

Sveučilište u Zagrebu
Fakultet organizacije i informatike

Ivan Ković

***Poboljšanje djelotvornosti genetskog algoritma za optimizaciju rasporeda sati
odabirom genetskih operatora***

Zagreb, 2014.

Ovaj rad izrađen je na Fakultetu organizacije i informatike pod vodstvom dr.sc. Dijane Oreški i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2013./2014.

Sadržaj

1. Uvod	1
2. Teorijski okvir	2
2.1 Problem vremenskog raspoređivanja	2
2.2 Standardni genetski algoritam	4
2.2.1 Struktura genetskog algoritma.....	4
2.2.2 Funkcija dobrote	6
2.2.3 Genetski operatori	6
2.3 Rješenje problema vremenskog raspoređivanja pomoću genetskog algoritma.....	8
3. Opći i specifični ciljevi rada.....	9
4. Materijal i metode.....	12
4.1 Tehnološka platforma istraživanja.....	12
4.2 Kodiranje kromosoma	12
4.3 Pregled promatranih genetskih operatora.....	14
4.3.1 Uniformna mutacija.....	14
4.3.2 Mutacija s Gaussovom distribucijom vrijednosti	15
4.3.3 Mutacija zamjenom vrijednosti	16
4.3.4 Križanje s jednom točkom prekida	17
4.3.5 Modificirano križanje s jednom točkom prekida	18
4.3.6 Križanje s dvije točke prekida	18
4.3.7 Selekcija n najboljih kromosoma.....	19
4.3.8 Rulet selekcija	19
4.3.9 Turnirska selekcija	20
4.4 Metode istraživanja.....	20
5. Rezultati	22
5.1 Prosječna vrijednost funkcije dobrote najboljeg kromosoma	24
5.2 Prosječna vrijednost funkcije dobrote za pojedini operator	25
5.3 Prosječno vrijeme trajanja izvođenja algoritma	27
6. Rasprava	28
7. Zaključci	32
8. Zahvale	33
9. Popis literature	34
Sažetak	37
Summary.....	38

1. Uvod

Stvaranje rasporeda sati je problem praktične naravi s kojim se obrazovne institucije susreću svake godine. Uzevši u obzir brojna ograničenja koja se moraju zadovoljiti tokom formiranja jednog takvog rasporeda, lako je uvidjeti zahtjevnost rješavanja tog problema. Budući da je problem vremenskog raspoređivanja okarakteriziran kao NP-težak problem (Piola, 1992), algoritamsko rješavanje tog problema ograničeno je na upotrebu raznih heurističkih metoda, među kojima se ističe primjena genetskih algoritama. Ovaj rad bavi se poboljšanjem djelotvornosti takvih algoritama.

Poboljšanjem djelotvornosti nekog algoritma želi se omogućiti njegova provedba uz što manju potrošnju resursa, i to najčešće vremena. Jedan od mogućih načina poboljšanja djelotvornosti genetskih algoritama je odabir odgovarajućih genetskih operatora za neki problem (Fogel, 1999).

Cilj ovog rada je pronaći takvu kombinaciju genetskih operatora koja omogućava bolju djelotvornost genetskog algoritma za optimizaciju rasporeda sati u odnosu na ostale kombinacije genetskih operatora. Pronalazak takve kombinacije operatora omogućava provedbu daljnjih istraživanja kojima bi se uz zadanu kombinaciju genetskih operatora istraživalo daljnje povećanje djelotvornosti kroz promjenu ostalih varijabilnih dijelova genetskog algoritma. Na taj način potpuno bi se definirao oblik genetskog algoritma koji rješava problem optimizacije rasporeda sati uz najveću djelotvornost.

U narednom poglavlju definiran je teorijski okvir istraživanja koji sadrži opis problema vremenskog raspoređivanja, prikaz jednostavnog genetskog algoritma te pregled dosadašnjih rješenja problema optimizacije rasporeda sati pomoću genetskog algoritma. Potom se navode konkretna istraživačka pitanja čiji se odgovori traže u ovom radu. U poglavlju „Materijal i metode“ opisani su svi genetski operatori uključeni u istraživanje, te je opisan postupak mjerenja djelotvornosti algoritma. Nakon toga prikazani su dobiveni rezultati, čija se interpretacija nalazi u poglavlju „Rasprava“. Na kraju se daju odgovori na postavljena istraživačka pitanja te se postavljaju pitanja za buduća istraživanja.

2. Teorijski okvir

U ovom poglavlju ustanovljen je teorijski okvir provedenog istraživanja, te su definirani pojmovi bitni za razumijevanje provedenog istraživanja. Kontekst u kojem se istraživanje provodi je genetski algoritam primijenjen na problem vremenskog istraživanja, tako da se u ovom poglavlju govori o zahtjevnosti problema vremenskog raspoređivanja, temeljnim obilježjima standardnog genetskog algoritma te se na kraju pokazuje na koji način se genetski algoritam može prilagoditi za rješavanje problema vremenskog raspoređivanja.

2.1 Problem vremenskog raspoređivanja

Problem vremenskog raspoređivanja jedan je od najčešćih predstavnika skupine *optimizacijskih problema*. Za optimizacijske probleme karakteristično je postojanje skupa rješenja problema, unutar kojeg se za svaki element skupa može provjeriti kvaliteta rješenja. Kvaliteta rješenja najčešće se otkriva kao izračun troška ili cijene nekog rješenja. Zbog toga optimizaciju definiramo kao postupak pronalaženja najboljeg rješenja zadanog optimizacijskog problema. Boyd i Vanderberghe (2004) optimizacijski problem opisuju na sljedeći način:

minimiziraj $f_0(x)$ podložan:

$$g_i(x) \leq 0, \quad i = 1, \dots, m$$

$$h_i(x) = 0, \quad i = 1, \dots, p$$

Navedena notacija koristi se za opisivanje problema pronalaženja varijable x koja minimizira $f_0(x)$ između svih vrijednosti x koje zadovoljavaju uvjete $g_i(x)$ i $h_i(x)$. Varijablu x nazivamo *optimizacijskom varijablom*, funkciju $f_0(x)$ *funkcijom cilja*, a funkcije $g_i(x)$ i $h_i(x)$ *funkcijama ograničenja*. Prema tome, rješenje problema traži se minimiziranjem funkcije $f_0(x)$.

Skup mogućih rješenja zadanog problema sastoji se od svih instanci x koje udovoljavaju funkcijama ograničenja. Svaki element tog skupa može biti rješenje optimizacijskog problema, te se sami postupak optimizacije manifestira odabirom instance koja postiže najpoželjniju vrijednost funkcije cilja (minimalnu ili maksimalnu, ovisno o postavljenom problemu).

Optimizacijski problemi međusobno se razlikuju po navedenim funkcijama, tako da među njima možemo prepoznati više karakterističnih skupina problema poput *problema trgovačkog putnika* ili *problema zadovoljenja ograničenja* (engl. *Constraint Satisfaction Problem*).

Problem vremenskog raspoređivanja se može promatrati kao problem zadovoljenja ograničenja, koji se sastoji od konačnog niza varijabli od kojih je svaka povezana s odgovarajućom domenom, te niza ograničenja koje je potrebno zadovoljiti (Blanco i Kathib, 1998). Pritom varijable preuzimaju vrijednosti iz povezanih domena tako da zadovolje sva navedena ograničenja. Navedene vrijednosti su resursi u obliku osoba koje izvode nastavu, grupa studenata koje slušaju nastavu te dvorana u kojima se provodi nastava. Ograničenja specifična za problem rasporeda sati su najčešće neka od sljedećih (Burke et al., 1994):

- određeni profesor u jednom terminu može biti samo u jednoj dvorani
- određena grupa studenata u jednom terminu može biti samo u jednoj dvorani
- u određenoj dvorani u jednom terminu može se izvoditi nastava iz samo jednog kolegija
- za svaki kolegij definiran je broj sati koji se moraju odraditi u jednom tjednu

Navedena ograničenja je nužno zadovoljiti kako bi neki raspored sati bio upotrebljiv, odnosno formalno važeći. Uključivanjem neobveznih ograničenja poput preferencije jutarnjih termina nastave komplicira problem, budući da takva ograničenja nije obavezno zadovoljiti, ali utječu na kvalitetu potencijalnog rješenja problema.

Kao i većina drugih optimizacijskih problema, problem vremenskog raspoređivanja okarakteriziran je kao **NP-težak** problem (Piola, 1992), odnosno ne postoji ili nije poznat algoritam koji bi pronašao najbolje rješenje za navedeni problem u polinomskom odnosno prihvatljivom vremenu.

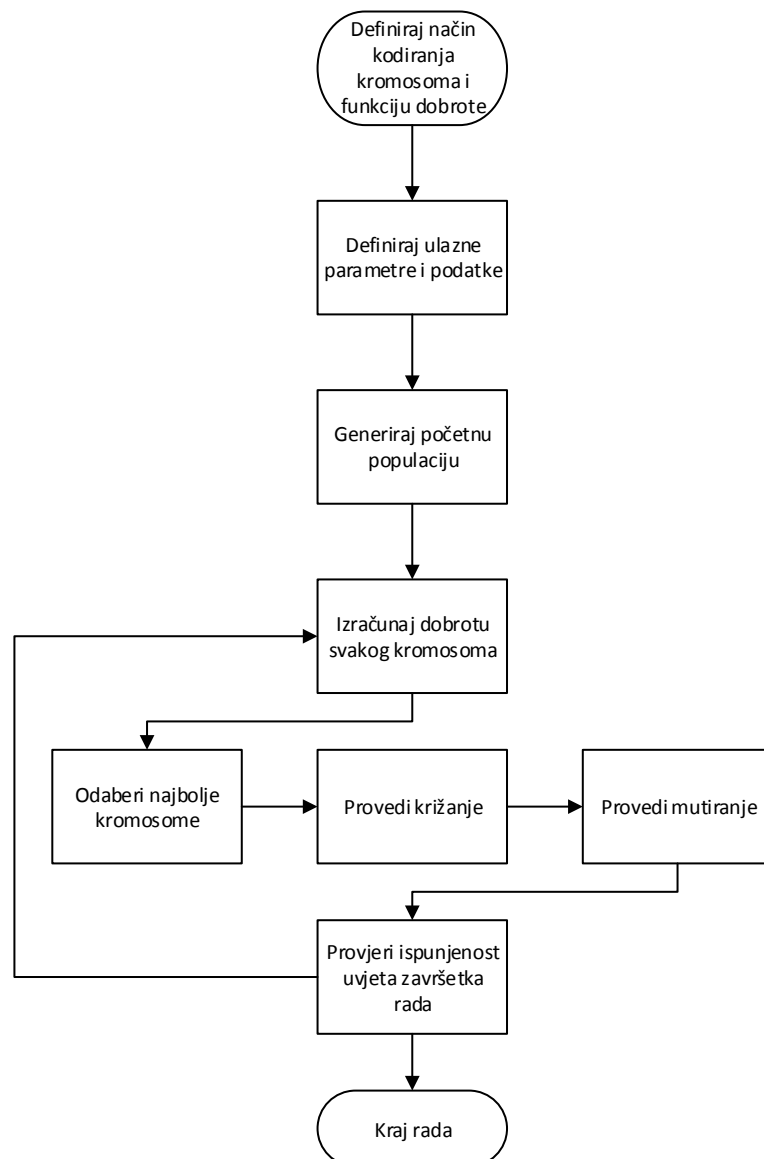
Nužan uvjet da bi neki problem bio NP-težak je mogućnost provjere ispravnosti nekog od mogućih rješenja u polinomskom vremenu. Zbog toga se problem vremenskog raspoređivanja može riješiti nekom od metoda pretraživanja prostora rješenja, a jedna od tih metoda su *genetski algoritmi* (Burke et al., 1995).

2.2 Standardni genetski algoritam

Beatty (1991) definira genetski algoritam kao heurističku metodu 'slučajnog i usmjerenog pretraživanja prostora rješenja koja imitira prirodni evolucijski proces'. Takav algoritam nad skupom potencijalnih rješenja zadanog problema ciklički provodi genetske operatore selekcije, križanja i mutacije dok se ne zadovolje sva zadana ograničenja, ili dok se ne ispuni neki drugi uvjet završetka algoritma. U nastavku su opisane navedene komponente genetskog algoritma te tijekom njegovog izvođenja.

2.2.1 Struktura genetskog algoritma

Struktura genetskog algoritma kakvu je definirao John Holland (1992) može se jednostavno prikazati na sljedeći način:



Slika 1: Koraci standardnog genetskog algoritma (Holland, 1992)

Prvi korak algoritma sastoji se od pripremnih aktivnosti koje ovise o optimizacijskom problemu koji se rješava pomoću genetskog algoritma, a konkretno se odnosi na odabir načina kodiranja kromosoma. **Kromosom** je osnovna jedinica nad kojom genetski algoritam obavlja operacije, te predstavlja potencijalno rješenje problema koji se rješava. U računalnoj implementaciji kromosom je najčešće neki oblik apstraktnog tipa podataka namijenjenog držanju gena. **Gen** je sastavna jedinica kromosoma koja nosi jedan dio rješenja problema, te određuje svojstva kromosoma.

Genetski algoritam u svom radu provodi razne operacije nad genetskim materijalom unutar kromosoma kako bi došao do rješenja, pa se svaki kromosom može nazvati i potencijalnim rješenjem problema (Blickle i Thiele, 1991).

Reprezentacija rješenja problema pomoću kromosoma ovisi o konkretnom problemu koji se rješava. Kumar (2013) donosi pregled danas najkorištenijih načina kodiranja kromosoma:

- kodiranje temeljeno na redoslijedu gena: *permutacijsko kodiranje*
- kodiranje temeljeno na redoslijedu i vrijednosti gena: *binarno kodiranje*
- kodiranje temeljeno na vrijednosti gena: *vrijednosno kodiranje*

U istom radu Kumar naglašava da je odabir kodiranja ključan za rad genetskog algoritma budući da o kodiranju ovisi i djelotvornost algoritma i odabir operatora, budući da pojedine operatore nije moguće provesti u slučaju korištenja određenog kodiranja.

Skup kromosoma ili **jedinki** čini populaciju. **Populacija** je skup svih mogućih rješenja problema, te je za nju karakteristično da se mijenja sa svakom iteracijom algoritmom. Ukoliko početnu populaciju označimo oznakom $P(0)$, nakon n provedenih iteracija možemo reći da promatramo populaciju $P(0 + n)$, odnosno n -tu **generaciju**.

Početna populacija se može generirati na različite načine:

- sve početne jedinke su jednake (uniformna populacija)
- slučajno generiranje jedinki
- početni skup rješenja dobiven nekim drugim optimizacijskim postupkom

Uniformna populacija je najjednostavnije rješenje, ali nosi i mnoge probleme: ukoliko u takvoj populaciji nije prisutno neko od mogućih svojstava, pitanje je koliko će se

puta provesti operator mutiranja prije negoli se pojavi traženo svojstvo. Time se bitno smanjuje djelotvornost algoritma, budući da je potrebno provesti veći broj evolucija prije nego se dođe do rješenja u odnosu na druge slučajeve.

Slučajno generiranje jedinki je najčešće korišten pristup, budući da postoji mogućnost da će se već u početnoj populaciji pronaći rješenje koje djelomično ili potpuno zadovoljava navedene uvjete.

Početni skup rješenja dobiven nekim drugim optimizacijskim postupkom olakšava rad algoritma, te je za očekivati da se rješenje problema takvim pristupom dobije s manjim brojem generacija u odnosu na prethodna dva pristupa (Corcoran, 1992).

2.2.2 Funkcija dobrote

Kako bi mogli ocijeniti u kolikoj mjeri neka jedinka zadovoljava unaprijed postavljene uvjete ili ograničenja, potrebno je definirati *funkciju dobrote* (engl. *fitness function*) koja vrednuje kvalitetu svake pojedine jedinke. Ta nam je informacija potrebna kako bi kasnije mogli usporediti kromosome po njihovoj dobroti, na temelju čega se biraju najbolji kromosomi, odnosno oni koji sadrže poželjna svojstva.

Ne postoji standardna implementacija funkcije dobrote, već je ona specifična za svaki problem (pa i za različita rješenja istog problema). Osnovni koraci koji se poduzimaju unutar funkcije dobrote su dekodiranje kromosoma te dodjeljivanje određene vrijednosti dobrote na temelju količine ispunjenosti zadanih uvjeta. Proces se provodi za svaku jedinku iz pojedine populacije.

2.2.3 Genetski operatori

Genetski operatori se mogu podijeliti u tri skupine s obzirom na njihovu namjenu:

- operatori selekcije
- operatori križanja
- operatori mutacije

Operator **selekcije** jednostavno rečeno služi za prenošenje dobrih svojstava iz jedne generacije u drugu, te eliminiranje loših svojstava. Svrha selekcije je odabir jedinki nad kojima će se provesti ostali genetski operatori, najčešće prema vrijednosti funkcije cilja povezanoj s pojedinom jedinkom.

Mitchell (1997) upozorava na problem *preuranjene konvergencije* koja se događa ukoliko grupa jedinki pridobije značajno veću vrijednost funkcije dobrote u odnosu na ostale jedinke u populaciji. Zbog toga će ta grupa dominirati populacijom i nakon određenog broja populacija potpuno će istisnuti jedinke s manjom vrijednosti funkcije dobrote. Time se značajno sužava prostor traženja, odnosno gube se dobra svojstva koja posjeduju loše jedinke. Zbog toga operatori selekcije rade probabilistički, odnosno boljim jedinkama dodjeljuju veću vjerojatnost da će biti odabrane, dok lošijim jedinkama dodjeljuju manju vjerojatnost odabiranja. Na taj način osigurava se da se u sljedećoj generaciji ne izgube i dobra svojstva prisutna u lošim jedinkama.

Operator **križanja** odnosi se na razmjenu genetskog materijala između dva kromosoma, te je analogan biološkom procesu razmnožavanja. Križanje se provodi nad skupom kromosoma odabranom pomoću operatora selekcije, i to tako da se prema zadanoj vjerojatnosti odabere određeni broj kromosoma koji će u parovima razmjenjivati genetski materijal i stvoriti određeni broj novih kromosoma koji će biti pridodani novoj generaciji jedinki.

Operator **mutacije** djeluje nad jednom jedinkom, te ovisno o izvedbi mijenja vrijednost jednog ili više gena unutar odabranog kromosoma. Može ga se konfigurirati definiranjem vjerojatnosti da će neki kromosom biti odabran za mutaciju.

Brojni autori radova na području genetskih algoritama naglašavaju važnost ravnoteže dvaju procesa: *istraživanje* i *iskorištavanje*. Beyer (1998) definira istraživanje kao 'sposobnost algoritma da se kreće u smjeru željenog poboljšanja'. Proces iskorištavanja odnosi se na poboljšavanje kvalitete rješenja kroz fokusiranje na najbolje jedinke. Istraživanje prostora traženja potiče raznovrsnost populacije, te sprječava upadanje u lokalne minimume ili maksimume. S druge strane, proces iskorištavanja smanjuje raznovrsnost ali fokusiranjem na najbolja svojstva poboljšava kvalitetu rješenja.

Herrera i Lozano (2000) smatraju kako je odnos između istraživanja i iskorištavanja 'ključan za određivanje ponašanja algoritma'. Važnost genetskih operatora očituje se upravo u činjenici da su oni odgovorni za procese istraživanja i iskorištavanja, a pritom se općenito smatra da je iskorištavanje u genetskim algoritmima odgovoran operator selekcije, a za istraživanje primarno operator mutacije, te u manjoj mjeri i operator križanja (Eiben i Schippers, 1998).

2.3 Rješenje problema vremenskog raspoređivanja pomoću genetskog algoritma

Više autora bavilo se rješavanjem problema vremenskog raspoređivanja pomoću genetskih algoritama, počevši od Burkea i ostalih (1995) do Lukasa i ostalih (2009). Navedeni znanstveni radovi problemu pristupaju na različit način, te sukladno tome predlažu i različite načine poboljšanja djelotvornosti postojećih algoritama. Neovisno o konkretnoj implementaciji, navedenim radovima zajednički je postupak prilagođavanja genetskog algoritma problemu vremenskog raspoređivanja. Kako bi se genetski algoritam prilagodio rješavanju nekog praktičnog problema, potrebno je provesti sljedeće korake:

- odabrati odgovarajući način kodiranja kromosoma,
- postaviti funkciju dobrote,
- odabrati genetske operatore,
- odrediti uvjete završetka rada algoritma.

Kodiranje kromosoma moguće je za navedeni problem obaviti na različite načine, te je bitno da se tokom kodiranja slijede sljedeći principi (Goldberg, 1989):

- Princip značajnih građevnih blokova,
- Princip minimalnog skupa znakova.

Prvi princip nalaže da se kromosom kodira na takav način da je udaljenost u pozicijama gena koji kontroliraju zajedničko obilježje što manja, dok drugi princip nalaže da se za vrijednost pojedinog gena uvijek koristi najmanji mogući skup znakova.

Funkcija dobrote mora obuhvatiti sva ograničenja koja će se morati zadovoljiti kako bi neko rješenje bilo važeće. Potrebno je implementirati provjere koje će dodjeljivati vrijednost dobrote pojedinom kromosomu prema tome koliko konflikata pri korištenju resursa postoji u danom rješenju.

Najčešći uvjet završetka rada je dosezanje zadane maksimalne vrijednosti funkcije dobrote, ili dosezanje zadanog broja evolucija.

3. Opći i specifični ciljevi rada

U prethodnom dijelu rada pokazano je zašto je problem vremenskog raspoređivanja posebno složen za rješavanje, te na koji način se mogu iskoristiti genetski algoritmi za njegovo rješenje. Pitanje kojim se ovaj istraživački rad bavi je *na koji način poboljšati djelatnost genetskog algoritma*.

Goldberg (2002) definira četiri opsežne kategorije načina poboljšavanja djelatnosti genetskih algoritama na sljedeći način:

- **paralelizacija** – poboljšavanje djelatnosti vidljivo je kroz skraćivanje vremenskog trajanja provođenja algoritma, koje se postiže povećanjem procesorske snage paralelnom obradom istog problema od strane većeg broja računala, odnosno procesora,
- **hibridizacija** – 'genetski algoritmi su moćan alat za pretragu cijelog prostora rješenja, ali su slabi u lokalnom traženju, te se prema tome djelatnost genetskog algoritma može povećati dodavanjem operatora za lokalno traženje koji se provodi nad svakim kromosom u generaciji nakon selekcije,
- **vremenski kontinuitet** – mogućnosti genetskih operatora mogu se iskoristiti na takav način da omogućavaju pronalaženje najboljeg mogućeg rješenja uz zadane računalne resurse,
- **opuštanje evaluacije** – funkcija vrednovanja dobrote najčešće je vremenski najsloženija komponenta genetskog algoritma. Povećanje djelatnosti algoritma može se postići odabirom manje precizne funkcije vrednovanja dobrote koje pokazuje manju vremensku složenost u odnosu na precizniju funkciju vrednovanja dobrote.

Proces poboljšavanja djelatnosti genetskih algoritama nije jednostavan. Budući da genetski algoritmi nisu determinističke prirode, oni ne moraju za isti ulaz dati uvijek isto konačno rješenje, čime je otežano samo mjerenje i uspoređivanje djelatnosti promjena nad algoritmom. Zbog toga se **djelatnost** može iskazati kao *prosječna vrijednost funkcije dobrote najboljeg kromosoma nakon određenog broja evolucija unutar više provedbi nekog genetskog algoritma* (Xu, 1999).

Formalno djelotvornost zapisujemo na sljedeći način:

$$d = \frac{\sum_{i=1}^n x}{n}$$

Varijabla d označava djelotvornost nekog genetskog algoritma, koja se računa kao suma vrijednosti funkcije dobrote (predstavljena varijablom x) najboljeg kromosoma nakon određenog broja evolucija podijeljena brojem provedbi algoritma, označenih kao n .

Povećanje djelotvornosti iskazane na navedeni način percipiramo kao smanjenje prosječnog broja evolucija potrebnih za dobivanje željene vrijednosti funkcije dobrote kod nekog genetskog algoritma. Drugim riječima, djelotvorniji genetski algoritam s većom vjerojatnošću ranije postiže željenu vrijednost funkcije dobrote u odnosu na manje djelotvorni genetski algoritam primijenjen na isti problem.

U ovom istraživanju zanemaruje se *vremenska složenost* genetskog algoritma, odnosno istražuje se na koji način postići što povoljniju vrijednost funkcije dobrote nekog genetskog algoritma kroz što manji broj evolucija, neovisno o samom trajanju izvođenja programa na računalu. Goldberg-ova kategorizacija dopušta oba pristupa povećanju djelotvornosti genetskih algoritama. Vremenska složenost algoritma utječe na brzinu provođenja jedne iteracije, dok se smanjenjem broja evolucija zapravo smanjuje broj iteracija unutar jednog provođenja algoritma.

Pregledom dosadašnjih istraživanja može se ustanoviti da su se mnogi autori bavili poboljšavanjem djelotvornosti genetskih algoritama promjenom parametara poput veličine populacije ili vjerojatnosti mutacije ili križanja, dok se jedan dio autora fokusirao na poboljšavanje djelotvornosti algoritma kroz odgovarajući odabir genetskih operatora.

Glavna pretpostavka na kojoj se temelji ovaj rad dolazi od Fogela (1999), prema kome ne postoji takva kombinacija genetskih operatora koja bi davala superiorne rezultate na nizu proizvoljnih optimizacijskih problema u odnosu na genetske algoritme s nekim drugim operatorima, odnosno ne postoji univerzalno najbolja kombinacija genetskih operatora. Zbog toga je opravdano istraživati koje su optimalne kombinacije operatora u kontekstu specifičnog problema koji se rješava pomoću genetskog algoritma.

Razali i Geraghty (2011) su istraživali djelotvornost genetskog algoritma s različitim operatorima selekcije pri rješavanju problema trgovačkog putnika, te su došli do zaključka da turnirska selekcija omogućava bolje performanse genetskog algoritma pri rješavanju problema trgovačkog putnika u odnosu na proporcionalnu i rangiranu rulet selekciju.

Na sličan su način Puljić i Manger (2013) usporedili utjecaj 8 različitih operatora križanja na djelotvornost genetskog algoritma prilikom rješavanja problema trgovačkog putnika i problema usmjeravanja vozila. Zaključili su da 'eksperimentalni rezultati jasno pokazuju da se ponašanje i učinkovitost operatora pri rješavanju problema usmjeravanja vozila razlikuju u odnosu na primjenu istih operatora na rješavanje problema trgovačkog putnika'.

Sukladni tome su i rezultati već spomenutog Xu-ovog istraživanja (1999), koji je na nizu različitih problema usporedio djelotvornost genetskih algoritama konfiguriranih s različitim genetskim operatorima, te je također ustanovio da ne postoji univerzalno najbolja kombinacija operatora.

Glavni cilj ovog istraživanja je dati odgovor na sljedeće istraživačko pitanje:

Postoji li takva kombinacija genetskih operatora koja omogućuje bolju djelotvornost genetskog algoritma za optimizaciju rasporeda sati u odnosu na genetski algoritam s nekom drugom kombinacijom operatora?

Ukoliko se odgovor na navedeno pitanje pokaže potvrdnim, postavlja se i pomoćno pitanje:

Koja kombinacija genetskih operatora omogućuje najbolju djelotvornost genetskog algoritma za optimizaciju rasporeda sati?

4. Materijal i metode

U ovom poglavlju dan je cjelovit opis načina provedbe istraživanja. U nastavku je dan pregled tehnološke platforme, odnosno razvojne okoline unutar koje je istraživanje provedeno. Posebno se ističe odabrani način kodiranja kromosoma, budući da je presudan za odabir genetskih operatora. Slijedi popis svih genetskih operatora uključenih u istraživanje zajedno s kratkim opisom operatora. Nakon toga navodi se metodologija upotrijebljena za mjerenje te se definiraju pokazatelji čije su vrijednosti mjerene. Na kraju je priložen skup testnih podataka nad kojima je provedeno istraživanje.

4.1 Tehnološka platforma istraživanja

Za potrebe provedbe istraživanja korištene su sljedeće tehnologije:

- Programski jezik **Java 7**,
- Integrirano razvojno okruženje **NetBeans 7.3.1**,
- Razvojni okvir **JGAP**.

JGAP (engl. *Java Genetic Algorithms Package*) je besplatan paket biblioteka i alata koji olakšavaju rad s genetičkim algoritmima te omogućavaju genetičko programiranje. JGAP korisniku omogućuje korištenje ugrađenih algoritama uz mogućnost konfiguriranja i proširivanja ugrađenih funkcionalnosti (Meffert, 2009).

4.2 Kodiranje kromosoma

Odabir načina kodiranja kromosoma temeljio se na sljedećim pretpostavkama:

- Nastava se izvodi od ponedjeljka do petka, dakle 5 dana tjedno,
- U jednom danu postoji 12 termina s trajanjem od sat vremena,
- Poznat je broj dvorana raspoloživih za nastavu,
- Poznat je broj kolegija zajedno s vezanim grupama studenata i profesora.

Neki fakultet ima dostupno n dvorana za održavanje nastave. Prema tome, u jednom danu moguće je održati 12 predavanja u svakoj dvorani. Dalje, u jednom tjednu nastava se izvodi od ponedjeljka do petka, tako da je u jednom tjednu ukupno moguće izvesti $5 \times 12 \times n$ predavanja.

Prema tome, kromosom definiram kao niz cijelih brojeva ukupne duljine $5 \times 12 \times n$. Svaki broj unutar tog niza odgovara brojčanom identifikatoru kolegija koji je unaprijed definiran. Kromosom se iščitava na sljedeći način: prvih 12 brojki predstavljaju identifikatore kolegija koji se izvode ponedjeljkom u prvoj dostupnoj dvorani. Ispod teksta prikazana je komponenta kromosoma koja predstavlja popis kolegija koji se izvode u određenoj dvorani na određeni dan.

Tabela 1: Kromosomski prikaz zauzetosti dvorane

kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID	kol_ID
Dvorana n											

Daljnjih $(n-1) \times 12$ brojki predstavljaju identifikatore kolegija koji se izvode ponedjeljkom u ostalim dostupnim dvoranama. Takav niz se ponavlja za svaki dan u tjednu zaključno s petkom, tako da se na kraju kromosom sastoji od $12 \times \text{broj_dvorana} \times 5$ vrijednosti, odnosno $\text{broj_dvorana} \times 5$ komponenti poput gornje komponente. Struktura kromosoma vizualno je predočena donjom tablicom.

Tabela 2: Prikaz kodiranja kromosoma za čitavi raspored sati

Ponedjeljak			...	Petak		
Dvorana 1	...	Dvorana n	...	Dvorana 1	...	Dvorana n

Prednost ovakvog kodiranja kromosoma očituje se u tome što se u samom početku ispunjava jedan čvrsti uvjet: nemoguće je zadati više od jednog predavanja u jednoj dvorani u pojedinom terminu. Time se eliminira potreba za odgovarajućom provjerom unutar algoritma, čime se skraćuje prosječno vrijeme izvođenja algoritma.

Nedostatak kodiranja predstavlja veza između prosječnog vremena izvođenja algoritma i broja dvorana – svaka dodatna dvorana povećava ukupnu duljinu kromosoma za 60 elemenata, što povećava vremensku složenost algoritma.

Osim toga, ovakav odabir kodiranja isključuje upotrebu određenih genetskih operatora, u prvom redu onih namijenjenih radu s bit-kodiranim kromosomima. U nastavku slijedi pregled svih operatora uključenih u istraživanje.

4.3 Pregled promatranih genetskih operatora

U istraživanje je bilo uključeno 9 različitih genetskih operatora – 3 operatora mutacije, 3 operatora križanja te 3 operatora selekcije. Pretpostavka je korištenje po jednog operatora iz svake kategorije. Prema tome, ukupni broj mogućih kombinacija navedenih operatora iznosi **27**. Slijedi opis pojedinih operatora.

4.3.1 Uniformna mutacija

Djelovanje operatora uniformne mutacije može se opisati sljedećim nizom koraka:

1. Preuzmi sve kromosome iz trenutne generacije,
2. Slučajnim odabirom odaberi kromosome koji će mutirati,
3. Za svaki odabrani kromosom mutiraj svaki gen prema **uniformnoj** distribuciji vrijednosti,
4. Dodaj mutirane kromosome među kandidate za sljedeću generaciju.

U prvom koraku operator jednostavno preuzima sve kromosome iz trenutne generacije $G(n)$. Nakon toga operator bira određen broj kromosoma nad kojima će se provesti mutacija, te formira skup M , za koji vrijedi $M \subseteq G$. Ukupan broj kromosoma koji su odabrani za mutaciju ovisi o parametru koji se naziva *vjerojatnost mutacije*, te ga označavamo s p_m . Vjerojatnost da će kromosom biti odabran za mutaciju jednaka je za sve kromosome, neovisno o njihovoj kvaliteti, odnosno vrijednosti funkcije dobrote.

U trećem koraku operator iz skupa M za svaki kromosom mutira gene po sljedećoj formuli:

$$x'_i = \begin{cases} x_i + x_i r, & dg \leq x_i + x_i r \leq gg \\ gg, & x_i + x_i r > gg \\ dg, & x_i + x_i r < dg \end{cases}$$

$$r_{uniform} = rand(\{-1, 1\})$$

Varijabla x_i predstavlja vrijednost i -tog gena u nekom kromosomu iz skupa M . Varijabla r predstavlja slučajan broj generiran prema *uniformnoj* razdiobi kojim se množi vrijednost x_i . Konačna vrijednost mutiranog gena x'_i izračunava se kao zbroj vrijednosti x_i s umnoškom $x_i r$. Pritom x'_i mora biti unutar prethodno definiranog skupa dopuštenih vrijednosti $A[dg, gg]$. Varijable dg i gg predstavljaju gornju i donju granicu skupa dopuštenih vrijednosti gena.

4.3.2 Mutacija s Gaussovom distribucijom vrijednosti

Operator mutacije Gaussovom razdiobom funkcionira na jedan način kao i prethodno opisani operator uniformne mutacije, uz razliku u distribuciji slučajne varijable, koja je u ovom slučaju normalna, odnosno Gaussova:

1. Preuzmi sve kromosome iz trenutne generacije
2. Slučajnim odabirom odaberi kromosome koji će mutirati
3. Za svaki odabrani kromosom mutiraj svaki gen prema **normalnoj** distribuciji vrijednosti
4. Dodaj mutirane kromosome među kandidate za sljedeću generaciju

U prvom koraku operator jednostavno preuzima sve kromosome iz trenutne generacije $G(n)$. Nakon toga operator bira određen broj kromosoma nad kojima će se provesti mutacija, te formira skup M , za koji vrijedi $M \subseteq G$. Ukupan broj kromosoma koji su odabrani za mutaciju ovisi o parametru koji se naziva *vjerojatnost mutacije*, te ga označavamo s p_m . Vjerojatnost da će kromosom biti odabran za mutaciju jednaka je za sve kromosome, neovisno o njihovoj kvaliteti, odnosno vrijednosti funkcije dobrote.

U trećem koraku operator iz skupa M za svaki kromosom mutira gene po sljedećoj formuli:

$$x'_i = \begin{cases} x_i + x_i r, & dg \leq x_i + x_i r \leq gg \\ gg, & x_i + x_i r > gg \\ dg, & x_i + x_i r < dg \end{cases}$$

$$r_{gauss} = rand(\{-1, 1\})$$

Varijabla x_i predstavlja vrijednost i -tog gena u nekom kromosomu iz skupa M . Varijabla r predstavlja slučajan broj generiran prema *normalnoj* razdiobi kojim se množi vrijednost x_i . Konačna vrijednost mutiranog gena x'_i izračunava se kao zbroj vrijednosti x_i s umnoškom $x_i r$. Pritom x'_i mora biti unutar prethodno definiranog skupa dopuštenih vrijednosti $A[dg, gg]$. Varijable dg i gg predstavljaju gornju i donju granicu skupa dopuštenih vrijednosti gena.

4.3.3 Mutacija zamjenom vrijednosti

Mutacija zamjenom vrijednosti često se primjenjuje kod rješavanja problema koji se kodiraju permutacijski, poput problema trgovačkog putnika (Grefenstette et al, 1985). U 2. poglavlju naveo sam da je za permutacijsko kodiranje specifično da se niti jedna vrijednost ne smije pojaviti na više od jednog mjesta unutar kromosoma. Zbog toga se mutacija kod permutacijskih problema ne može izvoditi na klasičan način (generiranjem slučajnih vrijednosti), odnosno mutacijom se može dobiti neispravan kromosom koji ne zadovoljava pravilo jedinstvenih vrijednosti.

Budući da je za rješenje problema rasporeda sati pomoću genetskog algoritma korišteno kodiranje u kojem je osim vrijednosti gena bitan i njihov redoslijed, mutacija zamjenom vrijednosti također se može primijeniti i na problem optimizacije rasporeda sati. Operator se provodi na sljedeći način:

1. Preuzmi sve kromosome iz trenutne generacije
2. Slučajnim odabirom odaberi kromosome koji će mutirati
3. Za svaki odabrani kromosom odaberi par gena i zamijeni im mjesta
4. Dodaj mutirane kromosome među kandidate za sljedeću generaciju

U prvom koraku operator jednostavno preuzima sve kromosome iz trenutne generacije $G(n)$. Nakon toga operator bira određen broj kromosoma nad kojima će se provesti mutacija, te formira skup M , za koji vrijedi $M \subseteq G$. Ukupan broj kromosoma koji su odabrani za mutaciju ovisi o parametru koji se najčešće naziva *vjerojatnost mutacije*, te ga označavamo s p_m . Vjerojatnost da će kromosom biti odabran za mutaciju jednaka je za sve kromosome, neovisno o njihovoj kvaliteti, odnosno vrijednosti funkcije dobrote.

U trećem koraku operator izračunava ukupnu duljinu kromosoma, te generira dva slučajna broja p i q u rasponu od $[1, \text{duljina kromosoma})$. Potom unutar kromosoma pronalazi gene x_p i x_q , te im međusobno zamijeni vrijednosti tako da x_p poprima vrijednost x_q , a x_q poprima početnu vrijednost od x_p .

Navedeni korak ponavlja se za svaki kromosom iz skupa M , te se na kraju navedeni skup pridružuje ostalim kandidatima za postupak selekcije.

4.3.4 Križanje s jednom točkom prekida

Za razliku od dosad prikazanih unarnih operatora mutacije, operatori križanja su binarni, odnosno za križanje su uvijek potrebna dva roditelja. Križanje s jednom točkom prekida je jedan od operatora koje je John Holland definirao 1975. godine, te se provodi na sljedeći način:

1. Preuzmi sve kromosome iz trenutne generacije
2. Odaberi n parova kromosoma za križanje
3. Za svaki odabrani par kromosoma slučajno odaberi točku presjeka i stvori dva nova kromosoma kao kombinaciju genetskog materijala roditelja
4. Dodaj kromosome djecu među kandidate za sljedeću generaciju

U prvom koraku operator jednostavno preuzima sve kromosome iz trenutne generacije $G(n)$. Nakon toga operator bira određen broj parova kromosoma za križanje, te formira skup K , za koji vrijedi $K \subseteq G$. Ukupan broj parova kromosoma koji su odabrani ovisi o parametru koji se naziva *vjerojatnost križanja*, te ga označavamo s p_k . Vjerojatnost da će kromosom biti odabran za mutaciju jednaka je za sve kromosome, neovisno o njihovoj kvaliteti, odnosno vrijednosti funkcije dobrote.

Križanje se provodi tako da se slučajno odabere broj r iz raspona $[1, duljina\ kromosoma)$, te se na temelju toga odredi točka prekida kromosoma roditelja. Potom se kromosomi roditelja prekidaju na točki prekida, te se rekombiniraju tako da se prvom roditelju na točku prekida nadodaje preostali dio drugog roditelja, dok se drugom roditelju na točku prekida nadodaje preostali dio prvog roditelja. Postupak je pregledno prikazan sljedećom ilustracijom:

Tabela 3: Kromosomi roditelji kod križanja s jednom točkom prekida

Roditelj 1		
Roditelj 2		

Križanjem gornjih roditelja nastaju sljedeća djeca:

Tabela 4: Kromosomi djeca kod križanja s jednom točkom prekida

Dijete 1		
Dijete 2		

4.3.5 Modificirano križanje s jednom točkom prekida

Modificirani operator križanja s jednom točkom prekida radi na isti način kao i prethodno opisani operator križanja s jednom točkom prekida, uz razliku u koraku u kojem se generira točka prekida – u ovom slučaju točke prekida za svaki par kromosoma generiraju se samo tokom prve provedbe operatora, te se spremaju kako bi se ponovo upotrijebile tokom sljedećih iteracija.

4.3.6 Križanje s dvije točke prekida

Križanje s dvije točke prekida također se provodi na jednak način kao i križanje s jednom točkom prekida, samo što se nakon definiranja prve točke prekida definira još jedna točka nakon koje prestaje razmjena genetskog materijala između dva roditelja.

Križanje se provodi tako da se generiraju dva slučajna broja p i q u rasponu od $[1, duljina\ kromosoma)$. Broj p predstavlja indeks gena u kromosomu na kojem će se napraviti prva točka prekida, a broj q indeks gena na kojem će se napraviti druga točka prekida.

Prvi kromosom djeteta prima prvih p gena od prvog roditelja, sljedećih $q - p$ gena od drugog roditelja, te posljednjih $duljina\ kromosoma - q$ gena opet od prvog roditelja. Postupak za drugo dijete je identičan, uz zamjenu uloga prvog i drugog roditelja.

Postupak je pregledno prikazan sljedećom ilustracijom:

Tabela 5: Kromosomi roditelji kod križanja s dvije točke prekida

Roditelj 1			
Roditelj 2			

Križanjem gornjih roditelja nastaju sljedeća djeca:

Tabela 6: Kromosomi djeca kod križanja s dvije točke prekida

Dijete 1			
Dijete 2			

4.3.7 Selekcija n najboljih kromosoma

Operator selekcije najboljih n kromosoma najjednostavniji je oblik selekcije. Radi na sljedeći način:

1. Preuzmi listu kandidata za sljedeću generaciju
2. Rangiraj listu prema vrijednosti funkcije dobrote svakog kandidata
3. Odaberi n kromosoma za sljedeću generaciju
4. Preostale kromosome odbaci

Ovaj operator može se konfigurirati parametrom n kojim se određuje broj kromosoma koji prelaze u sljedeću generaciju. Ukoliko se odredi da je broj jedinki u generaciji fiksna, onda je n jednak postavljenom broju jedinki u generaciji.

4.3.8 Rulet selekcija

Rulet selekcija (engl. *Roulette Wheel Selection*) je način selekcije kromosoma u kojem je vjerojatnost da će kromosom prijeći u sljedeću generaciju proporcionalna njegovoj vrijednosti funkcije dobrote. Selekcija se provodi na sljedeći način:

1. Preuzmi listu kandidata za sljedeću generaciju
2. Konstruiraj kotač za rulet s određenim brojem slotova
3. Dodijeli slotove kromosomima tako da kromosomi s većom vrijednosti funkcije dobrote dobiju veći broj slotova, a oni s manjom vrijednosti manji broj slotova
4. Zavrti kotač onoliko puta koliko je potrebno da se dobije n kromosoma
5. Odbaci preostale kromosome

Ovaj operator može se konfigurirati parametrom n kojim se određuje broj kromosoma koji prelaze u sljedeću generaciju. Ukoliko se odredi da je broj jedinki u generaciji fiksna, onda je n jednak postavljenom broju jedinki u generaciji.

Kotač se mora zavrtjeti bar onoliko puta koliko iznosi n , budući da se najbolji kromosomi nalaze na više slotova, te ih je zbog toga moguće više puta odabrati, pa je često potrebno više od n rotacija.

Ovakav pristup omogućava da se u sljedećoj generaciji pojave i slabiji kromosomi, ali u isto vrijeme i dopušta mogućnost da se izostave neki od najboljih kromosoma, odnosno ne garantira selekciju najboljeg kromosoma.

4.3.9 Turnirska selekcija

Turnirska selekcija radi na drugačijem principu od prethodno opisanih vrsta selekcije – umjesto da selekcija neke jedinke izravno ovisi o njenoj vrijednosti funkcije dobrote, selekcija ovisi o relativnoj poziciji u odnosu na neku drugu jedinku, ukoliko bi jedinke usporedili po vrijednosti funkcije dobrote (Bäck, 1995). Dakle, jedinke se uspoređuju po rangu umjesto po vrijednosti funkcije dobrote.

Tijek turnirske selekcije je sljedeći:

1. Preuzmi listu kandidata za sljedeću generaciju
2. Slučajno odaberi n kromosoma
3. Iz skupa odabranih kromosoma odaberi najbolji kromosom za sljedeću generaciju
4. Ponavljaj od 2. koraka dok se ne dosegne željena veličina populacije
5. Odbaci preostale kromosome

Operator se može konfigurirati parametrom n kojim se određuje broj kromosoma koji sudjeluju u jednom turniru. Broj turnira koji se moraju odigrati da bi se dosegla željena veličina populacije jednak je traženoj veličine populacije, budući da se u sljedeću generaciju šalje samo pobjednik pojedinog turnira.

4.4 Metode istraživanja

Provedba istraživanja osnovana je na već spomenutom Xu-ovom radu (1999), koji se bavio usporedbom djelotvornosti genetskih operatora uz različite zadane probleme. U istraživanju su promatrana sljedeća obilježja:

- **broj evolucija,**
- **vrijednost funkcije dobrote najboljeg kromosoma pri određenoj generaciji,**
- **vrijeme izvršavanja algoritma na računalu.**

Broj evolucija je potrebno evidentirati, budući da je svrha istraživanja pronaći kombinaciju genetskih operatora koja omogućuje postizanje što veće vrijednosti funkcije dobrote unutar što manjeg broja evolucija.

Iz istog razloga potrebno je i pratiti *vrijednost dobrote najboljeg kromosoma pri određenog generaciji*.

Vrijeme izvršavanja algoritma na računalu ne utječe na rezultat istraživanja, ali praćenje tog pokazatelja omogućuje stjecanje dojma o odnosu vremenskih složenosti pojedinih kombinacija genetskih operatora.

Računalni program koji je kreiran za provedbu istraživanja radi sljedeći niz koraka:

1. Preuzmi testne podatke,
2. Konfiguriraj genetski algoritam za izvođenje s određenom kombinacijom genetskih operatora,
3. Provedi algoritam zadani broj puta, i za svaku provedbu evidentiraj promatrana obilježja,
4. Ponovi od 2. koraka za svaku kombinaciju genetskih operatora.

Djelotvornost algoritma definirana je u 3. poglavlju kao prosječna vrijednost funkcije dobrote najboljeg kromosoma pri određenoj generaciji, nakon više provođenja algoritma. Vodeći se Xu-ovim istraživanjem (1999), promatra se prosječna vrijednost funkcije dobrote najboljeg kromosoma pri **3000**-toj generaciji, i to nakon što se algoritam izvrši **20** puta.

Testni skup podataka nad kojima ćemo izvršiti algoritam sastoji se od elemenata prikazanih u donjoj tablici:

Tabela 7: Popis testnih podataka

Broj profesora	7
Broj dvorana	3
Broj grupa studenata	3
Broj različitih kolegija	15

Problem optimizacije rasporeda sati zadan s gornjim skupom podataka ne smatra se posebno složenim, te je očekivana prosječna vrijednost funkcije dobrote najboljeg kromosoma pri 3000-toj generaciji veća od 800.

5. Rezultati

Zbog velikog broja genetskih operatora i njihovih mogućih kombinacija, za potrebe prikaza rezultata definirane su kratice prikazane u donjoj tablici.

Tabela 8: Objašnjenje kratica korištenih u prikazima rezultata

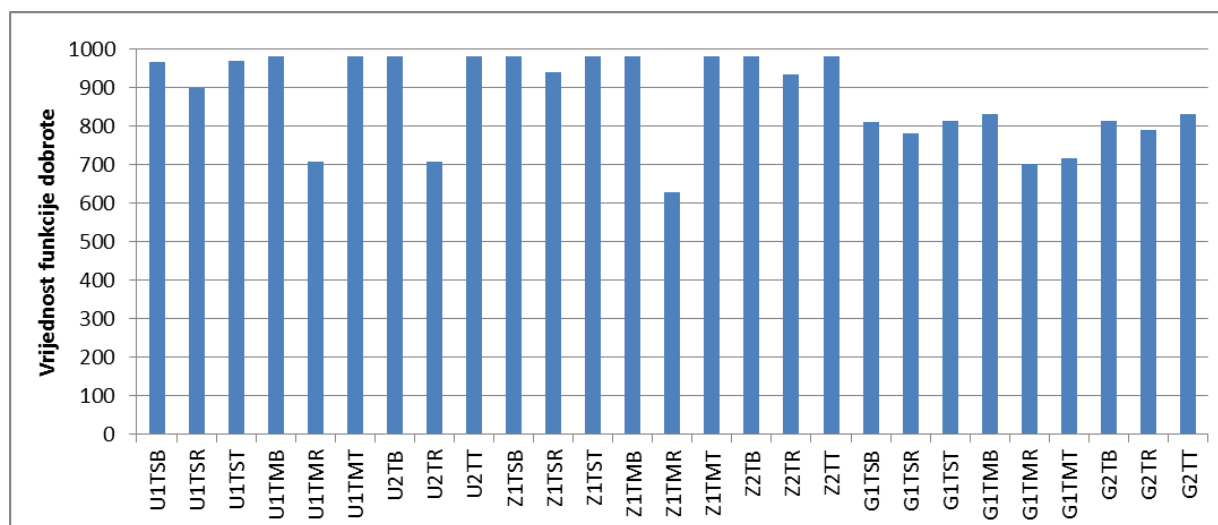
Operator mutacije	Operator križanja	Operator selekcije	Kratice
Uniformna mutacija	Križanje s jednom točkom prekida	Selekcija n najboljih kromosoma	U1TSB
Uniformna mutacija	Križanje s jednom točkom prekida	Rulet selekcija	U1TSR
Uniformna mutacija	Križanje s jednom točkom prekida	Turnirska selekcija	U1TST
Uniformna mutacija	Modificirano križanje s jednom točkom prekida	Selekcija n najboljih kromosoma	U1TMB
Uniformna mutacija	Modificirano križanje s jednom točkom prekida	Rulet selekcija	U1TMR
Uniformna mutacija	Modificirano križanje s jednom točkom prekida	Turnirska selekcija	U1TMT
Uniformna mutacija	Križanje s dvije točke prekida	Selekcija n najboljih kromosoma	U2TB
Uniformna mutacija	Križanje s dvije točke prekida	Rulet selekcija	U2TR
Uniformna mutacija	Križanje s dvije točke prekida	Turnirska selekcija	U2TT
Mutacija zamjenom vrijednosti	Križanje s jednom točkom prekida	Selekcija n najboljih kromosoma	Z1TSB
Mutacija zamjenom vrijednosti	Križanje s jednom točkom prekida	Rulet selekcija	Z1TSR
Mutacija zamjenom vrijednosti	Križanje s jednom točkom prekida	Turnirska selekcija	Z1TST
Mutacija zamjenom vrijednosti	Modificirano križanje s jednom točkom prekida	Selekcija n najboljih kromosoma	Z1TMB

Mutacija zamjenom vrijednosti	Modificirano križanje s jednom točkom prekida	Rulet selekcija	Z1TMR
Mutacija zamjenom vrijednosti	Modificirano križanje s jednom točkom prekida	Turnirska selekcija	Z1TMT
Mutacija zamjenom vrijednosti	Križanje s dvije točke prekida	Selekcija n najboljih kromosoma	Z2TB
Mutacija zamjenom vrijednosti	Križanje s dvije točke prekida	Rulet selekcija	Z2TR
Mutacija zamjenom vrijednosti	Križanje s dvije točke prekida	Turnirska selekcija	Z2TT
Gaussova mutacija	Križanje s jednom točkom prekida	Selekcija n najboljih kromosoma	G1TSB
Gaussova mutacija	Križanje s jednom točkom prekida	Rulet selekcija	G1TSR
Gaussova mutacija	Križanje s jednom točkom prekida	Turnirska selekcija	G1TST
Gaussova mutacija	Modificirano križanje s jednom točkom prekida	Selekcija n najboljih kromosoma	G1TMB
Gaussova mutacija	Modificirano križanje s jednom točkom prekida	Rulet selekcija	G1TMR
Gaussova mutacija	Modificirano križanje s jednom točkom prekida	Turnirska selekcija	G1TMT
Gaussova mutacija	Križanje s dvije točke prekida	Selekcija n najboljih kromosoma	G2TB
Gaussova mutacija	Križanje s dvije točke prekida	Rulet selekcija	G2TR
Gaussova mutacija	Križanje s dvije točke prekida	Turnirska selekcija	G2TT

Gore navedene kratice koriste se u jednakom obliku u svim narednim prikazima rezultata istraživanja.

5.1 Prosječna vrijednost funkcije dobrote najboljeg kromosoma

U donjem dijagramu prikazane su prosječne vrijednosti funkcije dobrote najboljeg kromosoma u generaciji nakon 3000 evolucija, odnosno njihova djelotvornost.



Slika 2: Usporedba djelotvornosti različitih kombinacija genetskih operatora

Najveća izmjerena prosječna vrijednost funkcije dobrote najboljeg kromosoma u generaciji nakon 3000 evolucija iznosi **982**, i izmjerena je kod sljedećih kombinacija genetskih operatora: U1TMB, U1TMT, U2TB, U2TT, Z1TSB, Z1TST, Z1TMB, Z2TB i Z2TT. Najniža izmjerena vrijednost iznosi **626,86**, i to kod kombinacije Z1TMR. Aritmetička sredina svih izmjerenih vrijednosti iznosi **875,99**, te je kod 15 kombinacija operatora zabilježena veća vrijednost od navedene prosječne vrijednosti, dok je kod ostalih 12 kombinacija vrijednost manja od prosječne. U donjoj tablici prikazane su sve izmjerene vrijednosti, zajedno s pripadajućom mjerom standardne devijacije:

Tabela 9: Izmjerene prosječne vrijednosti funkcije dobrote i standardna devijacija

Naziv	Djelotvornost	σ	Naziv	Djelotvornost	σ
U1TSB	966,79	5,59	Z1TMT	981,79	0,12
U1TSR	899,79	9,73	Z2TB	982,00	0,00
U1TST	967,64	5,98	Z2TR	932,33	11,97
U1TMB	982,00	0,00	Z2TT	982,00	0,00
U1TMR	706,92	23,62	G1TSB	809,40	14,31
U1TMT	982,00	0,00	G1TSR	782,30	12,98
U2TB	982,00	0,00	G1TST	813,70	14,31
U2TR	707,73	19,05	G1TMB	829,70	16,92
U2TT	982,00	0,00	G1TMR	700,82	9,64
Z1TSB	982,00	0,00	G1TMT	715,06	15,16
Z1TSR	939,00	11,38	G2TB	812,33	15,37
Z1TST	982,00	0,00	G2TR	789,60	17,74
Z1TMB	982,00	0,00	G2TT	831,94	11,22
Z1TMR	626,86	49,70			

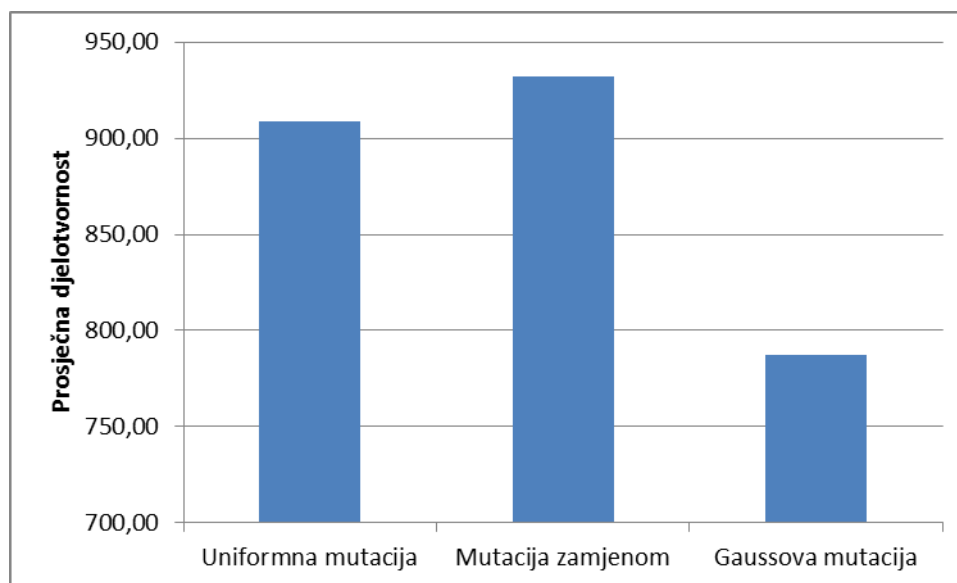
5.2 Prosječna vrijednost funkcije dobrote za pojedini operator

Na temelju dobivenih podataka izračunata je i prosječna djelotvornost za svako pojavljivanje pojedinog operatora, odnosno prosječna djelotvornost za sve kombinacije u kojima sudjeluje navedeni operator. Dobiveni podaci su prikazani u tablici:

Tabela 10: Prosječna djelotvornost genetskih operatora

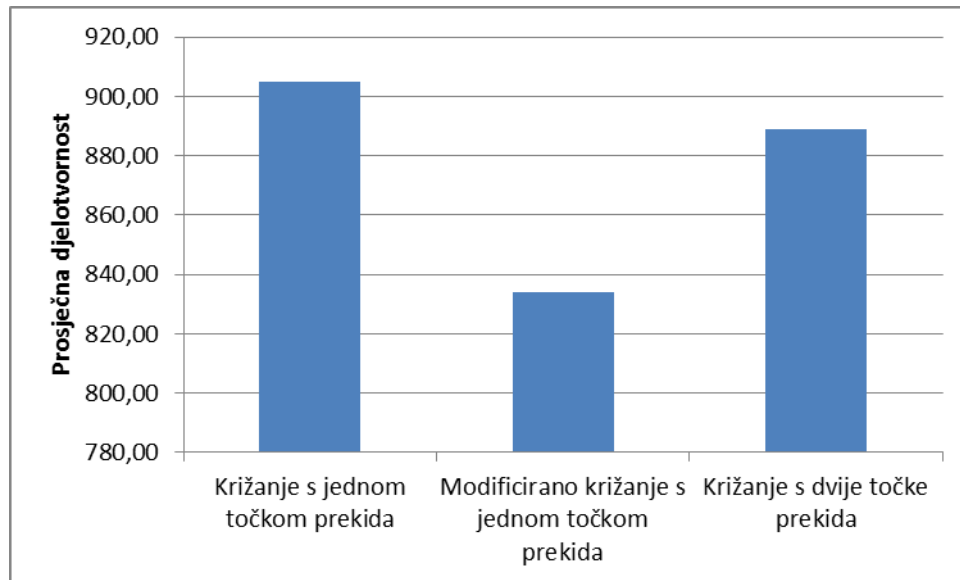
Genetski operator	Prosječna djelotvornost
Uniformna mutacija	908,54
Mutacija zamjenom	932,22
Gaussova mutacija	787,21
Križanje s jednom točkom prekida	904,74
Modificirano križanje s jednom točkom prekida	834,13
Križanje s dvije točke prekida	889,10
Selekcija n najboljih kromosoma	925,36
Rulet selekcija	787,26
Turnirska selekcija	915,35

Vizualni pregled rezultata prema kategorijama genetskih operatora dan je u sljedećim dijagramima.



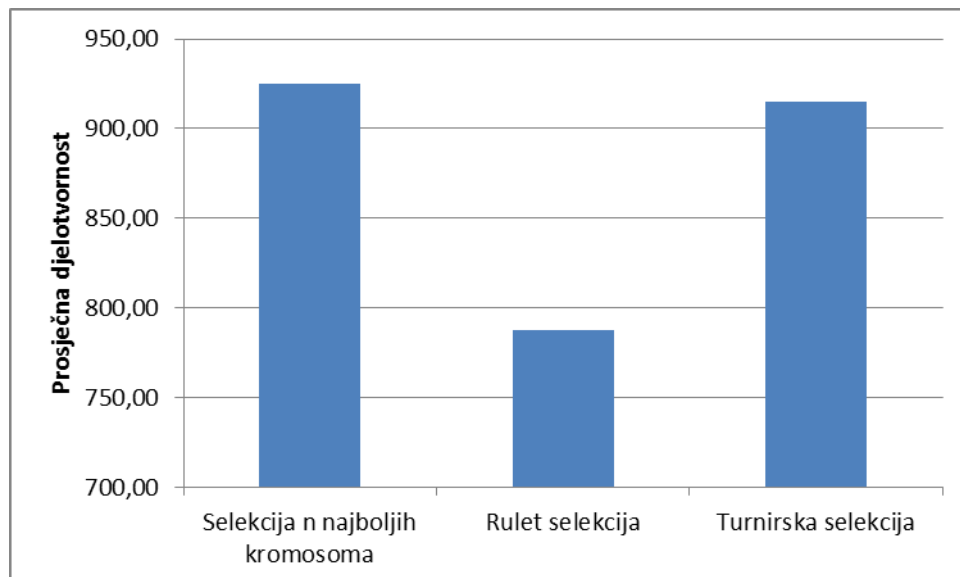
Slika 3: Prosječna djelotvornost mutacijskih operatora

Od genetskih operatora mutacije najveću prosječnu djelotvornost pokazuje operator mutacije zamjenom, s iznosom od 932,22, dok najmanju prosječnu djelotvornost ostvaruje operator Gaussove mutacije.



Slika 4: Prosječna djelotvornost operatora križanja

Od genetskih operatora križanja najveću prosječnu djelotvornost pokazuje operator križanja s jednom točkom prekida, s iznosom od 904,74, dok najmanju prosječnu djelotvornost ostvaruje modificirani operator križanja s jednom točkom prekida.



Slika 5: Prosječna djelotvornost operatora selekcije

Prema prosječnoj djelotvornosti, najbolji operator selekcije je selekcija n najboljih kromosoma s iznosom od 925,36, a najlošiji operator je rulet selekcija (787,26).

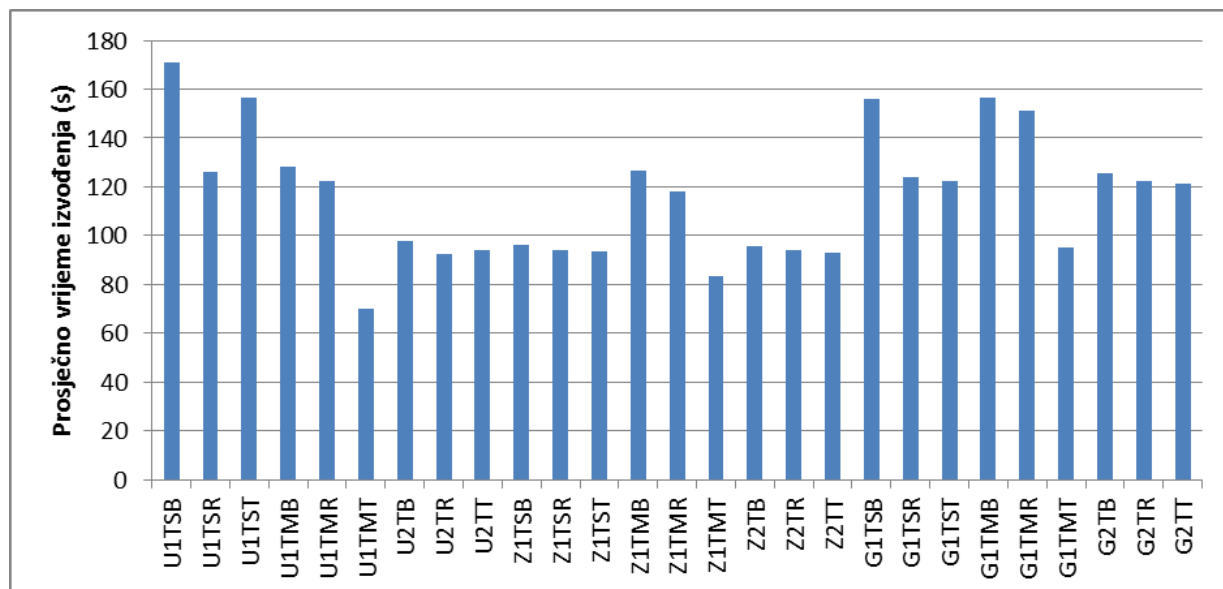
5.3 Prosječno vrijeme trajanja izvođenja algoritma

Provođenjem istraživanja zabilježena su sljedeća prosječna vremena izvođenja algoritma, zaključno s 3000-tom generacijom:

Tabela 11: Prosječno vrijeme trajanja izvođenja algoritma

Naziv	Trajanje (s)	Naziv	Trajanje (s)	Naziv	Trajanje (s)
U1TSB	170,81	Z1TSB	96	G1TSB	155,98
U1TSR	126,13	Z1TSR	94,12	G1TSR	124,21
U1TST	156,8	Z1TST	93,57	G1TST	122,29
U1TMB	128,3	Z1TMB	126,45	G1TMB	156,44
U1TMR	122,3	Z1TMR	118,13	G1TMR	151,47
U1TMT	70	Z1TMT	83,34	G1TMT	94,92
U2TB	98	Z2TB	95,65	G2TB	125,68
U2TR	92,42	Z2TR	93,91	G2TR	122,36
U2TT	94,2	Z2TT	93,1	G2TT	121,25

Usporedba vremena trajanja izvođenja algoritma s različitim kombinacijama genetskih operatora prikazana je u donjem dijagramu:



Slika 6: Odnos vremena trajanja izvođenja algoritma

6. Rasprava

Rezultati predstavljeni u prethodnom poglavlju konzistentni su s rezultatima proteklih istraživanja drugih autora, poput Puljića i Mangera (2013) koji su usporedbom djelotvornosti genetskih algoritama konfiguriranih s različitim kombinacijama genetskih operatora zaključili da određeni genetski operatori ne omogućavaju jednaku djelotvornost genetskog algoritma primijenjenog na rješavanje različitih problema. Dakle, potrebno je za svaki specifični problem pronaći takvu kombinaciju genetskih operatora koja genetskom algoritmu omogućava najbolju djelotvornost.

U ovom slučaju, prilikom rješavanja problema optimizacije rasporeda sati genetskim algoritmom, **9** različitih kombinacija genetskih operatora pokazalo se uspješnijim od ostalih kombinacija genetskih operatora uključenih u istraživanje. Te kombinacije genetskih operatora prikazane su u donjoj tablici.

Tabela 12: Najbolje kombinacije genetskih operatora

Operator mutacije	Operator križanja	Operator selekcije
Uniformna mutacija	Modificirano križanje s jednom točkom prekida	Selekcija n najboljih kromosoma
Uniformna mutacija	Modificirano križanje s jednom točkom prekida	Turnirska selekcija
Uniformna mutacija	Križanje s dvije točke prekida	Selekcija n najboljih kromosoma
Uniformna mutacija	Križanje s dvije točke prekida	Turnirska selekcija
Mutacija zamjenom	Križanje s jednom točkom prekida	Selekcija n najboljih kromosoma
Mutacija zamjenom	Križanje s jednom točkom prekida	Turnirska selekcija
Mutacija zamjenom	Modificirano križanje s jednom točkom prekida	Selekcija n najboljih kromosoma
Mutacija zamjenom	Križanje s dvije točke prekida	Selekcija n najboljih kromosoma
Mutacija zamjenom	Križanje s dvije točke prekida	Turnirska selekcija

Prvo što se može primijetiti u gornjoj tablici je izostanak genetskih operatora mutacije s Gaussovom raspodjelom vrijednosti te ruletske selekcije. Dakle, niti jedna kombinacija genetskih operatora koja je sadržavala jedan od ta dva operatora nije uspjela doseći djelotvornost najboljih kombinacija. Prosječna djelotvornost kombinacija genetskih operatora koje u sebi sadrže operator mutacije s Gaussovom raspodjelom vrijednosti iznosi 787,21, što je najgori rezultat zabilježen u ovom istraživanju. Shodno tome, prosječna djelotvornost operatora ruletske selekcije iznosi 787,26.

Nije teško zaključiti zašto operator mutacije s Gaussovom raspodjelom primijenjen na problem rasporeda sati ostvaruje loše rezultate. Budući da u primijenjenom načinu kodiranja kromosoma vrijednost gena odgovara identifikatoru nekog kolegija, mutacija prema Gaussovoj raspodjeli s većom vjerojatnosti generira vrijednosti u sredini raspona vrijednosti gena, što rezultira čestim pojavama kolegija s oznakama u tom rasponu. Time se negativno utječe i na vrijednost funkcije dobrote kromosoma, jer je veća vjerojatnost da će se prekršiti ograničenje broja sati unutar rasporeda za neki kolegij. Osim toga, izračun standardne devijacije pokazuje i da je Gaussova mutacija jedan od najmanje konzistentnih operatora.

Problem vezan za ruletsku selekciju je nemogućnost garantiranja očuvanja najboljeg kromosoma u generaciji. Budući da je kod takve selekcije vjerojatnost prelaska u sljedeću generaciju za neki kromosom proporcionalna njegovoj dobroti, moguće je da najbolji kromosom ne prođe u sljedeću generaciju. Posljedica toga je da dobrotu najboljeg kromosoma u nekoj generaciji bude lošija u odnosu na prethodnu generaciju. Neumann i ostali (2009) izvršili su teorijsku analizu ruletske selekcije, te su izrazili sumnju u prikladnost odabira tog operatora za rješavanje optimizacijskih problema. Blickle i Thiele (1995) usporedili su 8 različitih selekcijskih operatora i ustvrdili da ruletska selekcija ostvaruje slabije rezultate u odnosu na ostale promatrane operatore. Prema tome, dobiveni rezultati konzistentni su s rezultatima i zaključcima prethodnih istraživanja.

Usporede li se prosječne djelotvornosti operatora mutacije, najvišu djelotvornost ostvaruje operator mutacije zamjenom, s prosječnom djelotvornosti od 932,22. Nešto lošije rezultate pokazuje operator mutacije s uniformnom raspodjelom, s iznosom od 908,54. Ukoliko promotrimo 9 najboljih kombinacija operatora, operator mutacije

zamjenom pojavljuje se u 5, a operator mutacije s uniformnom raspodjelom u 4 kombinacije. Gledajući samo navedene pokazatelje, mutacija zamjenom čini se optimalnim izborom operatora mutacije. Međutim, ukoliko se razmisli o načinu rada tog operatora, može se postaviti pitanje o tome hoće li operator jednaku djelotvornost postizati i prilikom rješavanja složenijeg problema, za koji je potreban mnogo veći broj evolucija od onog koji je uključen u ovo istraživanje.

Budući da operator mutacije zamjenom ne stvara nove vrijednosti, već samo premješta postojeće vrijednosti gena unutar nekog kromosoma, moguće je da algoritam zapadne u lokalni maksimum u slučaju da se selekcijom izgubi neko od svojstava. Operator mutacije je odgovoran za vraćanje takvih svojstava, međutim mutacija zamjenom taj problem ne može riješiti. Zato možemo zaključiti da mutacija zamjenom zaista jest optimalan odabir za operator mutacije kod rješavanja jednostavnih oblika problema rasporeda sati, ali potrebno je daljnjim istraživanjima ispitati ponašanje tog operatora pri rješavanju složenijih problema.

Operator uniformne mutacije očekivano je postigao dobre rezultate, te je moguće daljnjim istraživanjima provjeriti hoće li na složenijim problemima taj operator postići bolje rezultate u odnosu na operator mutacije zamjenom vrijednosti.

Najmanja razlika u djelotvornosti zabilježena je kod genetskih operatora križanja. Unutar 9 najboljih kombinacija operatora zastupljene su sve 3 promatrane vrste operatora križanja. Razlika između prosječne djelotvornosti najboljeg i najlošijeg operatora križanja iznosi 70, odnosno najbolji operator križanja samo je za 8% bolji od najlošijeg operatora križanja. Modificirani operator križanja s jednom točkom prekida ostvaruje prosječno najlošije rezultate što se tiče djelotvornosti, ali zbog *cache*-iranja točaka prekida ostvaruje prosječno kraće trajanje izvođenja u odnosu na druge operatore križanja. Prema rezultatima istraživanja, križanje s jednom točkom prekida najbolji je odabir prilikom rješavanja problema optimizacije rasporeda sati.

Od operatora selekcije, među najboljim kombinacijama operatora javljaju se operatori selekcije n najboljih kromosoma, te turnirska selekcija. Djelotvornost spomenutih operatora je poprilično slična ukoliko promatramo prosječnu djelotvornost među svim slučajevima korištenja operatora. Prosječna djelotvornost operatora selekcije n najboljih kromosoma iznosi 925,36 dok prosječna djelotvornost operatora turnirske selekcije iznosi 915,35, što čini razliku od samo 1%. Takav rezultat je i očekivan, s

obzirom na to da jednostavni operator selekcije n najboljih kromosoma u obzir uzima samo najbolje kromosome, čime osigurava da u svakoj sljedećoj generaciji budu sadržani svi najbolji kromosomi iz prethodne generacije. S druge strane, operator turnirske selekcije ne garantira pojavljivanje najboljeg kromosoma u sljedećoj generaciji.

Slično kao i s operatorima mutacije, opet je otvoreno pitanje rada sa složenim problemima i velikim brojem generacija. Budući da selekcija n najboljih kromosoma ne uzima u obzir slabije kromosome, moguće je da se eliminacijom slabih kromosoma izgubi i neko poželjno obilježje koje nije prisutno u najboljim kromosomima.

Povod daljnjem razmišljanju daje i istraživanje Goluba i ostalih (2003), koji su ustanovili da za rješavanje problema optimizacije rasporeda sati moraju definirati dvije različite kombinacije genetskih operatora, ovisno o složenosti zadanog problema. Kao genetski operator selekcije u slučaju složenijeg problema odabrali su turnirsku selekciju, koja 'omogućava rad s većom populacijom bez usporavanja rada algoritma'.

Zanimljiva tema za buduća istraživanja bilo bi istraživanje o odabiru odgovarajućeg operatora mutacije uz selekciju n najboljih kromosoma pri složenim problemima, kako bi se povećala vjerojatnost da se mutacijom dobije neko od poželjnih obilježja.

Prema rezultatima istraživanja, na jednostavnim problemima rasporeda sati, najbolji odabir operatora selekcije je selekcija najboljih n kromosoma.

Kada se uzme u obzir sve dosad navedeno, opravdano se može odgovoriti na glavno istraživačko pitanje: *postoji kombinacija genetskih operatora koja omogućava najbolju djelotvornost genetskog algoritma za optimizaciju rasporeda sati!*

Potom se može dati odgovor i na pomoćno istraživačko pitanje: *kombinacija genetskih operatora koja omogućuje najbolju djelotvornost genetskog algoritma za optimizaciju sati sastoji se od operatora **mutacije zamjenom, križanja s jednom točkom prekida te selekcije n najboljih kromosoma.***

7. Zaključci

Ovo istraživanje temelji se na prethodnim radovima koji su dokazali da ne postoji takav skup genetskih operatora koji je konzistentno bolji u rješavanju proizvoljnog niza problema u odnosu na ostale skupove genetskih operatora (Fogel, 1999). Vodeći se tom pretpostavkom, mnogi autori pronašli su dokaze da se djelotvornost genetskog algoritma konfiguriranog s određenim skupom genetskih operatora mijenja ovisno o problemu na koji je primijenjen.

Ovim istraživanjem potvrđene su navedene pretpostavke, te je u skladu s njima pronađena najbolja kombinacija genetskih operatora za primjenu na optimizaciji rasporeda sati pomoću genetskog algoritma. Određeni operatori poput mutacije s Gaussovom raspodjelom ili ruletske selekcije pokazali su se iznimno neprimjerenima za primjenu na problemu rasporeda sati. S druge strane, kombinacije genetskih operatora koje su genetskom algoritmu omogućavale najbolju djelotvornost u sebi su sadržavale neke od 7 preostalih operatora. Daljnjom analizom utvrđeno je da najbolju djelotvornost genetskom algoritmu za optimizaciju rasporeda sati omogućuje sljedeća kombinacija genetskih operatora: *mutacija zamjenom, križanje s jednom točkom prekida te selekcija n najboljih kromosoma*.

Analizom podataka dobivenih provedbom istraživanja otvorila su se i neka zanimljiva pitanja za buduća istraživanja. Zbog određenih svojstava korištenih operatora, postoji mogućnost da optimalna kombinacija genetskih operatora nije jednaka na svim veličinama problema ili postavkama parametara genetskog algoritma, čemu svjedoče i prethodna istraživanja (Golub et al, 2003). Dakle, bilo bi opravdano provesti slično istraživanje koje bi se bavilo rješavanjem složenijeg problema optimizacije rasporeda sati, za čije rješenje bi se trebalo provesti veći broj evolucija.

S druge strane, djelotvornost genetskog algoritma može se poboljšati i postavljanjem optimalnih vrijednosti parametara poput veličine populacije, vjerojatnosti mutacije ili vjerojatnosti križanja (Goldberg, 2002). Provođenjem novog istraživanja moglo bi se utvrditi koje su vrijednosti parametara potrebne za najbolju djelotvornost genetskog algoritma primijenjenog na problem optimizacije rasporeda sati.

8. Zahvale

Najveću zahvalu upućujem mentorici dr.sc. Dijani Oreški, koja me je na kolegiju *Inteligentni sustavi* na Fakultetu organizacije i informatike upoznala s osnovama rada genetskih algoritama, te je tokom cjelokupne izrade ovog rada uložila veliki trud i vrijeme na savjete i konstruktivne kritike, što je značajno utjecalo na kvalitetu samog rada.

Posebno bih se zahvalio i kolegama Draganu Marinkoviću, Ivanu Petroviću i Ivanu Raspoviću, s kojima sam na gore spomenutom kolegiju kao projektni zadatak izradio programsko rješenje koje koristi genetski algoritam za optimizaciju rasporeda sati. Navedeni program poslužio je kao osnova programskog rješenja izrađenog za potrebe ovog rada.

Na kraju bih se zahvalio svima koji su na bilo koji način dali doprinos ovom znanstvenom radu, a nisu obuhvaćeni gornjim rečenicama.

9. Popis literature

1. Bäck, T. (1995) 'Generalized Convergence Models for Tournament- and (m,l)-Selection', *Proceedings of the Sixth International Conference on Genetic Algorithms*, San Francisco, CA, pp. 2-8.
2. Beaty, S.J. (1991) *Genetic Algorithms and Instruction Scheduling*, Microarchitecture, pp. 206-211.
3. Beyer, H. G. (1998) 'On the explorative power of ES/EP-like algorithms', *Proceedings of the 7th Annual Conference on Evolutionary Programming*, Springer, Berlin.
4. Blanco, J. J. i Kathib, L. (1998) 'Course Scheduling as a Constraint Satisfaction Problem', *PACT98 Proceedings*, International Conference on Parallel Architectures and Compilation Techniques, Pariz.
5. Blickle, T., Thiele, L. (1991) 'A Mathematical Analysis of Tournament Selection', *Proceedings of the Sixth International Conference on Genetic Algorithms*, San Francisco, pp. 2 - 8.
6. Blickle, T. i Thiele, L. (1995) 'A Comparison of Selection Schemes Used in Genetic Algorithms', *TIK-Report*, vol. 11, Zurich.
7. Boyd, S. P. i Vandenberghe, L. (2004) *Convex Optimization*, Cambridge University Press, pp. 129.
8. Burke, E., Elliman, D. i Weare, R. (1994). 'A Genetic Algorithm Based University Timetabling System', *East-West Conference on Computer Technologies in Education*, Krim, Ukrajina, pp. 35 – 40.
9. Burke, E. K., Elliman D. G. i Weare, R. F. (1995) 'A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems', *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufman, pp. 605 – 610.
10. Corcoran, A.L., Wainwright, R.L. (1992) 'A Genetic Algorithm for Packing in Three Dimensions', *Applied computing, Technological challenges of the 1990's*, vol. 2, pp. 1021-1030.
11. Eiben, A. E., Schippers, C. A. (1998) 'On Evolutionary Exploration and Exploitation', *Fundamenta Informaticae*, vol. 35, pp. 35-50.

12. Fogel, D.B., (1999) 'Some Recent Important Foundational Results in Evolutionary Computation', *Evolutionary Algorithms in Engineering and Computer Science*, Miettinen, K., Neittaanmaki, P., Makela, M.M., Periaux, J., John Wiley & sons, LTD.
13. Grefenstette, J., Gopal, R., Rosmaita, R. i Gucht, D. (1985) 'Genetic algorithms for the traveling salesman problem', *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Mahwah, NJ.
14. Goldberg, D. (1989) *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
15. Goldberg, D. (2002) *Lessons from and for Competent Genetic Algorithms*, Kluwer, Boston, MA.
16. Golub, M., Sigl, B. i Mornar, V. (2003), 'Solving Timetable Scheduling Problem by Using Genetic Algorithms', *Information Technology Interfaces*, Cavtat, pp. 519-524.
17. Herrera, F. i Lozano M. (2000) 'Gradual Distributed Real-Coded Genetic Algorithms', *IEEE Trans. Evolutionary Computation*, vol. 4, no. 1, pp. 43-63.
18. Holland, J. (1992) *Adaptation in Natural and Artificial Systems*, Massachusetts: MIT Press Cambridge.
19. Kumar, A. (2013) 'Encoding Schemes in Genetic Algorithm', *International Journal of Advanced Research in IT and Engineering*, vol. 2, no. 3.
20. Lukas, S., Aribowo, A. i Muchri, M. (2009) 'Genetic algorithm and heuristic search for solving timetable problem case study: Universitas Pelita Harapan timetable', *Proceedings of Conference on Applications of Digital Information and Web Technologies*, London, pp. 629-633.
21. Meffert, K. (2009) 'Genetische Algorithmen mit JGAP', *Java Magazin*, vol.6.
22. Mitchell, T M. (1997) *Machine Learning*, MIT Press and The McGraw-Hill Companies, Massachusetts.
23. Neumann, F., Oliveto, P. S. i Witt, C. (2009) 'Theoretical Analysis of Fitness Proportional Selection: Landscapes and Efficiency', *GECCO*, ACM, pp. 835-842.
24. Piola, R. (1992) *Evolutionary solutions to a highly constrained combinatorial problem*, Dipartimento di informatica, Università degli di Torino, Italiija.
25. Puljić, K. i Manger, R. (2013) 'Comparison of Eight Evolutionary Crossover Operators for the Vehicle Routing Problem', *Mathematical Communications*, vol. 18, pp. 359 – 375.

26. Razali, N. M. i Geraghty, J. (2011) 'Genetic Algorithm Performance with Different Selection Strategies in Solving TSP', *Proceedings of the World Congress on Engineering 2011*, WCE 2011, London.
27. Xu, H. (1999) 'Comparison of Genetic Operators on a General Genetic Algorithm Package', magistarski rad, Oklahoma State University, Norman.

Sažetak

Autor: Ivan Ković

Naslov rada: Poboljšanje djelotvornosti genetskog algoritma za optimizaciju rasporeda sati odabirom genetskih operatora

Sažetak:

U ovom radu predstavljen je način poboljšanja djelotvornosti genetskog algoritma za optimizaciju rasporeda sati pomoću odabira odgovarajućih genetskih operatora. S obzirom na prethodna istraživanja i općenita znanja iz područja evolucijskih algoritama, temeljna pretpostavka istraživanja glasila je da će djelotvornost genetskih operatora biti različita ovisno o tome na koji su problem primijenjeni. Za istraživanje je odabrano 9 genetskih operatora, i to po 3 operatora iz skupina operatora mutiranja, križanja i selekcije. Promatrana je prosječna vrijednost funkcije dobrote najboljeg kromosoma u 3000-toj generaciji kroz 20 provođenja algoritma za svih 27 mogućih kombinacija genetskih operatora. Rezultati su pokazali da se za genetski algoritam za optimizaciju rasporeda sati kombinacija genetskih operatora koja algoritmu omogućava najbolju djelotvornost sastoji od operatora mutacije zamjenom vrijednosti, križanja s jednom točkom prekida te selekcije n najboljih kromosoma.

Ključne riječi: genetski algoritmi, vremensko raspoređivanje, djelotvornost, odabir genetskih operatora

Summary

Author: Ivan Kovic

Title: Improvement of efficiency of genetic algorithm for timetable optimization through selection of genetic operators

Summary:

In this paper, a method of improving the efficiency of genetic algorithm for timetable optimization through selection of genetic operators is introduced. Considering both the general knowledge from the field of evolutionary algorithms and prior research, the basic presumption of this research says that genetic operators will yield different levels of efficiency when applied to different problems. Three different genetic operators from each category of genetic operators were selected, so the total number of operators included in research was 9. Average fitness function value of the best chromosome after 3000 evolutions through 20 algorithm runs was calculated and compared for all of the possible 27 different combinations of genetic operators. The results showed that the best combination of genetic operators for the genetic algorithm for timetable optimization consists of swapping mutation, single point crossover and best n chromosomes selection.

Keywords: genetic algorithms, timetabling problem, algorithmic efficiency, genetic operator selection