

Univerzitet u Beogradu

Matematički fakultet

Analiza uticaja operatora ukrštanja na performanse genetskog
algoritma za rešavanje prostog lokacijskog problema

master rad

Student:
Kosana Protić

Mentor:
prof. dr Zorica Stanimirović

Beograd
2015.

Mentor: **prof. dr Zorica Stanimirović**
Matematički fakultet
Univerzitet u Beogradu

Članovi komisije: **prof. dr Dušan Tošić**
Matematički fakultet
Univerzitet u Beogradu

doc. dr Miroslav Marić
Matematički fakultet
Univerzitet u Beogradu

Datum odbrane: _____

Analiza uticaja operatora ukrštanja na performanse genetskog algoritma za rešavanje prostog lokacijskog problema

Rezime - Adekvatan izbor operatora ukrštanja ima važnu ulogu za uspešno rešavanje zadanog problema optimizacije genetskim algoritmom. U ovom radu biće razmatrano nekoliko operatora ukrštanja kao jednog od elementata genetskog algoritma za rešavanje prostog lokacijskog problema koji je poznati NP-težak problem iz literature. Biće analizirani slučajevi primene različitih operatora ukrštanja za binarne i nebinarne reprezentacije rešenja. Dobijeni rezultati biće upoređeni u cilju adekvatnog izbora operatora ukrštanja za svaku od razmatranih reprezentacija rešenja.

Ključne reči: *genetski algoritam, prost lokacijski problem, problem optimizacije, operatori ukrštanja, NP-teški problemi*

The analysis of impact of crossover operator on the performance of a genetic algorithm for solving Simple Plant Location Problem

Abstract - An adequate selection of crossover operators plays an important role in a successful solving of a given optimisation problem by genetic algorithm. In this paper, several crossover operators will be examined as the elements of a genetic algorithm for solving a Simple Plant Location Problem, which is known as an NP-hard problem in the literature. Different crossover operators for binary and non-binary representations will be examined. The results will be compared in order that an adequate crossover operator is chosen for each of the considered solution representations.

Keywords: *Genetic Algorithm, Simple Plant Location Problem, Optimization problems, Crossover operator, NP-hard problems*

PREDGOVOR

Većina optimizacionih problema pripada klasi NP-teških problema. Za neke od njih postoje algoritmi čija je složenost, u najboljem slučaju, eksponencijalna. U praksi je zapaženo da često ipak nije potrebno egzaktno rešenje nekog problema već je dovoljno i približno rešenje. To za posledicu ima razvoj raznih heurističkih i metaheurističkih metoda. Međutim, ove metode ne pružaju garanciju da će se uvek doći do optimalnog rešenja. Ipak, one za relativno kratko vreme mogu dati kvalitetna rešenja koja često i jesu optimalna ali se to ne može verifikovati (verifikacija rešenja je najčešće višestruko složenija od samog dobijanja takvog rešenja).

Lokacijski problemi predstavljaju posebnu klasu zadataka optimizacije. U širem smislu, odnose se na određivanje pozicije jednog ili grupe objekata u prostoru određene dimenzije. U užem smislu, ovi problemi odnose se na lociranje resursa, skladišnih objekata, terminala, pretovarnih mesta, odnosno reč je o zadacima lociranja tačke u dvodimenzionom prostoru. Rešavanje ovih zadataka povezano je kako sa visokim investicionim ulaganjima tako i sa značajnim rizicima (finansijski, ekološki, ...). Svest o značaju lokacijskih problema inicirala je veliki broj istraživanja, a rezultat toga je više od 3000 publikovanih radova iz ove oblasti u svetskoj literaturi ([Bar97], [Top05], [Vyg05], [Dre06], [Ban10]). U radu [Rev05] može se pronaći pregled većine poznatih lokacijskih modela i metoda za njihovo rešavanje.

Problemi neograničenih kapaciteta pretpostavljaju da svako postrojenje može proizvesti i distribuirati neograničenu količinu robe koja se razmatra. Više matematičkih modela bilo je predloženo za ovaj problem, a prvi koji se u literaturi pominje datira iz 60-ih godina XX veka, kada je definisan prost lokacijski problem (Simple Plant Location Problem, [Kue63], [Bal65]). 1983. godine Krarup i Pruzan su objavili obiman pregled radova o prostom lokacijskom problemu kao i svojista dotadašnjih rešenja.

U ovom radu, za rešavanje prostog lokacijskog problema, korišćena je metaheuristika genetski algoritam. Cilj rada je analiza uticaja primene različitih tipova operatora ukrštanja na performanse genetskog algoritma. Kako je odgovarajući način kodiranja jedan od najbitnijih aspekata genetskog algoritma, u ovom radu predstavljena su tri načina kodiranja rešenja: binarno kodiranje, koje se prirodno nameće za ovaj problem, zatim celobrojno kodiranje i kombinacija prethodna dva. Za svaki od razmatranih načina kodiranja korišćeni su različiti operatori ukrštanja i analiziran je njihov uticaj na performanse algoritma.

Rad se sastoji od predgovora, 7 poglavlja, spiska literature i sadržaja. U prvom poglavlju opisan je prost lokacijski problem i dat je pregled postojećih metoda iz literature za njegovo rešavanje. Detaljan opis genetskog algoritma i pregled primena u praksi nalazi se u drugom poglavlju. Način implementacije GA sa binarnim kodiranjem i različitim operatorima ukrštanja, uz pregled postignutih rezultata, opisan je u trećem poglavlju. U četvrtom, odnosno petom poglavlju prikazan je način implementacije GA sa celobrojnim, odnosno mešovitim kodiranjem i različitim operatorima ukrštanja, uz pregled postignutih rezultata. Analiza performansi GA sa

različitim načinima kodiranja u odnosu na operator ukrštanja predstavljena je u šestom poglavlju. Zaključna razmatranja data su u sedmom poglavlju.

Želim da istaknem zahvalnost mentoru, prof. dr Zorici Stanimirović koja me je zainteresovala za genetske algoritme i korisnim savetima i razumevanjem pomogla u realizaciji ovog rada.

Takođe, dugujem zahvalnost prof. dr Dušanu Tošiću i doc. dr Miroslavu Mariću na sugestijama koje su doprinele poboljšanju kvaliteta rada.

Posebno se zahvaljujem svojoj porodici i prijateljima na ogromnoj podršci koja mi je nemerljivo značila tokom izrade ovog rada.

Beograd, 2015.

Kandidat
Kosana Protić 1047/2012

1. PROST LOKACIJSKI PROBLEM

Prost lokacijski problem u literaturi se javlja sa više različitih naziva na engleskom jeziku - Simple Plant Location Problem (SPLP), Uncapacitated Facility Location Problem (UFLP), Uncapacitated Warehouse Location Problem (UWLP), Uncapacitated Plant Location Problem (UPLP). Cilj problema je za dati skup korisnika (kupaca) izabrati skladišta (snabdevače) tako da je suma fiksnih troškova uspostavljanja skladišta i transportnih troškova do svakog korisnika minimalna. Michel Balinski je prvi formulisao prost lokacijski problem u radu [Bal65].

1.1 Matematička formulacija problema

Neka je $I = \{1, 2, \dots, n\}$ dati skup potencijalnih lokacija snabdevača i $J = \{1, 2, \dots, m\}$ dati skup korisnika. Uspostavljanje snabdevača na potencijalnu lokaciju $i \in I$ predstavlja fiksni trošak f_i . Svaki korisnik $j \in J$ potražuje određenu količinu robe b_j , dok c_{ij} podrazumeva cenu transporta robe (po jedinici količine robe) sa i -te lokacije do j -og korisnika. Ne gubeći na opštosti, možemo transportne troškove c_{ij} pomnožiti traženom količinom robe b_j .

Potrebno je rasporediti snabdevače na neke od potencijalnih lokacija i svakog korisnika pridružiti nekom od uspostavljenih snabdevača tako da ukupni troškovi budu minimalni. Pri tome, ukupni troškovi podrazumevaju fiksne troškove uspostavljanja snabdevača i promenljive troškove transporta do svakog korisnika [Cor90].

Matematička formulacija problema koristi 2 skupa promenljivih za lociranje snabdevača i pridruživanje korisnika uspostavljenim snabdevačima:

$$x_{ij} = \begin{cases} 1, & \text{ako je korisniku } j \text{ dodeljen snabdevač } i \\ 0, & \text{u suprotnom} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{ako je snabdevač } i \text{ uspostavljen} \\ 0, & \text{u suprotnom} \end{cases}$$

Koristeći navedenu notaciju SPLP se matematički može zapisati na sledeći način:

$$\min \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n f_i y_i \quad (1)$$

pri uslovima:

$$\sum_{i=1}^n x_{ij} = 1, \text{ za svako } j \in J \quad (2)$$

$$0 \leq x_{ij} \leq y_i, \text{ za svako } i \in I, j \in J \quad (3)$$

$$y_i \in \{0, 1\}, \text{ za svako } i \in I. \quad (4)$$

Formulom (1) predstavljen je osnovni cilj prostog lokacijskog problema odnosno minimizacija troškova uspostavljanja snabdevača i transporta robe do svakog korisnika. Uslov (2) označava da je jednom korisniku dodeljen najviše jedan snabdevač. Uslovom (3) i (4) definiše se binarna priroda promenljivih x_{ij} i y_i .

1.2 Pregled metoda iz literature za rešavanje SPLP-a

U literaturi je poznato da prost lokacijski problem pripada klasi NP-kompletnih problema, a dokaz ovog tvrđenja može se pronaći u radu [Cor90]. Uprkos tome, više desetina egzaktnih i heurističkih metoda je predloženo za njegovo rešavanje.

Većina egzaktnih algoritama su bazirani na metodama rešavanja iz matematičkog programiranja. U radovima [Aqe90] i [Grs94] prikazano je rešavanje SPLP-a u polinomskom vremenu izvršavanja. Beasley je u radu [Bes93] predstavio rezultate eksperimenata sa Lagranžovim heuristikama za slučajeve prostog lokacijskog problema.

Körkel [Kör89] predlaže algoritme bazirane na poboljšanju dualne metode penjanja (Dual Ascent) i odgovarajuće dualne metode poravnanja (Dual Adjustment) za rešavanje dualnog problema relaksacije linearnog programiranja (LP Dual) za SPLP, uz eventualno korišćenje metode grananja i ograničavanja (Branch and Bound) za dobijanje optimalnog rešenja [Erl78]. Rešavanje pomoću dualne metode grananja i ograničavanja prikazano je u radovima [Sim89] i [Hlm95]. Hibridizacija heuristike dodavanja (Add-heuristic) i dualnih metoda prikazana je u radu [Tch88].

SPLP je uspešno rešavan i heurističkim metodama: genetskim algoritmom (Genetic Algorithm) u radovima [Kra99], [Kra01], [Fil00], metodom promenljivih okolina (Variable Neighborhood Search) u radu [Han07], metodom simuliranog kaljenja (Simulated Annealing) u radu [Yig03], Lagranžovom relaksacijom (Lagrangean Relaxation) u radu [Crr06], metodom tabu pretraživanja (Tabu search) u radu [Sun06], metodom mravljih kolonija (Ant Colony) u radu [Kol14], metodom rojeva čestica (Particle Swarm) u radu [Suj11].

2. GENETSKI ALGORITAM

Genetski algoritam (Genetic Algorithm - GA) je metaheuristika koja se koristi za rešavanje brojnih problema kombinatorne i globalne optimizacije simulirajući mehanizam prirodne evolucije populacije jedinki. GA je zasnovan na Darwinovoj teoriji o postanku vrsta [Dar59] nastaloj krajem XIX veka.

Evolucija je prirodni proces prilagođavanja živih bića na svoju okolinu i zakone u prirodi. U prirodi, od skupa jedinki, preživljavaju one koje su najbolje prilagođene snalaženju u okolini siromašnoj sredstvima za život. Najsposobnije jedinke dobijaju priliku da dominiraju slabijima i učestvuju u reprodukciji. Ako poistovetimo meru sposobnosti jedinke da preživi sa naslednim materijalom koji nosi u sebi, genima, tada možemo reći da geni dominantnih jedinki opstaju dok geni slabijih izumiru jer nemaju potomstvo. Pored te pojave, u prirodi pri svakoj reprodukciji dolazi do rekombinacije gena koja uzrokuje različitost među jedinkama iste vrste, ali i sličnosti sa roditeljima jedinke. Takođe, uočava se još jedna pojava ali znatno ređa. Reč je o mutaciji, odnosno slučajnoj promeni genetskog materijala koja nastaje pod dejstvom spoljašnjih uticaja.

Pojavom Holandove knjige „Adaptacija u prirodnim i veštačkim sistemima” [Hil75] počinje aktivno proučavanje prirodnog fenomena adaptacije u računarskim naukama, kao i razvoj algoritama za rešavanje specifičnih problema zasnovanih na toj ideji.

Danas, 40 godina kasnije, genetski algoritmi se u velikoj meri primenjuju za rešavanje različitih klasa problema u kombinatornoj optimizaciji. Oni daju dobre rezultate u području učenja kod neuronskih mreža kao i pri rešavanju brojnih problema kombinatorne i globalne optimizacije. Neki od problema optimizacije koji su u literaturi rešavani genetskim algoritmima su: brojni lokacijski problemi, hab lokacijski problemi, problem pretraživanja, problem trgovačkog putnika, transportni problemi, problemi raspoređivanja procesa, optimizacija upita nad bazom podataka, itd. Pregled osobina genetskih algoritama može se naći u radovima [BD93a], [BD93b], [Müh97], [Kra00], [Sta04], [SZ07a], [Mar10], [Sta11], itd.

2.1 Opis GA

Osnovna jedinica genetskog algoritma jeste jedinka, dok je njegoa osnovna konstrukcija populacija jedinki. U toku rešavanja problema, obično je veličina populacije (broj jedinki) fiksna za sve generacije (često nekoliko desetina ili stotina). Svaka jedinka u populaciji sadrži genetski materijal, odnosno genetski kod i predstavlja moguće rešenje problema u pretraživačkom prostoru rešenja za dati problem. Jedinke mogu biti reprezentovane proizvoljnom strukturom podataka koja opisuje njihova svojstva, a najčešća i najuspešnija reprezentacija jedinki je u vidu nizova bitova (binarno kodiranje). U praksi se koriste i drugi načini reprezentacije jedinki, npr. kodiranje celim ili realnim brojevima.

Nakon adekvatnog izbora načina kodiranja rešenja, sledeći korak GA jeste generisanje početne populacije jedinki. Najčešće se početna populacija generiše na slučajan način, a ponekad se deo ili čak čitava populacija generiše koristeći neki optimizacioni metod. Nakon generisanja početne populacije sledeći bitan korak jeste ocenjivanje kvaliteta jedinke. Funkcija prilagođenosti (fitness function, funkcija cilja) se na određeni način dodeljuje svakoj jedinki i njena vrednost predstavlja meru kvaliteta jedinke. Naredni korak koji se prirodno nameće, prateći evolucionni proces iz prirode, jeste stvaranje nove generacije jedinki. Iz stare populacije formira se nova tako što se deo boljih jedinki, po nekom kriterijumu, izdvoji i prenosi u novu generaciju. Na ostatak populacije primenjuju se genetski operatori koji iz njih formiraju jedinke za novu generaciju. Genetski operatori se dele na unarne, koji stvaraju novu jedinku menjajući manji deo genetskog materijala (mutacija) i operatore višeg reda, koji kreiraju nove jedinke kombinujući osobine nekoliko jedinki (operatori ukrštanja). Nove generacije generišu se tako što GA uzastopno primenjuje operatore selekcije, ukrštanja i mutacije, pri čemu se očekuje da je prosečna prilagođenost jedinki sve bolja i bolja. Algoritam staje kada se ispuni zadati kriterijum zaustavljanja (parametar GA), pri čemu se najbolji član trenutne populacije proglašava rešenjem koje bi trebalo biti optimalno ili bar blizu optimalnog.

Operatorom selekcije deo jedinki se izdvaja za reprodukciju i generisanje nove generacije. Obično se izbor zasniva na funkciji prilagođenosti, i kao i u prirodi, prilagođenije jedinke opstaju dok manje prilagođene jedinke postepeno izumiru.

Operator ukrštanja razlikuje genetski algoritam od ostalih heurističkih metoda. On je binarni operator i deluje na dve (ili više) jedinki-roditelja koje razmenjuju genetski materijal iz čega nastaju kodovi novih (jedne ili više) jedinki-potomaka (sa nasleđenim svojstvima svojih roditelja). Zahvaljujući operatoru ukrštanja nova generacija može potencijalno posedovati raznovrsnije jedinke od onih u prethodnim generacijama.

Operator mutacije je unarni operator i predstavlja slučajnu promenu jednog ili više gena jedinke. Gen mutira sa malom verovatnoćom p_{mut} tako što se njegova vrednost (simbol) zameni nekim drugim simbolom iz azbuke kojom je kodirana jedinka. Mutacija je takođe bitan operator iz više razloga: mutacijom je moguće vratiti izgubljeni genetski materijal, njenom upotrebom unosi se raznovrsnost među jedinkama populacije kao i proširivanje prostora pretrage rešenja čime se sprečava preuranjena konvergencija GA u lokalni ekstremum.

Nakon više uzastopnih primena genetskih operatora i stvaranja konačnog broja generacija GA se zaustavlja kada je neki od kriterijuma zaustavljanja zadovoljen: maksimalan broj generacija, dostignut optimum (ukoliko je poznat), nepromenjen kvalitet rešenja posle unapred zadatog broja generacija, prekid od strane korisnika, itd.

Opšti koncept genetskog algoritma predstavljen je na Slici 2.1.

Algoritam GA:

Ulaz: učitavanje ulaznih podataka

Izlaz: najkvalitetnija jedinka u tekućoj populaciji

1. Generiši početnu populaciju potencijalnih rešenja;
2. Sve dok nije zadovoljen uslov zaustavljanje izvršavaj sledeću petlju:
 - Izračunaj prilagođenost svake jedinke u populaciji;
 - Primeni operator selekcije;
 - Primeni operator ukrštanja;
 - Primeni operator mutacije;
3. Vрати najkvalitetniju jedinku u tekućoj populaciji.

Slika 2.1 Opšti koncept GA

2.2 Prost GA i njegove karakteristike

Prost genetski algoritam (Simple Genetic Algorithm - SGA) je najjednostavniji oblik genetskog algoritma i sastoji se od proste rulet selekcije, jednopozicionog ukrštanja i proste mutacije [Hil75].

Moguće posledice primene prostog GA za rešavanje nekog problema jesu preuranjena konvergencija i spora konvergencija. Glavni uzrok oba nedostatka je korišćenje proste rulet selekcija. Ukoliko populacija postane dovoljno slična na početku izvršavanja algoritma, GA često pronalazi lokalni ekstremum. Međutim, ukoliko je situacija obrnuta tako da u kasnoj fazi izvršavanja jedinke postanu skoro indentične tada SGA gubi sve šanse za promene genetskog koda populacije kako bi se dostiglo optimalno rešenje, što za posledicu ima sporu konvergenciju [SZ07a].

2.3 Moderne tehnike prilagođavanja GA problemu

Nedostaci SGA, navedeni u prethodnom poglavlju, uzrokuju mali stepen primene na jednostavnijim problemima kombinatorne optimizacije. Danas su GA u velikoj meri zastupljeni u rešavanju optimizacionih problema ali su i isto tako razvijene različite tehnike koje oni koriste da bi se što bolje prilagodili prirodi problema koje rešavaju. Najbitnije tehnike su: različite vrste kodiranja rešenja i odgovarajućih vrednosnih funkcija, prilagođeni složeniji genetski operatori, više vrsta funkcija prilagođenosti, politika zamene generacije, fiksna ili adaptivna promena parametara tokom izvršavanja GA.

2.3.1 Kodiranje i funkcija prilagođenosti

Jedan od najbitnijih faktora uspešnosti GA pri rešavanju zadatog problema jesu način kodiranja rešenja i funkcija prilagođenosti. Za uspešnu primenu GA ne postoje neki specijalni

uslovi koji se odnose na funkciju prilagođenosti. U idealnim uslovima ona je neprekidna, glatka i nema suviše mnogo lokalnih ekstremuma niti suviše izolovan globalni ekstremum [MM08a]. Međutim, najčešće postavljeni problemi nisu ni blizu idealne situacije i za njihovo rešavanje druge metode daju izuzetno loše rezultate ili ih je čak nemoguće primeniti. U svim tim slučajevima jedino rešenje je primena GA koji, ako ne uspe optimalno, bar nađe nekakvo rešenje. Za računanje funkcije cilja postoji više načina a najčešće se koristi: direktno preuzimanje, linearno skaliranje, skaliranje u jedinični interval, sigma odsecanje ili kombinacija nekih od njih (videti [Kra00]).

Na uspešan izbor načina kodiranja najviše utiče priroda problema koji se rešava. GA su najuspešniji u rešavanju problema kod kojih se prirodno nameće binarni način kodiranja (nad binarnom azbukom $\{0,1\}$). Za rešavanje nekih problema često je kodiranje nad azbukom veće kardinalnosti dosta pogodnije, npr. kodiranje nad azbukom celih ili realnih brojeva (u radu [Jan91] prikazani su eksperimenti poređenja binarnog kodiranja GA i kodiranja GA nad azbukom realnih brojeva, dok je u radu [Kra07] opisano rešavanje problema pomoću GA korišćenjem celobrojnog kodiranja kao i kombinacije binarnog i celobrojnog kodiranja).

Kada je izabran adekvatan način kodiranja potrebno je dati ocenu kvaliteta svakoj jedinki tj. odrediti vrednost njene funkcije prilagođenosti. Najpogodnije je uspostaviti bijekciju između vrednosti jedinke i njeno genetskog koda. Ukoliko kodiranje nije ni bijektivno ni preslikavanje velike se šanse da GA proizvede nekorektne jedinke odnosno one koje ne odgovaraju nijednom rešenju [SZ07a]. I za ovaj problem, odnosno strategiju izbacivanja nekorektnih jedinki iz populacije, razvijeno je više tehnika: takvim jedinkama postaviti vrednost funkcije prilagođenosti na nulu, primenom neke metode popraviti nekorektne jedinke (opis u radu [Kra08]) ili koristiti kaznenu funkciju koja će smanjiti prilagođenost nekorektnih jedinki (opis u radu [Đur07]).

2.3.2 Selekcija

Selekcija je operator kojim se biraju jedinke za učestovanje u postupku stvaranja nove generacije. Ona osigurava veći ili bar isti kvalitet svake naredne generacije. Osnovna ideja je u tome da kvalitetnije jedinke, sa većom vrednošću funkcije prilagođenosti, budu izabrane za stvaranje nove generacije. Postoji više tipova operatora selekcije: prosta rulet selekcija, selekcija zasnovana na rangu, turnirska selekcija i poboljšana varijanta turnirske selekcije - fino gradirana turnirska selekcija.

Prosta rulet selekcija je najjednostavniji oblik ovog operatora. Jedinke učestvuju na ruletu sa izvesnom verovatnoćom da budu odabrane, pri čemu je ta verovatnoća proporcionalna vrednosti funkcije prilagođenosti.

Kod selekcije zasnovane na rangu unapred se za svaku poziciju jedinke u populaciji odredi njen rang (nenegativna vrednost). Jedinke učestvuju na ruletu sa verovatnoćom

proporcionalnoj njihovom rangui koja jedino zavisi od pozicije jedinke u populaciji (a ne od vrednosti njene funkcije prilagođenosti).

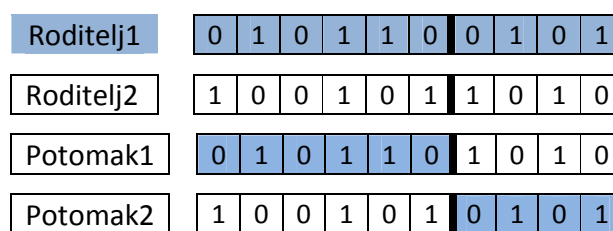
Turnirska selekcija je popularniji operator od prehodno opisana dva operatora selekcije. Ideja je organizovanje turnira (malih takmičenja) na kojima učestvuju jedinke radi prelaska u narednu generaciju. Prvi korak je generisanje podskupova od po N_{tur} jedinki (N_{tur} je parametar selekcije i predstavlja veličinu turnira), a zatim se najprilagođenija jedinka iz podskupa (pobednik turnira) bira za prelazak u novu generaciju i učešće u reprodukciji ([Kra00]).

Fino gradirana turnirska selekcija predstavlja unapređenje operatora turnirske selekcije. Veličina turnira nije jedinstvena u okviru populacije i zadaje se kao realan parametar algoritma. Detaljan opis ovog operatora selekcije i njegove primene pri rešavanju različitih vrsta problema dat je u radovima [Fil00], [Fil03] i [Fil06].

2.3.3 Ukrštanje

Ukrštanje je binarni operator u kome učestvuju dve jedinke-roditelji koje razmenjuju delove genetskog koda i tako stvaraju kodove 2 nove jedinke-potomka. Ukrštanje jedinki može biti jednopoziciono, dvopoziciono, višepoziciono, uniformno, aritmetičko ali postoje i složeniji tipovi ovog operatora o kojima se može pročitati u [Müh97]. Kod svih njih prvi korak je isti, odnosno na slučajan način se iz populacije biraju parovi jedinki-roditelja koji će učestovati u ukrštanju.

Za jednopoziciono ukrštanje broj k se bira na slučajan način iz skupa $\{0, 1, \dots, n-1\}$ (n je dužina genetskog koda) i predstavlja tačku ukrštanja odnosno poziciju u kodovima roditelja u odnosu na koju će razmeniti sadržaj. Od pozicije $k+1$ do poslednje pozicije $n-1$, simboli u oba koda razmenjuju mesta. Na Slici 2.2 prikazano je jednopoziciono ukrštanje 2 jedinke i 2 potomka koja su nastala kao rezultat.



Slika 2.2 Jednopoziciono ukrštanje

Za dvopoziciono ukrštanje na slučajan način se biraju brojevi k_1 i k_2 iz skupa $\{0, \dots, n-1\}$ koji predstavljaju tačke ukrštanja tj. pozicije u kodovima roditelja u odnosu na koju će razmeniti sadržaj (n je dužina genetskog koda). Od pozicije k_1+1 do k_2 simboli u oba koda razmenjuju mesta. Na Slici 2.3 prikazano je dvopoziciono ukrštanje 2 jedinke i 2 potomka koja su nastala kao rezultat.

Roditelj1	0	1	0	1	1	0	0	1	0	1
Roditelj2	1	0	0	1	0	1	1	0	1	0
Potomak1	0	1	0	1	0	1	0	1	0	1
Potomak2	1	0	0	1	1	0	1	0	1	0

Slika 2.3 Dvopoziciono ukrštanje

Za operator višepozicionog ukrštanja na slučajan način se bira više tačaka ukrštanja iz skupa $\{0, \dots, n-1\}$. Obično se za broj tačaka ukrštanja uzima vrednost $\lceil \sqrt{n} \rceil$. Na Slici 2.4 prikazano je višepoziciono ukrštanje 2 jedinke (dužina genetskog koda $n=10$, $\lceil \sqrt{n} \rceil = 3$) i 2 potomka koja su nastala kao rezultat.

Roditelj1	0	1	0	1	1	0	0	1	0	1
Roditelj2	1	0	0	1	0	1	1	0	1	0
Potomak1	0	1	0	1	0	1	0	1	0	0
Potomak1	1	0	0	1	1	0	1	0	1	1

Slika 2.4 Višepoziciono ukrštanje

Kod uniformnog ukrštanja za svaki par roditelja generiše se binarni niz iste dužine kao genetski kod jedinki tj. maska. Na mestima gde maska ima vrednost 1 roditelji zadržavaju svoje gene, a na pozicijama gde maska ima vrednost 0 roditelji razmenjuju gene. Na Slici 2.5 prikazano je uniformno ukrštanje 2 jedinke i 2 potomka koja su nastala kao rezultat.

Roditelj1	0	1	0	1	1	0	0	1	0	1
Roditelj2	1	0	0	1	0	1	1	0	1	0
Maska	1	0	1	0	0	0	1	1	0	0
Potomak1	0	0	0	1	0	1	0	1	1	0
Potomak2	1	1	0	1	1	0	1	0	0	1

Slika 2.5 Uniformno ukrštanje

Kod aritmetičkog ukrštanja, nakon što se na slučajan način izaberu roditelji, vrednost gena u potomcima se računa pomoću funkcije. Na Slici 2.6 prikazane su 2 jedinke-roditelja i 2 potomka.

Roditelj1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Roditelj2	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
Potomak1	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
Potomak2	p'_1	p'_2	p'_3	p'_4	p'_5	p'_6	p'_7	p'_8	p'_9	p'_{10}

Slika 2.6 Aritmetičko ukrštanje

Vrednost gena na poziciji 1 za Potomak 1 se računa na sledeći način:

$$p_1 = \lambda * x_1 + (1 - \lambda) * y_1,$$

vrednost gena na poziciji 1 za Potomak 2 se računa na sledeći način:

$$p'_1 = \lambda * y_1 + (1 - \lambda) * x_1,$$

gde je $\lambda \in [0,1]$, uz pretpostavku da su jedinke binarno kodirane.

Ukoliko je vrednost $p_1 < 0.5$ gen na toj poziciji dobija vrednost 0, u suprotnom 1. Aritmetički operator obezbeđuje da je svaki gen u potomku nastao od odgovarajućih gena oba roditelja.

U ovom radu biće razmatran uticaj izbora jednog od opisanih operatora ukrštanja na performanse genetskog algoritma pri rešavanju prostog lokacijskog problema. Iz dosadašnjih analiza zaključeno je da se operator ukrštanja bira u odnosu na cilj koji želimo da postignemo. Tako, korišćenjem jednopozicionog ukrštanja čuva se struktura genetskog koda. Dvopozicioni ili višepozicioni operator ukrštanja koristi se ukoliko je potrebno u većoj meri izmešati delove genetskog koda roditelja, dok se uniformno ukrštanje obično koristi u situacijama kada su geni međusobno nezavisni. Ukoliko se želi podjednako učešće gena roditelja za stvaranje gena potomaka tada se korisni aritmetičko ukrštanje [Pen04].

2.3.4 Mutacija

Mutacija je unarni operator čija je uloga u genetskim algoritmima da spreči gubitak genetskog materijala i da unese raznovrsnost među jedinkama u populaciji kada one postanu suviše slične.

Parametar operatora mutacije je p_{mut} i on predstavlja verovatnoću da gen mutira. Gen mutira tako što se njegova vrednost menja drugom vrednošću iz azbuke kodiranja (prosta mutacija). Za vrednost parametra p_{mut} najčešće se bira 0.1 ili 0.2. Ukoliko bi se on drastično povećao tada bi GA izgubio svoj smisao i predstavljao bi algoritam slučajne pretrage.

Korišćenjem binomne ili normalne raspodele moguće je postići ubrzanje realizacije proste mutacije. Detaljnije informacije se mogu pronaći u radovima [Kra00] i [Mar10].

2.3.5 Kriterijum zaustavljanja

Genetski algoritam se izvršava sve dok nije zadovoljen neki uslov zaustavljanja. Najčešće se koriste sledeći kriterijumi zaustavljanja [MM08a]:

- maksimalni broj generacija;
- najbolja jedinka je ponovljena maksimalni broj puta;
- algoritam je dostigao optimalno rešenje (ukoliko je to moguće);
- ograničeno vreme izvršavanja GA;
- prekid od strane korisnika;
- kombinacija nekih od prethodno navedenih kriterijuma.

2.3.6 Zamena generacije

Najčešće strategije zamene generacije su:

- generacijska (u svakoj generaciji sve jedinke će biti zamenjene);
- stacionarna (deo jedinki se prenosi iz prethodne generacije dok se preostale jedinke generišu u svakoj generaciji);
- elitistička (deo populacije se direktno prenosi u novu generaciju a nad ostatom se primenjuju genetski operatori ukršanje i mutacija u svakoj generaciji).

U praksi se najčešće koristi elitistička strategija zamene generacije jer povoljno utiče na performanse GA - skraćuje vreme izvršavanja i čuva dobra rešenja [SZ07a].

2.3.7 Parametri genetskog algoritma

Za performanse genetskog algoritma (kvalitet rešenja i brzina dolaženja do rešenja) često je od ključne važnosti izbor njegovih parametara. Ne postoji jedinstvena kombinacija parametara koja je najbolja za sve probleme ([SZ07a], [Sta11]). Najčešće se optimizovanje parametara rešava izvođenjem velikog broja eksperimenata. Parametri koji se najčešće koriste su: veličina populacije, broj generacija, nivo selekcije, nivo ukrštanja, nivo mutacije, broj izvršavanja algoritma, itd. Parametri mogu biti fiksni ili se menjati tokom izvršavanja algoritma (prikazano u radovima [Béc92], [Béc93] i [Sri94]). Menjanje parametara tokom izvršavanja GA moguće je implementirati tako što se unapred definiše formula po kojoj se promena vrši, kao i njen smer (povećanje ili smanjenje vrednosti), ili adaptivno vršiti promene, odnosno ne po unapred definisanoj formuli već na osnovu dosadašnje uspešnosti i rezultata za dati parametar koji se menja.

2.4 Primena GA u praksi

Genetski algoritmi se uspešno primenjuju na velikom broju optimizacionih problema. Uzrok toga je laka implementacija i postizanje kvalitetnih rešenja. Pregled velikog broja primena GA i praksi može se naći u [Ala08]. Neke od primena genetskog algoritma su za rešavanje:

- diskretnih lokacijskih problema [Fil00], [Kra00], [Kra01], [MM08a], [MM08b], [Sta04], [SZ07b], [Sta12], [SZ14a], [Mar15], [Sta15];
- hab lokacijskih problema [Fil06], [Kra06], [Kra08], [SZ07a], [Sta10], [Sta11], [Stn15];
- raspodele poslova [Sav08], [Law12], [Sav12];
- dizajna računarskih mreža [Kra02], [Val15];
- otpimizacije upita [Owa15];
- pretraživanja društvenih mreža [Sta13], [SZ14b].

3. GA IMPLEMENTACIJA KOJA KORISTI BINARNO KODIRANJE

3.1 Reprezentacija rešenja

Genetski kod jedinke sastoji se od n bitova (broj potencijalnih lokacija), pri čemu svaki bit odgovara jednoj potencijalnoj lokaciji za uspostavljenog snabdevača. Svaki bit može imati vrednost 0 ili 1.

Primer 3.1: Za $n = 8$, genetski kod jedinke 10110101 označava da:

- su na lokacijama 1,3,4, 6 i 8 uspostavljeni snabdevači;
- na lokacijama 2, 5 i 7 nisu uspostavljeni snabdevači.

Vrednosti binarnih promenljivih y_i ($i = 1, 2, \dots, n$) označavaju da li je na odgovarajućoj potencijalnoj poziciji uspostavljen snabdevač (vrednost 1) ili ne (vrednost 0). Iz genetskog koda jedinke lako se čitaju lokacije na kojima su uspostavljeni snabdevači. Ukoliko su poznate uspostavljene lokacije svaki korisnik može da izabere sebi najbližeg snabdevača čime se postiže minimizacija svih troškova (kapacitet snabdevača nije ograničen).

3.2 Računanje funkcije cilja

Računanje funkcije cilja moguće je znatno ubrzati korišćenjem dodatnog memorijskog prostora. Iz tog razloga, za svakog korisnika određena je uređena lista indeksa potencijalnih lokacija u neopadajućem poretku troškova transporta robe. Za uređivanje liste indeksa koristi se ugrađena funkcija $qsort()$. Nakon što su učitani ulazni podaci i izvršene sve pripreme (kreiranje uređenih lista indeksa za svakog korisnika, generisanje početne populacije), za svaku jedinku se računa funkcija cilja. Nakon što se odrede lokacije uspostavljenih snabdevača i saberu fiksni troškovi uspostavljanja f_i za $y_i=1$, za svakog korisnika potrebno je naći najpovoljnijeg snabdevača i cenu transporta između njih dodati sumi troškova. Najpovoljniji snabdevač se određuje tako što se za datog korisnika pronalazi njegova uređena lista (najbližih) lokacija i pretražuje sve do prve pojave y_i odnosno uspostavljenog snabdevača.

3.3 Genetski operatori

3.3.1 Selekcija

Nakon eksperimentisanja sa različitim operatorima selekcije najbolje rezultate je pokazala fino gradirana turnirska selekcija sa veličinom turnira jednakim 5.4. Najboljih 50 jedinki direktno prolazi u novu generaciju, dok preostalih 100 učestvuje u selekciji (elitistička strategija zamene generacije).

3.3.2 Ukrštanje

Implementirano je pet različitih operatora ukrštanja za binarnu reprezentaciju jedinki – jednopoziciono, dvopoziciono, višepoziciono, uniformno i aritmetičko ukrštanje sa parametrom ukrštanja $p_{cross} = 0.85$. To znači da će 85% jedinki, odabranih operatorom selekcije, učestvovati u ukrštanju, dok za 15% jedinki neće biti ukrštanja i potomci će biti identični roditeljima. Kod uniformnog ukrštanja verovatnoća razmene proizvoljnog bita je $p_{unif} = 0.6$, što znači da se približno 60% bitova razmenjuje između jedinki.

3.3.3 Mutacija

Za operator mutacije korišćena je prosta mutacija sa parametrom mutacije bita $p_{mut} = 0.1$, što znači da je verovatnoća da bit mutira jednaka 10%.

3.4 Ostali aspekti GA

Početna populacija je generisana na slučajan način. Genetski kod jedunki sadrži približno isti broj 0 i 1 ($p_0 = 1/2$, $p_1 = 1/2$).

Broj jedinki u populaciji je 150. Korišćena je elitistička strategija tj. u svakoj generaciji 1/3 populacije (50 jedinki sa najboljim vrednostima funkcije prilagođenosti) direktno prelazi u sledeću generaciju, dok se nad preostalim 2/3 (100 jedinki) primenjuju genetski operatori ukrštanja i mutacije.

Za kriterijum završetka GA korišćena je kombinacija 2 kriterijuma zaustavljenja: dostignut maksimalni broj generacija (ograničen na 1000) i uzastopno ponavljanje najbolje jedinice maksimalan broj puta (300).

3.5 Eksperimentalni rezultati

3.5.1 Ulazni podaci

Za testiranje su GA korišćene su standardne ORLIB instance [Bea96]. Dimenzije korišćenih instanci i broj instanci za svaku dimenziju dati su u Tabeli 3.1. Instance su podeljene u 2 grupe – prvu grupu čine instance kod kojih je maksimalni broj lokacija 20, a maksimalni broj korisnika 40, dok preostale instance čine drugu grupu (kod kojih je broj lokacija 20 ili 30, a broj korisnika između 50 i 90).

Tabela 3.1 – Karakteristike instanci

Dimenzija instance (nxm)	Broj instanci
5x10	1
10x10	1
10x20	6
15x30	11
20x40	8
20x50	8
30x60	8
30x75	8
30x90	8

3.5.2 Rezultati testiranja

Genetski algoritam za rešavanje prostog lokacijskog problema implementiran je u programskom jeziku C za sve tri vrste kodiranja. Svi algoritmi su testirani na računaru sa Intel Core i7, 2.60 GHz procesorom i 12GB radne memorije. Svi algoritmi pokrenuti su po 20 puta. Pri svakom izvršavanju pamti se:

- sol_i najbolje rešenje koje je GA dobio u i -tom izvršavanju;
- t_i vreme za koje je GA prvi put dobio najbolje rešenje sol_i , tzv. početno vreme;
- t_{tot} ukupno vreme i -tog izvršavanja GA (do zadovoljenja nekog kriterijuma);
- gen_i broj generacija GA u i -tom izvršavanju.

Na kraju svih 20 ($k=20$) izvršavanja najbolje rešenje GA označi se sa $Best_{sol}$. Nakon toga računa se procentualno odstupanje po formuli:

$gap_i = 100 * \frac{|sol_i - Best_{sol}|}{|Best_{sol}|}$ za svako $i = 1, \dots, 20$. U slučaju da je optimalno rešenje opt_{sol} za datu instancu poznato, procentualno odstupanje se računa po sledećoj formuli:

$gap_i = 100 * \frac{|sol_i - opt_{sol}|}{|opt_{sol}|}$ za svako $i = 1, \dots, 20$.

Zatim se računaju srednje vrednosti:

- $t = \frac{1}{k} \sum_{i=1}^k t_i$ srednje početno vreme (u sekundama);
- $t_{tot} = \frac{1}{k} \sum_{i=1}^k t_{tot_i}$ srednje ukupno vreme izvršavanja (u sekundama);
- $gen = \frac{1}{k} \sum_{i=1}^k t_{gen_i}$ srednji broj generacija;
- $agap = \frac{1}{k} \sum_{i=1}^k gap_i$ srednje odstupanje (u procentima);
- $\sigma = \sqrt{\sum_{i=1}^k (gap_i - agap)^2}$ standardna devijacija (u procentima).

U tabelama su prikazani rezultati testiranja obe grupe instanci za različite tipove operatora ukrštanja. Kolona *R.br.* predstavlja redni broj instance. Kolona *n x m* označava dimenziju problema (broj potencijalnih lokacija x broj korisnika). Kolona *opt_{sol}* predstavlja vrednost funkcije cilja rešenja dobijenog pomoću IBM ILOG CPLEX 12.1 rešavača (za svaku testiranu instancu CPLEX je našao optimalno rešenje), dok kolona *best_{sol}* predstavlja vrednost funkcije cilja najboljeg rešenja koje je GA dostigao tokom 20 pokretanja. Kolona *t(s)* predstavlja prosečno vreme potrebno da GA dostigne rešenje. Kolona *t_{tot}(s)* predstavlja prosečno ukupno vreme potrebno za završetak GA. Kolona *gen* predstavlja prosečan ukupan broj generacija. Kolona *agap(%)* predstavlja prosečno odstupanje rešenja (u procentima) od najboljeg rešenja dobijenog genetskim algoritmom nakon 20 ponavljanja. Kolona $\sigma(\%)$ predstavlja standardnu devijaciju najboljeg GA rešenja od optimalnog, odnosno najboljeg rešenja iz literature.

U nastavku su Tabele 3.2 - 3.13 u kojima su prikazani detaljni rezultati izvršavanja opisanog GA sa binarnim kodiranjem i različitim tipovima ukrštanja za svaku od testiranih grupa instanci.

Tabela 3.2 Rezultati testiranja instanci iz prve grupe – jednopoziciono ukrštanje

<i>R.br.</i>	<i>n x m</i>	<i>opt_{sol}</i>	<i>best_{sol}</i>	<i>t(s)</i>	<i>t_{tot}(s)</i>	<i>gen</i>	<i>agap(%)</i>	$\sigma(\%)$
1	5 x 10	2970.000	opt	0.002	0.019	300	0.000	0.000
2	10 x 10	2872.000	opt	0.001	0.027	305	0.000	0.000
3	10 x 20	4556.000	opt	0.001	0.029	313	0.000	0.000
4	10 x 20	5895.000	opt	0.001	0.029	307	0.000	0.000
5	10 x 20	7411.000	opt	0.002	0.034	300	0.000	0.000
6	10 x 20	8113.000	opt	0.002	0.031	306	0.000	0.000
7	10 x 20	10160.000	opt	0.001	0.029	315	0.000	0.000
8	10 x 20	5297.000	opt	0.002	0.029	309	0.000	0.000
9	15 x 30	9675.000	opt	0.002	0.047	326	0.000	0.000
10	15 x 30	9693.000	opt	0.002	0.046	307	0.000	0.000
11	15 x 30	6434.000	opt	0.002	0.04	309	0.000	0.000
12	15 x 30	14772.000	opt	0.003	0.055	331	0.000	0.000
13	15 x 30	6892.000	opt	0.002	0.045	316	0.000	0.000
14	15 x 30	8160.000	opt	0.002	0.043	316	0.000	0.000
15	15 x 30	7809.000	opt	0.002	0.043	311	0.000	0.000
16	15 x 30	10667.000	opt	0.003	0.051	316	0.000	0.000
17	15 x 30	6361.000	opt	0.003	0.052	347	0.000	0.000
18	15 x 30	7685.000	opt	0.003	0.054	342	0.000	0.000
19	15 x 30	15491.000	opt	0.003	0.051	320	0.000	0.000
20	20 x 40	13809.000	opt	0.004	0.085	396	0.000	0.000
21	20 x 40	14559.000	opt	0.004	0.087	382	0.000	0.000
22	20 x 40	14734.000	opt	0.004	0.087	374	0.000	0.000
23	20 x 40	8503.000	opt	0.003	0.066	348	0.000	0.000
24	20 x 40	8358.000	opt	0.004	0.079	391	0.000	0.000
25	20 x 40	9362.000	opt	0.004	0.073	333	0.000	0.000
26	20 x 40	12435.000	opt	0.004	0.076	358	0.000	0.000
27	20 x 40	13614.000	opt	0.006	0.106	449	0.000	0.000

Tabela 3.3 Rezultati testiranja istanci iz druge grupe – jednopoziciono ukrštanje

R. br.	$n \times m$	opt_{sol}	$best_{sol}$	$t(s)$	$t_{tot}(s)$	gen	$agap(\%)$	$\sigma(\%)$
1	20 x 50	9223.000	opt	0.005	0.082	393	0.000	0.000
2	20 x 50	17916.000	opt	0.005	0.113	425	0.000	0.000
3	20 x 50	14275.000	opt	0.004	0.081	326	0.000	0.000
4	20 x 50	14552.000	opt	0.004	0.080	343	0.000	0.000
5	20 x 50	11360.000	opt	0.005	0.088	418	0.000	0.000
6	20 x 50	9718.000	opt	0.004	0.086	413	0.000	0.000
7	20 x 50	14276.000	opt	0.004	0.115	455	0.000	0.000
8	20 x 50	22625.000	opt	0.005	0.110	418	0.000	0.000
9	30 x 60	14783.000	opt	0.007	0.153	440	0.015	0.015
10	30 x 60	12992.000	opt	0.011	0.199	585	0.017	0.014
11	30 x 60	21946.000	opt	0.017	0.223	602	0.010	0.010
12	30 x 60	20526.000	opt	0.011	0.189	511	0.031	0.023
13	30 x 60	25983.000	opt	0.022	0.174	460	0.032	0.016
14	30 x 60	25467.000	opt	0.010	0.160	428	0.010	0.008
15	30 x 60	27451.000	opt	0.010	0.167	440	0.052	0.029
16	30 x 60	32414.000	opt	0.009	0.205	542	0.064	0.046
17	30 x 75	16829.000	opt	0.019	0.233	595	0.011	0.018
18	30 x 75	17919.000	opt	0.010	0.230	577	0.011	0.009
19	30 x 75	22591.000	opt	0.018	0.212	511	0.028	0.015
20	30 x 75	25691.000	opt	0.012	0.230	529	0.022	0.011
21	30 x 75	33821.000	opt	0.013	0.284	646	0.005	0.006
22	30 x 75	30325.000	opt	0.025	0.293	660	0.036	0.039
23	30 x 75	48555.000	opt	0.011	0.216	465	0.029	0.028
24	30 x 75	31900.000	opt	0.012	0.203	463	0.065	0.043
25	30 x 90	21143.000	opt	0.012	0.326	694	0.008	0.007
26	30 x 90	19629.000	opt	0.024	0.267	592	0.020	0.016
27	30 x 90	29358.000	opt	0.010	0.210	432	0.026	0.032
28	30 x 90	29931.000	opt	0.023	0.259	543	0.016	0.007
29	30 x 90	39005.000	opt	0.009	0.305	602	0.040	0.027
30	30 x 90	35576.000	opt	0.016	0.255	514	0.013	0.014
31	30 x 90	49324.000	opt	0.025	0.263	518	0.046	0.047
32	30 x 90	49755.000	opt	0.025	0.309	602	0.060	0.037

Tabela 3.4 Rezultati testiranja istanci iz prve grupe – dvopoziciono ukrštanje

<i>R. br.</i>	<i>n x m</i>	<i>opt_{sol}</i>	<i>best_{sol}</i>	<i>t(s)</i>	<i>t_{tot}(s)</i>	<i>gen</i>	<i>agap(%)</i>	<i>σ(%)</i>
1	5 x 10	2970.000	opt	0.001	0.017	300	0.000	0.000
2	10 x 10	2872.000	opt	0.001	0.027	306	0.000	0.000
3	10 x 20	4556.000	opt	0.001	0.029	307	0.000	0.000
4	10 x 20	5895.000	opt	0.001	0.030	308	0.000	0.000
5	10 x 20	7411.000	opt	0.002	0.035	307	0.000	0.000
6	10 x 20	8113.000	opt	0.002	0.033	313	0.000	0.000
7	10 x 20	10160.000	opt	0.001	0.031	314	0.000	0.000
8	10 x 20	5297.000	opt	0.001	0.029	300	0.000	0.000
9	15 x 30	9675.000	opt	0.002	0.048	327	0.000	0.000
10	15 x 30	9693.000	opt	0.002	0.050	336	0.000	0.000
11	15 x 30	6434.000	opt	0.002	0.045	339	0.000	0.000
12	15 x 30	14772.000	opt	0.003	0.052	307	0.000	0.000
13	15 x 30	6892.000	opt	0.002	0.049	324	0.000	0.000
14	15 x 30	8160.000	opt	0.002	0.043	321	0.000	0.000
15	15 x 30	7809.000	opt	0.002	0.043	323	0.000	0.000
16	15 x 30	10667.000	opt	0.002	0.050	312	0.000	0.000
17	15 x 30	6361.000	opt	0.003	0.052	342	0.000	0.000
18	15 x 30	7685.000	opt	0.003	0.054	339	0.000	0.000
19	15 x 30	15491.000	opt	0.003	0.056	330	0.000	0.000
20	20 x 40	13809.000	opt	0.004	0.079	357	0.000	0.000
21	20 x 40	14559.000	opt	0.004	0.082	356	0.000	0.000
22	20 x 40	14734.000	opt	0.004	0.083	359	0.000	0.000
23	20 x 40	8503.000	opt	0.003	0.066	341	0.000	0.000
24	20 x 40	8358.000	opt	0.003	0.079	391	0.000	0.000
25	20 x 40	9362.000	opt	0.004	0.080	370	0.000	0.000
26	20 x 40	12435.000	opt	0.003	0.076	353	0.000	0.000
27	20 x 40	13614.000	opt	0.005	0.079	343	0.000	0.000

Tabela 3.5 Rezultati testiranja istanci iz druge grupe – dvopoziciono ukrštanje

R. br.	$n \times m$	opt_{sol}	$best_{sol}$	$t(s)$	$t_{tot}(s)$	gen	$agap(\%)$	$\sigma(\%)$
1	20 x 50	9223.000	opt	0.004	0.075	366	0.000	0.000
2	20 x 50	17916.000	opt	0.005	0.096	357	0.000	0.000
3	20 x 50	14275.000	opt	0.004	0.086	337	0.000	0.000
4	20 x 50	14552.000	opt	0.004	0.085	355	0.000	0.000
5	20 x 50	11360.000	opt	0.004	0.076	356	0.000	0.000
6	20 x 50	9718.000	opt	0.003	0.077	360	0.000	0.000
7	20 x 50	14276.000	opt	0.005	0.098	382	0.000	0.000
8	20 x 50	22625.000	opt	0.005	0.096	361	0.000	0.000
9	30 x 60	14783.000	opt	0.019	0.272	790	0.017	0.010
10	30 x 60	12992.000	opt	0.008	0.180	528	0.024	0.017
11	30 x 60	21946.000	opt	0.008	0.191	511	0.028	0.025
12	30 x 60	20526.000	opt	0.022	0.242	664	0.017	0.012
13	30 x 60	25983.000	opt	0.012	0.222	585	0.018	0.011
14	30 x 60	25467.000	opt	0.008	0.209	565	0.025	0.013
15	30 x 60	27451.000	opt	0.012	0.241	620	0.024	0.021
16	30 x 60	32414.000	opt	0.011	0.220	573	0.074	0.050
17	30 x 75	16829.000	opt	0.016	0.191	476	0.007	0.005
18	30 x 75	17919.000	opt	0.015	0.261	650	0.006	0.004
19	30 x 75	22591.000	opt	0.014	0.206	502	0.025	0.014
20	30 x 75	25691.000	opt	0.014	0.247	579	0.008	0.012
21	30 x 75	33821.000	opt	0.011	0.237	538	0.024	0.014
22	30 x 75	30325.000	opt	0.022	0.308	702	0.024	0.019
23	30 x 75	48555.000	opt	0.013	0.270	597	0.031	0.022
24	30 x 75	31900.000	opt	0.011	0.247	565	0.044	0.025
25	30 x 90	21143.000	opt	0.008	0.213	469	0.011	0.007
26	30 x 90	19629.000	opt	0.013	0.296	667	0.037	0.016
27	30 x 90	29358.000	opt	0.013	0.267	552	0.012	0.007
28	30 x 90	29931.000	opt	0.011	0.260	550	0.016	0.011
29	30 x 90	39005.000	opt	0.017	0.303	599	0.071	0.044
30	30 x 90	35576.000	opt	0.015	0.253	511	0.016	0.011
31	30 x 90	49324.000	opt	0.019	0.281	552	0.049	0.038
32	30 x 90	49755.000	opt	0.015	0.233	451	0.053	0.029

Tabela 3.6 Rezultati testiranja istanci iz prve grupe – višepoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.001	0.016	300	0.000	0.000
2	10 x 10	2872.000	opt	0.001	0.027	314	0.000	0.000
3	10 x 20	4556.000	opt	0.001	0.028	300	0.000	0.000
4	10 x 20	5895.000	opt	0.001	0.028	302	0.000	0.000
5	10 x 20	7411.000	opt	0.002	0.035	300	0.000	0.000
6	10 x 20	8113.000	opt	0.002	0.032	305	0.000	0.000
7	10 x 20	10160.000	opt	0.001	0.030	300	0.000	0.000
8	10 x 20	5297.000	opt	0.002	0.030	309	0.000	0.000
9	15 x 30	9675.000	opt	0.002	0.047	318	0.000	0.000
10	15 x 30	9693.000	opt	0.002	0.047	322	0.000	0.000
11	15 x 30	6434.000	opt	0.002	0.044	330	0.000	0.000
12	15 x 30	14772.000	opt	0.003	0.053	313	0.000	0.000
13	15 x 30	6892.000	opt	0.002	0.046	311	0.000	0.000
14	15 x 30	8160.000	opt	0.002	0.043	315	0.000	0.000
15	15 x 30	7809.000	opt	0.002	0.042	318	0.000	0.000
16	15 x 30	10667.000	opt	0.003	0.051	314	0.000	0.000
17	15 x 30	6361.000	opt	0.003	0.053	344	0.000	0.000
18	15 x 30	7685.000	opt	0.003	0.052	329	0.000	0.000
19	15 x 30	15491.000	opt	0.003	0.054	331	0.000	0.000
20	20 x 40	13809.000	opt	0.004	0.077	354	0.000	0.000
21	20 x 40	14559.000	opt	0.005	0.090	384	0.000	0.000
22	20 x 40	14734.000	opt	0.008	0.091	385	0.000	0.000
23	20 x 40	8503.000	opt	0.003	0.065	337	0.000	0.000
24	20 x 40	8358.000	opt	0.005	0.074	360	0.000	0.000
25	20 x 40	9362.000	opt	0.003	0.070	315	0.000	0.000
26	20 x 40	12435.000	opt	0.003	0.069	316	0.000	0.000
27	20 x 40	13614.000	opt	0.004	0.095	409	0.000	0.000

Tabela 3.7 Rezultati testiranja istanci iz druge grupe – dvopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	opt	0.004	0.079	382	0.000	0.000
2	20 x 50	17916.000	opt	0.005	0.100	374	0.000	0.000
3	20 x 50	14275.000	opt	0.005	0.088	346	0.000	0.000
4	20 x 50	14552.000	opt	0.004	0.089	372	0.000	0.000
5	20 x 50	11360.000	opt	0.004	0.080	366	0.000	0.000
6	20 x 50	9718.000	opt	0.005	0.080	373	0.000	0.000
7	20 x 50	14276.000	opt	0.005	0.101	393	0.000	0.000
8	20 x 50	22625.000	opt	0.006	0.101	376	0.000	0.000
9	30 x 60	14783.000	opt	0.012	0.184	519	0.024	0.011
10	30 x 60	12992.000	opt	0.008	0.144	420	0.021	0.016
11	30 x 60	21946.000	opt	0.008	0.168	442	0.021	0.023
12	30 x 60	20526.000	opt	0.006	0.177	484	0.028	0.021
13	30 x 60	25983.000	opt	0.008	0.189	500	0.017	0.010
14	30 x 60	25467.000	opt	0.008	0.212	564	0.008	0.013
15	30 x 60	27451.000	opt	0.011	0.255	663	0.018	0.020
16	30 x 60	32414.000	opt	0.010	0.232	592	0.059	0.043
17	30 x 75	16829.000	opt	0.009	0.268	676	0.002	0.002
18	30 x 75	17919.000	opt	0.013	0.186	446	0.014	0.012
19	30 x 75	22591.000	opt	0.012	0.240	578	0.020	0.011
20	30 x 75	25691.000	opt	0.016	0.257	593	0.039	0.022
21	30 x 75	33821.000	opt	0.012	0.263	592	0.029	0.028
22	30 x 75	30325.000	opt	0.009	0.193	433	0.046	0.043
23	30 x 75	48555.000	opt	0.013	0.251	548	0.033	0.022
24	30 x 75	31900.000	opt	0.016	0.232	524	0.026	0.027
25	30 x 90	21143.000	opt	0.012	0.236	501	0.022	0.010
26	30 x 90	19629.000	opt	0.009	0.269	599	0.037	0.022
27	30 x 90	29358.000	opt	0.021	0.270	560	0.006	0.007
28	30 x 90	29931.000	opt	0.023	0.291	615	0.015	0.009
29	30 x 90	39005.000	opt	0.010	0.301	593	0.062	0.050
30	30 x 90	35576.000	opt	0.010	0.241	488	0.023	0.018
31	30 x 90	49324.000	opt	0.026	0.267	510	0.056	0.036
32	30 x 90	49755.000	opt	0.026	0.343	662	0.037	0.022

Tabela 3.8 Rezultati testiranja istanci iz prve grupe – uniformno ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.001	0.017	300	0.000	0.000
2	10 x 10	2872.000	opt	0.001	0.028	306	0.000	0.000
3	10 x 20	4556.000	opt	0.001	0.031	304	0.000	0.000
4	10 x 20	5895.000	opt	0.002	0.031	312	0.000	0.000
5	10 x 20	7411.000	opt	0.002	0.037	309	0.000	0.000
6	10 x 20	8113.000	opt	0.002	0.035	308	0.000	0.000
7	10 x 20	10160.000	opt	0.002	0.032	306	0.000	0.000
8	10 x 20	5297.000	opt	0.002	0.030	302	0.000	0.000
9	15 x 30	9675.000	opt	0.003	0.053	334	0.000	0.000
10	15 x 30	9693.000	opt	0.003	0.051	322	0.000	0.000
11	15 x 30	6434.000	opt	0.002	0.046	322	0.000	0.000
12	15 x 30	14772.000	opt	0.003	0.057	310	0.000	0.000
13	15 x 30	6892.000	opt	0.003	0.050	333	0.000	0.000
14	15 x 30	8160.000	opt	0.003	0.046	307	0.000	0.000
15	15 x 30	7809.000	opt	0.002	0.045	315	0.000	0.000
16	15 x 30	10667.000	opt	0.003	0.056	326	0.000	0.000
17	15 x 30	6361.000	opt	0.003	0.052	315	0.000	0.000
18	15 x 30	7685.000	opt	0.003	0.056	318	0.000	0.000
19	15 x 30	15491.000	opt	0.003	0.062	334	0.000	0.000
20	20 x 40	13809.000	opt	0.004	0.082	342	0.000	0.000
21	20 x 40	14559.000	opt	0.005	0.094	377	0.000	0.000
22	20 x 40	14734.000	opt	0.005	0.094	378	0.000	0.000
23	20 x 40	8503.000	opt	0.003	0.074	368	0.000	0.000
24	20 x 40	8358.000	opt	0.004	0.078	360	0.000	0.000
25	20 x 40	9362.000	opt	0.004	0.095	411	0.000	0.000
26	20 x 40	12435.000	opt	0.004	0.083	350	0.000	0.000
27	20 x 40	13614.000	opt	0.005	0.103	418	0.000	0.000

Tabela 3.9 Rezultati testiranja istanci iz druge grupe – uniformno ukrštanje

R.br.	$n \times m$	opt_{sol}	$best_{sol}$	$t(s)$	$t_{tot}(s)$	gen	$agap(\%)$	$\sigma(\%)$
1	20 x 50	9223.000	opt	0.004	0.088	367	0.000	0.000
2	20 x 50	17916.000	opt	0.006	0.098	347	0.000	0.000
3	20 x 50	14275.000	opt	0.004	0.089	335	0.000	0.000
4	20 x 50	14552.000	opt	0.004	0.091	364	0.000	0.000
5	20 x 50	11360.000	opt	0.004	0.089	391	0.000	0.000
6	20 x 50	9718.000	opt	0.004	0.084	372	0.000	0.000
7	20 x 50	14276.000	opt	0.005	0.096	354	0.000	0.000
8	20 x 50	22625.000	opt	0.005	0.099	343	0.000	0.000
9	30 x 60	14783.000	opt	0.018	0.197	526	0.014	0.009
10	30 x 60	12992.000	opt	0.007	0.231	623	0.008	0.009
11	30 x 60	21946.000	opt	0.017	0.219	536	0.011	0.009
12	30 x 60	20526.000	opt	0.019	0.223	557	0.009	0.011
13	30 x 60	25983.000	opt	0.010	0.223	558	0.018	0.010
14	30 x 60	25467.000	opt	0.014	0.264	652	0.015	0.014
15	30 x 60	27451.000	opt	0.009	0.241	591	0.017	0.013
16	30 x 60	32414.000	opt	0.016	0.218	502	0.030	0.022
17	30 x 75	16829.000	opt	0.013	0.293	685	0.004	0.002
18	30 x 75	17919.000	opt	0.018	0.223	506	0.008	0.008
19	30 x 75	22591.000	opt	0.015	0.222	501	0.007	0.009
20	30 x 75	25691.000	opt	0.009	0.207	455	0.026	0.016
21	30 x 75	33821.000	opt	0.010	0.222	471	0.025	0.020
22	30 x 75	30325.000	opt	0.016	0.262	564	0.019	0.020
23	30 x 75	48555.000	opt	0.014	0.287	599	0.033	0.022
24	30 x 75	31900.000	opt	0.009	0.248	534	0.061	0.032
25	30 x 90	21143.000	opt	0.008	0.176	367	0.011	0.007
26	30 x 90	19629.000	opt	0.023	0.231	493	0.026	0.018
27	30 x 90	29358.000	opt	0.018	0.317	624	0.013	0.010
28	30 x 90	29931.000	opt	0.013	0.234	472	0.009	0.007
29	30 x 90	39005.000	opt	0.009	0.230	421	0.042	0.029
30	30 x 90	35576.000	opt	0.014	0.269	511	0.025	0.014
31	30 x 90	49324.000	opt	0.012	0.290	527	0.036	0.029
32	30 x 90	49755.000	opt	0.010	0.293	540	0.063	0.034

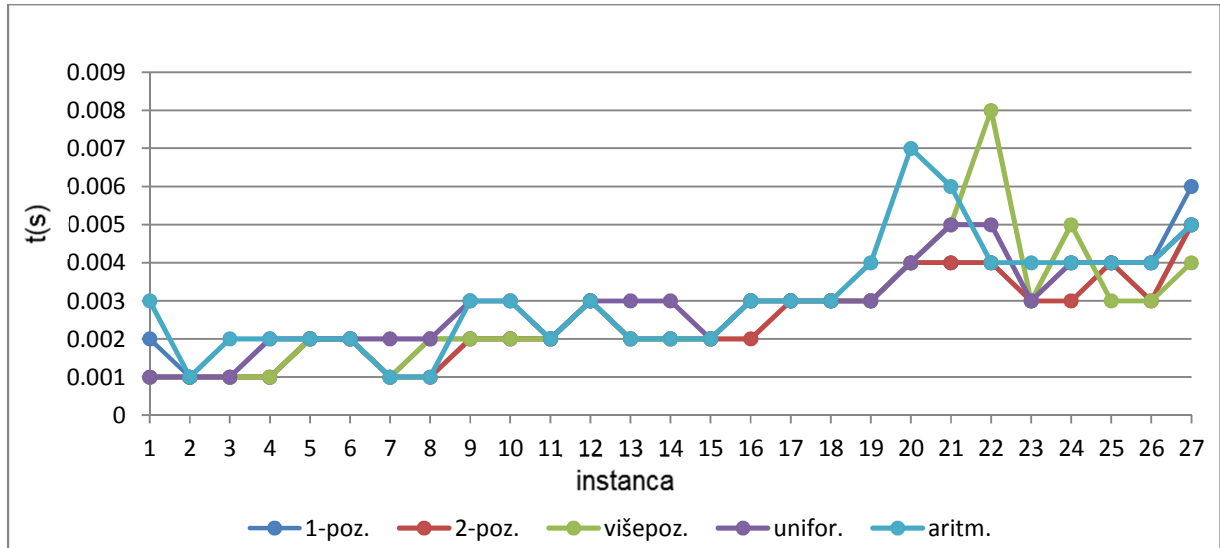
Tabela 3.10 Rezultati testiranja istanci iz prve grupe – aritmetičko ukrštanje

R. br.	$n \times m$	opt_{sol}	$best_{sol}$	$t(s)$	$t_{tot}(s)$	gen	$agap(\%)$	$\sigma(\%)$
1	5 x 10	2970.000	opt	0.003	0.020	300	0.000	0.000
2	10 x 10	2872.000	opt	0.001	0.029	318	0.000	0.000
3	10 x 20	4556.000	opt	0.002	0.030	305	0.000	0.000
4	10 x 20	5895.000	opt	0.002	0.030	310	0.000	0.000
5	10 x 20	7411.000	opt	0.002	0.035	304	0.000	0.000
6	10 x 20	8113.000	opt	0.002	0.034	306	0.000	0.000
7	10 x 20	10160.000	opt	0.001	0.030	304	0.000	0.000
8	10 x 20	5297.000	opt	0.001	0.030	306	0.000	0.000
9	15 x 30	9675.000	opt	0.003	0.050	327	0.000	0.000
10	15 x 30	9693.000	opt	0.003	0.049	317	0.000	0.000
11	15 x 30	6434.000	opt	0.002	0.045	325	0.000	0.000
12	15 x 30	14772.000	opt	0.003	0.055	317	0.000	0.000
13	15 x 30	6892.000	opt	0.002	0.046	308	0.000	0.000
14	15 x 30	8160.000	opt	0.002	0.045	321	0.000	0.000
15	15 x 30	7809.000	opt	0.002	0.045	327	0.000	0.000
16	15 x 30	10667.000	opt	0.003	0.053	325	0.000	0.000
17	15 x 30	6361.000	opt	0.003	0.052	329	0.000	0.000
18	15 x 30	7685.000	opt	0.003	0.058	330	0.000	0.000
19	15 x 30	15491.000	opt	0.004	0.060	330	0.000	0.000
20	20 x 40	13809.000	opt	0.007	0.087	377	0.000	0.000
21	20 x 40	14559.000	opt	0.006	0.091	380	0.000	0.000
22	20 x 40	14734.000	opt	0.004	0.081	341	0.000	0.000
23	20 x 40	8503.000	opt	0.004	0.075	374	0.000	0.000
24	20 x 40	8358.000	opt	0.004	0.076	351	0.000	0.000
25	20 x 40	9362.000	opt	0.004	0.086	379	0.000	0.000
26	20 x 40	12435.000	opt	0.004	0.099	442	0.000	0.000
27	20 x 40	13614.000	opt	0.005	0.094	391	0.000	0.000

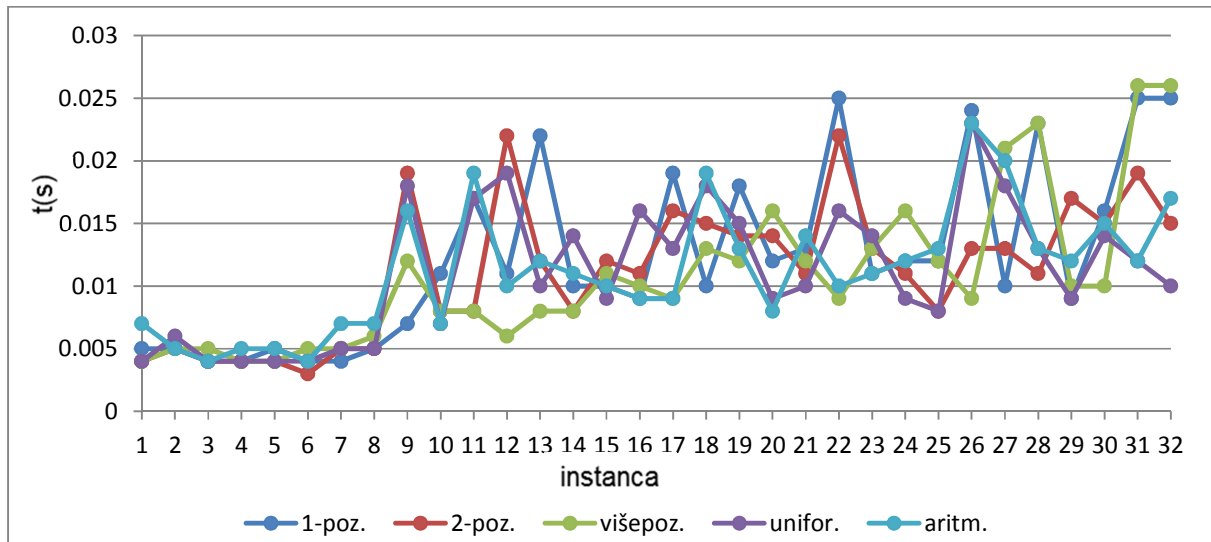
Tabela 3.11 Rezultati testiranja istanci iz druge grupe – aritmetičko ukrštanje

R. br.	$n \times m$	opt_{sol}	$best_{sol}$	$t(s)$	$t_{tot}(s)$	gen	$agap(\%)$	$\sigma(\%)$
1	20 x 50	9223.000	opt	0.007	0.097	412	0.000	0.000
2	20 x 50	17916.000	opt	0.005	0.133	477	0.000	0.000
3	20 x 50	14275.000	opt	0.004	0.093	354	0.000	0.000
4	20 x 50	14552.000	opt	0.005	0.099	384	0.000	0.000
5	20 x 50	11360.000	opt	0.005	0.109	469	0.000	0.000
6	20 x 50	9718.000	opt	0.004	0.106	480	0.000	0.000
7	20 x 50	14276.000	opt	0.007	0.126	477	0.000	0.000
8	20 x 50	22625.000	opt	0.007	0.110	392	0.000	0.000
9	30 x 60	14783.000	opt	0.016	0.265	726	0.021	0.013
10	30 x 60	12992.000	opt	0.007	0.165	460	0.018	0.014
11	30 x 60	21946.000	opt	0.019	0.174	437	0.028	0.024
12	30 x 60	20526.000	opt	0.010	0.197	513	0.018	0.019
13	30 x 60	25983.000	opt	0.012	0.177	445	0.027	0.016
14	30 x 60	25467.000	opt	0.011	0.211	539	0.017	0.018
15	30 x 60	27451.000	opt	0.010	0.231	578	0.018	0.017
16	30 x 60	32414.000	opt	0.009	0.215	520	0.044	0.025
17	30 x 75	16829.000	opt	0.009	0.205	498	0.016	0.012
18	30 x 75	17919.000	opt	0.019	0.254	594	0.021	0.013
19	30 x 75	22591.000	opt	0.013	0.181	410	0.039	0.020
20	30 x 75	25691.000	opt	0.008	0.206	455	0.041	0.018
21	30 x 75	33821.000	opt	0.014	0.232	483	0.033	0.024
22	30 x 75	30325.000	opt	0.010	0.213	443	0.040	0.022
23	30 x 75	48555.000	opt	0.011	0.312	637	0.028	0.016
24	30 x 75	31900.000	opt	0.012	0.198	422	0.040	0.043
25	30 x 90	21143.000	opt	0.013	0.274	559	0.023	0.013
26	30 x 90	19629.000	opt	0.023	0.285	609	0.043	0.030
27	30 x 90	29358.000	opt	0.020	0.278	547	0.020	0.013
28	30 x 90	29931.000	opt	0.013	0.225	462	0.021	0.014
29	30 x 90	39005.000	opt	0.012	0.360	671	0.041	0.026
30	30 x 90	35576.000	opt	0.015	0.247	475	0.025	0.017
31	30 x 90	49324.000	opt	0.012	0.300	555	0.055	0.027
32	30 x 90	49755.000	opt	0.017	0.306	553	0.029	0.015

U svih pet varijanti (u odnosu na tip operatora ukrštanja) GA je postigao optimalna rešenja za sve instance. Što se tiče vremena izvršavanja tu je situacija drugačija. Na Graficima 3.1. i 3.2. prikazano je poređenje srednjeg početnog vremena GA implementacije sa binarnim kodiranjem i različitim tipovima ukrštanja.

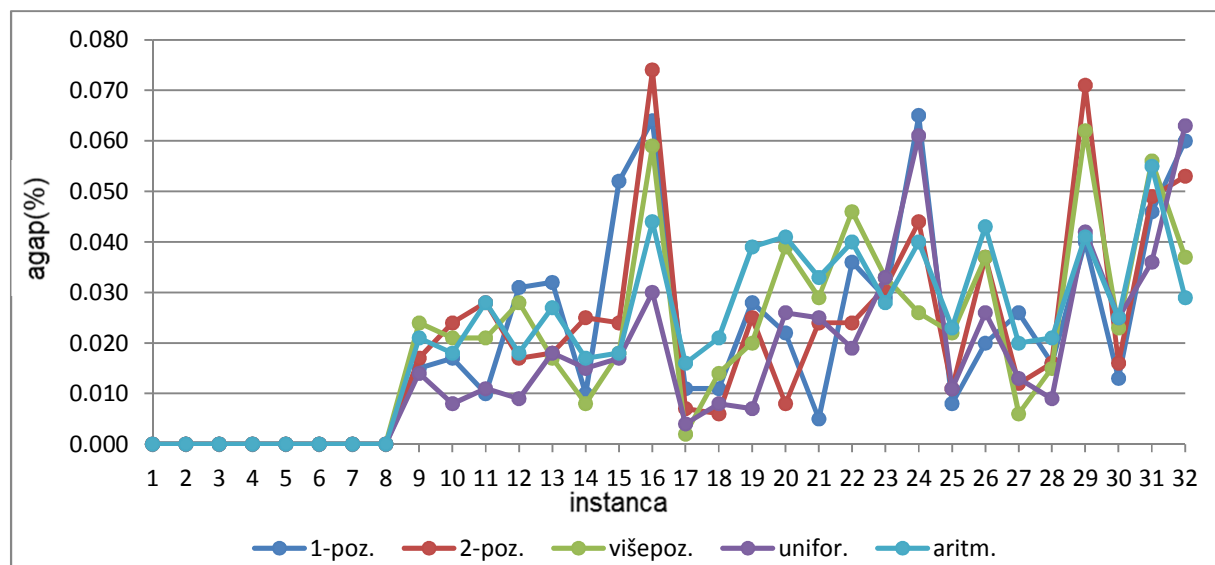


Grafik 3.1 Prikaz poređenja vremena izvršavanja GA sa različitim operatorima ukrštanja za instance prve grupe



Grafik 3.2 Prikaz poređenja vremena izvršavanja GA sa različitim operatorima ukrštanja za instance druge grupe

Na Grafiku 3.3 prikazano je prosečno odstupanje najboljeg GA rešenja od optimalnog za svih pet varijanti implementiranog GA. Za prvu grupu instanci nema prosečnog odstupanja od optimalnog rešenja, odnosno GA je dostigao sva optimalna rešenja (*agap* je 0.000% u svim slučajevima). U drugoj grupi instanci sa grafika se može primetiti da GA sa dvopozicionim operatorom ukrštanja ima najizraženija odstupanja u odnosu na ostale varijante GA.



Grafik 3.3 Prikaz poredjenja prosecnog odstupanja rešenja GA sa razlicitim operatorima ukrštanja za instance druge grupe

Iz prikazanih rezultata možemo izvući sledeće zaključke za performanse GA implementacija sa binarnim kodiranjem:

- Za instance prve grupe, genetski algoritam u kome su implementirani višepozicioni i aritmetički operator ukrštanja za par instanci imaju nešto duže vreme izvršavanja u odnosu na GA implementacije koje koriste druge operatore ukrštanja (jednopolozicioni, dvopolozicioni i višepozicioni);
- Za instance druge grupe, genetski algoritam u kome su implementirani jednopolozicioni i višepozicioni operator ukrštanja imaju nešto duže vreme izvršavanja u odnosu na ostale GA implementacije;
- Za prvu grupu instanci prosečno odstupanje od optimalnog rešenja - *agap* je 0.000% za sve testirane GA implementacije sa binarnim kodiranjem;
- Sa Grafika 3.3 se može primetiti da za drugu grupu instanci, GA sa dvopolozicionim operatorom ukrštanja ima najveća odstupanja u odnosu na ostale implementacije GA;
- GA sa binarnim kodiranjem i uniformnim operatorom ukrštanja ima najkraće vreme izvršavanja za instance iz obe grupe;
- GA sa binarnim kodiranjem i uniformnim operatorom ukrštanja ima najmanje prosečno odstupanje za instance iz obe grupe.

4. GA IMPLEMENTACIJA KOJA KORISTI CELOBROJNO KODIRANJE

4.1 Reprezentacija rešenja

Genetski kod jedinke je string dužine n (broj potencijalnih lokacija). Elementi stringa predstavljaju lokacije uspostavljenih snabdevača i uzimaju vrednost iz skupa $\{0, 1, \dots, n-1\}$. Svaki element stringa u genetskom kodu jedinke, čija je vrednost x , označava da je na lokaciji x uspostavljen snabdevač.

Primer 4.1: Za $n = 8$, genetski kod jedinke 75053275 označava da:

- su uspostavljeni snabdevači na lokacijama 1, 3, 4, 6 i 8;
- na lokacijama 2, 5 i 7 nisu uspostavljeni snabdevači.

Iz genetskog koda jedinke čitaju se uspostavljene lokacije (vrednost svakog y_i predstavlja lokaciju koja je uspostavljena). Isto kao i kod binarnog kodiranja, kada su poznate uspostavljene lokacije svaki korisnik može da izabere sebi najbližeg snabdevača.

4.2 Računanje funkcije cilja

Računanje funkcija cilja implementirano je na isti način kao i kod binarnog kodiranja, odnosno korišćen je dodatni memorijski prostor za uređene liste indeksa lokacija za svakog korisnika. Kada su poznate uspostavljene lokacije snabdevača, za svakog korisnika se određuje njemu najpovoljniji snabdevač tako što se pronade njegova uređena lista (najbližih) lokacija i pretražuje do prve pojave y_i .

4.3 Genetski operatori

4.3.1 Selekcija

Za operator selekcije najbolje rezultate je pokazala fino gradirana turnirska selekcija sa parametrom veličina turnira jednakim 5.4. Najboljih 50 jedinki direktno prolazi u novu generaciju, dok preostalih 100 učestvuje u selekciji (elitistička strategija zamene generacije).

4.3.2 Ukrštanje

Implementirano je pet različitih operatora ukrštanja za celobrojnu reprezentaciju rešenja: jednopoziciono, dvopoziciono, višepoziciono, uniformno i aritmetičko ukrštanje sa parametrom ukrštanja $p_{cross} = 0.85$. Kod uniformnog ukrštanja verovatnoća razmene proizvoljnog bita je

$p_{unif} = 0.6$, što znači da se približno 60% bitova razmenjuje između jedinki. Operator aritmetičkog ukrštanja je modiškovan u skladu sa kodiranjem odnosno:

- vrednost gena na poziciji 1 za Potomak 1 se dobija na sledeći način:

$$p_1 = \lambda * x_1 + (n - 1 - \lambda) * y_1 ;$$

- vrednost gena na poziciji 1 za Potomak 2 se dobija na sledeći način:

$$p'_1 = \lambda * y_1 + (n - 1 - \lambda) * x_1 ,$$

gde su $\lambda \in [0, n - 1]$, a x_1 i y_1 vrednosti gena roditelja na poziciji 1. Konačne vrednosti za potomke se dobijaju kao ostatak pri deljenju p_1 i p'_1 sa vrednošću n (n je broj potencijalnih lokacija).

4.3.3 Mutacija

Za operator mutacije korišćena je prosta mutacija sa parametrom mutacije bita $p_{mut} = 0.2$. Implementacija ovog operatora je takođe modifikovana zbog različitog načina kodiranja - ako član niza, koji predstavlja genetski kod jedinke, mutira onda on može biti zamenjen nekom slučajnom vrednošću iz niza $\{0, 1, \dots, n-1\}$, s tim da prednost imaju vrednosti koje se već nalaze u genetskom kodu jedinke (već je uspostavljena lokacija).

4.4 Ostali aspekti GA

Početna populacija je generisana na slučajan način. Za svaki član niza y_i , koji predstavlja genetski kod jedinke, na slučajan način je izabrana vrednost iz skupa $\{0, 1, \dots, n-1\}$, gde je n broj potencijalnih lokacija.

Broj jedinki u populaciji je 150. Korišćena je elitistička strategija tj. u svakoj generaciji 1/3 populacije (50 jedinki sa najboljim vrednostima funkcije prilagođenosti) direktno prelazi u sledeću generaciju, dok se nad preostalim 2/3 (100 jedinki) primenjuju genetski operatori ukrštanja i mutacije.

Za kriterijum završetka GA korišćena je kombinacija 2 kriterijuma zaustavljenja: dostignut maksimalni broj generacija (ograničen na 1000) i uzastopno ponavljanje najbolja jedinka maksimalan broj puta (300).

4.5 Eksperimentalni rezultati

Za testiranje GA koji koristi celobrojno kodiranje korišćene su iste instance i eksperimenti su vršeni na istom računaru. Rezultati u tabelama su prikazani istim redosledom i na isti način kao u Poglavlju 3. I u ovom slučaju, genetski algoritam je izvršavan 20 puta za svaku instancu.

U nastavku su Tabele 4.2 - 4.10 u kojima su prikazani detaljni rezultati izvršavanja opisanog GA sa celobrojnim kodiranjem i različitim tipovima ukrštanja za svaku od testiranih grupa instanci.

Tabela 4.1 Rezultati testiranja istanci iz prve grupe – jednopoziciono ukrštanje

<i>R. br.</i>	<i>n x m</i>	<i>opt_{sol}</i>	<i>best_{sol}</i>	<i>t(s)</i>	<i>t_{tot}(s)</i>	<i>gen</i>	<i>agap(%)</i>	<i>σ(%)</i>
1	5 x 10	2970.000	opt	0.002	0.023	300	0.000	0.000
2	10 x 10	2872.000	opt	0.002	0.049	303	0.011	0.023
3	10 x 20	4556.000	opt	0.004	0.067	307	0.000	0.000
4	10 x 20	5895.000	opt	0.003	0.067	306	0.022	0.018
5	10 x 20	7411.000	opt	0.003	0.068	305	0.001	0.001
6	10 x 20	8113.000	opt	0.004	0.081	306	0.011	0.011
7	10 x 20	10160.000	opt	0.003	0.066	300	0.000	0.000
8	10 x 20	5297.000	opt	0.003	0.067	300	0.002	0.003
9	15 x 30	9675.000	opt	0.007	0.140	326	0.004	0.006
10	15 x 30	9693.000	opt	0.007	0.139	323	0.008	0.007
11	15 x 30	6434.000	opt	0.006	0.118	312	0.012	0.011
12	15 x 30	14772.000	opt	0.010	0.215	338	0.024	0.029
13	15 x 30	6892.000	opt	0.006	0.118	315	0.014	0.020
14	15 x 30	8160.000	opt	0.006	0.116	316	0.007	0.008
15	15 x 30	7809.000	opt	0.006	0.120	323	0.000	0.000
16	15 x 30	10667.000	opt	0.007	0.145	317	0.030	0.034
17	15 x 30	6361.000	opt	0.007	0.145	312	0.008	0.018
18	15 x 30	7685.000	opt	0.008	0.159	335	0.018	0.021
19	15 x 30	15491.000	opt	0.009	0.175	332	0.014	0.013
20	20 x 40	13809.000	opt	0.013	0.293	328	0.029	0.029
21	20 x 40	14559.000	opt	0.023	0.413	339	0.040	0.045
22	20 x 40	14734.000	opt	0.021	0.420	353	0.039	0.044
23	20 x 40	8503.000	opt	0.011	0.229	328	0.007	0.008
24	20 x 40	8358.000	opt	0.012	0.267	327	0.041	0.042
25	20 x 40	9362.000	opt	0.016	0.325	335	0.028	0.016
26	20 x 40	12435.000	opt	0.015	0.281	333	0.018	0.015
27	20 x 40	13614.000	opt	0.031	0.452	344	0.087	0.078

Tabela 4.2 Rezultati testiranja istanci iz druge grupe – jednopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	opt	0.014	0.290	317	0.015	0.012
2	20 x 50	17916.000	opt	0.027	0.506	342	0.041	0.041
3	20 x 50	14275.000	opt	0.019	0.400	333	0.007	0.022
4	20 x 50	14552.000	opt	0.017	0.350	333	0.012	0.023
5	20 x 50	11360.000	opt	0.015	0.288	323	0.026	0.020
6	20 x 50	9718.000	opt	0.014	0.277	329	0.017	0.016
7	20 x 50	14276.000	opt	0.023	0.634	331	0.035	0.030
8	20 x 50	22625.000	opt	0.026	0.506	334	0.007	0.006
9	30 x 60	14783.000	opt	0.034	0.740	348	0.070	0.069
10	30 x 60	12992.000	opt	0.034	0.690	355	0.036	0.026
11	30 x 60	21946.000	opt	0.066	1.350	394	0.153	0.108
12	30 x 60	20526.000	opt	0.055	1.170	405	0.148	0.072
13	30 x 60	25983.000	opt	0.094	1.643	440	0.109	0.052
14	30 x 60	25467.000	opt	0.060	1.373	404	0.095	0.106
15	30 x 60	27451.000	opt	0.116	1.834	456	0.193	0.149
16	30 x 60	32414.000	opt	0.086	2.095	511	0.322	0.225
17	30 x 75	16829.000	opt	0.045	0.827	345	0.024	0.027
18	30 x 75	17919.000	opt	0.041	0.891	349	0.035	0.040
19	30 x 75	22591.000	opt	0.063	1.025	356	0.073	0.050
20	30 x 75	25691.000	opt	0.077	1.351	373	0.093	0.073
21	30 x 75	33821.000	opt	0.127	2.140	491	0.129	0.126
22	30 x 75	30325.000	opt	0.079	2.025	470	0.190	0.146
23	30 x 75	48555.000	49223.000	0.108	2.773	502	0.236	0.129
24	30 x 75	31900.000	opt	0.087	2.294	496	0.318	0.200
25	30 x 90	21143.000	opt	0.048	0.989	349	0.020	0.017
26	30 x 90	19629.000	opt	0.047	0.871	345	0.068	0.044
27	30 x 90	29358.000	opt	0.086	1.946	435	0.085	0.075
28	30 x 90	29931.000	opt	0.072	1.206	358	0.054	0.052
29	30 x 90	39005.000	opt	0.116	2.368	444	0.258	0.184
30	30 x 90	35576.000	opt	0.113	1.797	423	0.085	0.073
31	30 x 90	49324.000	opt	0.143	2.736	454	0.302	0.196
32	30 x 90	49755.000	opt	0.120	3.264	507	0.321	0.322

Tabela 4.3 Rezultati testiranja istanci iz prve grupe – dvopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.003	0.024	300	0.000	0.000
2	10 x 10	2872.000	opt	0.002	0.051	309	0.008	0.012
3	10 x 20	4556.000	opt	0.003	0.067	300	0.000	0.000
4	10 x 20	5895.000	opt	0.003	0.065	301	0.020	0.018
5	10 x 20	7411.000	opt	0.003	0.071	310	0.000	0.000
6	10 x 20	8113.000	opt	0.004	0.079	311	0.000	0.000
7	10 x 20	10160.000	opt	0.003	0.067	304	0.000	0.000
8	10 x 20	5297.000	opt	0.003	0.066	300	0.005	0.002
9	15 x 30	9675.000	opt	0.007	0.137	315	0.027	0.030
10	15 x 30	9693.000	opt	0.007	0.139	319	0.000	0.000
11	15 x 30	6434.000	opt	0.006	0.119	321	0.006	0.017
12	15 x 30	14772.000	opt	0.011	0.232	332	0.072	0.073
13	15 x 30	6892.000	opt	0.006	0.116	312	0.000	0.000
14	15 x 30	8160.000	opt	0.006	0.120	319	0.000	0.000
15	15 x 30	7809.000	opt	0.006	0.115	315	0.000	0.000
16	15 x 30	10667.000	opt	0.007	0.147	323	0.000	0.000
17	15 x 30	6361.000	opt	0.007	0.156	327	0.013	0.008
18	15 x 30	7685.000	opt	0.009	0.174	336	0.003	0.006
19	15 x 30	15491.000	opt	0.009	0.180	327	0.013	0.017
20	20 x 40	13809.000	opt	0.016	0.314	332	0.034	0.040
21	20 x 40	14559.000	opt	0.020	0.426	347	0.048	0.038
22	20 x 40	14734.000	opt	0.020	0.406	350	0.033	0.042
23	20 x 40	8503.000	opt	0.013	0.233	319	0.013	0.020
24	20 x 40	8358.000	opt	0.013	0.257	326	0.025	0.030
25	20 x 40	9362.000	opt	0.017	0.316	330	0.008	0.012
26	20 x 40	12435.000	opt	0.014	0.283	333	0.030	0.028
27	20 x 40	13614.000	opt	0.023	0.373	343	0.037	0.051

Tabela 4.4 Rezultati testiranja istanci iz druge grupe – dvopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	opt	0.013	0.266	319	0.024	0.022
2	20 x 50	17916.000	opt	0.028	0.485	335	0.054	0.037
3	20 x 50	14275.000	opt	0.019	0.404	329	0.029	0.041
4	20 x 50	14552.000	opt	0.019	0.348	330	0.014	0.019
5	20 x 50	11360.000	opt	0.014	0.276	332	0.002	0.006
6	20 x 50	9718.000	opt	0.013	0.270	320	0.025	0.027
7	20 x 50	14276.000	opt	0.022	0.451	338	0.028	0.023
8	20 x 50	22625.000	opt	0.020	0.485	343	0.016	0.021
9	30 x 60	14783.000	opt	0.039	0.741	357	0.027	0.026
10	30 x 60	12992.000	opt	0.032	0.663	360	0.021	0.025
11	30 x 60	21946.000	opt	0.055	1.319	398	0.118	0.096
12	30 x 60	20526.000	opt	0.074	1.349	433	0.113	0.077
13	30 x 60	25983.000	opt	0.064	1.542	432	0.089	0.052
14	30 x 60	25467.000	opt	0.057	1.275	402	0.096	0.081
15	30 x 60	27451.000	opt	0.114	2.032	490	0.125	0.106
16	30 x 60	32414.000	opt	0.094	2.333	545	0.328	0.246
17	30 x 75	16829.000	opt	0.044	0.821	348	0.023	0.021
18	30 x 75	17919.000	opt	0.051	0.883	348	0.039	0.040
19	30 x 75	22591.000	opt	0.055	1.134	363	0.063	0.044
20	30 x 75	25691.000	opt	0.067	1.409	382	0.085	0.072
21	30 x 75	33821.000	opt	0.074	1.900	440	0.128	0.122
22	30 x 75	30325.000	opt	0.110	1.951	424	0.230	0.174
23	30 x 75	48555.000	49223.000	0.127	2.140	491	0.129	0.126
24	30 x 75	31900.000	opt	0.079	2.203	476	0.225	0.160
25	30 x 90	21143.000	opt	0.049	1.063	350	0.029	0.025
26	30 x 90	19629.000	opt	0.048	0.910	350	0.053	0.033
27	30 x 90	29358.000	opt	0.088	1.785	399	0.077	0.060
28	30 x 90	29931.000	opt	0.060	1.315	364	0.055	0.049
29	30 x 90	39005.000	opt	0.107	2.965	521	0.238	0.189
30	30 x 90	35576.000	opt	0.103	2.265	458	0.092	0.062
31	30 x 90	49324.000	opt	0.187	2.915	514	0.164	0.087
32	30 x 90	49755.000	opt	0.133	3.342	507	0.329	0.238

Tabela 4.5 Rezultati testiranja istanci iz prve grupe – višepoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.003	0.024	300	0.000	0.000
2	10 x 10	2872.000	opt	0.002	0.051	309	0.008	0.012
3	10 x 20	4556.000	opt	0.003	0.067	300	0.000	0.000
4	10 x 20	5895.000	opt	0.003	0.065	301	0.020	0.018
5	10 x 20	7411.000	opt	0.003	0.071	310	0.000	0.000
6	10 x 20	8113.000	opt	0.004	0.079	311	0.000	0.000
7	10 x 20	10160.000	opt	0.003	0.067	304	0.000	0.000
8	10 x 20	5297.000	opt	0.003	0.066	300	0.005	0.002
9	15 x 30	9675.000	opt	0.007	0.137	315	0.027	0.030
10	15 x 30	9693.000	opt	0.007	0.139	319	0.000	0.000
11	15 x 30	6434.000	opt	0.006	0.119	321	0.006	0.017
12	15 x 30	14772.000	opt	0.011	0.232	332	0.072	0.073
13	15 x 30	6892.000	opt	0.006	0.116	312	0.000	0.000
14	15 x 30	8160.000	opt	0.006	0.120	319	0.000	0.000
15	15 x 30	7809.000	opt	0.006	0.115	315	0.000	0.000
16	15 x 30	10667.000	opt	0.007	0.147	323	0.000	0.000
17	15 x 30	6361.000	opt	0.007	0.156	327	0.013	0.008
18	15 x 30	7685.000	opt	0.009	0.174	336	0.003	0.006
19	15 x 30	15491.000	opt	0.009	0.180	327	0.013	0.017
20	20 x 40	13809.000	opt	0.016	0.314	332	0.034	0.040
21	20 x 40	14559.000	opt	0.020	0.426	347	0.048	0.038
22	20 x 40	14734.000	opt	0.020	0.406	350	0.033	0.042
23	20 x 40	8503.000	opt	0.013	0.233	319	0.013	0.020
24	20 x 40	8358.000	opt	0.013	0.257	326	0.025	0.030
25	20 x 40	9362.000	opt	0.017	0.316	330	0.008	0.012
26	20 x 40	12435.000	opt	0.014	0.283	333	0.030	0.028
27	20 x 40	13614.000	opt	0.023	0.373	343	0.037	0.051

Tabela 4.6 Rezultati testiranja istanci iz druge grupe – višepoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	opt	0.014	0.298	328	0.009	0.019
2	20 x 50	17916.000	opt	0.031	0.559	337	0.041	0.042
3	20 x 50	14275.000	opt	0.023	0.435	333	0.042	0.055
4	20 x 50	14552.000	opt	0.018	0.341	324	0.015	0.025
5	20 x 50	11360.000	opt	0.013	0.284	322	0.027	0.033
6	20 x 50	9718.000	opt	0.014	0.286	333	0.023	0.018
7	20 x 50	14276.000	opt	0.020	0.450	329	0.040	0.026
8	20 x 50	22625.000	opt	0.033	0.550	332	0.027	0.040
9	30 x 60	14783.000	opt	0.050	0.845	359	0.032	0.029
10	30 x 60	12992.000	opt	0.031	0.688	361	0.021	0.022
11	30 x 60	21946.000	opt	0.058	1.323	397	0.137	0.086
12	30 x 60	20526.000	opt	0.064	1.241	428	0.135	0.102
13	30 x 60	25983.000	26174.000	0.058	1.566	419	0.100	0.067
14	30 x 60	25467.000	opt	0.059	1.513	423	0.139	0.116
15	30 x 60	27451.000	opt	0.060	1.518	427	0.215	0.116
16	30 x 60	32414.000	33261.000	0.100	1.982	484	0.375	0.247
17	30 x 75	16829.000	opt	0.046	0.977	350	0.035	0.026
18	30 x 75	17919.000	opt	0.047	0.932	343	0.040	0.044
19	30 x 75	22591.000	opt	0.058	1.066	364	0.086	0.052
20	30 x 75	25691.000	opt	0.083	1.626	389	0.104	0.065
21	30 x 75	33821.000	opt	0.083	2.148	438	0.267	0.253
22	30 x 75	30325.000	opt	0.105	2.201	491	0.218	0.129
23	30 x 75	48555.000	49181.000	0.117	3.160	559	0.222	0.171
24	30 x 75	31900.000	opt	0.128	2.546	550	0.343	0.186
25	30 x 90	21143.000	opt	0.051	1.011	346	0.042	0.038
26	30 x 90	19629.000	opt	0.047	1.015	339	0.087	0.052
27	30 x 90	29358.000	opt	0.076	1.745	389	0.108	0.088
28	30 x 90	29931.000	opt	0.072	1.353	359	0.056	0.044
29	30 x 90	39005.000	opt	0.203	3.014	506	0.187	0.127
30	30 x 90	35576.000	opt	0.079	2.031	399	0.185	0.118
31	30 x 90	49324.000	opt	0.124	2.874	490	0.313	0.180
32	30 x 90	49755.000	52318.000	0.169	2.831	486	0.303	0.192

Tabela 4.7 Rezultati testiranja istanci iz prve grupe – uniformno ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.001	0.025	300	0.000	0.000
2	10 x 10	2872.000	opt	0.003	0.059	304	0.000	0.000
3	10 x 20	4556.000	opt	0.004	0.076	300	0.000	0.000
4	10 x 20	5895.000	opt	0.004	0.075	311	0.000	0.000
5	10 x 20	7411.000	opt	0.004	0.075	303	0.001	0.001
6	10 x 20	8113.000	opt	0.004	0.081	309	0.000	0.000
7	10 x 20	10160.000	opt	0.003	0.073	306	0.000	0.000
8	10 x 20	5297.000	opt	0.004	0.075	300	0.002	0.004
9	15 x 30	9675.000	opt	0.008	0.151	313	0.009	0.006
10	15 x 30	9693.000	opt	0.007	0.150	312	0.008	0.012
11	15 x 30	6434.000	opt	0.006	0.132	316	0.012	0.015
12	15 x 30	14772.000	opt	0.010	0.230	323	0.052	0.068
13	15 x 30	6892.000	opt	0.006	0.126	309	0.000	0.000
14	15 x 30	8160.000	opt	0.006	0.130	315	0.000	0.000
15	15 x 30	7809.000	opt	0.006	0.134	320	0.009	0.012
16	15 x 30	10667.000	opt	0.008	0.164	317	0.004	0.006
17	15 x 30	6361.000	opt	0.008	0.168	323	0.026	0.022
18	15 x 30	7685.000	opt	0.009	0.182	333	0.025	0.022
19	15 x 30	15491.000	opt	0.009	0.201	326	0.018	0.021
20	20 x 40	13809.000	opt	0.015	0.328	327	0.026	0.034
21	20 x 40	14559.000	opt	0.021	0.383	338	0.027	0.024
22	20 x 40	14734.000	opt	0.024	0.447	339	0.047	0.029
23	20 x 40	8503.000	opt	0.012	0.264	324	0.003	0.005
24	20 x 40	8358.000	opt	0.015	0.301	327	0.009	0.009
25	20 x 40	9362.000	opt	0.018	0.358	323	0.015	0.028
26	20 x 40	12435.000	opt	0.017	0.319	330	0.005	0.007
27	20 x 40	13614.000	opt	0.021	0.474	363	0.080	0.053

Tabela 4.8 Rezultati testiranja istanci iz druge grupe – uniformno ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	opt	0.015	0.294	333	0.005	0.008
2	20 x 50	17916.000	opt	0.026	0.546	342	0.045	0.046
3	20 x 50	14275.000	opt	0.025	0.456	333	0.001	0.004
4	20 x 50	14552.000	opt	0.020	0.393	330	0.025	0.064
5	20 x 50	11360.000	opt	0.017	0.321	328	0.015	0.012
6	20 x 50	9718.000	opt	0.015	0.321	332	0.020	0.022
7	20 x 50	14276.000	opt	0.022	0.399	328	0.031	0.025
8	20 x 50	22625.000	opt	0.027	0.554	341	0.025	0.033
9	30 x 60	14783.000	opt	0.043	0.820	355	0.032	0.031
10	30 x 60	12992.000	opt	0.041	0.715	361	0.032	0.035
11	30 x 60	21946.000	opt	0.070	1.416	406	0.109	0.095
12	30 x 60	20526.000	opt	0.079	1.185	403	0.141	0.080
13	30 x 60	25983.000	26383.000	0.068	1.377	406	0.072	0.057
14	30 x 60	25467.000	25535.000	0.065	1.529	429	0.088	0.061
15	30 x 60	27451.000	27773.000	0.088	1.885	496	0.203	0.119
16	30 x 60	32414.000	33261.000	0.108	1.659	444	0.290	0.253
17	30 x 75	16829.000	33261.000	0.046	0.854	349	0.019	0.028
18	30 x 75	17919.000	opt	0.043	0.889	343	0.044	0.042
19	30 x 75	22591.000	opt	0.056	1.016	359	0.072	0.047
20	30 x 75	25691.000	opt	0.068	1.446	390	0.060	0.044
21	30 x 75	33821.000	opt	0.174	2.202	498	0.108	0.098
22	30 x 75	30325.000	opt	0.106	2.094	465	0.212	0.128
23	30 x 75	48555.000	49181.000	0.114	2.659	487	0.226	0.114
24	30 x 75	31900.000	opt	0.133	2.552	545	0.269	0.153
25	30 x 90	21143.000	opt	0.044	0.942	346	0.015	0.024
26	30 x 90	19629.000	opt	0.046	0.969	339	0.059	0.048
27	30 x 90	29358.000	opt	0.096	1.819	413	0.068	0.066
28	30 x 90	29931.000	opt	0.060	1.199	360	0.046	0.048
29	30 x 90	39005.000	opt	0.127	2.630	495	0.158	0.142
30	30 x 90	35576.000	opt	0.093	1.982	417	0.110	0.098
31	30 x 90	49324.000	51493.000	0.121	2.653	490	0.173	0.147
32	30 x 90	49755.000	51145.000	0.149	3.205	521	0.267	0.185

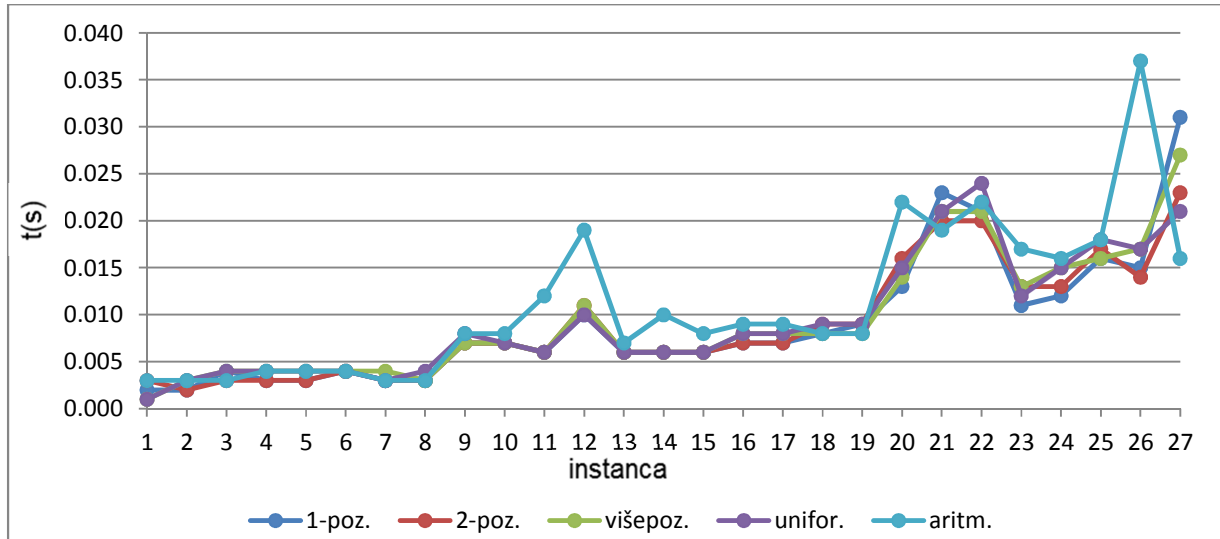
Tabela 4.9 Rezultati testiranja istanci iz prve grupe – aritmetičko ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.003	0.029	300	0.000	0.000
2	10 x 10	2872.000	opt	0.003	0.053	304	0.000	0.000
3	10 x 20	4556.000	opt	0.003	0.072	318	0.000	0.000
4	10 x 20	5895.000	opt	0.004	0.071	308	0.000	0.000
5	10 x 20	7411.000	opt	0.004	0.076	305	0.013	0.008
6	10 x 20	8113.000	opt	0.004	0.081	304	0.000	0.000
7	10 x 20	10160.000	opt	0.003	0.074	317	0.000	0.000
8	10 x 20	5297.000	opt	0.003	0.068	300	0.000	0.000
9	15 x 30	9675.000	opt	0.008	0.216	489	0.004	0.007
10	15 x 30	9693.000	opt	0.008	0.171	384	0.002	0.005
11	15 x 30	6434.000	opt	0.012	0.178	462	0.000	0.000
12	15 x 30	14772.000	opt	0.019	0.269	419	0.042	0.038
13	15 x 30	6892.000	opt	0.007	0.199	508	0.005	0.007
14	15 x 30	8160.000	opt	0.010	0.202	509	0.007	0.010
15	15 x 30	7809.000	opt	0.008	0.192	494	0.002	0.004
16	15 x 30	10667.000	opt	0.009	0.181	362	0.000	0.000
17	15 x 30	6361.000	opt	0.009	0.172	342	0.012	0.004
18	15 x 30	7685.000	opt	0.008	0.161	317	0.007	0.004
19	15 x 30	15491.000	opt	0.008	0.171	322	0.011	0.011
20	20 x 40	13809.000	13984.000	0.022	0.462	491	0.075	0.031
21	20 x 40	14559.000	opt	0.019	0.508	471	0.073	0.048
22	20 x 40	14734.000	14767.000	0.022	0.486	379	0.070	0.035
23	20 x 40	8503.000	opt	0.017	0.397	522	0.021	0.018
24	20 x 40	8358.000	opt	0.016	0.329	390	0.019	0.019
25	20 x 40	9362.000	9523.000	0.018	0.511	490	0.042	0.046
26	20 x 40	12435.000	opt	0.037	0.402	441	0.051	0.021
27	20 x 40	13614.000	opt	0.016	0.444	411	0.099	0.079

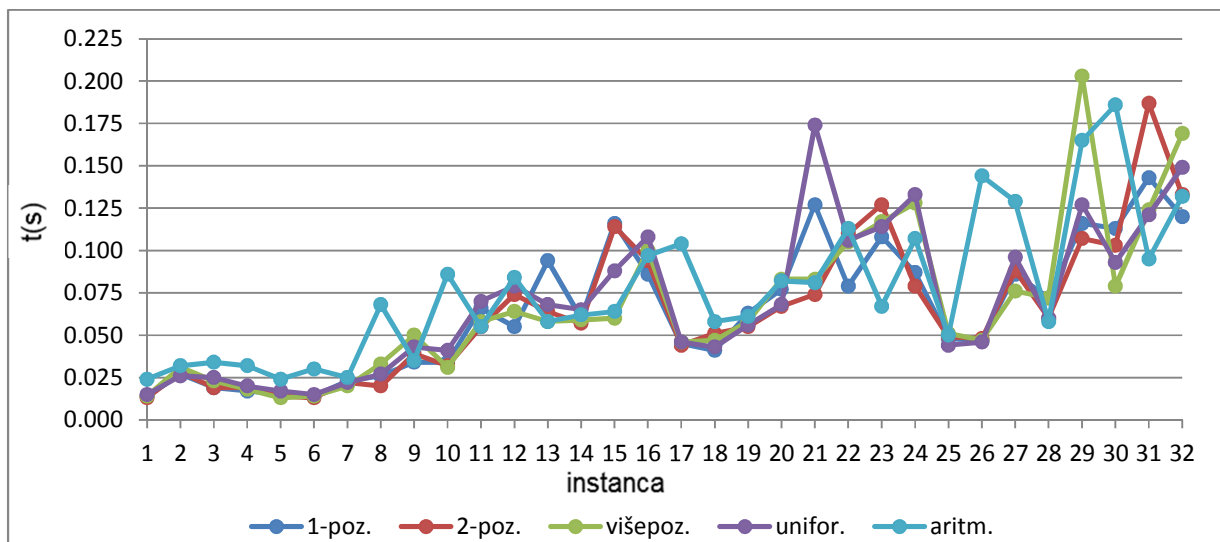
Tabela 4.10 Rezultati testiranja istanci iz druge grupe – aritmetičko ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	opt	0.024	0.450	478	0.015	0.010
2	20 x 50	17916.000	18470.000	0.032	0.716	548	0.059	0.054
3	20 x 50	14275.000	opt	0.034	0.603	496	0.089	0.045
4	20 x 50	14552.000	opt	0.032	0.551	491	0.044	0.051
5	20 x 50	11360.000	11457.000	0.024	0.488	496	0.041	0.019
6	20 x 50	9718.000	opt	0.030	0.412	422	0.016	0.010
7	20 x 50	14276.000	14502.000	0.025	0.553	460	0.047	0.044
8	20 x 50	22625.000	opt	0.068	0.712	476	0.035	0.034
9	30 x 60	14783.000	15239.000	0.035	1.165	583	0.081	0.038
10	30 x 60	12992.000	13444.000	0.086	1.195	558	0.104	0.061
11	30 x 60	21946.000	22880.000	0.055	1.633	507	0.226	0.133
12	30 x 60	20526.000	21908.000	0.084	1.635	528	0.153	0.057
13	30 x 60	25983.000	27184.000	0.058	1.331	414	0.169	0.082
14	30 x 60	25467.000	26417.000	0.062	1.738	526	0.183	0.078
15	30 x 60	27451.000	opt	0.064	1.878	493	0.136	0.098
16	30 x 60	32414.000	39240.000	0.097	2.434	483	0.593	0.283
17	30 x 75	16829.000	17149.000	0.104	1.566	597	0.070	0.063
18	30 x 75	17919.000	18573.000	0.058	1.638	595	0.112	0.077
19	30 x 75	22591.000	23707.000	0.061	1.206	428	0.096	0.041
20	30 x 75	25691.000	27682.000	0.082	1.717	453	0.166	0.066
21	30 x 75	33821.000	35837.000	0.081	2.317	551	0.218	0.111
22	30 x 75	30325.000	31836.000	0.113	2.170	448	0.372	0.134
23	30 x 75	48555.000	49223.000	0.067	2.269	439	0.165	0.153
24	30 x 75	31900.000	36645.000	0.107	2.361	379	0.135	0.091
25	30 x 90	21143.000	22888.000	0.050	1.182	370	0.118	0.035
26	30 x 90	19629.000	opt	0.144	1.803	615	0.107	0.068
27	30 x 90	29358.000	30910.000	0.129	2.096	533	0.156	0.079
28	30 x 90	29931.000	29986.000	0.058	2.067	581	0.119	0.108
29	30 x 90	39005.000	43377.000	0.165	3.044	459	0.363	0.149
30	30 x 90	35576.000	37444.000	0.186	2.638	555	0.228	0.081
31	30 x 90	49324.000	53905.000	0.095	2.943	451	0.325	0.142
32	30 x 90	49755.000	56782.000	0.132	3.120	430	0.296	0.115

Na Graficima 4.1 i 4.2 prikazano je poredjenje srednjeg početnog vremena izvršavanja GA za instance prve i druge grupe za svih pet varijanti implementacije GA sa celobrojnim kodiranjem. GA sa aritmetičkim operatorom ukrštanja je nešto sporiji u odnosu na ostale ostale implementacije GA, dok GA sa uniformnim operatorom ukrštanja najbrže dolazi do rešenja.

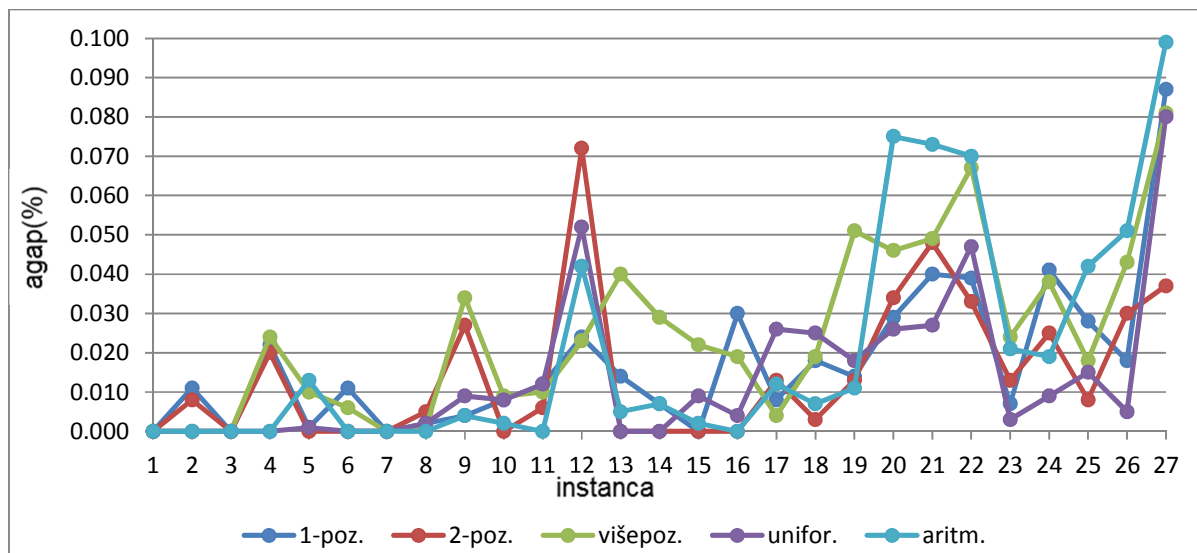


Grafik 4.1 Prikaz poredjenja vremena izvršavanja GA sa razlicitim operatorima ukrštanja za instance prve grupe

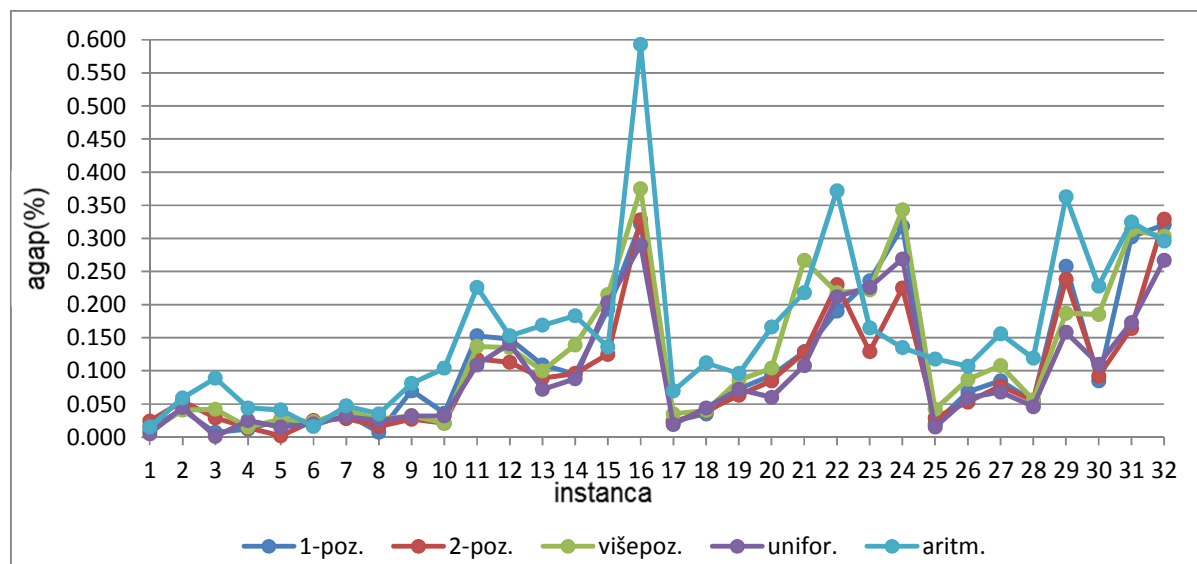


Grafik 4.2 Prikaz poredjenja vremena izvršavanja GA sa razlicitim operatorima ukrštanja za instance druge grupe

Na Grafcima 4.3 i 4.4 predstavljeno je poređenje prosečnog odstupanje najboljeg GA rešenja od optimalnog za instance prve, odnosno druge grupe. Za instance iz prve grupe najveća odstupanja najboljeg GA rešenja od optimalnog su u slučaju GA sa dvopozicionim i aritmetičkim operatorom ukrštanja, dok za instance druge grupe najveća odstupanja su u slučaju GA sa aritmetičkim operatorom ukrštanja.



Grafik 4.3 Prikaz poredjenja prosecnog odstupanja rešenja GA sa razlicitim operatorima ukrštanja za instance prve grupe



Grafik 4.4 Prikaz poredjenja prosecnog odstupanja rešenja GA sa razlicitim operatorima ukrštanja za instance druge grupe

Iz prikazanih rezultata možemo izvući sledeće zaključke za performanse GA implementacija sa celobrojnim kodiranjem:

- GA sa celobrojnim kodiranjem nije dostigao optimalna rešenja u svim varijantama. GA sa jednopozicionim i dvopozicionim ukrštanjem je za veći broj instanci dao optimalna rešenja. Ostale varijante GA su za instance iz prve grupe dostigla optimalna rešenja, dok u slučaju instanci iz druge grupe to nije uvek bio slučaj;
- GA sa aritmetičkim operatorom ukrštanja nešto sporije daje rešenja u odnosu na ostale ostale implementacije GA;
- GA sa uniformnim operatorom ukrštanja najbrže dolazi do rešenja;
- GA sa aritmetičkim operatorom ukrštanja ima najveće prosečno odstupanje najboljeg GA rešenja od optimalnog;
- GA sa uniformnim operatorom ukrštanja daje najbolja rešenja sa najmanjim prosečnim odstupanjem od optimalnih rešenja.

5. GA IMPLEMENTACIJA KOJA KORISTI MEŠOVITO KODIRANJE

5.1 Reprezentacija rešenja

Genetski kod jedinke je string dužine $2n$, gde je n broj potencijalnih lokacija snabdevača, odnosno sastoji se od n podstringova dužine 2 (n parova). Prvi član para uzima vrednost 0 ili 1 i predstavlja informaciju da li je snabdevač uspostavljen ili ne, dok je drugi član para ceo broj, može imati vrednost iz skupa $\{0, 1, \dots, n-1\}$ i predstavlja snabdevača odnosno lokaciju.

Primer 5.1: Za $n = 8$, genetski kod jedinke 07|13|16|05|12|11|04|10|01|14| označava da:

- su uspostavljeni snabdevači na lokacijama 2, 3, 5, 6 i 8
- na lokacijama 1, 4 i 7 nisu uspostavljeni snabdevači.

Iz genetskog koda jedinke čitaju se uspostavljene lokacije – u paru u kome je prvi član 1 vrednost drugog člana predstavlja uspostavljenog snabdevača. Kada su poznate uspostavljene lokacije svaki korisnik može sa izabere sebi najbližeg snabdevača.

5.2 Računanje funkcije cilja

Računanje funkcije cilja implementirano je na isti način kao i kod binarnog kodiranja odnosno korišćen je dodatni memorijski prostor za uređene liste indeksa lokacija za svakog korisnika.

5.3 Genetski operatori

5.3.1 Selekcija

Za operator selekcije najbolje rezultate je pokazala fino gradirana turnirska selekcija sa parametrom veličina turnira jednakim 5.4. Najboljih 50 jedinki direktno prolazi u novu generaciju, dok preostalih 100 učestvuje u selekciji (elitistička strategija zamene generacije).

5.3.2 Ukrštanje

Implementirano je pet različitih operatora ukrštanja za mešovitu reprezentaciju jedinki: jednopoziciono, dvopoziciono, višepoziciono, uniformno i aritmetičko ukrštanje sa parametrom ukrštanja $p_{cross} = 0.85$. Kod uniformnog ukrštanja verovatnoća razmene proizvoljnog bita je $p_{unif} = 0.3$, što znači da se približno 30% bitova razmenjuje između jedinki. Operatori ukrštanja su modifikovani u skladu sa kodiranjem rešenja. Za jednopoziciono, dvopoziciono i višepoziciono ukrštanje slučajaj izbor pozicija k tj. tački gde se vrši ukrštanje množi se sa dva da

bi se izbeglo presecanje genetskog materijala u tački koja razdvaja članove jednog para. Uniformno ukrštanje je modifikovano u smislu da se generiše maska dužine $2n$. Operator aritmetičkog ukrštanja je modifikovan na sledeći način: ukoliko se računa vrednost gena na neparnoj poziciji (da li je snabdevač uspostavljen ili ne) koristi se funkcija koja je implementirana u aritmetičkom operatoru za binarno kodiranje. Ukoliko se računa vrednost gena na parnoj poziciji koristi se funkcija koja je implementirana u aritmetičkom operatoru za celobrojno kodiranje.

5.3.3 Mutacija

Za operator mutacije korišćena je prosta mutacija sa parametrom mutacije bita $p_{mut} = 0.1$. Implementacija ovog operatora je takođe modifikovana zbog različitog načina kodiranja - ako mutira gen na neparnoj poziciji njegova nova vrednost može biti 0 ili 1, u suprotnom kada mutira gen na parnoj poziciji njegova nova vrednost može biti jedna iz skupa $\{0, 1, \dots, n\}$.

5.4 Ostali aspekti GA

Početna populacija je generisana na slučajan način. Za svaki neparan član niza y_i , koji predstavlja genetski kod jedinke, na slučajan način je izabran broj 0 ili 1 ali sa nejednakim verovatnoćama – verovatnoća da bude izabrana vrednost 0 je 70%, dok je 30% verovatnoća da će biti izabrana vrednost 1. Svaki paran član niza y_i može biti jedna od vrednosti iz skupa $\{0, 1, \dots, n-1\}$ koja se već ne nalazi u genetskom kodu jedinke. Na ovaj način je obezbeđeno da su sve potencijalne lokacije u genetskom kodu tj. svaka ima svoj par u kome je ona drugi član, dok prvi član određuje da li je snabdevač uspostavljen ta toj lokaciji ili ne (vrednost 0 ili 1).

Broj jedinki u populaciji je 150. Korišćena je elitistička strategija tj. u svakoj generaciji 1/3 populacije (50 jedinki sa najboljim vrednostima funkcije prilagođenosti) direktno prelazi u sledeću generaciju, dok se nad preostalim 2/3 (100 jedinki) primenjuju genetski operatori ukrštanja i mutacije.

Za kriterijum završetka GA korišćena je kombinacija 2 kriterijuma zaustavljenja: dostignut maksimalni broj generacija (ograničen na 1000) i ponavljanje najbolja jedinka maksimalan broj puta (200).

5.5 Eksperimentalni rezultati

Za testiranje GA koji koristi mešovito kodiranje korišćene su iste instance i eksperimenti su vršeni na istom računaru. Rezultati u tablici su prikazani istim redosledom i algoritam je izvršavan 20 puta za svaku instancu.

U nastavku su Tabele 5.1 - 5.10 u kojima su prikazani detaljni rezultati izvršavanja opisanog GA sa celobrojnim kodiranjem i različitim tipovima ukrštanja za svaku od testiranih grupa instanci.

Tabela 5.1 Rezultati testiranja istanci iz prve grupe – jednopoziciono ukrštanje

<i>R. br.</i>	<i>n x m</i>	<i>opt_{sol}</i>	<i>best_{sol}</i>	<i>t(s)</i>	<i>t_{tot}(s)</i>	<i>gen</i>	<i>agap(%)</i>	<i>σ(%)</i>
1	5 x 10	2970.000	opt	0.006	0.067	201	0.000	0.000
2	10 x 10	2872.000	2949.000	0.034	0.672	1000	0.038	0.038
3	10 x 20	4556.000	opt	0.045	0.901	1000	0.122	0.056
4	10 x 20	5895.000	opt	0.045	0.906	1000	0.090	0.055
5	10 x 20	7411.000	opt	0.045	0.897	1000	0.042	0.031
6	10 x 20	8113.000	opt	0.045	0.903	1000	0.048	0.035
7	10 x 20	10160.000	opt	0.052	1.031	1000	0.041	0.021
8	10 x 20	5297.000	5328.000	0.045	0.899	1000	0.084	0.054
9	15 x 30	9675.000	9703.000	0.099	1.964	1000	0.111	0.038
10	15 x 30	9693.000	opt	0.088	1.730	1000	0.096	0.040
11	15 x 30	6434.000	6827.000	0.099	1.964	1000	0.109	0.055
12	15 x 30	14772.000	opt	0.090	1.793	1000	0.093	0.042
13	15 x 30	6892.000	7004.000	0.100	1.978	1000	0.146	0.051
14	15 x 30	8160.000	8295.000	0.087	1.726	1000	0.162	0.060
15	15 x 30	7809.000	8022.000	0.101	1.990	1000	0.163	0.073
16	15 x 30	10667.000	10805.000	0.102	2.018	1000	0.108	0.056
17	15 x 30	6361.000	opt	0.100	1.967	1000	0.112	0.049
18	15 x 30	7685.000	opt	0.101	1.986	1000	0.113	0.048
19	15 x 30	15491.000	15844.000	0.089	1.762	1000	0.054	0.029
20	20 x 40	13809.000	14770.000	0.162	3.216	1000	0.089	0.042
21	20 x 40	14559.000	14724.000	0.147	2.958	1000	0.133	0.065
22	20 x 40	14734.000	15218.000	0.167	3.315	1000	0.095	0.045
23	20 x 40	8503.000	8868.000	0.159	3.162	1000	0.108	0.047
24	20 x 40	8358.000	8949.000	0.157	3.183	1000	0.104	0.048
25	20 x 40	9362.000	9661.000	0.163	3.244	1000	0.182	0.079
26	20 x 40	12435.000	13648.000	0.203	3.746	1000	0.127	0.063
27	20 x 40	13614.000	opt	0.148	3.158	1000	0.211	0.077

Tabela 5.2 Rezultati testiranja istanci iz druge grupe – jednopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	9791.000	0.183	3.656	1000	0.125	0.044
2	20 x 50	17916.000	18470.000	0.194	3.832	1000	0.126	0.056
3	20 x 50	14275.000	opt	0.190	3.808	1000	0.211	0.081
4	20 x 50	14552.000	15610.000	0.166	3.301	1000	0.179	0.068
5	20 x 50	11360.000	12227.000	0.183	3.654	1000	0.116	0.045
6	20 x 50	9718.000	10203.000	0.185	3.672	1000	0.105	0.046
7	20 x 50	14276.000	14750.000	0.188	3.908	1000	0.101	0.049
8	20 x 50	22625.000	22925.000	0.193	3.848	1000	0.078	0.040
9	30 x 60	14783.000	15609.000	0.331	6.608	1000	0.145	0.059
10	30 x 60	12992.000	14283.000	0.329	6.560	1000	0.164	0.045
11	30 x 60	21946.000	25808.000	0.353	6.954	1000	0.111	0.055
12	30 x 60	20526.000	23493.000	0.346	6.916	1000	0.161	0.073
13	30 x 60	25983.000	27992.000	0.357	7.145	1000	0.108	0.051
14	30 x 60	25467.000	25960.000	0.405	7.080	1000	0.254	0.087
15	30 x 60	27451.000	30651.000	0.322	6.252	1000	0.179	0.073
16	30 x 60	32414.000	34932.000	0.347	6.941	1000	0.544	0.206
17	30 x 75	16829.000	18305.000	0.383	7.653	1000	0.098	0.054
18	30 x 75	17919.000	20116.000	0.390	7.724	1000	0.106	0.050
19	30 x 75	22591.000	24795.000	0.359	7.181	1000	0.072	0.037
20	30 x 75	25691.000	28790.000	0.405	8.090	1000	0.141	0.047
21	30 x 75	33821.000	36986.000	0.406	8.107	1000	0.346	0.137
22	30 x 75	30325.000	35583.000	0.364	7.364	1000	0.202	0.096
23	30 x 75	48555.000	54016.000	0.405	8.135	1000	0.236	0.127
24	30 x 75	31900.000	42613.000	0.405	8.243	1000	0.184	0.089
25	30 x 90	21143.000	23098.000	0.446	8.908	1000	0.151	0.055
26	30 x 90	19629.000	21847.000	0.613	12.295	1000	0.135	0.049
27	30 x 90	29358.000	32050.000	0.414	8.272	1000	0.164	0.065
28	30 x 90	29931.000	33377.000	0.410	8.162	1000	0.079	0.044
29	30 x 90	39005.000	49121.000	0.419	8.364	1000	0.245	0.129
30	30 x 90	35576.000	36092.000	0.416	8.327	1000	0.309	0.090
31	30 x 90	49324.000	58725.000	0.470	9.319	1000	0.282	0.146
32	30 x 90	49755.000	58777.000	0.418	8.356	1000	0.430	0.185

Tabela 5.3 Rezultati testiranja istanci iz prve grupe – dvopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.004	0.067	201	0.000	0.000
2	10 x 10	2872.000	opt	0.037	0.760	1000	0.091	0.034
3	10 x 20	4556.000	opt	0.050	1.000	1000	0.125	0.076
4	10 x 20	5895.000	opt	0.047	0.933	1000	0.077	0.054
5	10 x 20	7411.000	opt	0.048	0.955	1000	0.046	0.032
6	10 x 20	8113.000	opt	0.046	0.919	1000	0.076	0.038
7	10 x 20	10160.000	opt	0.050	1.018	1000	0.046	0.019
8	10 x 20	5297.000	5328.000	0.048	0.935	1000	0.116	0.043
9	15 x 30	9675.000	opt	0.089	1.778	1000	0.100	0.057
10	15 x 30	9693.000	9813.000	0.102	1.969	1000	0.081	0.050
11	15 x 30	6434.000	6617.000	0.100	1.976	1000	0.120	0.065
12	15 x 30	14772.000	opt	0.104	2.052	1000	0.091	0.044
13	15 x 30	6892.000	7223.000	0.100	1.982	1000	0.113	0.050
14	15 x 30	8160.000	opt	0.090	1.814	1000	0.219	0.086
15	15 x 30	7809.000	8349.000	0.099	1.958	1000	0.111	0.059
16	15 x 30	10667.000	10770.000	0.102	2.008	1000	0.109	0.053
17	15 x 30	6361.000	opt	0.090	1.776	1000	0.085	0.043
18	15 x 30	7685.000	7757.000	0.101	2.002	1000	0.093	0.051
19	15 x 30	15491.000	15844.000	0.102	2.011	1000	0.054	0.037
20	20 x 40	13809.000	14712.000	0.164	3.241	1000	0.119	0.037
21	20 x 40	14559.000	15458.000	0.165	3.300	1000	0.094	0.052
22	20 x 40	14734.000	15126.000	0.167	3.323	1000	0.083	0.044
23	20 x 40	8503.000	8831.000	0.159	3.180	1000	0.117	0.055
24	20 x 40	8358.000	8758.000	0.142	2.817	1000	0.132	0.070
25	20 x 40	9362.000	10021.000	0.143	2.854	1000	0.144	0.067
26	20 x 40	12435.000	13626.000	0.163	3.242	1000	0.130	0.067
27	20 x 40	13614.000	opt	0.167	3.330	1000	0.188	0.071

Tabela 5.4 Rezultati testiranja istanci iz druge grupe – dvopoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	9791.000	0.183	3.656	1000	0.125	0.044
2	20 x 50	17916.000	19537.000	0.195	3.834	1000	0.072	0.041
3	20 x 50	14275.000	14559.000	0.190	3.773	1000	0.204	0.079
4	20 x 50	14552.000	16054.000	0.187	3.724	1000	0.120	0.062
5	20 x 50	11360.000	12284.000	0.185	3.686	1000	0.142	0.045
6	20 x 50	9718.000	10094.000	0.184	3.664	1000	0.105	0.039
7	20 x 50	14276.000	14750.000	0.188	3.908	1000	0.101	0.049
8	20 x 50	22625.000	22761.000	0.193	3.839	1000	0.081	0.041
9	30 x 60	14783.000	16385.000	0.331	6.577	1000	0.101	0.042
10	30 x 60	12992.000	14395.000	0.340	6.979	1000	0.137	0.051
11	30 x 60	21946.000	24938.000	0.361	7.243	1000	0.136	0.065
12	30 x 60	20526.000	23225.000	0.354	7.063	1000	0.164	0.063
13	30 x 60	25983.000	28308.000	0.346	6.961	1000	0.117	0.059
14	30 x 60	25467.000	28553.000	0.407	7.568	1000	0.118	0.084
15	30 x 60	27451.000	31155.000	0.361	7.130	1000	0.196	0.106
16	30 x 60	32414.000	35794.000	0.347	6.972	1000	0.604	0.248
17	30 x 75	16829.000	18111.000	0.400	7.865	1000	0.132	0.045
18	30 x 75	17919.000	19445.000	0.464	8.477	1000	0.154	0.050
19	30 x 75	22591.000	24118.000	0.397	7.968	1000	0.112	0.052
20	30 x 75	25691.000	27682.000	0.416	8.297	1000	0.157	0.053
21	30 x 75	33821.000	38636.000	0.430	8.511	1000	0.246	0.101
22	30 x 75	30325.000	34472.000	0.407	8.127	1000	0.314	0.108
23	30 x 75	48555.000	52038.000	0.427	8.427	1000	0.286	0.123
24	30 x 75	31900.000	42388.000	0.438	8.870	1000	0.177	0.094
25	30 x 90	21143.000	23107.000	0.455	8.976	1000	0.147	0.060
26	30 x 90	19629.000	22468.000	0.459	9.055	1000	0.098	0.034
27	30 x 90	29358.000	31854.000	0.642	12.906	1000	0.138	0.062
28	30 x 90	29931.000	33377.000	0.410	8.162	1000	0.079	0.044
29	30 x 90	39005.000	49121.000	0.419	8.364	1000	0.245	0.129
30	30 x 90	35576.000	40499.000	0.417	8.338	1000	0.164	0.077
31	30 x 90	49324.000	60003.000	0.550	9.880	1000	0.311	0.103
32	30 x 90	49755.000	59908.000	0.418	8.358	1000	0.351	0.170

Tabela 5.5 Rezultati testiranja istanci iz prve grupe – višepoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.007	0.069	200	0.000	0.000
2	10 x 10	2872.000	opt	0.037	0.748	1000	0.066	0.035
3	10 x 20	4556.000	opt	0.047	0.928	1000	0.106	0.056
4	10 x 20	5895.000	opt	0.084	1.663	1000	0.083	0.052
5	10 x 20	7411.000	opt	0.045	0.898	1000	0.051	0.035
6	10 x 20	8113.000	opt	0.052	1.287	1000	0.060	0.031
7	10 x 20	10160.000	opt	0.068	1.224	1000	0.034	0.016
8	10 x 20	5297.000	5374.000	0.045	0.901	1000	0.072	0.052
9	15 x 30	9675.000	9790.000	0.089	1.762	1000	0.095	0.044
10	15 x 30	9693.000	opt	0.104	2.099	1000	0.082	0.050
11	15 x 30	6434.000	6565.000	0.126	2.100	1000	0.134	0.047
12	15 x 30	14772.000	opt	0.090	1.802	1000	0.096	0.058
13	15 x 30	6892.000	7226.000	0.087	1.731	1000	0.100	0.057
14	15 x 30	8160.000	8757.000	0.102	1.981	1000	0.099	0.065
15	15 x 30	7809.000	8000.000	0.088	1.733	1000	0.152	0.066
16	15 x 30	10667.000	10819.000	0.101	2.007	1000	0.100	0.046
17	15 x 30	6361.000	opt	0.088	1.756	1000	0.093	0.065
18	15 x 30	7685.000	opt	0.100	1.979	1000	0.093	0.056
19	15 x 30	15491.000	opt	0.153	3.029	1000	0.074	0.047
20	20 x 40	13809.000	14865.000	0.237	4.734	1000	0.103	0.045
21	20 x 40	14559.000	opt	0.245	4.879	1000	0.176	0.070
22	20 x 40	14734.000	opt	0.148	2.947	1000	0.137	0.045
23	20 x 40	8503.000	8821.000	0.161	3.204	1000	0.125	0.049
24	20 x 40	8358.000	8582.000	0.161	3.201	1000	0.181	0.065
25	20 x 40	9362.000	9853.000	0.164	3.272	1000	0.150	0.073
26	20 x 40	12435.000	13005.000	0.165	3.270	1000	0.177	0.070
27	20 x 40	13614.000	14370.000	0.168	3.364	1000	0.147	0.059

Tabela 5.6 Rezultati testiranja istanci iz druge grupe – višepoziciono ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	9848.000	0.186	3.708	1000	0.105	0.053
2	20 x 50	17916.000	18899.000	0.194	3.858	1000	0.111	0.057
3	20 x 50	14275.000	opt	0.276	5.565	1000	0.218	0.092
4	20 x 50	14552.000	16349.000	0.277	5.523	1000	0.097	0.052
5	20 x 50	11360.000	12257.000	0.164	3.316	1000	0.119	0.048
6	20 x 50	9718.000	10248.000	0.269	5.421	1000	0.103	0.048
7	20 x 50	14276.000	14750.000	0.177	3.423	1000	0.103	0.057
8	20 x 50	22625.000	22774.000	0.194	4.425	1000	0.084	0.045
9	30 x 60	14783.000	15905.000	0.346	7.612	1000	0.126	0.044
10	30 x 60	12992.000	13687.000	0.397	7.352	1000	0.186	0.052
11	30 x 60	21946.000	24259.000	0.403	7.802	1000	0.172	0.065
12	30 x 60	20526.000	23771.000	0.473	8.683	1000	0.123	0.057
13	30 x 60	25983.000	28370.000	0.396	7.797	1000	0.110	0.051
14	30 x 60	25467.000	27958.000	0.359	7.257	1000	0.178	0.060
15	30 x 60	27451.000	29638.000	0.361	7.194	1000	0.214	0.097
16	30 x 60	32414.000	40652.000	0.349	6.929	1000	0.443	0.214
17	30 x 75	16829.000	18255.000	0.410	9.561	1000	0.116	0.045
18	30 x 75	17919.000	19477.000	0.389	7.771	1000	0.145	0.049
19	30 x 75	22591.000	24191.000	0.413	8.405	1000	0.116	0.056
20	30 x 75	25691.000	27015.000	0.407	8.122	1000	0.197	0.063
21	30 x 75	33821.000	37657.000	0.427	8.421	1000	0.321	0.131
22	30 x 75	30325.000	32275.000	0.420	8.424	1000	0.314	0.131
23	30 x 75	48555.000	49181.000	0.439	8.542	1000	0.342	0.145
24	30 x 75	31900.000	34092.000	0.421	8.656	1000	0.459	0.172
25	30 x 90	21143.000	23985.000	0.446	9.084	1000	0.118	0.056
26	30 x 90	19629.000	22804.000	0.442	8.791	1000	0.087	0.033
27	30 x 90	29358.000	32295.000	0.479	9.581	1000	0.124	0.063
28	30 x 90	29931.000	32437.000	0.585	11.402	1000	0.117	0.042
29	30 x 90	39005.000	47193.000	0.506	11.930	1000	0.373	0.188
30	30 x 90	35576.000	37717.000	0.495	9.888	1000	0.270	0.093
31	30 x 90	49324.000	59617.000	0.471	9.380	1000	0.256	0.142
32	30 x 90	49755.000	54853.000	0.529	10.515	1000	0.455	0.201

Tabela 5.7 Rezultati testiranja istanci iz prve grupe – uniformno ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.005	0.068	200	0.000	0.000
2	10 x 10	2872.000	opt	0.038	0.756	1000	0.058	0.043
3	10 x 20	4556.000	opt	0.050	1.021	1000	0.126	0.057
4	10 x 20	5895.000	opt	0.050	1.038	1000	0.101	0.056
5	10 x 20	7411.000	opt	0.045	0.943	1000	0.049	0.030
6	10 x 20	8113.000	opt	0.050	1.012	1000	0.058	0.031
7	10 x 20	10160.000	opt	0.050	1.020	1000	0.042	0.022
8	10 x 20	5297.000	5385.000	0.052	1.044	1000	0.075	0.042
9	15 x 30	9675.000	9818.000	0.089	1.770	1000	0.093	0.045
10	15 x 30	9693.000	9909.000	0.100	1.998	1000	0.084	0.043
11	15 x 30	6434.000	6563.000	0.100	2.040	1000	0.143	0.068
12	15 x 30	14772.000	opt	0.104	2.063	1000	0.079	0.053
13	15 x 30	6892.000	7154.000	0.150	3.046	1000	0.126	0.048
14	15 x 30	8160.000	8757.000	0.101	2.012	1000	0.107	0.068
15	15 x 30	7809.000	8253.000	0.104	2.018	1000	0.133	0.057
16	15 x 30	10667.000	10805.000	0.102	2.016	1000	0.109	0.062
17	15 x 30	6361.000	6419.000	0.100	1.990	1000	0.073	0.043
18	15 x 30	7685.000	opt	0.144	2.980	1000	0.105	0.064
19	15 x 30	15491.000	opt	0.102	2.034	1000	0.090	0.039
20	20 x 40	13809.000	14984.000	0.165	3.271	1000	0.090	0.048
21	20 x 40	14559.000	opt	0.168	3.332	1000	0.158	0.059
22	20 x 40	14734.000	15126.000	0.174	3.533	1000	0.098	0.053
23	20 x 40	8503.000	9042.000	0.161	3.221	1000	0.090	0.047
24	20 x 40	8358.000	8849.000	0.165	3.291	1000	0.158	0.063
25	20 x 40	9362.000	10117.000	0.163	3.256	1000	0.140	0.070
26	20 x 40	12435.000	13527.000	0.169	3.265	1000	0.111	0.055
27	20 x 40	13614.000	14370.000	0.242	4.921	1000	0.153	0.060

Tabela 5.8 Rezultati testiranja istanci iz druge grupe – uniformno ukrštanje

<i>R. br.</i>	<i>n x m</i>	<i>opt_{sol}</i>	<i>best_{sol}</i>	<i>t(s)</i>	<i>t_{tot}(s)</i>	<i>gen</i>	<i>agap(%)</i>	<i>σ(%)</i>
1	20 x 50	9223.000	9669.000	0.186	3.688	1000	0.155	0.052
2	20 x 50	17916.000	18645.000	0.193	3.852	1000	0.122	0.059
3	20 x 50	14275.000	15304.000	0.190	3.813	1000	0.131	0.069
4	20 x 50	14552.000	15987.000	0.278	5.575	1000	0.129	0.058
5	20 x 50	11360.000	12579.000	0.181	3.695	1000	0.091	0.040
6	20 x 50	9718.000	10283.000	0.185	3.683	1000	0.089	0.034
7	20 x 50	14276.000	14750.000	0.191	3.810	1000	0.119	0.055
8	20 x 50	22625.000	opt	0.194	3.838	1000	0.102	0.042
9	30 x 60	14783.000	15993.000	0.335	6.653	1000	0.118	0.064
10	30 x 60	12992.000	14803.000	0.342	7.155	1000	0.106	0.052
11	30 x 60	21946.000	23506.000	0.365	7.932	1000	0.192	0.087
12	30 x 60	20526.000	22434.000	0.361	7.222	1000	0.213	0.095
13	30 x 60	25983.000	28741.000	0.365	7.252	1000	0.102	0.047
14	30 x 60	25467.000	27745.000	0.345	6.905	1000	0.163	0.083
15	30 x 60	27451.000	29306.000	0.348	6.944	1000	0.234	0.117
16	30 x 60	32414.000	40092.000	0.347	6.925	1000	0.393	0.186
17	30 x 75	16829.000	18402.000	0.384	7.698	1000	0.122	0.046
18	30 x 75	17919.000	19560.000	0.407	8.280	1000	0.133	0.043
19	30 x 75	22591.000	24060.000	0.398	7.951	1000	0.105	0.049
20	30 x 75	25691.000	28524.000	0.406	8.109	1000	0.135	0.049
21	30 x 75	33821.000	38096.000	0.467	8.666	1000	0.289	0.125
22	30 x 75	30325.000	35932.000	0.409	8.143	1000	0.231	0.086
23	30 x 75	48555.000	51962.000	0.561	10.436	1000	0.287	0.145
24	30 x 75	31900.000	33400.000	0.409	8.233	1000	0.485	0.185
25	30 x 90	21143.000	22906.000	0.448	8.943	1000	0.146	0.070
26	30 x 90	19629.000	21985.000	0.440	8.785	1000	0.121	0.047
27	30 x 90	29358.000	32805.000	0.465	9.256	1000	0.113	0.056
28	30 x 90	29931.000	33763.000	0.494	11.333	1000	0.085	0.033
29	30 x 90	39005.000	50039.000	0.484	9.699	1000	0.235	0.109
30	30 x 90	35576.000	39170.000	0.468	9.345	1000	0.203	0.108
31	30 x 90	49324.000	61004.000	0.468	9.351	1000	0.212	0.096
32	30 x 90	49755.000	60343.000	0.469	10.098	1000	0.344	0.150

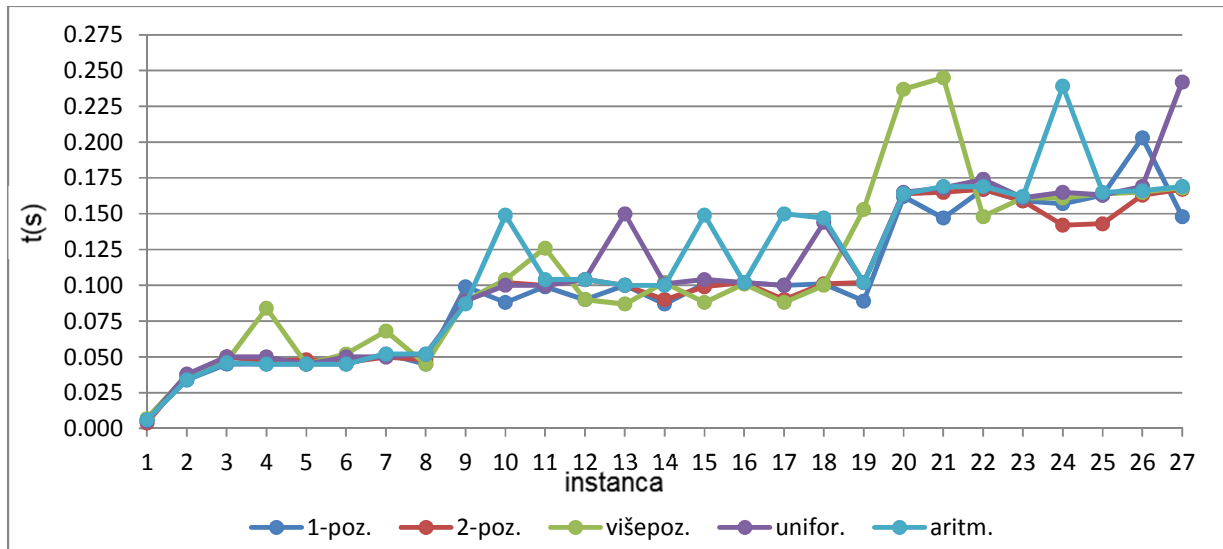
Tabela 5.9 Rezultati testiranja istanci iz prve grupe – aritmetičko ukrštanje

R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	5 x 10	2970.000	opt	0.006	0.069	200	0.000	0.000
2	10 x 10	2872.000	opt	0.034	0.686	1000	0.073	0.033
3	10 x 20	4556.000	4622.000	0.046	0.941	1000	0.083	0.068
4	10 x 20	5895.000	opt	0.045	0.906	1000	0.088	0.051
5	10 x 20	7411.000	opt	0.045	0.904	1000	0.035	0.033
6	10 x 20	8113.000	opt	0.045	0.910	1000	0.070	0.043
7	10 x 20	10160.000	opt	0.052	1.046	1000	0.038	0.021
8	10 x 20	5297.000	opt	0.052	1.056	1000	0.120	0.039
9	15 x 30	9675.000	9790.000	0.087	1.741	1000	0.109	0.046
10	15 x 30	9693.000	9927.000	0.149	3.012	1000	0.075	0.046
11	15 x 30	6434.000	6762.000	0.104	2.004	1000	0.103	0.046
12	15 x 30	14772.000	opt	0.104	2.069	1000	0.119	0.053
13	15 x 30	6892.000	7001.000	0.100	1.991	1000	0.160	0.060
14	15 x 30	8160.000	8388.000	0.100	2.006	1000	0.143	0.069
15	15 x 30	7809.000	8312.000	0.149	2.994	1000	0.112	0.067
16	15 x 30	10667.000	10770.000	0.102	2.028	1000	0.107	0.040
17	15 x 30	6361.000	opt	0.150	3.008	1000	0.093	0.058
18	15 x 30	7685.000	opt	0.147	3.015	1000	0.098	0.046
19	15 x 30	15491.000	15844.000	0.102	2.041	1000	0.045	0.031
20	20 x 40	13809.000	15273.000	0.164	3.273	1000	0.068	0.033
21	20 x 40	14559.000	15194.000	0.169	3.361	1000	0.116	0.064
22	20 x 40	14734.000	14960.000	0.169	3.378	1000	0.111	0.044
23	20 x 40	8503.000	8835.000	0.162	3.226	1000	0.123	0.053
24	20 x 40	8358.000	8836.000	0.239	4.768	1000	0.124	0.070
25	20 x 40	9362.000	10131.000	0.165	3.283	1000	0.123	0.064
26	20 x 40	12435.000	13315.000	0.166	3.283	1000	0.140	0.056
27	20 x 40	13614.000	13721.000	0.169	3.370	1000	0.197	0.068

Tabela 5.10 Rezultati testiranja istanci iz druge grupe – uniformno ukrštanje

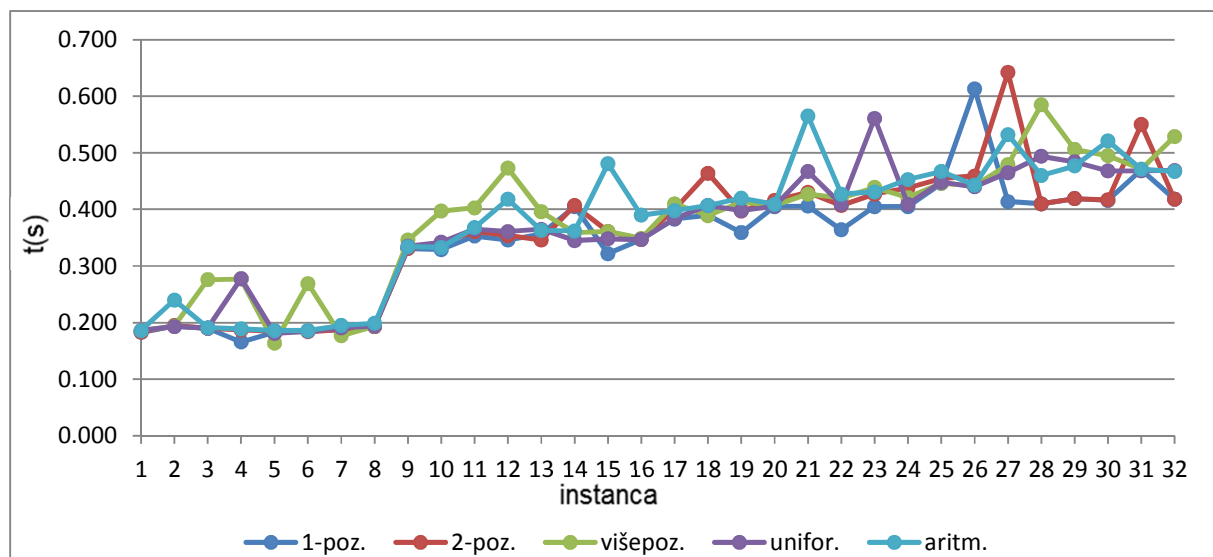
R. br.	n x m	opt_{sol}	best_{sol}	t(s)	t_{tot}(s)	gen	agap(%)	σ(%)
1	20 x 50	9223.000	9982.000	0.186	3.707	1000	0.123	0.050
2	20 x 50	17916.000	19281.000	0.240	4.200	1000	0.088	0.060
3	20 x 50	14275.000	15086.000	0.191	3.811	1000	0.187	0.070
4	20 x 50	14552.000	15505.000	0.189	3.774	1000	0.118	0.054
5	20 x 50	11360.000	12198.000	0.186	3.701	1000	0.130	0.053
6	20 x 50	9718.000	10332.000	0.186	3.688	1000	0.081	0.042
7	20 x 50	14276.000	14367.000	0.195	3.816	1000	0.153	0.054
8	20 x 50	22625.000	23191.000	0.199	3.860	1000	0.071	0.038
9	30 x 60	14783.000	15835.000	0.334	6.679	1000	0.133	0.046
10	30 x 60	12992.000	13935.000	0.333	6.661	1000	0.159	0.060
11	30 x 60	21946.000	24177.000	0.368	8.595	1000	0.158	0.060
12	30 x 60	20526.000	opt	0.418	7.080	1000	0.323	0.090
13	30 x 60	25983.000	28042.000	0.363	7.241	1000	0.129	0.058
14	30 x 60	25467.000	26731.000	0.362	7.286	1000	0.204	0.072
15	30 x 60	27451.000	31010.000	0.481	7.814	1000	0.183	0.078
16	30 x 60	32414.000	42670.000	0.390	7.396	1000	0.364	0.158
17	30 x 75	16829.000	18574.000	0.398	7.950	1000	0.121	0.040
18	30 x 75	17919.000	20186.000	0.407	8.133	1000	0.123	0.046
19	30 x 75	22591.000	23594.000	0.420	8.507	1000	0.147	0.043
20	30 x 75	25691.000	28111.000	0.409	8.184	1000	0.152	0.068
21	30 x 75	33821.000	37010.000	0.565	10.510	1000	0.309	0.147
22	30 x 75	30325.000	33406.000	0.427	10.476	1000	0.295	0.109
23	30 x 75	48555.000	52612.000	0.431	9.975	1000	0.268	0.129
24	30 x 75	31900.000	39797.000	0.453	8.674	1000	0.257	0.114
25	30 x 90	21143.000	23609.000	0.467	9.337	1000	0.128	0.047
26	30 x 90	19629.000	22472.000	0.443	8.851	1000	0.101	0.036
27	30 x 90	29358.000	31923.000	0.532	11.700	1000	0.146	0.063
28	30 x 90	29931.000	33097.000	0.460	9.182	1000	0.091	0.043
29	30 x 90	39005.000	55006.000	0.477	9.595	1000	0.156	0.099
30	30 x 90	35576.000	39088.000	0.521	10.027	1000	0.179	0.095
31	30 x 90	49324.000	59990.000	0.471	9.438	1000	0.229	0.131
32	30 x 90	49755.000	56782.000	0.467	9.524	1000	0.411	0.178

Na Graficima 5.1 i 5.2 prikazano je poređenje srednjeg početnog vremena GA implementacije sa celobrojnim kodiranjem i različitim tipovima ukrštanja za instance prve, odnosno druge grupe.



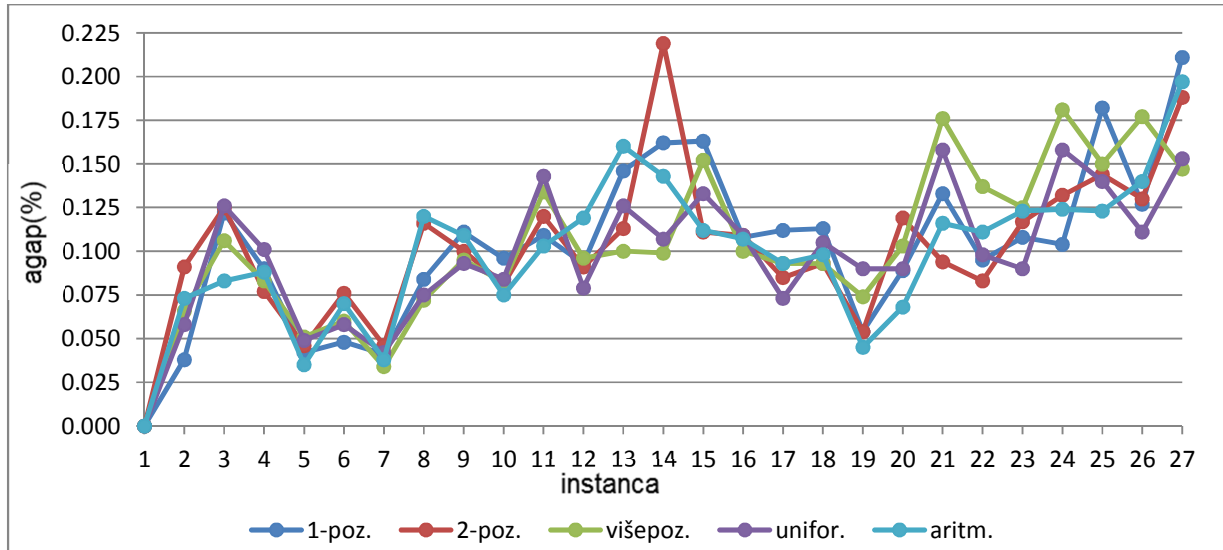
Grafik 5. 1 Prikaz poredjenja vremena izvršavanja GA sa različitim operatorima ukrštanja za instance prve grupe

GA sa višepozicionim operatorom ukrštanja nešto duže vreme izvršavanja u odnosu na druge implementacije GA.

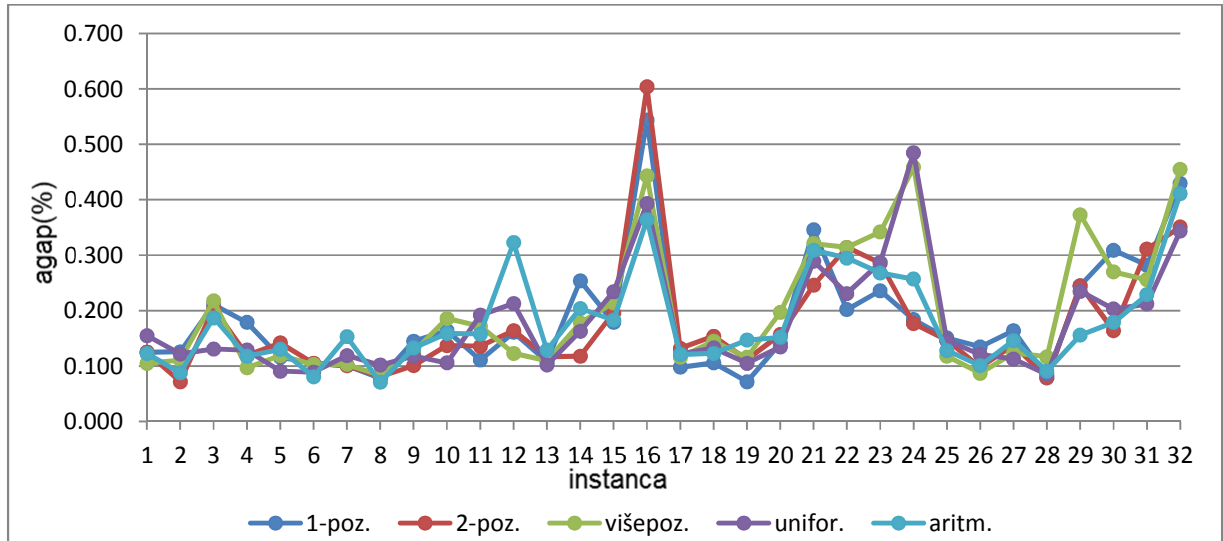


Grafik 5. 2 Prikaz poredjenja vremena izvršavanja GA sa različitim operatorima ukrštanja za instance druge grupe

Na Graficima 5.3 i 5.4 prikazano poređenje prosečnog odstupanja najboljeg GA rešenja od optimalnog za sve implementacije GA i obe grupe instanci. Sa Grafika 5.3 i 5.4 se može primetiti da GA koji implementira dvopozicioni operator ukrštanja ima najveće odstupanje GA rešenja od optimalnog.



Grafik 5.3 Prikaz poredjenja prosecnog odstupanja rešenja GA sa razlicitim operatorima ukrštanja za instance prve grupe



Grafik 5.4 Prikaz poredjenja prosecnog odstupanja rešenja GA sa razlicitim operatorima ukrštanja za instance druge grupe

Iz prikazanih rezultata možemo izvući sledeće zaključke za performanse GA implementacija sa mešovitim kodiranjem:

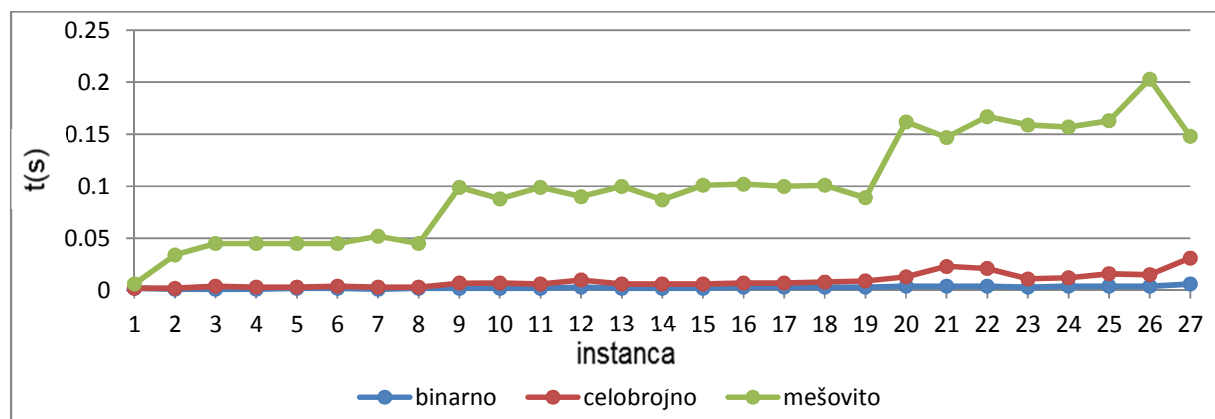
- GA sa mešovitim načinom kodiranja za većinu instanci nije dao optimalna rešenja;
- GA u kome je implementiran višepozicioni operator ukrštanja dao je najbolja rešenja najbliža optimalnim rešenjima;
- GA sa višepozicionim ukrštanjem najsporije dolazi do rešenja;
- GA sa uniformnim ukrštanjem najbrže dolazi do rešenja;
- GA sa dvopozicionim ukrštanjem proizvodi najbolja rešenja koja imaju najveće odstupanje od optimalnih rešenja;
- GA sa višepozicionim ukrštanjem daje najbolja rešenja koja imaju najmanje odstupanje od optimalnih rešenja.

6. POREĐENJE REZULTATA

U ovom poglavlju, predstavljena je uporedna analiza performansi GA implementacija koje koriste različite načine kodiranja i različite operatore ukrštanja. Za svaki od tipova ukrštanja upoređene su performanse GA sa različitim načinima kodiranja za instance iz obe grupe. Upoređeno je prosečno početno vreme izvršavanja GA i prosečno odstupanje najboljeg GA rešenja od optimalnog, a rezultati poređenja su prikazani Graficima 6.1–6.20. Na Ox osi Grafika 6.1-6.20 nalazi se redni broj instance iz prve, odnosno druge grupe, dok se na Oy osi nalazi prosečno početno vreme $t(s)$, odnosno prosečno odstupanje $agap(\%)$.

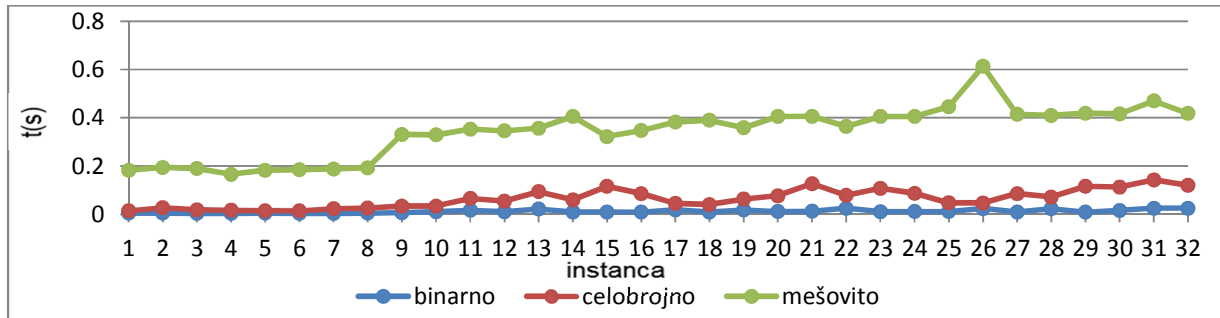
6.1 Prosečno početno vreme izvršavanja algoritma

Sa Grafika 6.1, koji se odnosi na jednopoziciono ukrštanje, može se videti da GA sa binarnim načinom kodiranja najbrže dolazi do rešenja za instance iz prve grupe. Za instance manjih dimenzija i GA sa celobrojnim načinom kodiranja radi relativno brzo. GA sa mešovitim kodiranjem znatno sporije dolazi do rešenja čak i za instance manjih dimenzija.



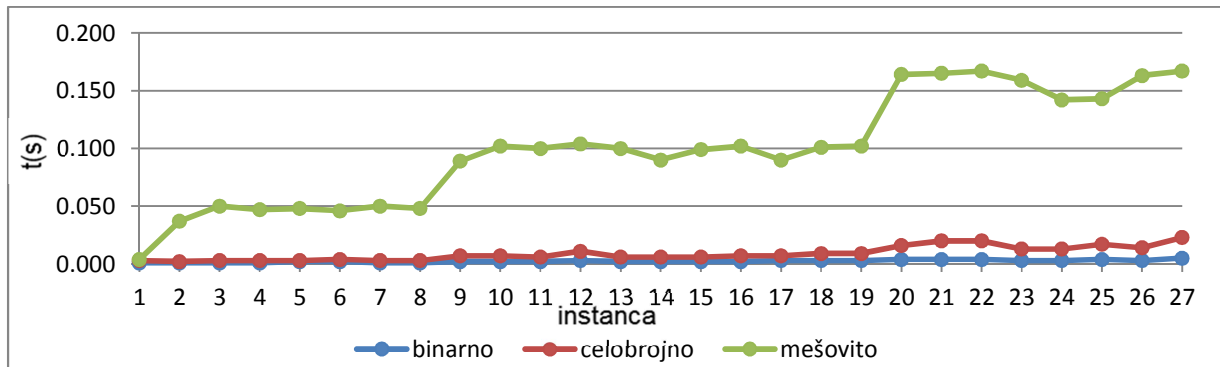
Grafik 6.1 Poređenje vremena izvršavanja $t(s)$ za instance prve grupe – GA sa jednopozicionim ukrštanjem

Sa Grafika 6.2 može se primetiti da GA sa celobrojnim načinom kodiranja i jednopozicionim ukrštanjem ne prati više brzinu izvršavanja GA sa binarnim kodiranjem, već postepeno usporava sa povećanjem dimenzije problema. GA sa mešovitim kodiranjem i jednopozicionim ukrštanjem je i dalje znatno sporiji u odnosu na GA sa binarnim, odnosno celobrojnim kodiranjem i jednopozicionim ukrštanjem.



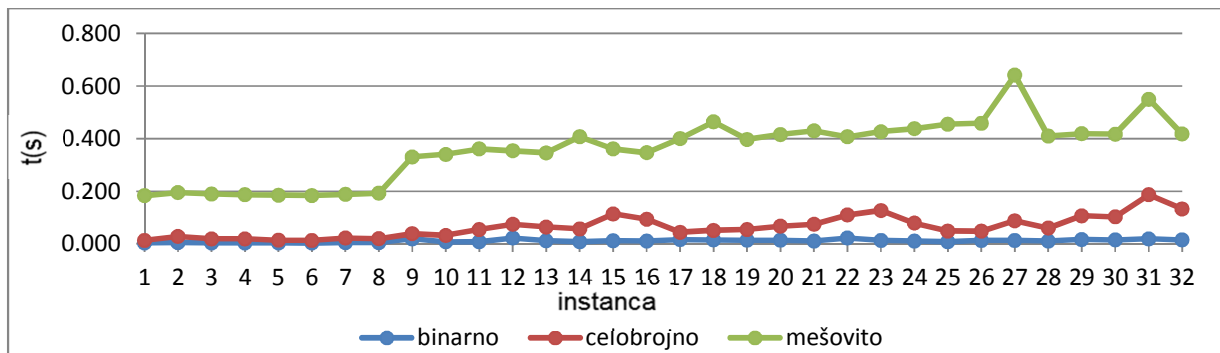
Grafik 6.2 Poređenje vremena izvršavanja $t(s)$ za instance druge grupe – GA sa jednopozicionim ukrštanjem

Na instancama iz prve grupe GA sa dvopozicionim operatorom ukrštanja dosta brzo dolazi do rešenja bez obzira da li koristi binarno ili celobrojno kodiranje, dok GA sa dvopozicionim ukrštanjem i mešovitim kodiranjem dosta sporo dolazi do rešenja koja pritom nisu zadovoljavajućeg kvaliteta (Grafik 6.3).



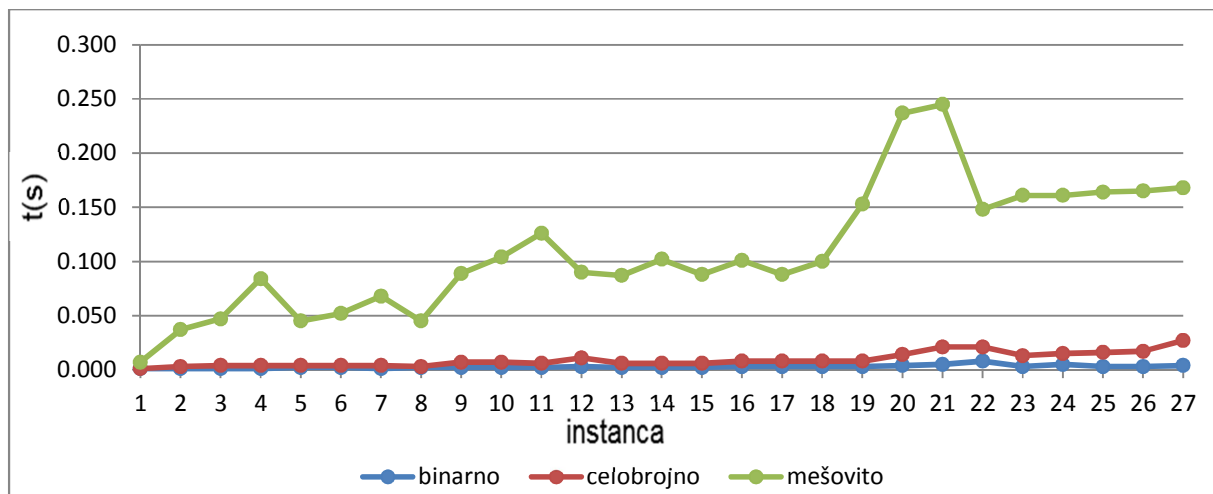
Grafik 6.3 Poređenje vremena izvršavanja $t(s)$ za instance prve grupe – GA sa dvopozicionim ukrštanjem

Na instancama iz druge grupe GA sa dvopozicionim operatorom ukrštanja i binarnim kodiranjem najbrže dolazi do kvalitetnih rešenja, dok GA sa celobrojnim kodiranjem postepeno usporava sa povećanjem dimanzije problema. GA sa mešovitim kodiranjem je znatno sporiji i daje nedovoljno kvalitetna rešenja (Grafik 6.4).



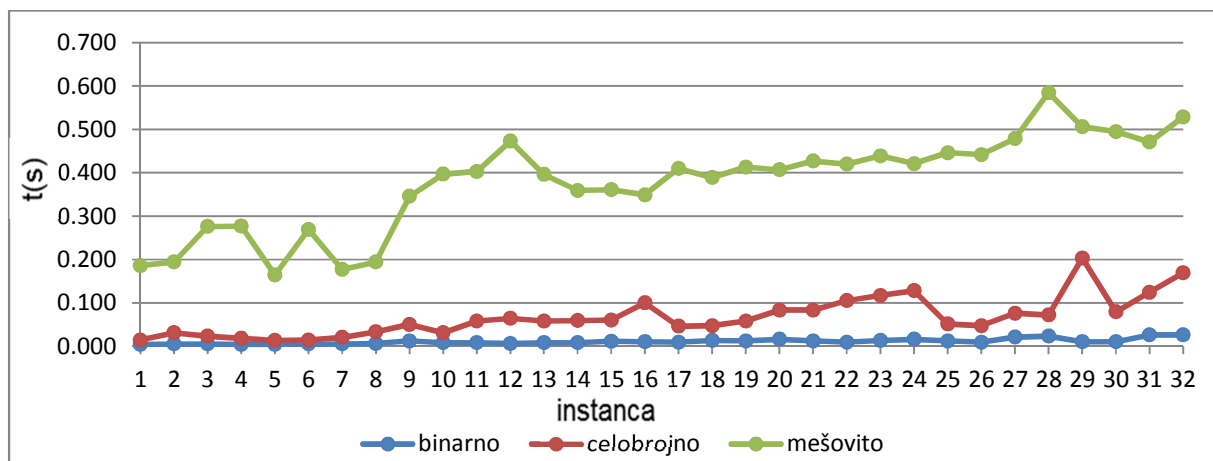
Grafik 6.4 Poređenje vremena izvršavanja $t(s)$ za instance druge grupe – GA sa dvopozicionim ukrštanjem

U slučaju instanci iz prve grupe GA sa višepozicionim operatorom ukrštanja dosta brzo dolazi do rešenja bez obzira da li koristi binarno ili celobrojno kodiranje. Primetimo da je ovde razlika u vremenu izvršavanja manja u odnosu na slučaj GA sa dvopozicionum ukrštanjem. GA sa mešovitim kodiranjem dosta sporo dolazi do rešenja koja nisu zadovoljavajućeg kvaliteta (Grafik 6.5).



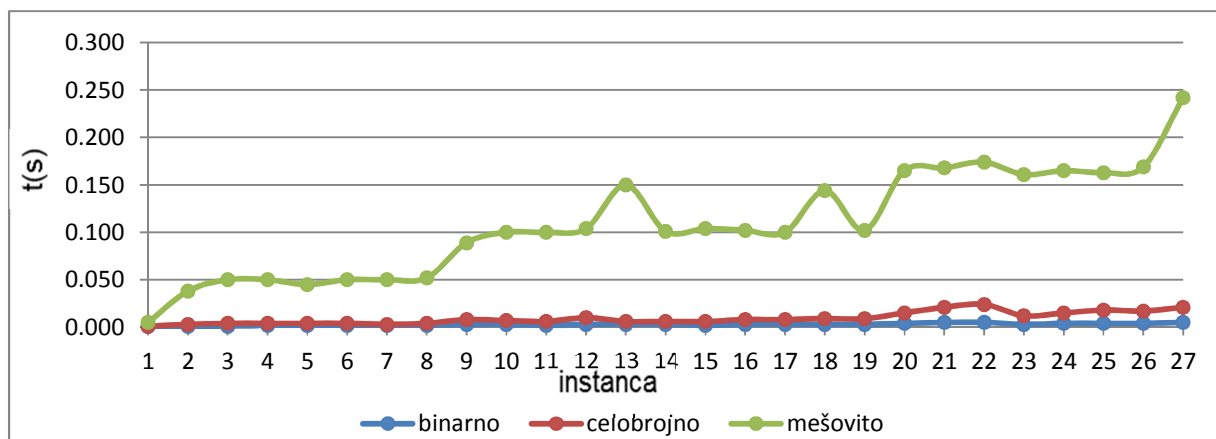
Grafik 6.5 Poređenje vremena izvršavanja $t(s)$ za instance prve grupe – GA sa višepozicionim ukrštanjem

Sa Grafika 6.6 može se primetiti da GA sa višepozicionim operatorom ukrštanja i binarnim kodiranjem najbrže dolazi do kvalitetnih rešenja, dok su zapažanja za GA sa celobrojnim i mešovitim kodiranjem ista kao u prethodnom primeru za GA sa dvopozicionim ukrštanjem.



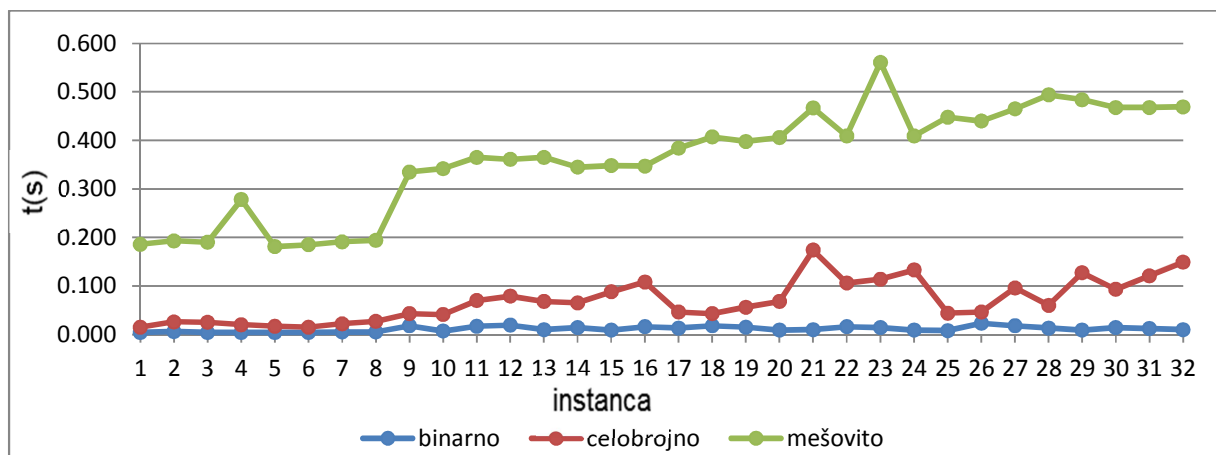
Grafik 6.6 Poređenje vremena izvršavanja $t(s)$ za instance druge grupe – GA sa višepozicionim ukrštanjem

Na instancama iz prve grupe, GA sa uniformnim operatorom ukrštanja dosta brzo dolazi do rešenja bez obzira da li koristi binarno ili celobrojno kodiranje (sa sličnim razlikama u vremenu izvršavanja kao GA sa dvopozicionim ukrštanjem). I u slučaju korišćenja uniformnog operatora ukrštanja, GA sa mešovitim kodiranjem sporo dolazi do rešenja koja nisu zadovoljavajućeg kvaliteta (Grafik 6.7).



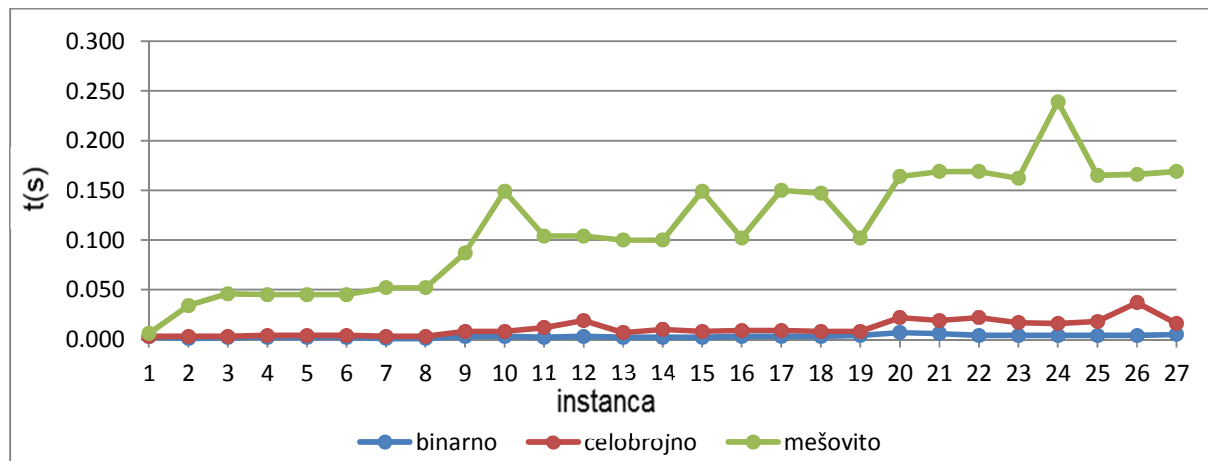
Grafik 6.7 Poređenje vremena izvršavanja $t(s)$ za instance prve grupe – GA sa uniformnim ukrštanjem

Grafik 6.8 pokazuje da na instancama iz druge grupe GA sa uniformnim operatorom ukrštanja i binarnim kodiranjem najbrže dolazi do kvalitetnih rešenja, dok su zapažanja za GA sa celobrojnim i mešovitim kodiranjem ista kao u prethodnom primeru za GA sa dvopozicionim ukrštanjem.



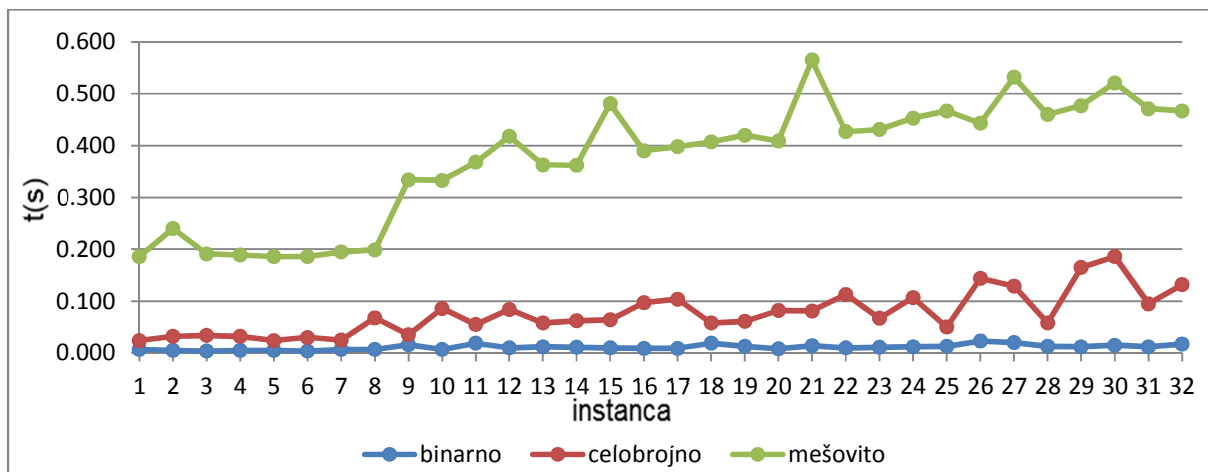
Grafik 6.8 Poređenje vremena izvršavanja $t(s)$ za instance druge grupe – GA sa uniformnim ukrštanjem

GA sa aritmetičkim operatorom ukrštanja, testiran na instancama iz prve grupe, najbrže dolazi do rešenja kada koristi binarno kodiranje, neznatno sporiji je kada koristi celobrojno kodiranje, a znatno sporiji sa mešovitim načinom kodiranja (Grafik 6.9).



Grafik 6.9 Poređenje vremena izvršavanja $t(s)$ za instance prve grupe – GA sa aritmetičkim ukrštanjem

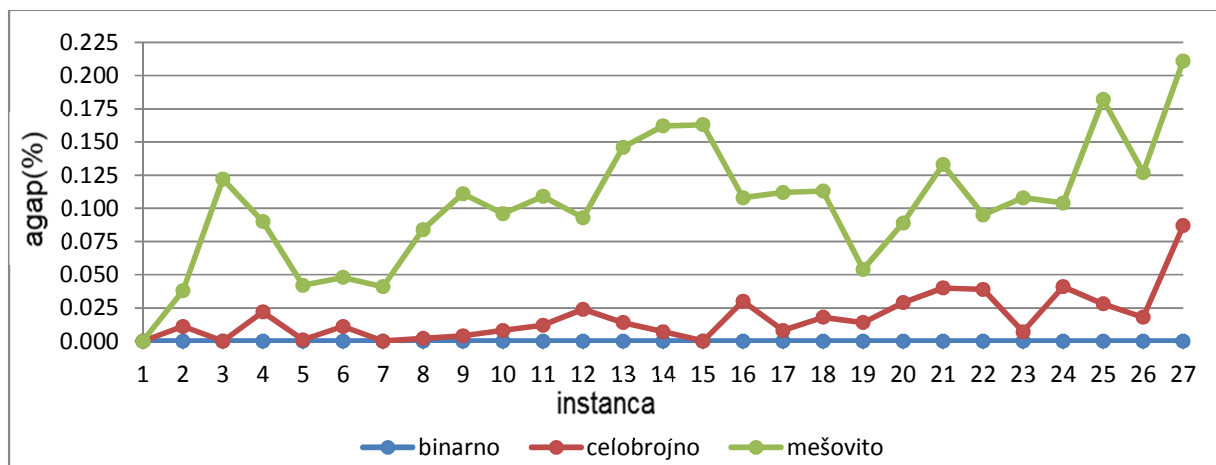
U slučaju instanci iz druge grupe, GA sa aritmetičkim operatorom ukrštanja najbrže daje rešenja kada se koristi binarno kodiranje. Sa povećanjem dimenzije problema GA sa celobrojnim kodiranjem postaje nešto sporiji, dok je GA sa mešovitim kodiranjem i aritmetičkim operatorom ukrštanja znatno sporiji u odnosu na GA sa ostala dva načina kodiranja (Grafik 6.10).



Grafik 6.10 Poređenje vremena izvršavanja $t(s)$ za instance druge grupe – GA sa aritmetičkim ukrštanjem

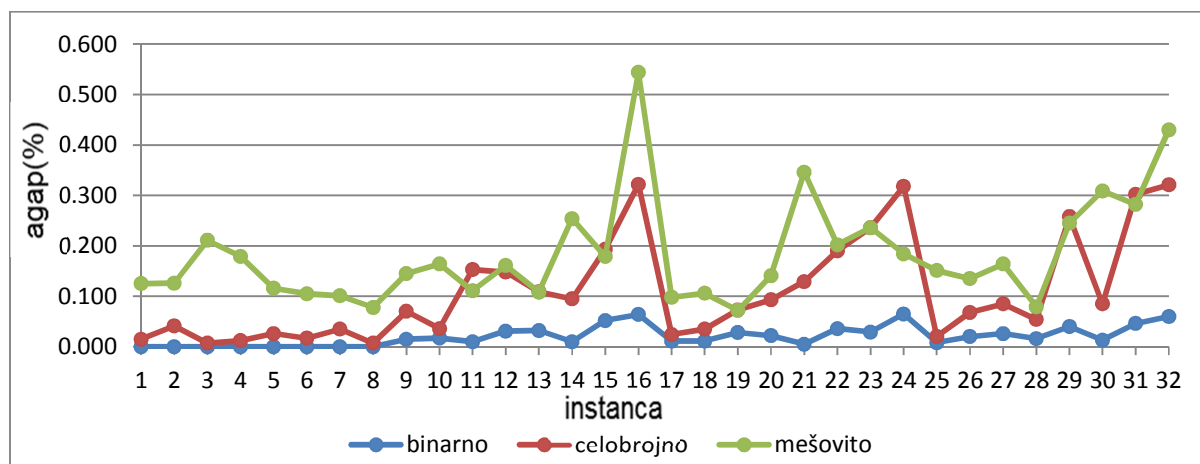
6.2 Prosečno odstupanje najboljeg GA rešenja od optimalnog

Posmatrajući Grafik 6.11, može se zaključiti da GA sa jednopozicionim ukrštanjem i binarnim kodiranjem, za instance iz prve grupe, dostiže sva optimalna rešenja (*agap* je 0.000%). GA sa celobrojnim kodiranjem ima neznatna odstupanja, dok GA sa mešovitim kodiranjem ima veća odstupanja iako ne daje dovoljno kvalitetna rešenja.



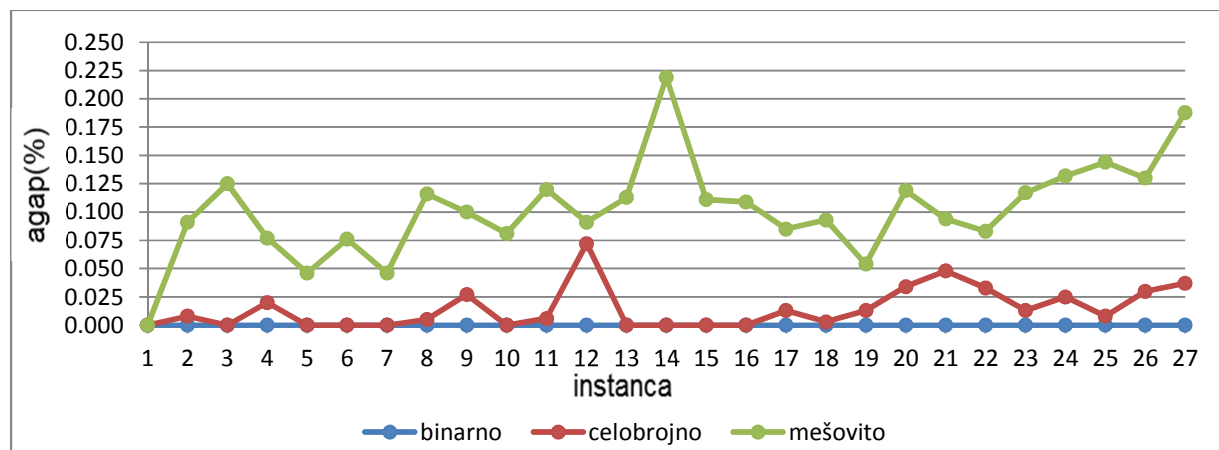
Grafik 6.11 Poređenje prosečnog odstupanja *agap*(%) za instance prve grupe – GA sa jednopozicionim ukrštanjem

Sa Grafika 6.12 se može videti da sa povećavanjem dimenzije problema dolazi i do većih odstupanja najboljih GA rešenja od optimalnih. Odstupanja su neznatna u slučaju GA sa binarnim kodiranjem, dok GA sa mešovitim načinom kodiranja ima najveća odstupanja.

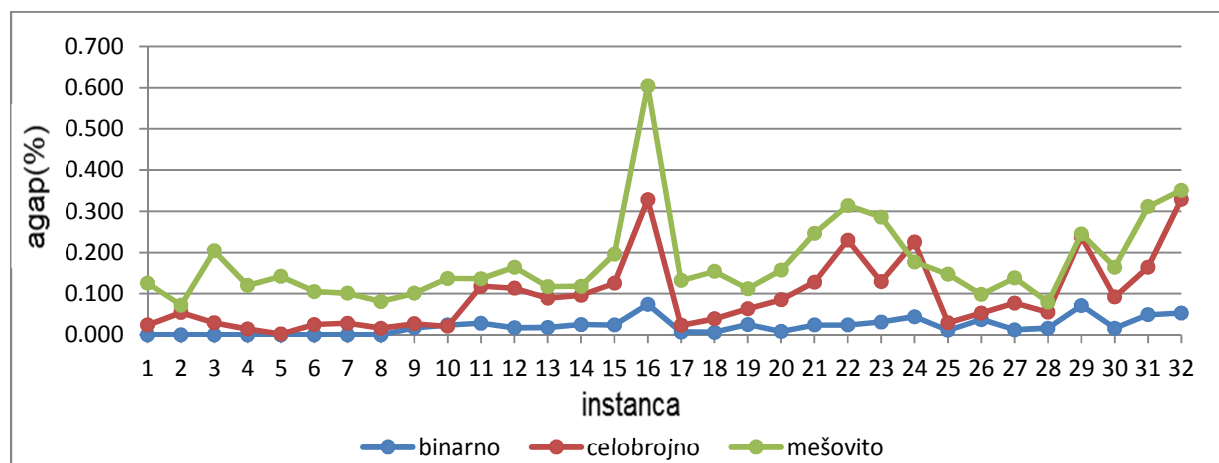


Grafik 6.12 Poređenje prosečnog odstupanja *agap*(%) za instance druge grupe – GA sa jednopozicionim ukrštanjem

Rešenja dobijena GA implementacijom sa dvopozicionim ukrštanjem u proseku najmanje odstupaju od optimalnih u slučaju binarnog kodiranja, dok su prosečna odstupanja u slučaju GA sa mešovito kodiranje značajnija (Grafici 6.13 i 6.14).

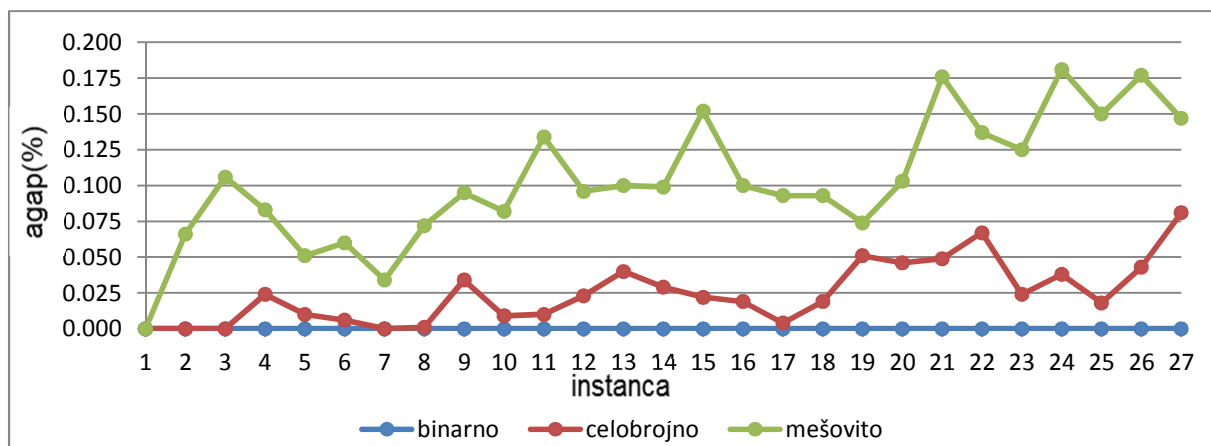


Grafik 6.13 Poređenje prosečnog odstupanja agap(%) za instance prve grupe – GA sa dvopozicionim ukrštanjem



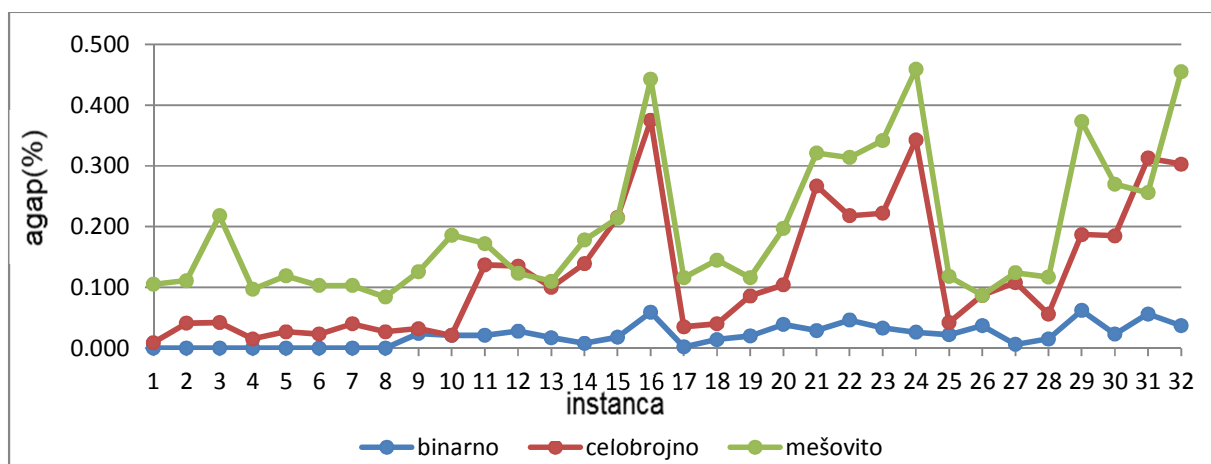
Grafik 6.14 Poređenje prosečnog odstupanja agap(%) za instance druge grupe – GA sa dvopozicionim ukrštanjem

Sa grafika 6.15 se mogu uočiti značajnije oscilacije prosečnog odstupanja rešenja od optimalnog čak i za instance manjih dimenzija. GA sa višepozicionim ukrštanjem i binarnim kodiranjem daje rešenja koja ne odstupaju od optimalnog rešenja, dok GA sa celobrojnim kodiranjem daje rešenja koja primetno odstupaju od optimalnih. GA sa mešovitim kodiranjem daje rešenja znatno lošijeg kvaliteta u odnosu na ostala dva načina kodiranja.



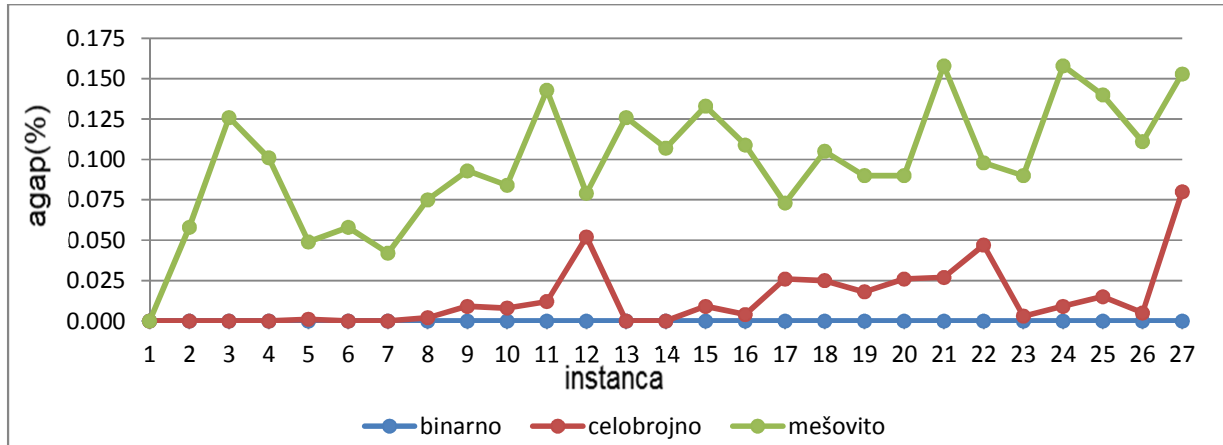
Grafik 6.15 Poređenje prosečnog odstupanja $agap(\%)$ za instance prve grupe – GA sa višepozicionim ukrštanjem

Sa Grafika 6.16 se može videti da GA sa višepozicionim ukrštanjem i binarnim kodiranjem ima malo prosečno odstupanje od najboljeg rešenja za instance većih dimenzija, dok preostale dve varijante GA imaju znatno veća prosečna odstupanja.



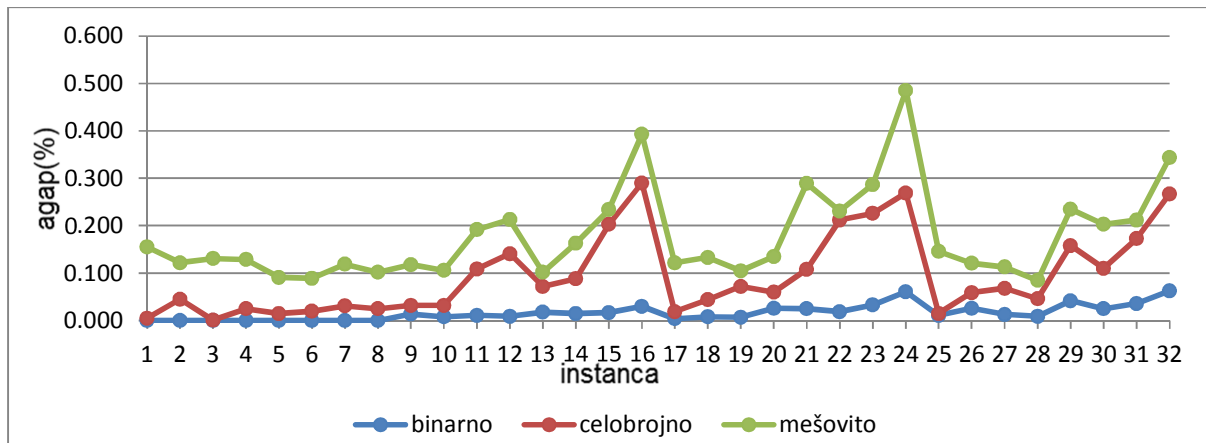
Grafik 6.16 Poređenje prosečnog odstupanja $agap(\%)$ za instance druge grupe – GA sa višepozicionim ukrštanjem

U slučaju GA sa uniformnim operatorom ukrštanja i binarnim kodiranjem prosečno odstupanje najboljeg rešenja od optimalnog je 0.000% za instance manjih dimenzija (Grafik 6.17). Za GA sa celobrojnim i mešovitim kodiranjem zapažanja su ista kao u prethodnom primeru.



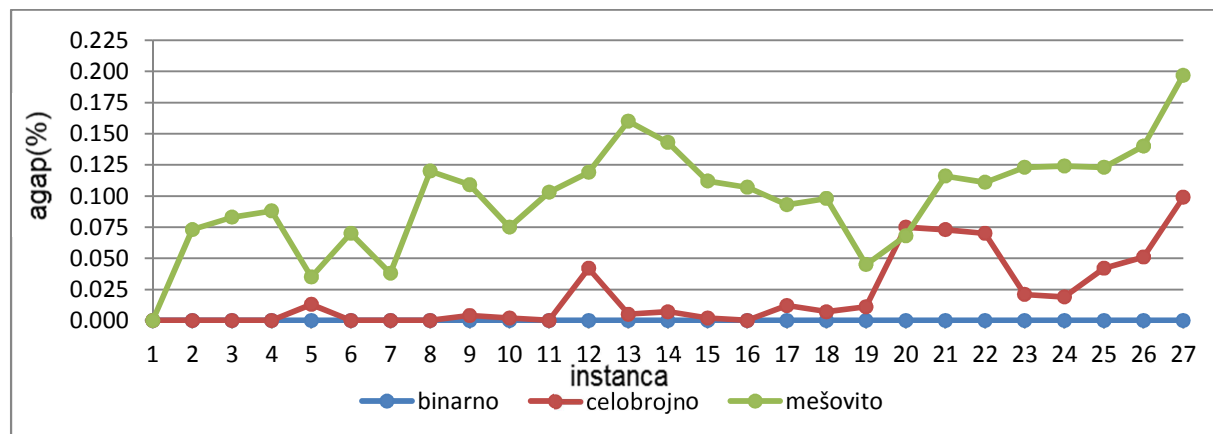
Grafik 6.17 Poređenje prosečnog odstupanja $agap(\%)$ za instance prve grupe – GA sa uniformnim ukrštanjem

Sa Grafika 6.18 se može videti da GA sa uniformnim operatorom ukrštanja, za instance većih dimenzija, najmanje prosečno odstupanje rešenja ima za binarno kodiranje, dok su za celobrojno i mešovito kodiranje zapažanja ista kao u prethodnom primeru.



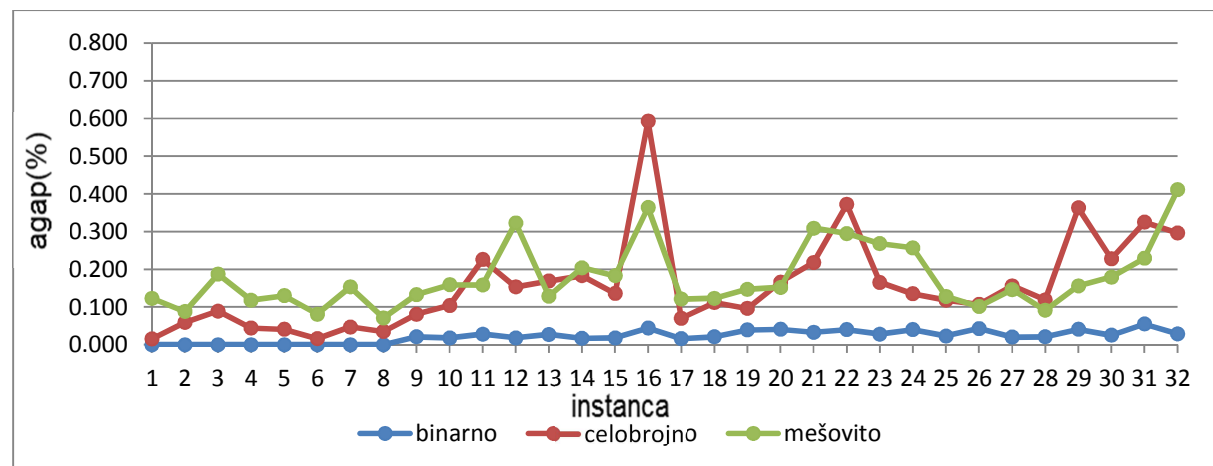
Grafik 6.18 Poređenje prosečnog odstupanja $agap(\%)$ za instance druge grupe – GA sa uniformnim ukrštanjem

U slučaju GA sa aritmetičkim operatorom ukrštanja i binarnim kodiranjem prosečno odstupanje najboljeg rešenja od optimalnog je 0.000% za instance manjih dimenzija (Grafik 6.19). Za GA sa celobrojnim kodiranjem odstupanja su neznatna, dok GA sa mešovitim ima veća prosečna odstupanja najboljeg GA rešenja od optimalnih.



Grafik 6.19 Poređenje prosečnog odstupanja $agap(\%)$ za instance prve grupe – GA sa aritmetičkim ukrštanjem

GA sa aritmetičkim operatorom ukrštanja i binarnim kodiranjem ima mala prosečna odstupanja za instance većih dimenzija (Grafik 6.20). U slučaju GA sa celobrojnim kodiranjem i aritmetičkim operatorom ukrštanja prosečna odstupanja rešenja od optimalnog su izraženija i za par instance veća nego u slučaju GA sa mešovitim kodiranjem, koji je do sada bio najlošiji u pogledu svih analiziranih aspekata.



Grafik 6.20 Poređenje prosečnog odstupanja $agap(\%)$ za instance druge grupe – GA sa aritmetičkim ukrštanjem

7. ZAKLJUČAK

U ovom radu je razmatran prost lokacijski problem i predložen je genetski algoritam kao metaheuristička metoda za njegovo rešavanje. U zavisnosti od načina kodiranja implementirane su tri varijante genetskog algoritma - GA sa binarnim kodiranjem, GA sa celobrojnim kodiranjem i GA sa kombinacijom prethodne dve vrste kodiranja (mešovito kodiranje). U svakoj od pomenutih varijanti GA implementirano je pet različitih operatora ukrštanja – jednopoziciono, dvopoziciono, višepoziciono, uniformno i aritmetičko ukrštanje. Analizirane su performanse predloženih varijanti GA sa aspekta prosečnog početnog vremena izvršavanja i prosečnog odstupanja najboljeg GA rešenja od optimalnog. Izvršena su poređenja vremena izvršavanja i kvaliteta rešenja za GA sa istim kodiranjem i različitim operatorom ukrštanja, kao i poređenja GA implementacija sa istim tipom operatora ukrštanja ali različitim kodiranjem.

Eksperimentalni rezultati još jednom potvrđuju da je izbor načina kodiranja jedan od najbitnijih aspekata genetskog algoritma. Pri rešavanju prostog lokacijski problem prirodno se nameće binarno kodiranje i ovim radom je još jednom potvrđeno da to i jeste najbolji izbor sa aspekta kvaliteta rešenja i vremena izvršavanja. Upoređujući performanse u odnosu na GA sa celobrojnim kodiranjem zaključuje se da na instancama manjih dimenzija algoritam sa celobrojnim kodiranjem daje rešenja u relativno kratkom vremenu, ali sa izvesnim (malim) prosečnim odstupanjima rešenja od optimalnih, dok je na instancama većih dimenzija GA sa binarnim kodiranjem bolji u svim aspektima. GA sa mešovitim kodiranjem je u slučaju samo nekoliko instanci dao dovoljno dobra rešenja ali kroz dosta dugo vreme izvršavanja, dok je na ostalim instancama eksperimentalno potvrđeno da je ovo loš izbor kodiranja sa aspekata kvaliteta rešenja i vremena izvršavanja.

U pogledu izbora operatora ukrštanja, razlike u performansama GA implementacija sa istim načinom kodiranja i različitim operatorima ukrštanja su znatno manje. Ipak, za GA sa binarnim načinom kodiranja utvrđeno je da najbolje performanse obezbeđuje uniformni operator ukrštanja u pogledu kvaliteta rešenja i brzine izvršavanja. Za GA sa celobrojnim kodiranjem, utvrđeno je da se najbolji rezultati u pogledu vremena i prosečnog odstupanja dobijaju kada se koristi dvopozicioni operator ukrštanja. Iako se mešovito kodiranje pokazalo kao loš izbor prilikom rešavanja prostog lokacijskog problema, GA sa mešovitim kodiranjem ima najkraće vreme izvršavanja kada koristi dvopoziciono ukrštanje, dok se najmanje prosečno odstupanje rešenja postiže sa uniformnim ukrštanjem.

Doprinos ovog rada ogleda se u činjenici da se prvi put analiziraju performanse genetskog algoritma za rešavanje prostog lokacijskog problema u odnosu na različite operatore ukrštanja i različite načine kodiranja. Dalja istraživanja i unapređenja rada moguća su poboljšanjem GA algoritma kroz dinamičko podešavanje parametara tokom izvršavanja, detaljnijom analizom ostalih operatora i aspekata GA, hibridizacijom sa egzaktnim ili drugim heurističkim metodama uz paralelno izvršavanje, itd.

LITERATURA

- [Ala08] **Alander, J.T.**, “An indexed bibliography of genetic algorithms“, *Technical Report DRAFT 2008/05/21*, Department of Electrical Engineering and Automation, University of Vaasa, Finland (2008).
- [Aqe90] **Aqeev A.A., Beresnev V.S.**, “Polynomially Solvable Cases of the Simple Plant Location Problem”, In: *Proceeding of the First Integer Programming and Combinatorial Optimization Conference*, University of Waterloo Press, Ont., Canada, pp. 1-6 (1990).
- [Bal65] **Balinski M.L.**, “Integer programming: methods, uses, computation”, *Management Science*, University of Pennsylvania and MATHEMATICA, pp. 253-313 (1965).
- [Bar97] **Barros A.I., Frenk J.B.G., Gromicho J.**, “Fractional Location Problems”, *Location Science*, Vol. 5, No.1, pp. 47-58 (1997).
- [Ban10] **Baron O., Milner J.**, “Facility Location: A Robust Optimization Approach”, *Production and Operations Management*, Rotman School of Management, University of Toronto, Canada (2010).
- [Bea96] **Beasley J.E.**, “Obtaining Test Problems via Internet“, *Journal of Global Optimization*, Vol. 8, pp. 429-433 (1996).
- [BD93a] **Beasley D., Bull D.R., Martin R.R.**, “An Overview of Genetic Algorithms, Part 1, Fundamentals“, *University Computing*, Vol. 15, No. 2, pp.58-69 (1993).
- [BD93b] **Beasley D., Bull D.R., Martin R.R.**, “An Overview of Genetic Algorithms, Part 2, Research Topics“, *University Computing*, Vol. 15, No. 4, pp. 170-181 (1993).
- [Bes93] **Beasley J.E.**, “Lagrangian Heuristics for Location Problems”, *European Journal of Operational Research*, pp. 383–399 (1993).
- [Bëc92] **Bëck T.**, “Self-Adaptation in Genetic Algorithms“, in: *Proceedings of the First European Conference on Artificial Life*, MIT Press (1992).
- [Bëc93] **Bëck T.**, “Optimal Mutation Rates in Genetic Search“, in: *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, pp. 2-8 (1993).
- [Cor90] **Cornuejols G., Nemhauser G.L., Wolsey L.A.**, “The Uncapacitated Facility Location Problem”, In: *Francis RL, Mirchandani PB (Eds.) Discrete Location Theory*, New York: Wiley-Interscience, pp. 119–171 (1990).

- [Crr06] **Correa F. A., Lorena L. A., N. Senne E. L. F.**, “Lagrangean Relaxation with Clusters for the Uncapacitated Facility Location Problem“, *Proceedings of the XIII Congreso Latino Iberoamericano de Investigacion Operativa - CLAIO*, Montevideo, Uruguay, (2006).
- [Dar59] **Darwin C.**, “The origin of species“, London (1859).
- [Dre06] **Drezner Z.**, “A general global optimization approach for solving location problems in the plane“, *Journal of Global Optimization*, Vol. 37, pp. 305-319 (2007).
- [Đur07] **Đurić B., Kratica J., Tošić D., Filipović V.**, “Solving the maximally balanced connected partition problem in graphs by using genetic algorithm“, *Computing and Informatics*, Vol. 27, pp. 341-354 (2007).
- [Erl78] **Erlenkotter D.**, “A dual-based procedure for uncapacitated facility location“, *Operations Research*, Vol. 26, pp. 992-1009 (1978).
- [Fil00] **Filipović V., Kratica J., Tošić D., Ljubić I.**, “Fine Grained Tournament Selection for Simple Plant Location Problem“, *Proceedings of th 5th Online Word Conference on Computing Methods in Industrial Applications – WSC5*, pp. 152-158, (2000).
- [Fil03] **Filipović, V.** “Fine-Grained Tournament Selection Operator in Genetic Algorithms“, *Computing and Informatics*, Vol. 22, pp. 143-161 (2003).
- [Fil06] **Filipović, V.**, „Operatori selekcije i migracije i WEB servisi kod paralelnih evolutivnih algoritama“, *Doktorska disertacija*, Matematički fakultet, Beograd (2006).
- [Grs94] **Grishukhin V.P.**, “On polynomial solvability conditions for the simplest plant location problem“, In: *Selected topics in discrete mathematics*, Kelmans A.K. and Ivanov S. (eds), American Mathematical Society, Providence, RI, pp. 37-46 (1994).
- [Han07] **Hansen P., Brimberg J., Urošević D., Mladenović N.**, “Primal-Dual Variable Neighborhood Search for the Simple Plant-Location Problem“, *INFORMS Journal on Computing* 19, pp. 552-564 (2007).
- [Hll75] **Holland J.H.**, “Adaptation in Natural and Artificial Systems“, *The University of Michigan Press*, Ann Arbor (1975).
- [Hlm95] **Holmberg K.**, “Experiments with primal-dual decomposition and subgradient methods for the uncapacitated facility location problem“, *Research Report LiTH-MAT/OPT-WP-1995-08*, Optimization Group, Department of Mathematics, Linköping Institute of Technology, Sweden (1995).
- [Jan91] **Janikow C.Z., Michalewicz Z.**, "An Experimental Comparasion of Binary and Floating Point Representations in Genetic Algorithms", In: *Proceedings of the Fourth International Conference on Genetic Algorithms - ICGA 1991*, Morgan Kaufmann, San Mateo, Calif., pp. 37-44 (1991).

- [Kol14] **Kole A., Chakrabarti P., Bhattacharyya S.**, “An Ant Colony Optimization Algorithm for Uncapacitated Facility Location Problem”, *Artificial intelligence and applications*, Vol.1, pp. 55-61, India (2014).
- [Kör89] **Körkel M.**, “On the Exact Solution of Large-Scale Simple Plant Location Problems”, *European Journal of Operational Research*, Vol. 39, pp. 157–173 (1989).
- [Krr83] **Krarp J., Pruzan P.M.**, “The simple plant location problem: Survey and synthesis“, *European Journal of Operational Research*, Vol. 12, pp. 36-81 (1983).
- [Kra99] **Kratica J.**, “Improvement of Simple Genetic Algorithm for Solving the Uncapacitated Warehouse Location Problem”, *Advance in Soft Computing – Engineering Design and Manufacturing*, R. Roy, T. Furuhashi and P.K. Chawdhry (Eds), Springer-Verlag London Limited, pp.390-402 (1999).
- [Kra00] **Kratica J.**, “Paralelizacija genetskih algoritama za rešavanje nekih NP – kompletnih problema”, *Doktorska disertacija, Matematički fakultet*, Beograd (2000).
- [Kra01] **Kratica J., Tošić D., Filipović V., Ljubić I.**, “Solving the Simple Plant Location Problem by Genetic Algorithm”, *RAIRO Operations Research*, Vol. 73, No. 1, pp. 127-142 (2001).
- [Kra02] **Kratica J., Tošić D., Filipović V., Ljubić I.**, “A Genetic Algorithm for the Uncapacitated Network Design Problem“, *Soft Computing in Industry - Recent Applications, Engineering series*, pp. 329-338. Springer (2002).
- [Kra06] **Kratica J., Stanimirović Z.**, “Solving the Uncapacitated Multiple Allocation p-Hub Center Problem by Genetic Algorithm“, *Asia-Pacific Journal of Operational Research*, Vol 23. No. 4, pp 425- 437 (2006).
- [Kra07] **Kratica J., Stanimirović Z., Tošić D., Filipović V.**, “Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem“, *European Journal of Operational Research* Vol. 182, No. 1, pp 15-28 (2007).
- [Kra08] **Kratica J., Kovačević-Vujčić V., Čangalović M.**, “Computing the Metric Dimension of Graphs by Genetic Algorithms“, *Yugoslav Journal of Operations Research*, Vol. 18, pp. 143-151 (2008).
- [Kue63] **Kuehn A.A., Hamburger M.J.**, “A heuristic program for location warehouses”, *Management Science* 9, pp. 643-666 (1963).
- [Law12] **Lawryniewicz A.**, “Genetic Algorithms for Solving Scheduling Problems in Manufacturing Systems”, *Foundations of Management, ISSN*, Vol. 3, pp. 7-26 (2012).
- [MM08a] **Marić M.**, „Rešavanje nekih NP–teških hijerarhijsko–lokacijskih problema primenom genetskih algoritama”, *Doktorska disertacija, Matematički fakultet*, Beograd (2008).

- [MM08b] **Marić M., Kratica J., Tuba M.**, “Parameter Adjustment for Genetic Algorithm for Two-Level Hierarchical Covering Location Problem“ *WSEAS Transactions*, Vol. 7, pp. 746 – 755 (2008).
- [Mar10] **Marić M.**, “An Efficient Genetic Algorithm For Solving The Multi-Level Uncapacitated Facility Location Problem“, *Computing and Informatics*, Vol. 29, pp. 183-201 (2010).
- [Mar15] **Marić M., Stanimirović Z., Božović S.**, “Hybrid metaheuristic method for determining locations for long-term health care facilities“, *Annals of Operations Research*, 227(1), pp. 3-23, DOI: 10.1007/s10479-013-1313-8 (2015).
- [Mit96] **Mitchell M.**, “An Introduction to Genetic Algorithms“, *MIT Press* (1996).
- [Müh97] **Mühlenbein H.**, “Genetic Algorithms“, *Local Search in Combinatorial Optimization*, eds. Aarts E.H.L., Lenstra J.K., John Wiley & Sons Ltd., pp. 137-172 (1997).
- [Owa15] **Owais S.S.J., Krómer P., Snášel V.**, “Query Optimization by Genetic Algorithms“, Department of Computer Science, Technical University of Ostrava, Czech Republic (2015).
- [Pen04] **Pendharkar C. P., Rodger A.J.**, “An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification“, *Computers&Operations Research* 31, pp. 481-498 (2004).
- [Rev05] **ReVelle C.S., Eiselt H.A.**, “Location analysis: A synthesis and survey“, *European Journal of Operational Research*, Vol. 165, pp. 1-19 (2005).
- [Sav08] **Savić A., Tošić D., Marić M., Kratica J.**, “An genetic algorithm approach for solving the task assignment problem“, *Serdica Journal of Computing*, pp. 267-276 (2008).
- [Sav12] **Savić A.**, “An genetic algorithm approach for solving the machine-job assignment with controllable processing times“, *Computing and Informatics*, Vol. 31, pp. 827-845 (2012).
- [Sim89] **Simao H.P., Thizy J.M.**, “A dual simplex algorithm for the canonical representation of the uncapacitated facility location problem“, *Operations Research Letters*, Vol. 8, No.5, pp. 279-286 (1989).
- [Sri94] **Srinivas M., Patnaik L.M.**, “Genetic Algorithms: A Survey“, *IEEE Computer*, pp. 17-26 (June 1994).
- [Sta04] **Stanimirović Z.**, „Rešavanje nekih diskretnih lokacijskih problema primenom genetskih algoritama“, *Magistarski rad*, Matematički fakultet, Beograd (2004).
- [SZ07a] **Stanimirović Z.**, „Genetski algoritmi za rešavanje nekih NP-teških hab lokacijskih problema“, *Doktorska disertacija*, Matematički fakultet, Beograd (2007).

- [SZ07b] **Stanimirović Z., Kratica J., Dugošija Đ.**, “Genetic algorithms for solving the discrete ordered median problem“, *European Journal of Operational Research*, Vol. 182, No. 3, pp. 983-1001 (2007).
- [Sta10] **Stanimirović Z.**, “A genetic algorithm approach for the capacitated single allocation p-hub median problem“, *Computing and Informatic*, Vol. 27, No.1, pp. 117-132 (2010).
- [Sta11] **Stanimirović Z., Kratica J., Filipović V., Tošić D.**, „Evolutivni pristup za rešavanje hab lokacijskih problema“, *Zavod za udžbenike i nastavna sredstva*, Beograd, ISBN: 978-86-17-17501 (2011).
- [Sta12] **Stanimirović Z., Marić M., Božović S., Stanojević P.**, “An Efficient Evolutionary Algorithm for Locating Long-Term Care Facilities”, *Information Technology and Control*, Vol.41, No.1, pp. 77 – 89 (2012).
- [Sta13] **Stanimirović Z., Mišković S.**, “Efficient Metaheuristic Approaches for Exploration of Online Social Networks“ (chapter) In: Wen-Chen Hu, Naima Kaabouch (Eds.); *Data Management, Technologies and Applications*, Chapter 10, pp. 222-269, ISBN: 1-466-64699-3, IGI Global (2013).
- [SZ14a] **Stanimirović Z., Marić M., Radojičić N., Božović S.**, “Two Efficient Hybrid Metaheuristic Methods for Solving the Load Balance Problem”, *Applied and Computational Mathematics*, Vol.13, No. 3, pp. 332-349 (2014).
- [SZ14b] **Stanimirović Z., Grujičić I., Trifunović D.**, “A hybrid evolutionary algorithm for efficient exploration of online social networks“, *Computing and Informatics*, Vol 33, No.2, pp. 410-430 (2014).
- [Sta15] **Stanimirović Z., Grujičić I., Trifunović D.**, “Modeling the emergency service network of police special forces units for high-risk law enforcement operations”, *Information System and Operational Research – INFOR*, to appear (2015).
- [Stn15] **Stanojević P., Marić M., Stanimirović Z.**, “A hybridization of an evolutionary algorithm and a parallel branch and bound for solving the capacitated single allocation hub location problem“, *Applied Soft Computing*, DOI information:10.1016/j.asoc.2015.04.018 (2015).
- [Owa15] **Owais S.S.J., Krómer P., Snášel V.**, “Query Optimization by Genetic Algorithms”, Department of Computer Science, Technical University of Ostrava, Czech Republic (2015).
- [Sun06] **Sun M.**, “Solving the uncapacitated facility location problem using tabu search“, *Computers and Operations Research*, Vol. 33, No. 9, pp. 2563-2589 (2006).
- [Suj11] **Sujay S.D.K, Kole A.**, “A modified continuous particle swarm optimization algorithm for uncapacitated facility problem”, *AIM 2011*, pp. 305-311, Springer-Verlag Berlin Heidelberg (2011).

[Tch88] **Tcha D.W., Ro H.B., Yoo C.B.**, “Dual-based Add Heuristic for Uncapacitated Facility Location”, *Journal of Operational Research Society*, Vol 39, No. 9, pp. 873-878 (1988).

[Top05] **Topcuoglu H., Corut F., Ermis M., Yilmaz G.**, “Solving the Uncapacitated Hub Location Problem Using Genetic Algorithms”, *Computers & Operations Research*, Vol. 32, pp.967-984 (2005).

[Val15] **Vallejo J.F.C., Ortiz J.M., Ramos F.L., Rodriguez R.P.**, “A Genetic Algorithm for the Bi-Level Topological Design of Local Area Networks ”, *PLoS ONE* 10(6) (2015).

[Vyg05] **Vygen J.**, “Approximation Algorithm for Facility Location Problems”, Research Institute for Discrete Mathematics, University of Born, Germany (2005).

[Yig03] **Yigit V., Turkbey O.**, “An approach to the facility location problems with hill-climbing and simulated annealing“, *Journal of The Faculty of Engineering and Architecture of Gazi University*, Vol. 18, No. 4, pp. 45-56 (2003).

SADRŽAJ

PREDGOVOR.....	5
1. PROST LOKACIJSKI PROBLEM.....	7
1.1 Matematička formulacija problema	7
1.2 Pregled metoda iz literature za rešavanje SPLP-a.....	8
2. GENETSKI ALGORITAM.....	9
2.1 Opis GA.....	9
2.2 Prost GA i njegove karakteristike	11
2.3 Moderne tehnike prilagođavanja GA problemu.....	11
2.3.1 Kodiranje i funkcija prilagođenosti	11
2.3.2 Selekcija.....	12
2.3.3 Ukrštanje.....	13
2.3.4 Mutacija.....	15
2.3.5 Kriterijum zaustavljanja.....	16
2.3.6 Zamena generacije.....	16
2.3.7 Parametri genetskog algoritma.....	16
2.4 Primena GA u praksi.....	17
3. GA IMPLEMENTACIJA KOJA KORISTI BINARNO KODIRANJE.....	18
3.1 Reprezentacija rešenja.....	18
3.2 Računanje funkcije cilja	18
3.3 Genetski operatori.....	18
3.3.1 Selekcija.....	18
3.3.2 Ukrštanje.....	19
3.3.3 Mutacija.....	19
3.4 Ostali aspekti GA.....	19
3.5 Eksperimentalni rezultati.....	19
3.5.1 Ulazni podaci	19
3.5.2 Rezultati testiranja.....	20
4. GA IMPLEMENTACIJA KOJA KORISTI CELOBROJNO KODIRANJE.....	33
4.1 Reprezentacija rešenja.....	33
4.2 Računanje funkcije cilja	33
4.3 Genetski operatori.....	33

4.3.1	Selekcija.....	33
4.3.2	Ukrštanje.....	33
4.3.3	Mutacija.....	34
4.4	Ostali aspekti GA.....	34
4.5	Eksperimentalni rezultati.....	34
5.	GA IMPLEMENTACIJA KOJA KORISTI MEŠOVITO KODIRANJE.....	48
5.1	Reprezentacija rešenja.....	48
5.2	Računanje funkcije cilja	48
5.3	Genetski operatori.....	48
5.3.1	Selekcija.....	48
5.3.2	Ukrštanje.....	48
5.3.3	Mutacija.....	49
5.4	Ostali aspekti GA.....	49
5.5	Eksperimentalni rezultati.....	49
6.	POREĐENJE REZULTATA	63
6.1	Prosečno početno vreme izvršavanja algoritma.....	63
6.2	Prosečno odstupanje najboljeg GA rešenja od optimalnog	68
7.	ZAKLJUČAK	73