija rješenja i isparavanje feromonskih tragova provode se na jednak način kao i kod algoritma AS. Eksperimenti koje su proveli Dorigo, Maniezzo i Colorni 1996. godine pokazali su kako algoritam EAS pronalazi bolja rješenja od algoritma AS te ih pronalazi nakon manjeg broja iteracija.

3.1.3. Rangirani sustav mrava

Rangirani sustav mrava je algoritam koji je također nastao kao modifikacija algoritma AS. Kod ove verzije algoritma, rješenja iz zadnje iteracije se sortiraju prema dobroti i tada se rangiraju tako da najbolje rješenje poprimi rang 1. Ako se pojave rješenja jednake dobrote, tada se rang tim rješenjima može dodijeliti slučajno ili nekim drugim principima. Kod ovog algoritma, konstruiranje rješenja i isparavanje feromonskog traga provodi se jednako kao i kod algoritma AS, dok se pojačavanje tragova provodi za $w-1$ rješenja s najnižim rangovima te za najbolje rješenje od početka algoritma s^{gb}. Parametar w je broj koliko se najboljih rješenja koristi za ažuriranje tragova, dok se ostala rješenja čiji je rang veći ili jednak w ne koriste. Feromonski tragovi kod ovog algoritma pojačavaju se prema formuli:

$$\tau_c = \tau_c + \sum_{r=1}^{w-1} (w-r)\Delta\tau_c^{(r)} + w \cdot \Delta\tau_c^{(gb)},$$

tako da mravi s rješenjem manjeg ranga r ostavljaju više feromona. Količina feromona koji se dodaje ovisi i o vrijednosti dobrote rješenja.

3.1.4. Sustav kolonije mrava

Algoritam sustav kolonije mrava razlikuje se od algoritma sustava mrava prema načinu na koji se konstruiraju rješenja. Kod odabira komponente $c_l \in C^P$koja će biti dodana u parcijalno rješenje s^P koristi se slučajna varijabla q. Ta slučajna varijabla je uniformno distribuirana na intervalu $[0,1]$ prema pseudo-slučajnom pravilu

$$c_l = \begin{cases} \arg\max c \in C^P \left\{\tau_c \cdot \eta_c^{\beta}\right\}, q < q_0 \\ \frac{\tau_{ij} \cdot \eta_{ij}^{\beta}}{\sum_{c \in C^P}(\tau_c \cdot \eta_c^{\beta})}, inače \end{cases}.$$

Ako je varijabla q manja od parametra q_0, odabire se komponenta za koju se dobiva najveći umnožak feromonskog traga i heurističke informacije podignute na potenciju β. U suprotnom, koristi se proporcionalno-slučajno pravilo kao i kod algoritma AS uz fiksiran parametar $\alpha = 1$. Nakon dodavanja komponente c_l u svoje parcijalno rješenje, svaki mrav provodi lokalno isparavanje za pripadni feromonski trag prema formuli:

$$\tau_{c(l)} = (1 - \xi) \cdot \tau_{c(l)},$$

gdje je ξ parametar lokalnog isparavanja u intervalu $\langle 0,1 \rangle$.
Globalno ažuriranje feromonskih tragova provodi se nakon svake iteracije, samo za najbolje rješenje konstruirano od početka algoritma prema formuli:

$$\tau_c = (1 - \rho) \cdot \tau_c + \Delta\tau_c^{(gb)}, \forall c \in s^{gb}.$$

Za ostale feromonske tragove ne provodi se ni isparavanje, ni ažuriranje tragova.

3.1.5. MAKS-MIN sustav mrava

Algoritam MAKS-MIN sustav mrava 2000. godine predložili su Stützle i Hoos, a dobio je ime po tome što se vrijednosti feromonskih tragova izražavaju u intervalu između unaprijed definirane minimalne i maksimalne vrijednosti. Ovaj algoritam koristi istu metodu za konstruiranje rješenja kao i algoritam AS, dok se razlikuje u načinu ažuriranja feromonskih tragova jer se kod ovog algoritma koristi samo najbolje rješenje za pojačavanje tragova. Potrebno je osigurati da kod ažuriranja feromonskih tragova vrijednosti ne poprime vrijednosti veće od τ_{max}, a da kod isparavanja vrijednosti ne smiju poprimiti vrijednost manju od donje granice zadane s τ_{min}. Formula za isparavanje glasi:

$$\tau_c = \max\{\tau_{min}, (1 - \rho) \cdot \tau_c\}, \forall c \in C,$$

dok je formula za pojačavanje:

$$\tau_c = \min\left\{\tau_{max}, \tau_c + \Delta\tau_c^{(best)}\right\}, \forall c \in s^{best},$$

gdje je s^{best} najbolje rješenje od početka algoritma ili najbolje rješenje u zadnjoj iteraciji.

Gornja granična vrijednost feromonskih tragova τ_{max} postavlja se na asimptotsku vrijednost feromonskog traga koju bi mogla poprimiti komponenta koja se u svakom koraku pojačava kao da je sastavni dio optimalnog rješenja. Ta vrijednost izračunava se prema:

$$\tau_{max} = \frac{1}{\rho \cdot f(s^*)},$$

gdje je $f(s^*)$ dobrota optimalnog rješenja s^*. U pravilu, dobrota optimalnog rješenja nije poznata pa se za početnu vrijednost τ_{max} uzima procjena dobrote optimalnog rješenja koja se može dobiti nekim jednostavnim postupkom. Dodatna odlika algoritma MMAS je što postoji mogućnost da se kod stagnacije algoritma, odnosno kada velikim brojem iteracija ne dolazimo do boljeg rješenja, provede reinicijalizacija feromonskih tragova na početne vrijednosti.

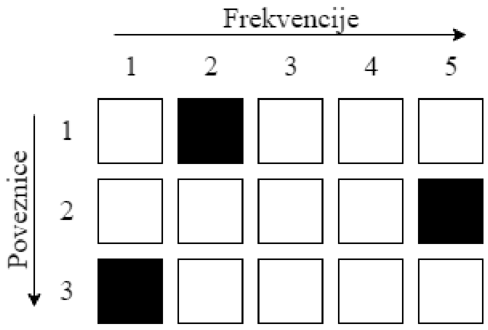
4. Algoritamsko rješenje za odabranu vrstu problema

Kao što je ranije pojašnjeno, problem dodjeljivanja frekvencije moguće je riješiti primjenom optimizacije kolonijom mrava. Problem koji je u ovome radu riješen ovim postupkom je problem dodjeljivanja frekvencija u mobilnim mrežama. Problem se javlja kada jedna bazna stanica mora dodijeliti frekvencije uređajima koji se spajaju na tu baznu stanicu. Uređaji koji se žele spojiti na istu baznu stanicu, moraju biti u geografskoj blizini te stanice pa tako su time i u blizini jedan drugoga. Kako bi se izbjegle smetnje kod prijenosa signala između bazne stanice i svakog uređaja, važno je osigurati posebnu frekvenciju za svaki uređaj. Ovaj problem zadaje se uz neke uvjete koje rješenje mora zadovoljavati. Problem se zadaje s brojem poveznica kojima treba dodijeliti frekvencije, s brojem dostupnih frekvencija koje se mogu dodijeliti i matricom koja određuje minimalni razmak frekvencija između poveznica kako bi se izbjegle smetnje. Slijedeći primjer prikazuje jedan takav problem i njegovo rješenje. Problem je zadan slijedećim vrijednostima:

- Broj poveznica je 3,
- Broj dostupnih frekvencija je 5
- Matrica je:

$$\begin{bmatrix} 5 & 1 & 0 \\ 2 & 5 & 4 \\ 0 & 3 & 5 \end{bmatrix}.$$

Vrijednosti na dijagonali matrice postavljaju se na jednake vrijednosti koje u slučaju ove varijante problema ne nose nikakvu informaciju. Ostale vrijednosti u matrici pokazuju minimalni razmak između frekvencija kako bi se izbjegle smetnje. Na primjer, prema matrici je vidljivo da poveznica 2 mora imati frekvenciju dodijeljenu koja je za minimalno 2 udaljena od frekvencije dodijeljene poveznici 1 i za 4 udaljena od frekvencije dodijeljene poveznici 3. Rješenje problema može se prikazati kao matrica veličine $brojPoveznica \times brojFrekvencija$ te se u matrici označe frekvencije koje su dodijeljene svakoj poveznici. Takvu matricu prikazuje slijedeća slika.



Slika 5. Primjer rješenja problema [autorski rad]

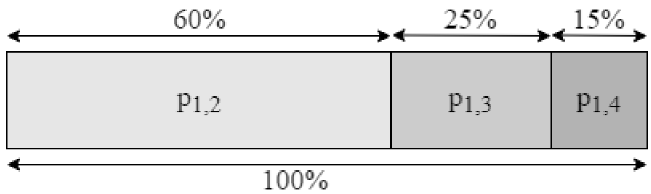
Na slici je vidljivo da je poveznici 1 dodijeljena frekvencija 2, poveznici 2 je dodijeljena frekvencija 5 te je poveznici 3 dodijeljena frekvencija 1. Rješenje se sastoji od parova poveznica i frekvencija koje su tim poveznicama dodijeljene. Za rješavanje ovog problema implementiran je algoritam optimizacije kolonijom mrava.

Algoritam prima određene parametre kako bi se moglo utjecati na njegov rad, to su parametri α koji označava važnost feromonskog traga kod odabira komponente u rješenje, β koji označava važnost heurističke informacije kod odabira komponente u rješenje i ρ koji određuje količinu feromona koja će ispariti nakon svake iteracije. Na početku izvođenja algoritma, unose se podaci o problemu, broj poveznica kojima treba dodijeliti frekvenciju, broj frekvencija, broj mrava koji će konstruirati rješenja i broj iteracija programa. Početni trag feromona postavlja se na određenu vrijednost, $\tau_0 = 1$. Vrijednosti traga feromona mogu postići najvišu vrijednost $\tau_{MAX} = 1000$ ili najmanju vrijednost $\tau_{MIN} = 0.001 * \tau_o$ koja se izračunava iz početne vrijednosti τ_0. Nakon unosa svih potrebnih parametara problema, potrebno je popuniti matricu koja sadrži podatke o tome koliki mora biti razmak između frekvencija.

Nakon unosa matrice, program kreće s radom te pokušava pronaći optimalno rješenje. U ovom koraku, svaki mrav kreira svoje rješenje koje se sastoji od parova poveznica i frekvencija koje su dodijeljene tim poveznicama. Kako bi mravi dodali neki par poveznice i frekvencije u svoje parcijalno rješenje, koriste vjerojatnosti da neka frekvencija bude dodijeljena nekoj poveznici. Prije konstruiranja rješenja, program prema vrijednostima iz matrice feromonskog traga i matrice heurističke informacije izračunava vjerojatnosti za odabir svakog para poveznice i frekvencije u parcijalno rješenje. Mravi tada odabiru poveznice prema tim vjerojatnostima. Vjerojatnosti se izračunavaju prema sljedećem izrazu:

$$p_{lf} = \frac{\tau_{lf}^{\alpha} \cdot \eta_{lf}^{\beta}}{\sum_{j=1}^{k} \left(\tau_{lj}^{\alpha} \cdot \eta_{lj}^{\beta} \right)}.$$

Vjerojatnost se računa kao omjer vrijednosti feromonskog traga za određenu kombinaciju poveznice i frekvencije dignut na potenciju α pomnožen s heurističkom informacijom za tu kombinaciju dignutu na potenciju β i sume umnoška vrijednosti feromonskog traga dignut na potenciju α s vrijednostima heurističke informacije dignute na potenciju β za sve frekvencije i odabranu poveznicu. Nakon što se izračuna ta vjerojatnost za sve poveznice i frekvencije, generira se pseudo slučajni broj u intervalu $[0,1]$ i prema tom broju se odabire koja će kombinacija poveznice i frekvencije biti dodana u rješenje. Frekvencije koje imaju jači trag feromona i heurističku informaciju, imaju veću vjerojatnost da budu odabrane.



Slika 6. Primjer vjerojatnosti odabira frekvencije [autorski rad]

Slika prikazuje kako odabrani pseudo slučajni broj određuje koju frekvenciju će mrav dodijeliti trenutnoj poveznici. Na primjeru, vjerojatnost da poveznici 1 dodijeli frekvenciju 2 je 60%, ako pseudo slučajni broj bude u intervalu od 0 do 0.6, poveznici 1 će biti dodijeljena frekvencija 2. Vjerojatnost da poveznici 1 bude dodijeljena frekvencija 3 je 25%, ako pseudo slučajni broj bude u intervalu između 0.6 do 0.85, poveznici 1 će biti dodijeljena frekvencija 3. Ako je pseudo slučajni broj veći od 0.85, poveznici 1 će biti dodijeljena frekvencija 4.

Kod implementacije, za čuvanje podataka o vjerojatnostima za odabir nekog para u parcijalno rješenje koristi se matrica veličine $brojPoveznica \times brojFrekvencija$ čiji su elementi decimalni brojevi u intervalu $[0,1]$. Sljedeća slika prikazuje primjer vrijednosti u matrici vjerojatnosti nakon prve iteracije.

matricaVjerojatnosti	{double[3, 6]}	double[,]
[0, 0]	0.0921687884799478	double
[0, 1]	0.18433757695989561	double
[0, 2]	0.36867515391979122	double
[0, 3]	0.073735030783958239	double
[0, 4]	0.18433757695989561	double
[0, 5]	0.096745872896511514	double
[1, 0]	0.092253238652150252	double
[1, 1]	0.1845064773043005	double
[1, 2]	0.369012954608601	double
[1, 3]	0.077467613478497371	double
[1, 4]	0.1845064773043005	double
[1, 5]	0.092253238652150252	double
[2, 0]	0.096745872896511514	double
[2, 1]	0.18433757695989561	double
[2, 2]	0.36867515391979122	double
[2, 3]	0.073735030783958239	double
[2, 4]	0.18433757695989561	double
[2, 5]	0.0921687884799478	double

Slika 7. Primjer vrijednosti u matrici vjerojatnosti [autorski rad]

Vjerojatnosti koje su sadržane u matrici izračunavaju se prolaskom kroz dvije petlje, koje uzimaju trenutnu vjerojatnost da neki par bude odabran u parcijalno rješenje i dijele je sa sumom svih vjerojatnosti za odabranu frekvenciju. Nakon toga, novo dobivene vrijednosti upisuju se u matricu. Ovaj postupak provodi se na početku svake iteracije, a vrijednosti se ažuriraju prema vrijednostima iz matrica traga i heurističke informacije koje se ažuriraju na kraju svake iteracije. Sljedeći isječak koda prikazuje opisanu funkcionalnost:

```
for (int l = 0; l < brojPoveznica; l++)
{
    for (int f = 0; f < brojFrekvencija; f++)
    {
        sumaVjerojatnosti = 0;
        vjerojatnost = Math.Pow(Tau[l, f],alfa)*
Math.Pow(etaMatrica[l,f],beta);
        for (int i = 0; i < brojFrekvencija; i++)
        {
            sumaVjerojatnosti += (Math.Pow(Tau[l, i],
alfa))*(Math.Pow(etaMatrica[l,i],beta));
        }
        matricaVjerojatnosti[l, f] = vjerojatnost / sumaVjerojatnosti;
    }
}
```

Nakon što su svi mravi konstruirali svoja rješenja odabirom frekvencija za poveznice prema vrijednostima iz matrice vjerojatnosti, potrebno je odrediti koje rješenje ima najbolju dobrotu, odnosno najmanju sumu kazne zbog preklapanja frekvencija. Ovo se provodi za sva rješenja, na način da se za svaki par poveznica iz rješenja provjeri je li razmak između dodijeljenih frekvencija veći od vrijednosti iz matrice koja sadrži podatke o potrebnom razmaku između frekvencija za te dvije poveznice. Kazna za preklapanje frekvencija izračunava se prema izrazu:

$$kazna_{i,j} = \begin{cases} 0,\ ako\ je\ |f_i - f_j| \geq c_{ij} \\ c_{ij} - |f_i - f_j|, ako\ je\ |f_i - f_j| < c_{ij} \end{cases},$$

gdje su c_{ij} vrijednosti iz matrice koja sadrži podatke o potrebnom razmaku između poveznica za dodijeljene poveznice i i j. Vrijednost kazne za svakog mrava zapisuje se u vektor $poljeSuma$ čiji elementi predstavljaju dobrotu rješenja za svakog mrava. Kako bi se ažurirao trag feromona, potrebno je odabrati najbolje rješenje iz trenutne iteracije. Najbolje rješenje je ono koje ima najmanju kaznu, odnosno najbolju dobrotu. Iz vektora $poljeSuma$ uzima se indeks najmanjeg elementa, mrav s tim indeksom konstruirao je najbolje rješenje te se prema tom rješenju ažurira trag feromona. Računanje suma kazni za preklapanje i pronalazak najmanje suma iz vektora $poljeSuma$ implementirano je na način kako je prikazano na sljedećem isječku koda:

```
for (int z = 0; z < brojMrava; z++)
{
    int prvaFreq, drugaFreq, prvaPoveznica, drugaPoveznica, kazna,
sumaKazni;
    prvaPoveznica = 0;
    drugaPoveznica = 0;
    kazna = 0;
    sumaKazni = 0;
    for (int i = 0; i < brojPoveznica; i++)
    {
        for (int k = 0; k < brojPoveznica; k++)
        {
            if (k == i)
            {
                continue;
            }
            prvaPoveznica = i;
            drugaPoveznica = k;
            prvaFreq = poljeMrava[z].skupRjesenja[i];
            drugaFreq = poljeMrava[z].skupRjesenja[k];
            if (Math.Abs(prvaFreq - drugaFreq) < poveznice[prvaPoveznica,
drugaPoveznica])
            {
                kazna = poveznice[prvaPoveznica, drugaPoveznica] -
Math.Abs(prvaFreq - drugaFreq);
                sumaKazni += kazna;
            }
        }
    }
    poljeSuma[z] = sumaKazni;
}

trenutnoNajbolje = 0;

for (int i = 0; i < brojMrava; i++)
{
    if (poljeSuma[i] < minSuma)
    {
        minSuma = poljeSuma[i];
        trenutnoNajbolje = i;
    }
}
```

Nakon što je pronađeno najbolje rješenje u trenutnoj iteraciji, prema tom rješenju ažurira se matrica traga. Najprije se provodi isparavanje traga za sve vrijednosti iz matrice traga prema parametru ρ koji smo zadali, dok se ažuriraju vrijednosti feromonskog traga samo za parove poveznica i frekvencija koje se nalaze u najboljem trenutnom rješenju. U sljedećem isječku koda nalaze se dvije *for* petlje, prva koja provodi isparavanje feromonskog traga za sve vrijednosti iz matrice traga prema izrazu $\tau_c = (1-\rho)\cdot\tau_c, \forall c \in C$ i druga koja ažurira matricu feromonskog traga prema najboljem rješenju u trenutnoj iteraciji.

```
for (int l = 0; l < brojPoveznica; l++)
{
    for (int f = 0; f < brojFrekvencija; f++)
    {
        double trenutnaVrijednost = Tau[l, f];
        Tau[l, f] = (1 - ro) * trenutnaVrijednost;
        if (Tau[l, f] < tauMin)
        {
            Tau[l, f] = tauMin;
        }
    }
}

for (int p = 0; p < brojPoveznica; p++)
{
    int indeks = poljeMrava[trenutnoNajbolje].skupRjesenja[p];
    deltaTau = 1 / minSuma;
    Tau[p, indeks]+=deltaTau;
    if (Tau[p, indeks] > tauMax)
    {
        Tau[p, indeks] = tauMax;
    }
}
```

Vrijednosti u matrici feromonskog traga mijenjaju se ovisno o tome koje rješenja se nagrađuju kao dobra, no kod pokretanja programa postavljene su vrijednosti $tauMin$ i $tauMax$ koje služe kao granice. Ukoliko se neki element matrice feromonskog traga pokuša pojačati iznad maksimalne granice, vrijednost za taj element postaviti će se na vrijednost $tauMax$. Isti je slučaj kada vrijednosti traga isparavaju, ukoliko bi vrijednost nekog elementa matrice pala ispod minimalne granice, vrijednost tog elementa će se postaviti na vrijednost $tauMin$.

Za izračunavanje vjerojatnosti odabir frekvencije za neku poveznicu se uz feromonski trag koristi i heuristička informacija. Vrijednosti heurističke informacija spremaju se u matricu veličine $brojPoveznica \times brojFrekvencija$ te se za svaki par poveznica i frekvencije čuva vrijednost heurističke informacije. Početne vrijednosti u matrici heurističke informacije postavljene su na $\eta_0 = 1$. Prilikom konstruiranja rješenja, za svaku frekvenciju bilježi se koliko je puta bila odabrana u rješenje i te vrijednosti se spremaju u vektor $brojPonavljanja$. Taj broj služi za ažuriranje matrice heurističke informacije na kraju svake iteracije prema izrazu:

$$\eta_{ij} = \frac{\eta_{ij}}{brojPonavljanja_j + 1}, \forall i \in poveznice, \forall j \in frekvencije.$$

Prema ovome, ako neka frekvencija nije odabrana ni u jedno rješenje u prvoj iteraciji na primjer, heuristička informacija za parove poveznica s tom frekvencijom iznositi će $\eta_{ij} = \frac{1}{0+1}$ jer je broj ponavljanja te frekvencije 0 pa će heuristička informacija ostati 1. Frekvencijama koje su odabrane u neko rješenje heuristička informacija će se ažurirati prema broju ponavljanja. Na primjer, na kraju prve iteracije kada je $\eta_{ij} = 1$ heuristička informacija će se ažurirati prema $\eta_{ij} = \frac{1}{N+1}$ gdje je N broj ponavljanja neke frekvencije u rješenjima. Ovaj dio implementiran je na način kako je prikazano na slijedećem isječku koda.

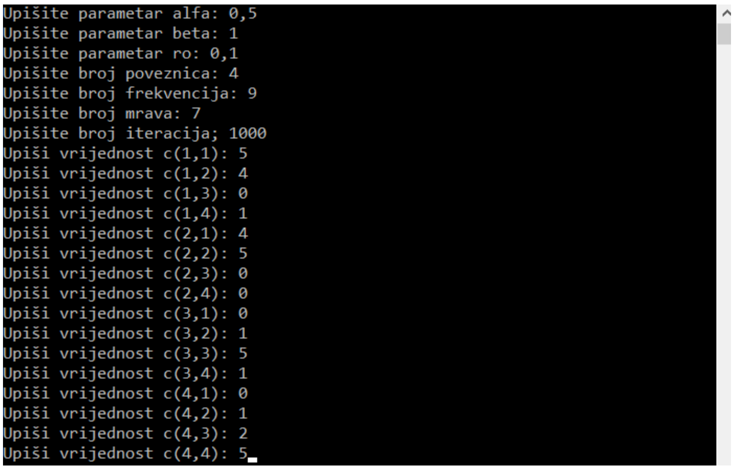
```
for (int l = 0; l < brojPoveznica; l++)
{
    for (int f = 0; f < brojFrekvencija; f++)
    {
        double vrijednost = etaMatrica[l, f];
        etaMatrica[l, f] = vrijednost / (brojPonavljanja[f] + 1);
    }
}
```

Optimalno rješenje prepoznaje se prema dobroti rješenja koje je određeni mrav izračunao. U svakoj iteraciji, izračunava se dobrota najboljeg rješenja te se taj podatak zapisuje u datoteku. Program završava s radom kada je izvršio onoliko iteracija koliko je bilo zadano ili ako je pronašao rješenje čija je cijena 0, tj. rješenje u kojem nema preklapanja. Rezultat izvršavanja ovog programa je datoteka u kojoj se nalazi dobrota najboljeg rješenja za svaku iteraciju. Cilj programa je da se svakom iteracijom približava optimalnom rješenju. Iz podataka koje je program zapisao u datoteku, možemo kreirati dijagram koji pokazuje kako se dobrota rješenja kreće iz iteracije u iteraciju.

U programskom rješenju ovog algoritma, koriste se razne strukture podataka kako bi program funkcionirao. Za pohranu vrijednosti parametara kao što su α, β, ρ te broj frekvencija, poveznica, mrava i iteracija koriste se jednostavne strukture podataka, ovisno o tome što trebamo pamtiti. Parametri α, β i ρ su decimalni brojevi pa se za njihovo spremanje koristi jednostavni tip podataka *double* (double), dok se za spremanje broja frekvencija, poveznica, mrava i iteracija koristi tip podataka *int (*integer*)* jer su ove vrijednosti cijeli brojevi. Za pohranu podataka o razmaku između frekvencija za bilo koje dvije poveznice koristi se matrica veličine $brojPoveznica \times brojPoveznica$, čije su vrijednosti cijeli brojevi. U programskom rješenju, ta se matrica implementira kao višedimenzionalno polje tipa *int* (integer). Za pohranu podataka o tragu feromona i heurističkoj informaciji također se koristi višedimenzionalno polje, no u ovom slučaju tip podataka koji se koristi je *double (*double) jer su vrijednosti koje se nalaze u ovim matricama decimalni brojevi. Matrice traga feromona i heurističke informacije veličine su $brojPoveznica \times brojFrekvencija$. Rješenja koja mravi konstruiraju spremaju se u obliku vektora, odnosno, jednodimenzionalnog polja veličine *brojPoveznica* jer se rješenje treba sastojati od parova poveznica i frekvencija. Vrijednosti koje se upisuju u vektor rješenja su brojevi frekvencija koje su dodijeljene poveznicama. Mravi su implementirani kao klasa *Mrav* koja sadrži svojstvo *rjesenje* koje je vektor jednostavnog tipa *int* (integer) veličine *brojPoveznica* koji sadrži informacije o tome koje je frekvencije taj mrav dodijelio kojim poveznicama. Za konstruiranje rješenja mravi koriste vjerojatnosti odabira neke frekvencije za određenu poveznicu čije se vrijednosti spremaju u matricu veličine $brojPoveznica \times brojFrekvencija$, dok su vrijednosti u toj matrici tipa *double* (double) jer su dobivene vrijednosti decimalni brojevi između 0 i 1.

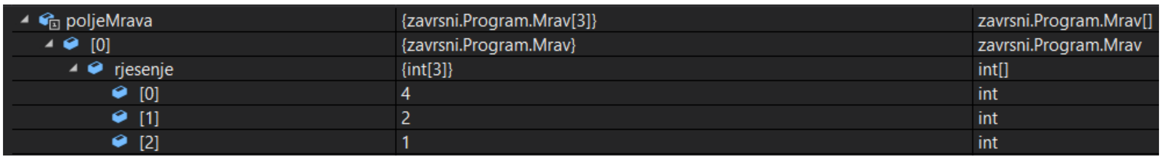
Nakon što su svi mravi završili s konstruiranjem rješenja, izračunava se dobrota svih rješenja te se prema najboljem rješenju ažurira trag feromona. Kako bi se u sljedećim iteracijama mogle odabrati kombinacije čiji je trag feromona slabiji, koristi se heuristička informacija. Kada se neka frekvencija odabere u rješenje, povećava se njen broj korištenja te se heuristička informacija ažurira prema tom broju, tako da se trenutna vrijednost heurističke informacije podijeli s brojem korištenja te frekvencije. Na taj način frekvencije koje imaju slabiji trag, mogu ući u rješenje, no ako su rješenja koja koriste takve frekvencije loša, trag feromona će slabjeti i heuristička informacija neće biti dovoljna da se ta frekvencija ponovo odabere. Opisani postupak konstruiranja rješenja i ažuriranja feromonskog traga ponavlja se dok se program ne izvede određeni broj puta, a taj broj je broj iteracija koji se unosi kod pokretanja programa. Iznimka ovom pravilu je da će program završiti prije ukoliko se pokaže da je našao optimalno rješenje gdje je suma preklapanja između svih poveznica i frekvencija jednaka nuli, odnosno nema preklapanja frekvencija koje su dodijeljene.

Na kraju svake iteracije, uzima se najbolje rješenje koje je konstruirano u toj iteraciji i dobrota tog rješenja, odnosno cijena preklapanja između poveznica i frekvencija u tom rješenju se zapisuje u datoteku. U datoteci se nalazi onolik broj podataka koliko se iteracija programa izvelo. Podaci se jednostavno mogu kopirati iz datoteke i iz njih se lako kreira dijagram koji prikazuje kako se mijenja vrijednost rješenja dok se ne postigne optimalno rješenje



Slika 8. Primjer zadavanja problema u programu [autorski rad]

Slika iznad prikazuje kako se zadaje problem u programu koji implementira optimizaciju kolonijom mrava, inačicu MAKS-MIN sustav mrava. Kada korisnik završi s unosom problema, program traži optimalno rješenje i podatke zapisuje u datoteku. Algoritam je implementiran u programskom jeziku *C#* korištenjem programskog alata *Microsoft Visual Studio 2017* na operacijskom sustavu *Microsoft Windows 10.*



Slika 9. Primjer rješenja za jednog mrava [autorski rad]

Druga slika prikazuje primjer rješenja koje je konstruirao neki mrav. Rješenje se sastoji od indeksa poveznice i indeksa frekvencije koja je dodijeljena toj poveznici. U ovom primjeru, vidimo da je mrav s indeksom 0 u polju mrava konstruirao rješenje u kojem je poveznici s indeksom 0 dodijelio frekvenciju s indeksom 4, poveznici s indeksom 1 je dodijelio frekvenciju s indeksom 2 i poveznici s indeksom 2 je dodijelio frekvenciju s indeksom 1.

5. Eksperimentalna istraživanja

Implementirani algoritam testirati ćemo na dva problema, jedan čije je optimalno rješenje jednako nuli, tj. nema preklapanja frekvencija koje su dodijeljene poveznicama u rješenju, dok je rješenje drugog problema minimalna suma kazne zbog preklapanja kada se poveznicama dodijele frekvencije. Parametri α, β i ρ biti će postavljeni na sljedeće vrijednosti: $\alpha = 1, \beta = 1, \rho = 0.2$. Korištenjem ovih parametara postignuto je da algoritam pronalazi optimalno rješenje unutar nekoliko stotina iteracija, ovisno o veličini problema koji smo zadali. Broj mrava koji će konstruirati rješenja postavljen je na 5, broj iteracija koje će program izvesti je postavljen na 1000, dok broj poveznica i broj frekvencija ovisi o problemu.

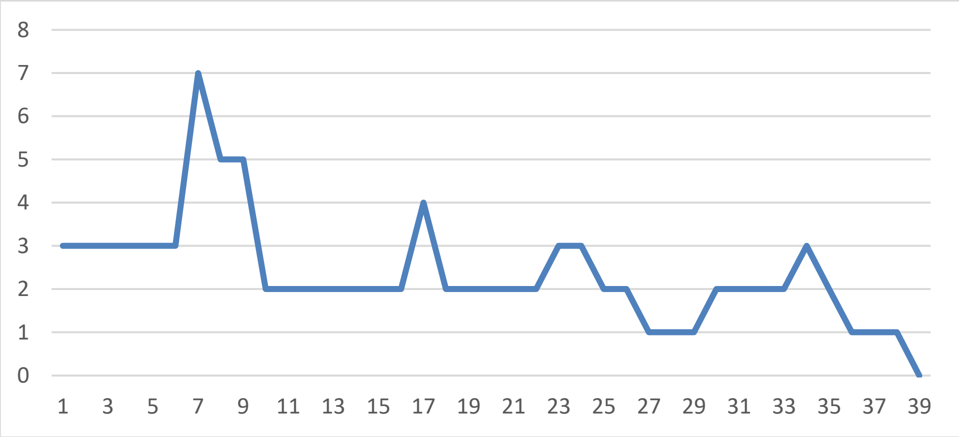
Za prvu vrstu problema, čija ukupna suma kazne za preklapanje frekvencija dosegne vrijednost nula, postavili smo broj poveznica na 5, a broj frekvencija koje je moguće dodijeliti na broj 9. Matrica koja sadrži podatke o tome koliki mora biti razmak između frekvencija kada odaberemo bilo koje dvije poveznice u ovom je slučaju veličine 5×5. Konačno rješenje koje će mravi konstruirati biti će vektor veličine 5 u kojem će se nalaziti brojevi frekvencija koje su dodijeljene poveznicama. U nastavku slijede podaci koji su korišteni za ovaj problem.

$$M = \begin{bmatrix} 10 & 2 & 0 & 1 & 0 \\ 1 & 10 & 0 & 0 & 1 \\ 0 & 3 & 10 & 0 & 4 \\ 2 & 0 & 1 & 10 & 1 \\ 3 & 4 & 0 & 1 & 10 \end{bmatrix}$$

$$brojPoveznica = 5, \quad brojFrekvencija = 9, \quad brojMrava = 5, \quad brojIteracija = 200,$$

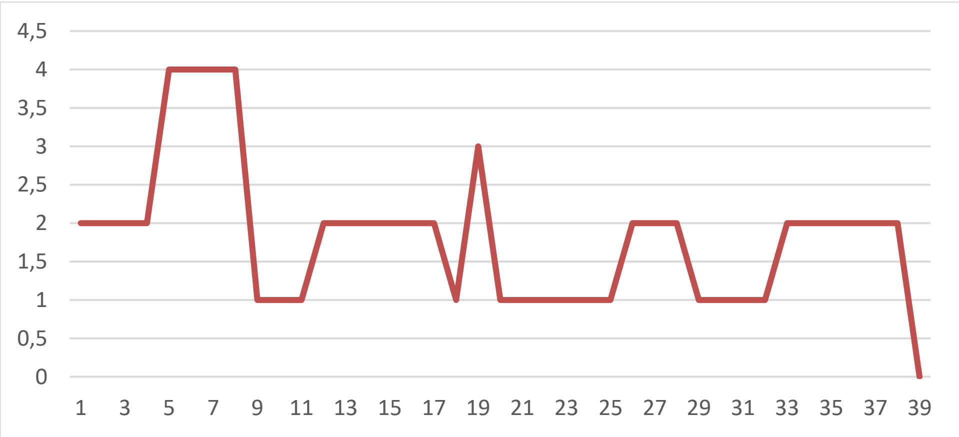
$$\alpha = 1, \quad \beta = 1, \quad \rho = 0.2$$

Program je pokrenut tri puta sa ovim postavkama, a rezultati koji su dobiveni svakim pokretanjem prikazani su na dijagramima. Prva tri dijagrama prikazuju rezultate pojedinačnih testova dok zadnji dijagram prikazuje rezultate sva tri eksperimenta zajedno.



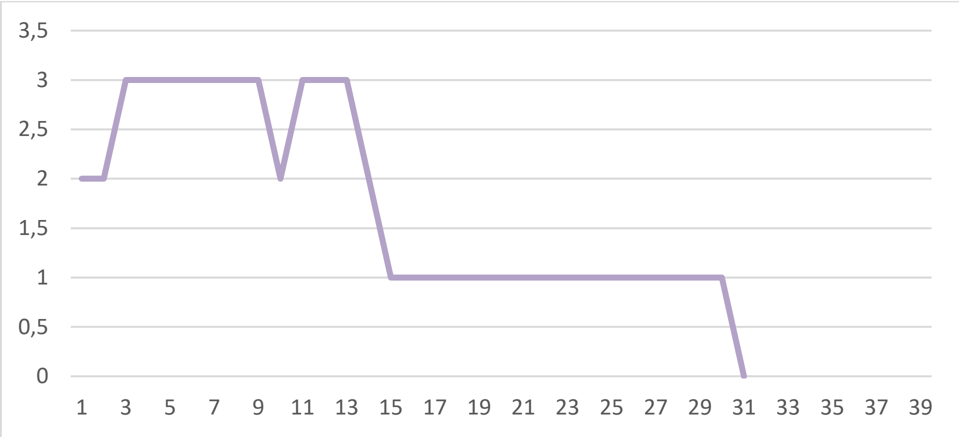
Dijagram 1. Prvi eksperiment – prvi problem

U ovom slučaju, najbolje rješenje koje je algoritam pronašao u prvoj iteraciji ima cijenu 3 te od tog rješenja ažurira feromonski trag. U sljedećim iteracijama koristi ažurirani feromonski trag za pronalaženje rješenja. Zbog heurističke informacije moguće je da se izabere lošije rješenje, upravo to se dogodilo kada je cijena najboljeg rješenja skočila na 7. Algoritam tada nastavlja rad i učenje, na kraju pronalazi optimalno rješenje unutar 40-ak iteracija, čija je cijena 0.



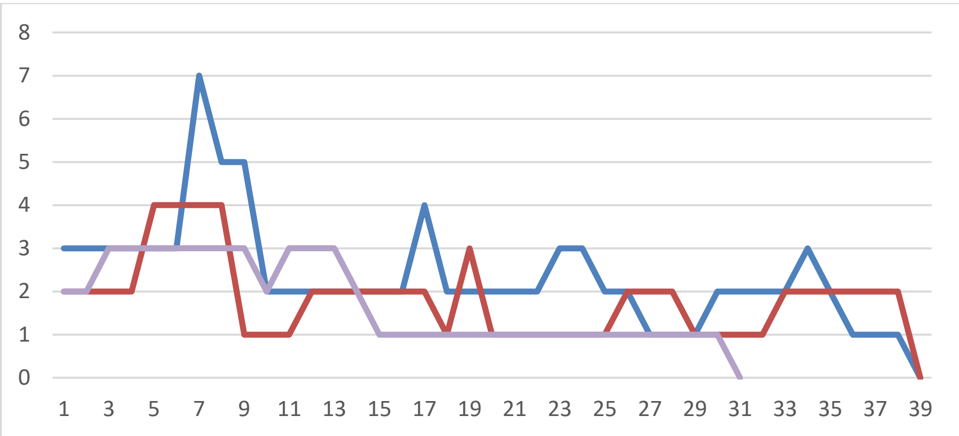
Dijagram 2. Drugi eksperiment – prvi problem

Drugi eksperiment započinje rješenjem čija je cijena 2. Algoritam izvođenjem dolazi do rješenja čija je cijena između 1 i 4, te se ta rješenja izmjenjuju dok se ne postigne optimalno rješenje.



Dijagram 3. Treći eksperiment – prvi problem

Treći eksperiment pokazao se kao najbrži, algoritam je krenuo s rješenjem čija je cijena 2 kao i kod drugog eksperimenta, ali mu je kasnije zbog boljeg odabira poveznica bilo potrebno izvršiti 31 iteraciju da dođe do optimalnog rješenja.



Dijagram 4. Svi eksperimenti – prvi problem

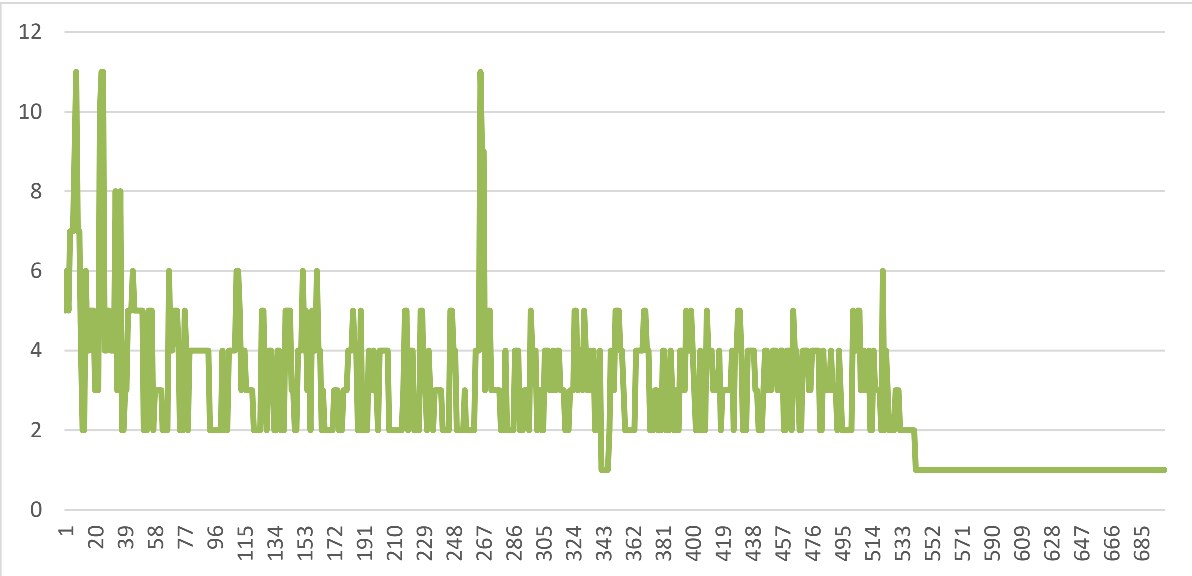
Na ovom dijagramu možemo vidjeti da je algoritam u svim eksperimentima krenuo od rješenja čija je cijena oko 2, te je birao rješenja čija se cijena kreće između 1 i 7 dok nije došao do optimalnog rješenja. Ovisno o izboru rješenja, za pronalazak optimalnog rješenja u nekim eksperimentima trebalo mu je više iteracija, dok mu je u drugim eksperimentima trebalo manje ali algoritam je s ovim postavkama za ovaj problem rješenje pronašao unutar 40 iteracija.

Druga vrsta problema je problem gdje ne postoji rješenje u kojem nema preklapanja frekvencija te je zadatak algoritma odrediti optimalno rješenje gdje je cijena preklapanja najmanja. Da bi se postigao takav problem, potrebno je u matrici koja sadrži podatke o cijeni preklapanja većinu vrijednosti postaviti da su različite od nule. Na taj način, suma nikako ne može postati nula, te će se program izvršavati dok ne izvrši onoliko iteracija koliko je zadano. Rješenje koje će program dati je rješenje u kojem je cijena preklapanja frekvencija minimalna. Za ovaj problem koristili smo sljedeće podatke:

$$M = \begin{bmatrix} 10 & 2 & 4 & 1 & 0 \\ 1 & 10 & 2 & 5 & 1 \\ 3 & 3 & 10 & 2 & 4 \\ 0 & 5 & 1 & 10 & 1 \\ 3 & 4 & 0 & 1 & 10 \end{bmatrix}$$

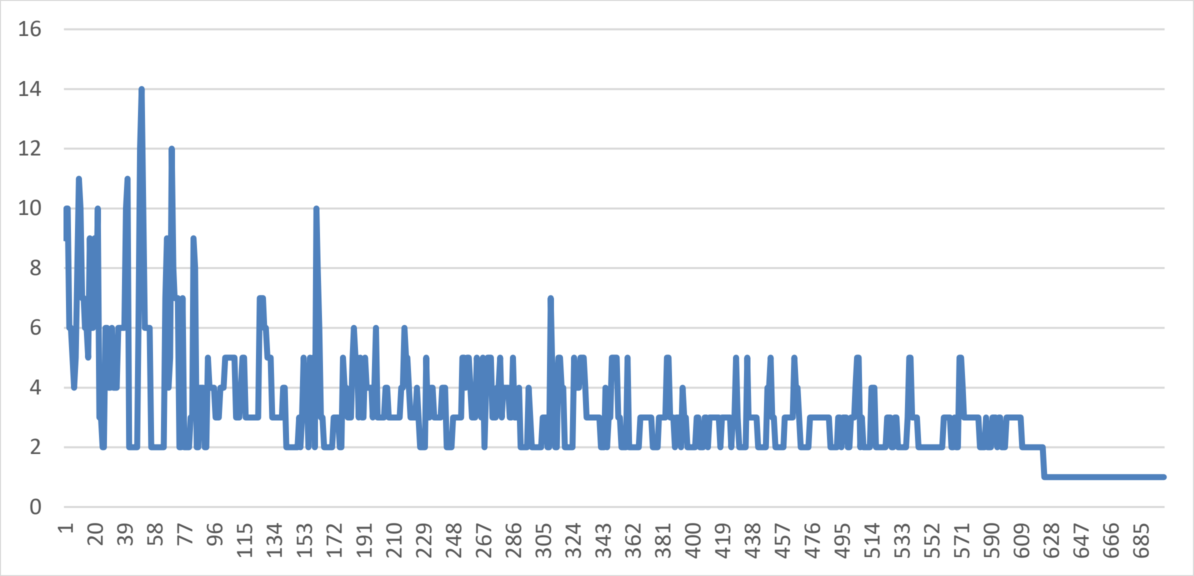
$$brojPoveznica = 5, \quad brojFrekvencija = 9, \quad brojMrava = 5, \quad brojIteracija = 1000,$$
$$\alpha = 1, \quad \beta = 1, \quad \rho = 0.2$$

Program je s ovim postavkama pokrenut tri puta, a rezultati su prikazani na sljedećim dijagramima. Na prva tri dijagrama prikazana su pojedinačna rješenja, dok su na četvrtom dijagramu prikazana sva tri rješenja kako bi se mogli lakše usporediti.



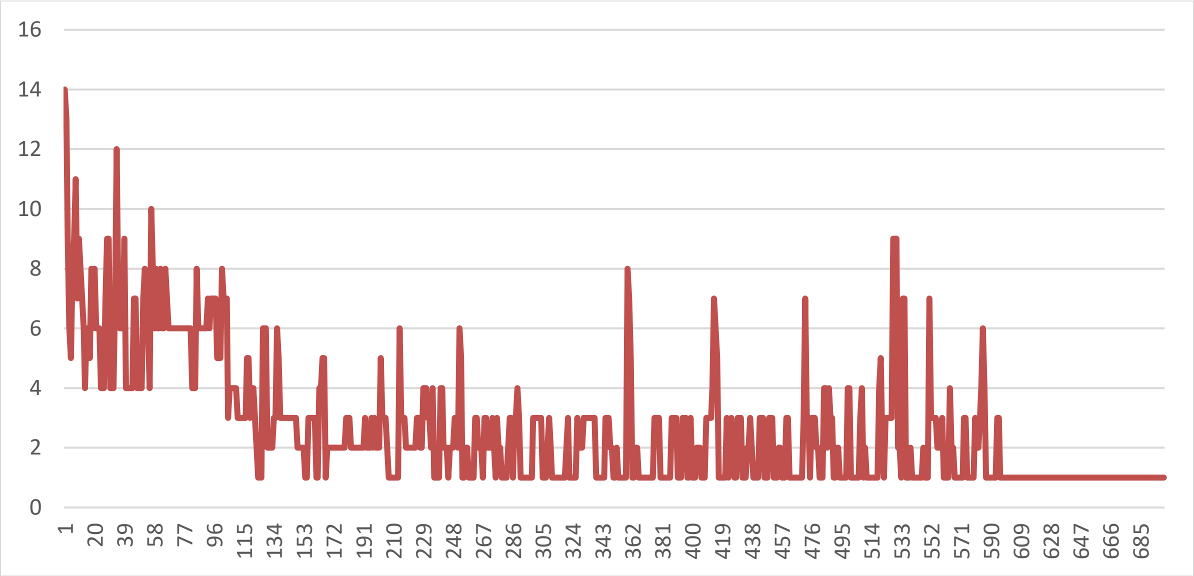
Dijagram 5. Prvi eksperiment – drugi problem

Prvi eksperiment pokazao je kako je optimalno rješenje za ovaj problem kada je cijena preklapanja jednaka 1. Ovaj eksperiment dostiže optimalno rješenje oko 600-te iteracije. Iako je u postavkama broj iteracija postavljen na 1000, na ovom dijagramu je prikazano samo prvih 700 iteracija jer se nakon toga ponavlja rješenje 1.



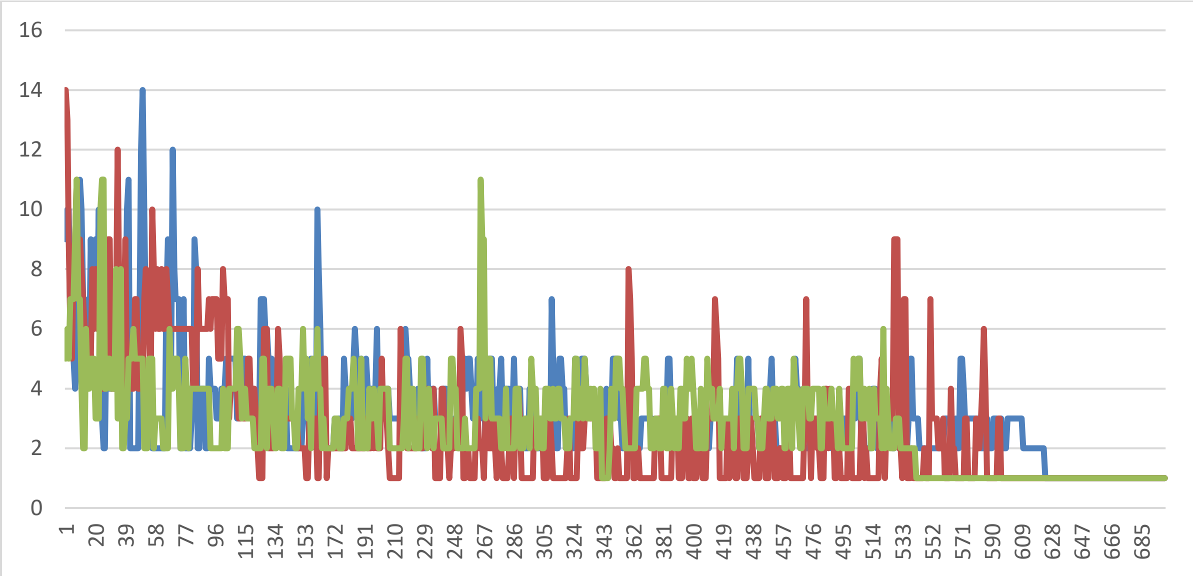
Dijagram 6. Drugi eksperiment – drugi problem

Drugi eksperiment također je doveo do optimalnog rješenja gdje je cijena 1. Na ovom dijagramu vidi se da heuristička informacija može nadjačati vrijednost traga te se u nekim iteracijama izabiru lošija rješenja. Također je prikazano samo prvih 700 iteracija zbog ponavljanja optimalnog rješenja.



Dijagram 7. Treći eksperiment – drugi problem

Treći eksperiment više je puta pronašao optimalno rješenje s cijenom 1, ali je u sljedećim iteracijama odabirao lošija rješenja sve dok nije krenuo ponavljati optimalno rješenje. Prikazano je samo prvih 700 iteracija zbog ponavljanja optimalnog rješenja.



Dijagram 8. Svi eksperimenti – drugi problem

Na ovom dijagramu prikazani su svi eksperimenti zajedno, vidljivo je kako su početne cijene algoritama između 6 i 14 te kako algoritam konstruira rješenja dok ne dođe do optimalnog. Za pronalazak optimalnog rješenja, algoritmu je bilo potrebno između 530 i 640 iteracija te je nakon toga krenuo ponavljati cijenu optimalnog rješenja do zadanog broja iteracija koji je bio 1000.

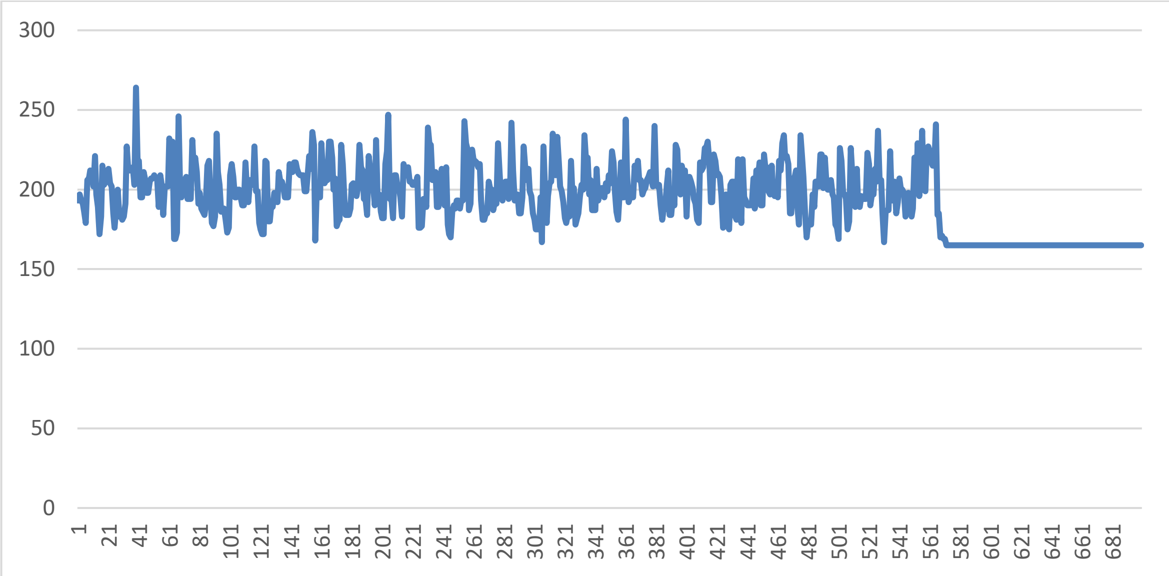
Prethodna dva problema su manjih dimenzija koji su pokazali kako algoritam radi i pronalazi rješenje. Problemi koji se rješavaju ovim pristupom većih su razmjera te je zbog toga proveden eksperiment na problemu većih dimenzija. Problem je zadan sljedećim podacima:

$$\begin{bmatrix} 20 & 12 & 6 & 7 & 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 20 & 3 & 6 & 4 & 0 & 18 & 0 & 3 & 2 \\ 10 & 19 & 20 & 2 & 0 & 3 & 1 & 16 & 13 & 0 \\ 3 & 3 & 15 & 20 & 4 & 14 & 0 & 0 & 0 & 6 \\ 0 & 0 & 2 & 1 & 20 & 0 & 3 & 13 & 14 & 5 \\ 5 & 17 & 5 & 6 & 17 & 20 & 0 & 3 & 0 & 4 \\ 7 & 8 & 9 & 0 & 0 & 3 & 20 & 11 & 1 & 5 \\ 9 & 5 & 1 & 3 & 4 & 7 & 16 & 20 & 13 & 12 \\ 0 & 7 & 16 & 0 & 5 & 8 & 9 & 11 & 20 & 0 \\ 16 & 0 & 0 & 9 & 8 & 1 & 6 & 0 & 6 & 20 \end{bmatrix}$$

$$brojPoveznica = 10,\ brojFrekvencija = 16,\ brojMrava = 5,\ brojIteracija = 1000,$$

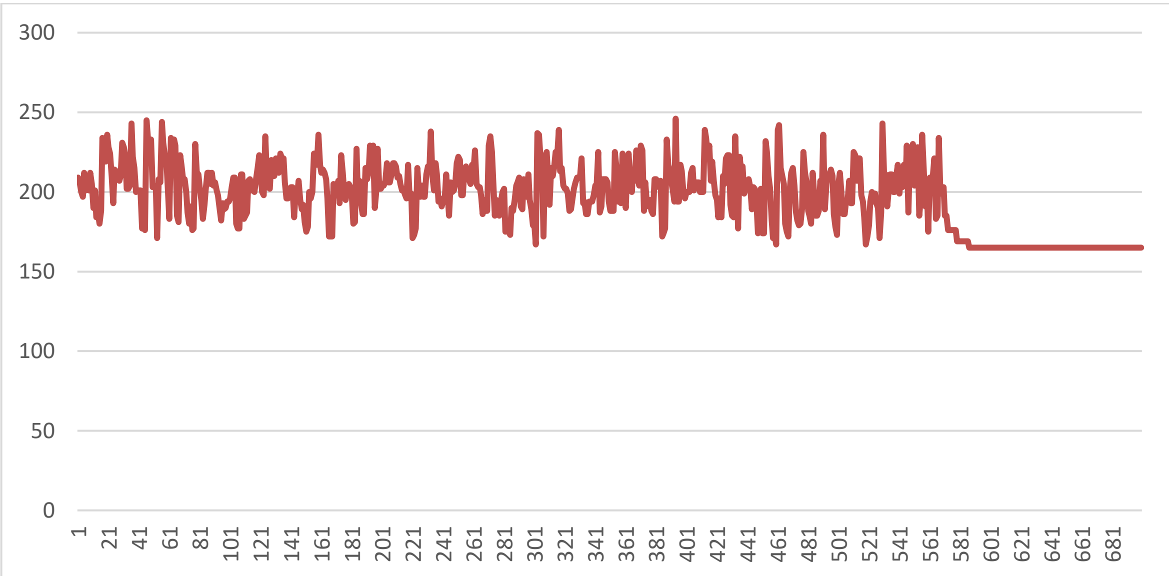
$$\alpha = 1, \qquad \beta = 1, \qquad \rho = 0.2$$

Program je ovim postavkama pokrenut tri puta, a rezultati dobiveni su prikazani na slijedećim dijagramima.



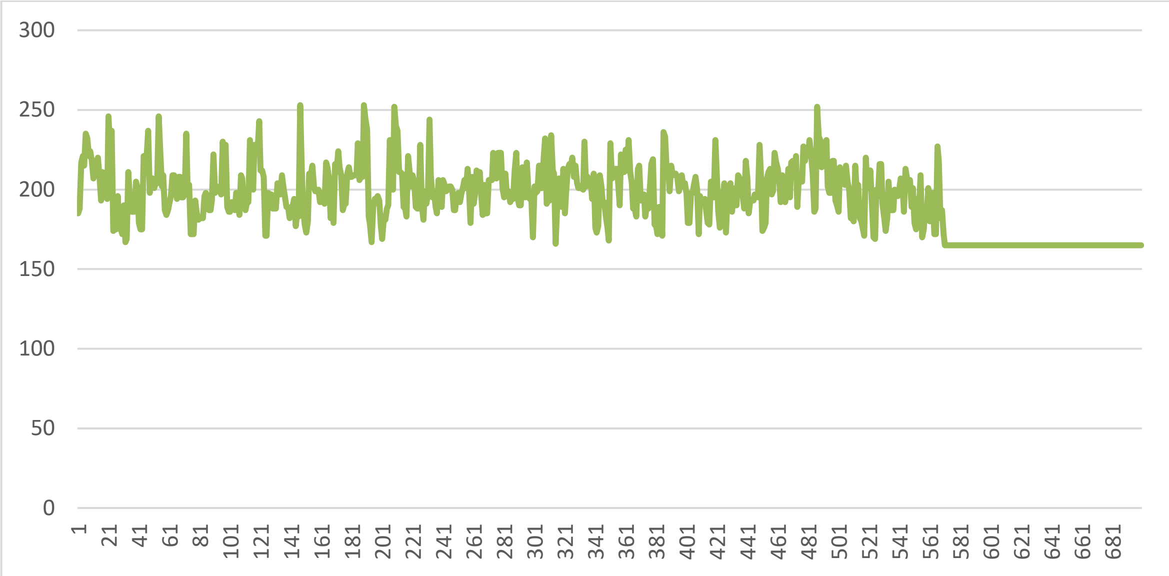
Dijagram 9. Prvi eksperiment – treći problem

U prvom eksperimentu pronađeno je najbolje rješenje čija je cijena 165 oko 580-te iteracije te se nakon toga ponavlja rješenje. Iz tog razloga na dijagramu je prikazano samo 700 iteracija od izvedenih 1000.



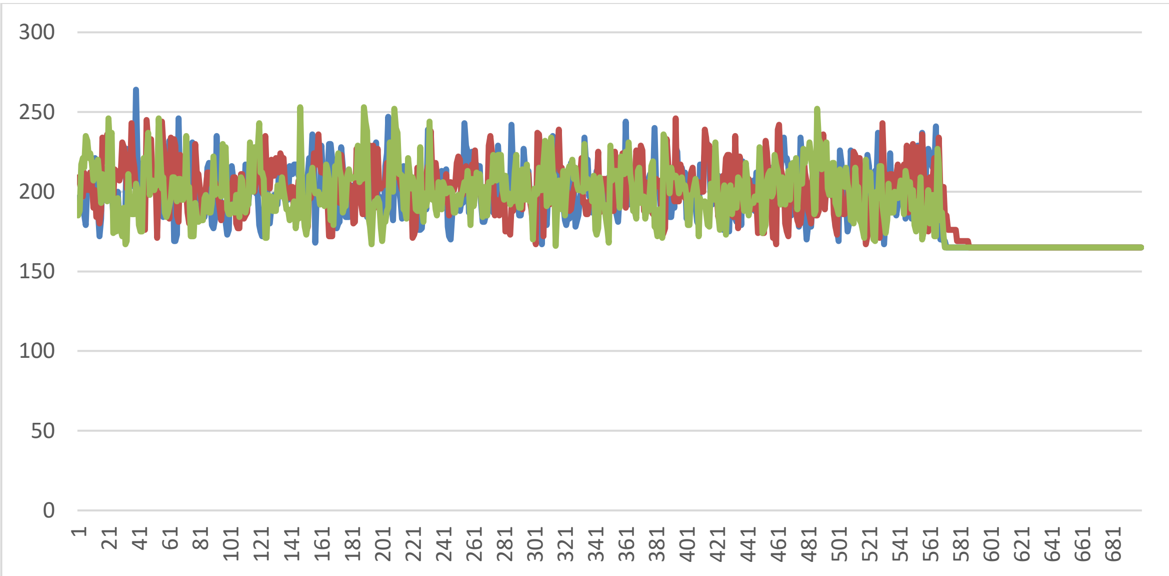
Dijagram 10. Drugi eksperiment – treći problem

U drugom eksperimentu najbolje rješenje pronađeno je oko 570-te iteracije te se to rješenje ponavlja do izvršenja 1000 iteracija. Iz tog razloga na dijagramu je prikazano samo 700 iteracija od izvedenih 1000.



Dijagram 11. Treći eksperiment – treći problem

U trećem eksperimentu program je do najboljeg rješenja došao nešto brže nego u prethodna dva eksperimenta, oko 550-te iteracije. Najbolje rješenje koje je program pronašao ima cijenu 165 te se to rješenje ponavlja do kraja. Iz tog razloga, na dijagramu je prikazano prvih 700 iteracija.



Dijagram 12. Svi eksperimenti – treći problem

Na ovom dijagramu prikazani su svi eksperimenti zajedno. Vidljivo je kako su se početna rješenja koja je program našao kretala oko cijene 200 te je program kroz daljnje iteracije pronalazio različita rješenja sve dok nije pronašao najbolje rješenje. Ovo rješenje možda nije optimalno, ali je najbolje koje je algoritam našao.

5.1. Performanse algoritma

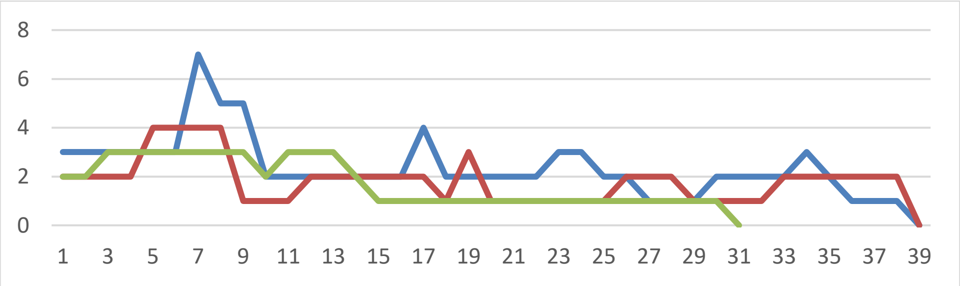
Prije izvršavanja programa unose se određeni parametri koji utječu na performanse algoritma. U prethodnim eksperimentima parametri α i β postavljeni su na 1, dok je ρ postavljen na 0.2. Broj mrava uvelike utječe na performanse algoritma, ako je veći broj mrava, veća je vjerojatnost da će jedno od rješenja koje su konstruirali biti optimalno. Broj poveznica i frekvencija također utječe na performanse, no ti parametri su vezani uz problem koji se rješava. Ovdje ćemo prikazati rezultate eksperimenata za prvi problem uz parametar $\alpha = 0.5$ kako bi pokazali kako parametar α utječe na performanse, a zatim ćemo broj mrava postaviti na 10 da vidimo kako broj mrava utječe na performanse algoritma. Promjenom parametra α problem izgleda ovako:

$$M = \begin{bmatrix} 10 & 2 & 0 & 1 & 0 \\ 1 & 10 & 0 & 0 & 1 \\ 0 & 3 & 10 & 0 & 4 \\ 2 & 0 & 1 & 10 & 1 \\ 3 & 4 & 0 & 1 & 10 \end{bmatrix}$$

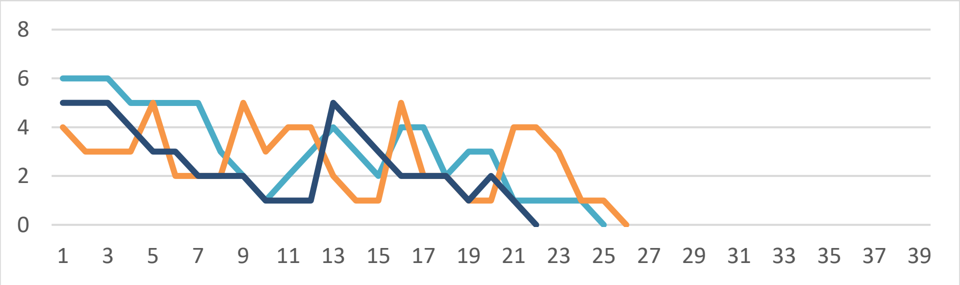
$$brojPoveznica = 5, \quad brojFrekvencija = 9, \quad brojMrava = 5, \quad brojIteracija = 200,$$

$$\alpha = 0.5, \quad \beta = 1, \quad \rho = 0.2$$

Program je pokrenut tri puta s ovim postavkama te su rezultati prikazani na jednom dijagramu dok su rezultati dobiveni ranije prikazani na drugom dijagramu. Usporedbom tih dijagrama vidljivo je da parametri koje unosimo imaju značajan utjecaj na performanse algoritma. Usporedbom ova dva dijagrama vidljivo je kako eksperimenti gdje je parametar α postavljen na 0.5 pronalaze optimalno rješenje brže, unutar 30 iteracija.



Dijagram 13. Rezultati prvog eksperimenta



Dijagram 14. Rezultati eksperimenta s parametrom $\alpha = 0.5$

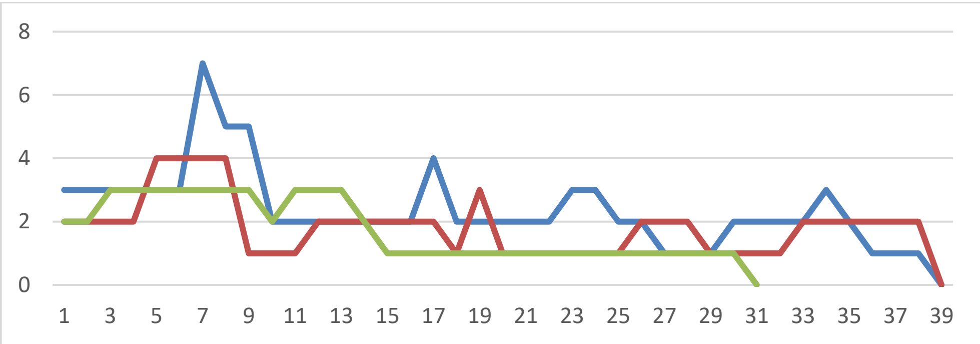
Na prethodnim eksperimentima pokazano je kako parametar α utječe na performanse algoritma. Proveden je još jedan eksperiment kojim je pokazano kako broj mrava koji konstruiraju rješenja utječe na performanse algoritma. Promjenom broja mrava problem dobiva slijedeći oblik:

$$M = \begin{bmatrix} 10 & 2 & 0 & 1 & 0 \\ 1 & 10 & 0 & 0 & 1 \\ 0 & 3 & 10 & 0 & 4 \\ 2 & 0 & 1 & 10 & 1 \\ 3 & 4 & 0 & 1 & 10 \end{bmatrix}$$

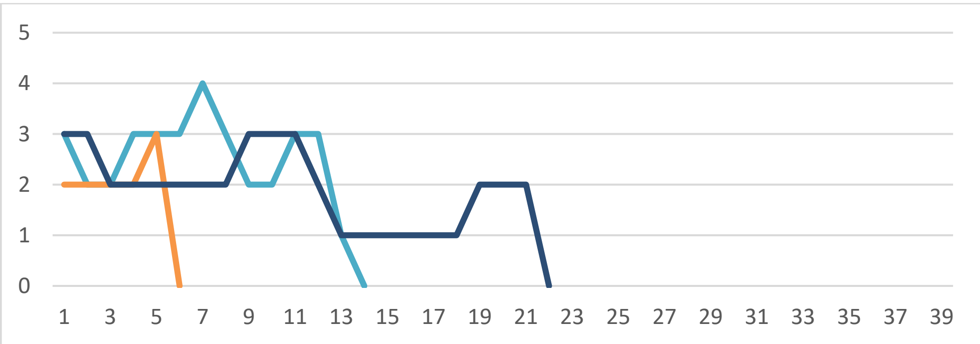
$$brojPoveznica = 5, \quad brojFrekvencija = 9, \quad brojMrava = 10, \quad brojIteracija = 200,$$

$$\alpha = 1, \quad \beta = 1, \quad \rho = 0.2$$

Program je pokrenut tri puta s ovim postavkama. Dobiveni rezultati prikazani su na jednom dijagramu te su na drugom prikazani rezultati ranijih eksperimenata za usporedbu. Usporedbom ova dva dijagrama vidljivo je kako algoritam pronalazi optimalno rješenje brže ako postavimo broj mrava na veći broj jer je vjerojatnost da će neki mrav pogoditi optimalno rješenje veća. Također je vidljivo da su cijene najboljih rješenja u svakoj iteraciji manje jer više mrava konstruira rješenja.



Dijagram 15. Rezultati prvog eksperimenta



Dijagram 16. Rezultati eksperimenta gdje je $brojMrava = 10$

6. Zaključak

Bežična komunikacija je u današnje vrijeme postala nezamisliva bez korištenja radiovalova za prijenos signala i poruka. Radiovalovi su vrsta elektromagnetskog zračenja koje se javlja protjecanjem električne struje kroz vodič. Kako se broj uređaja koji komuniciraju bežično povećavao sve više, postalo je jasno da će u nekom trenutku doći do zasićena i da će uređaji morati dijeliti frekvencije zbog ograničenosti frekvencijskog pojasa koji je dostupan za bežičnu komunikaciju. Tu se pojavio problem dodjeljivanja frekvencije komunikacijskom kanalu. Problem se sastoji od toga da postoji određeni broj poveznica kojima je potrebno dodijeliti frekvencije na način da se izbjegne preklapanje frekvencije u što većoj mjeri. Ometanje će se pojaviti ukoliko su uređaji koji bežično komuniciraju geografski blizu smješteni te koriste sličnu frekvenciju prema frekvencijskom pojasu. Kako bi se izbjeglo ometanje, važno je dobro rasporediti frekvenciju između uređaja. Kako je komunikacija dvosmjerna, potrebno je osigurati posebne frekvencije za svaki smjer komunikacije između odašiljača i prijemnika. Problem dodjeljivanja frekvencije javlja se u nekoliko primjena, a to su mobilna telefonija, vojne primjene, odašiljanje radio i TV signala te satelitska komunikacija. Za svaku od ovih primjena postoje specifični problemi koji se javljaju samo kod tih primjena. Iz svake primjene proizlazi nekoliko vrsta problema dodjeljivanja frekvencije komunikacijskom kanalu pa tako problem dodjeljivanja frekvencije ima mnogo oblika. Ovaj problem spada u grupu NP-teških problema, što znači da se ovakvi problemi rješavaju posebnim optimizacijskim algoritmima. Jedan od takvih algoritama je i optimizacija kolonijom mrava. Ovaj algoritam oponaša ponašanje mrava u potrazi za hranom. Mravi kod napuštanja gnijezda mogu krenuti u bilo kojem smjeru i tek kada nađu hranu vraćaju se u gnijezdo, pritom otpuštajući feromone kako bi drugim mravima dalji do znanja da su na tom putu pronašli hranu. Mravi koji izlaze iz gnijezda nakon što su se prvi vratili i ostavili trag, imaju izbor da prate njihov trag feromona ili da krenu svojim putem što odlučuju ovisno o jačini traga. Kako više mrava izabire isti put, trag feromona jača. Eksperiment s dvostrukim mostom pokazao je kako će u vrlo kratkom roku mravi izabrati kraći put do hrane. Sličan princip koristi i algoritam optimizacije kolonijom mrava. Ovaj algoritam koristi trag feromona kako bi ostavio više feromona na rješenjima koja su dala bolje rezultate, odnosno imala veću dobrotu. Kako trag feromona za dobra rješenja raste, algoritmi u kasnijim iteracijama odabiru baš rješenja koja imaju jači trag te tako dovode do optimalnog rješenja s najvećom dobrotom. S obzirom na konstrukciju rješenja i ažuriranje feromonskog traga, postoje razne inačice algoritma. Prvi algoritam optimizacije kolonijom mrava razvio je Marco Dorigo u svojoj doktorskoj disertaciji. Nakon njega, pojavile su se razne inačice algoritma koje su imale bolje performanse i davale bolja rješenja. U ovome radu implementiran je algoritam MAX-MIN sustav mrava (*MAX-MIN Ant System*) te su rezultati njegovog izvođenja prikazani na dijagramima. Dijagrami prikazuju kako algoritam uči te od početnog lošeg rješenja dolazi do optimalnog ili najboljeg mogućeg rješenja.

7. Popis literature

[1] E. Bonabeau, C. Meyer, „Swarm Intelligence: A Whole New Way to Think About Business“, 2001. [Na internetu]. Dostupno: https://hbr.org/2001/05/swarm-intelligence-a-whole-new-way-to-think-about-business [pristupano: 29.06.2018.]

[2] A.M.C.A Koster, „Frequency Assigment – Models and Algorithms“ [Doktorska disertacija]. Universiteit Maastricht, Nizozemska 1999, Dostupno: https://www.math2.rwth-aachen.de/files/mitarbeiter/koster/Koster1999.pdf [pristupano: 29.06.2018.]

[3] „Radiovalovi“ (bez dat.) u *Wikipedija, slobodna enciklopedija*. Dostupno: https://hr.wikipedia.org/wiki/Radiovalovi [pristupano: 29.06.2018.]

[4] N. Ivković, „Modeliranje, analiza i poboljšanje algoritama optimizacije kolonijom mrava“ [Doktorska disertacija], Zagreb 2014.

[5] D. Klemen, „Optimizacija kolonijom mrava“, Zagreb, 2008.

[6] K.I. Aardal, S.P.M. van Hoesel, A.M.C.A. Koster, C. Mannino, A. Sassano, „Models and solution techniques for frequency assignment problems“, *4OR* svibanj 2007, [Na internetu]. Dostupno: https://link.springer.com/content/pdf/10.1007%2Fs10479-007-0178-0.pdf [pristupano: 29.06.2018.]

[7] „NP-completeness“ (bez dat.) u *Wikipedija, slobodna enciklopedija*. Dostupno: https://en.wikipedia.org/wiki/NP-completeness [pristupano: 13.08.2018.]

[8] „Signal-to-noise ratio“ (bez dat.) u *Wikipedija, slobodna enciklopedija.* Dostupno: https://en.wikipedia.org/wiki/Signal-to-noise_ratio [pristupano: 14.08.2018.]

[9] I.A. Aldaya, G. Castanon, „Real-time traffic, handoff, and outage modeling in high RF pico cells“, 2011. [Na internetu]. Dostupno: https://www.researchgate.net/publication/220972053_Real-time_traffic_handoff_and_outage_modeling_in_high_RF_pico_cells [pristupano: 27.09.2018.]

[10] M.S. Maximiano, „Appying Multiobjective Metaheuristics to the Frequency Assignment Problem in GSM Networks“, [Doktorska disertacija], Cáceres, 2011.

8. Popis slika

9. Popis dijagrama