

OPTIMIZACIJA PUTANJE MANIPULATORA PRIMJENOM GENETSKOG ALGORITMA

MANIPULATOR ROUTING OPTIMIZATION USING GENETIC ALGORITHM

Hrvoje Rakić¹, Tomislav Brajković², Slobodan Obrenović³

¹Tehničko veleučilište u Zagrebu, Zagreb, Hrvatska

²Kel d.o.o., Kloštar Ivanić, Hrvatska

³Inovatus usluge d.o.o., Zagreb, Hrvatska

Sažetak

Problem trgovačkog putnika jedan je od najpoznatijih problema kombinatorne optimizacije. Spada u skupinu NP-potpunih problema i za veći broj gradova nerješiv je egzaktnim metodama u razumnom vremenu. U ovom radu putanja manipulatora optimizirana je pomoću genetskog algoritma. Istraživanja su pokazala da predloženi genetski algoritam pronalazi kraću rutu u realnom vremenu u usporedbi s postojećim modelom odabira putanje manipulatora. Također, istraživanja su pokazala da bi ovakav genetski algoritam trebalo upariti s tehnikama lokalnog pretraživanja prostora za postizanje konzistentno dobrih rješenja.

Ključne riječi: Optimizacija, genetski algoritam, problem trgovačkog putnika

Abstract

The traveling salesman problem is one of the most famous combinatorial optimization problems. It's categorized as one of the NP-complete problems and for the growing number of cities is unsolvable with the use of exact methods in a reasonable time. In this paper manipulator path has been optimized using a genetic algorithm. Studies have shown that the proposed genetic algorithm can find a shorter route in real-time, compared with the existing manipulator model of path selection. Also, studies have shown that the genetic algorithm should be paired with the local search techniques in order to be able to consistently achieve good solutions.

Keywords: Optimization, genetic algorithm, traveling salesman problem

1. Uvod

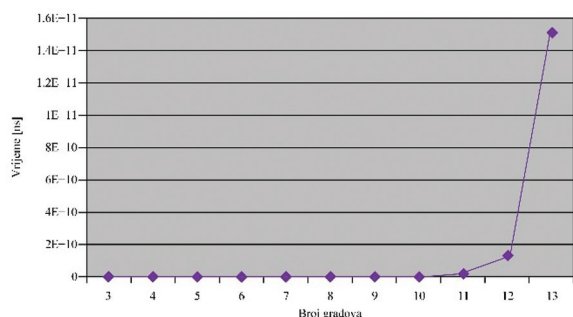
1. Introduction

Problem pronalaska najkraće rute često se javlja u inženjerskoj praksi. Pronalazak najkraćeg puta dostavnih vozila, bušenje sklopovskih pločica, analiziranje kristalnih struktura, raspoređivanje redosljedja poslova, samo su neki od problema. Govoreći o problemu optimizacije rute odnosno pronalasku najkraćeg puta, najčešće se misli na problem trgovačkog putnika (engl. *Traveling salesman problem, TSP*). Problem trgovačkog putnika definiran je na sljedeći način: zadan je skup gradova i njihovih međusobnih udaljenosti. Krećući iz proizvoljno odabranog grada, trgovački putnik mora obići sve gradove točno jednom i vratiti se u početni grad (ishodišni čvor). Cilj je pronaći zatvorenu rutu pri kojoj su troškovi putovanja najmanji, odnosno duljina puta najkraća [1], [2]. Matematički model problema trgovačkog putnika svodi se na traženje Hamiltonovog ciklusa najmanje težine u težinskom grafu, [3]. Težina zadanog brida može značiti visinu troška, udaljenost, vrijeme ili bilo koju drugu mjeru koja karakterizira taj brid. Problem trgovačkog putnika spada u skupinu NP-potpunih problema i jedan je od najsloženijih i najpoznatijih problema kombinatorne optimizacije. Često se koristi kao mjerilo performansi za optimizacijske algoritme poput simuliranog kaljenja, tabu pretraživanja, itd.

Ako je početni grad unaprijed određen, za n zadanih gradova broj mogućih ruta računa se prema izrazu (1) [1]:

$$\frac{(n-1)!}{2} \quad (1)$$

Vrijeme potrebno da se metodom iscrpne pretrage (engl. *brute force*) riješi problem od 16 gradova iznosi oko 4,7 dana (slika 1) [3]. Zbog nemogućnosti da se problem trgovačkog putnika riješi egzaktnim metodama u razumnom vremenu, za rješavanje problema koriste se heuristike. Općenito, heuristike su algoritmi koji ne garantiraju optimalno rješenje, ali mogu u razumnom vremenu pronaći rješenja vrlo blizu optimuma. Pregled nekih od najčešće korištenih heuristika nalazi se u [4] i [5].



Slika 1 Vrijeme pronalaska najkraćeg puta kod problema trgovačkog putnika

Figure 1 Time needed for finding the shortest route for the traveling salesman problem

Izvor: Čupić, M.; *Prirodom inspirirani optimizacijski algoritmi; Skripta, Verzija: 1.0.7.; Fakultet elektrotehnike i računarstva, Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave; p.1.; Zagreb, 2010.*

2. Genetski algoritam

2. Genetic algorithm

Genetski algoritam (dalje u tekstu: GA) pripada skupini evolucijskih algoritama. U tu skupinu pripadaju i evolucijske strategije, evolucijsko programiranje i genetsko programiranje. Neke od prednosti evolucijskih algoritama su: široka mogućnost primjene, hibridizacija s tradicionalnim metodama optimizacije, konceptualna jednostavnost, robusnost spram dinamičkih promjena, mogućnost rješavanja problema za koje ne postoji ljudska ekspertiza te mogućnost paralelizirane implementacije [6], [7], [8].

GA je metaheuristička metoda koja oponaša prirodan evolucijski proces. Po načinu djelovanja ubraja se u metode usmjerenog slučajnog pretraživanja prostora rješenja (engl. *guided random search techniques*) u potrazi za globalnim optimumom. GA najčešće se rješavaju optimizacijski problemi koji se mogu opisati na sljedeći način.

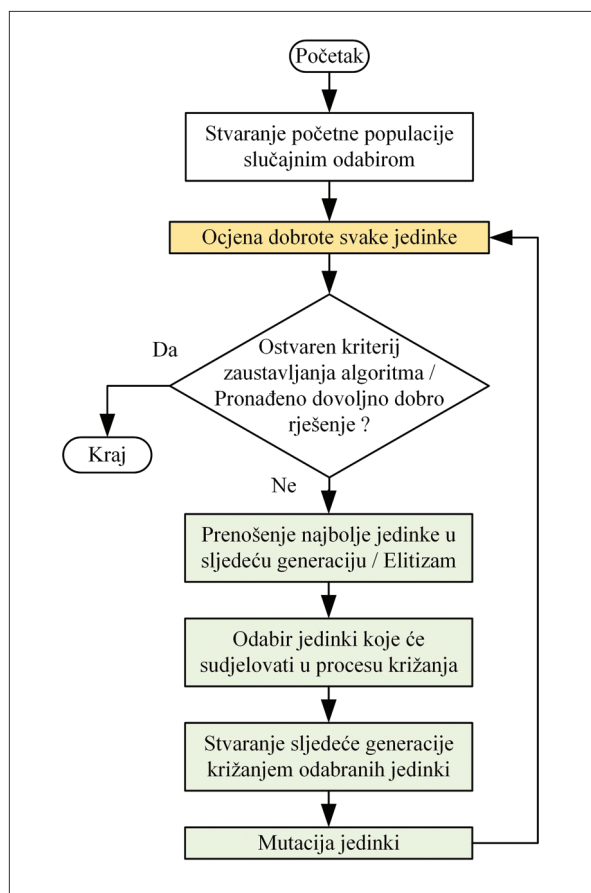
Zadana je funkcija cilja $f(\vec{x})$ gdje je

$$\vec{x} = (x_1, x_2, \dots, x_n).$$

Potrebno je pronaći \vec{x}^* koji maksimizira (ili minimizira) funkciju f [3]. Snaga GA je

sposobnost određivanja položaja globalnog optimuma u prostoru s više lokalnih ekstrema. Budući da oponašaju prirodan evolucijski proces, imaju osobine mutacije, selekcije i izmjene genetskog materijala (rekombinacije) [9].

Na slici 2 prikazan je dijagram toka općenitog GA. U općem slučaju, GA započinje svoj rad generiranjem početne populacije jedinki slučajnim odabirom. Za razliku od konvencionalnih optimizacijskih metoda, GA ne operira direktno na parametrima promatranog problema, već na kodiranim vrijednostima tih parametara.



Slika 2 Dijagram toka općenitog genetskog algoritma

Figure 2 Flowchart of conventional genetic algorithm

Početna (trenutna) populacija sadrži kromosome, pri čemu svaki kromosom predstavlja jednu početnu točku, odnosno jedno potencijalno rješenje problema koji se obrađuje. To može biti matematička funkcija, redosljed strojnih

operacija u nekom pogonu, optimizacija podatkovnih kanala telekomunikacijske infrastrukture i slično.

Konvencionalne optimizacijske metode započinju pretragu iz jedne točke, dok GA započinje pretragu iz više različitih točaka istovremeno [10]. Upravo to svojstvo istovremene pretrage iz više različitih točaka (istodobno se razmatra više različitih rješenja), pridonosi robusnosti GA jer povećava vjerojatnost pronalaska globalnog optimuma.

Nakon stvaranja početne populacije, za svako rješenje, odnosno za svaki kromosom, računa se dobrotu. Mjera kvalitete rješenja (dobrota) računa se pomoću funkcije cilja (funkcije dobrote). Ocjenom dobrote izdvajaju se bolje jedinke (bolja rješenja) od lošijih. Tada se iz stare populacije formira nova koristeći operatore križanja i mutacije [9].

Križanje je postupak u kojem se kombinacijom dviju jedinki (roditelji) iz trenutne populacije (generacije) stvara nova jedinka (potomak) koja pripada sljedećoj generaciji. Oni roditelji koji imaju veću dobrotu, imati će i više izgleda da sudjeluju u stvaranju potomaka. Nakon što je završeno stvaranje nove populacije, ona postaje trenutna populacija (smjena generacija) i postupak se ponavlja. Kada se zadovolje određeni kriteriji, a najčešće je to neki broj izvršenih generacija, čitav postupak se zaustavlja. Neke prednosti i nedostaci GA navedeni su u tablici 1 [6].

3. Prikaz rješenja i Operatora

3. *Solution and Operator representation*

U regalnom skladištu gotove robe, svaki tip proizvoda nalazi se u svom odjeljku. U tom skladištu nalazi se manipulator koji služi za dobavu proizvoda iz odgovarajućih odjeljaka. Za svaki odjeljak poznate su koordinate, a time i udaljenost odjeljka od polazne točke manipulatora. Svaki odjeljak sadrži više primjeraka proizvoda. Pomicanje manipulatora u 3D prostoru predstavlja vrlo složeno gibanje (rotacija, translacija, stupnjevi slobode gibanja, itd.). Ako se zanemari navedeno, kao i kapacitet palete manipulatora i slično, što i nije predmet ovog istraživanja, dobava svih zatraženih proizvoda iz odjeljaka svodi se na problem trgovačkog putnika. Manipulator mora obići sve odjeljke (sve proizvode, sve gradove), dohvatiti jedan ili više proizvoda iz svakog

odjeljka, staviti ih na svoju paletu i vratiti se na mjesto predviđeno za istovar (ishodište).

Tablica 1 Neke prednosti i nedostaci genetskog algoritma

Table 1 Some advantages and limitations of genetic algorithm

Prednosti GA
<ul style="list-style-type: none"> • Lakoća prilagodbe različitim optimizacijskim problemima • Mogućnost paralelizma • Dobre performanse u slučaju višekriterijske optimizacije • Mogućnost nadogradnje i povećanja učinkovitosti • Relativno dobro savladavanje šuma u podacima • Dobro rješavanje višedimenzionalnih problema jer vraća garnituru (skup) rješenja • Otpornost na "zaglavljivanje" u lokalnom optimumu • Diskontinuitet prostora rješenja ne narušava performanse • Ne zahtijeva poznavanje gradijenta prostora rješenja • Široka primjenjivost uz dobre performanse kod velikih optimizacijskih problema
Nedostaci GA
<ul style="list-style-type: none"> • Vrlo često potrebno je prilagoditi algoritam problemu ili problemu radu algoritma • Problem identifikacije i formulacije funkcije cilja • Prerana konvergencija • Veliki utjecaj parametara • Nemogućnost korištenja gradijenata • Teško identificiranje lokalnih optimuma, potrebno ga je kombinirati s tehnikama lokalnog pretraživanja prostora (hibridizacija) • Teška integracija podataka specifičnih za promatrani problem • Neučinkovitost za glatke unimodalne funkcije • Nema jasan kriterij zaustavljanja • Problem pri razlikovanju dubokih lokalnih optimuma od globalnog optimuma

Izvor: Sivanandam, S.N.; Deepa, S.N.; *Introduction to Genetic Algorithms*; ISBN: 978-3-540-73189-4; Springer Berlin Heidelberg, 2008.

Za prikaz rješenja trgovačkog putnika u ovom radu korišten je prikaz na bazi permutacija pomoću redoslijeda obilaska odjeljaka (engl. *path based representation*). Kod prikaza rješenja pomoću redoslijeda obilaska odjeljaka, n odjeljaka koje je potrebno obići postavljeni su u listu odjeljaka prema zadanom redoslijedu manipulatora. Početni (ujedno i završni) odjeljak fiksiran je za sva moguća rješenja, te se ne kodira eksplicitno u kromosomu. Npr. ruta:

1→5→7→8→6→1

jednostavno je prikazana rješenjem (kromosomom) kao:

(5 7 8 6)

Ovakav prikaz rješenja je intuitivan, a ujedno omogućuje upotrebu velikog broja različitih operatora križanja i mutacije. Budući da za ovakav prikaz rješenja klasični operatori križanja i mutacije koji se koriste kod GA nisu prikladni, znanstvenici su kreirali i razvili nove operatore. Razlog tome je što uslijed križanja može doći do pojave ilegalnih ruta. Neki odjeljci mogu se kod potomka javiti dva ili više puta, dok neki odjeljci mogu biti u potpunosti izbačeni iz rute.

Odabir roditelja koji će sudjelovati u stvaranju potomaka proveden je *k* turnirskom selekcijom na sljedeći način. Iz populacije se slučajnim odabirom sa zamjenom odabire *k* jedinki (veličina turnira). Zatim se između njih biraju dvije najbolje koje idu u bazen za reprodukciju. Ovaj postupak se ponavlja sve dok se ne dobije broj roditelja jednak veličini populacije, da bi se križanjem populacija zamijenila u potpunosti (svaki par roditelja, osim posljednjeg, daje križanjem dva potomka, dok posljednji par križanjem daje jednog potomka). Također, u rad algoritma ugrađen je mehanizam elitizma (jedinka s najvećom dobrotom iz trenutne generacije nepromijenjena se prenosi u sljedeću generaciju).

Za križanje roditelja odabran je postupak heurističkog križanja (engl. *heuristic crossover*), kojeg su prvi predložili Grefenstette et al. [10], [11], a radi na sljedeći način. Npr., neka su nakon turnira odabrana dva roditelja prikazana slikom 3.

Roditelj 1	(8	6	3	2	4	5	7)
Roditelj 2	(4	5	3	2	8	7	6)

Slika 3 Prikaz dvaju roditelja

Figure 3 Representation of two parents

Ovakav algoritam heurističkog križanja blago je modificiran. Odnosno, početni odjeljak ne odabire se slučajno, nego se za početni odjeljak rute uvijek odabire početni odjeljak u prvom roditelju (u ovom primjeru to je odjeljak 8), slika 4.

Potomak	(8	-	-	-	-	-	-)
---------	---	---	---	---	---	---	---	---	---

Slika 4 Prikaz potomka nakon umetanja prvog odjeljka

Figure 4 Representation of offspring after insertion of first compartment

Također, odabir sljedećeg odjeljka nije napravljen prema vjerojatnosnoj distribuciji definiranoj na temelju cijene bridova [10], nego mehanizmom heuristike najbližeg susjeda (najkraće udaljenosti do sljedećeg odjeljka). Odabirom sljedećeg odjeljka prema distribuciji udaljenosti ne bi se čuvala sličnost između poretka odjeljaka u roditeljima i djetetu, te bi u tom slučaju GA služio praktički samo kao način kodiranja problema. Uz heuristiku najbližeg susjeda čuva se svojstvo susjednosti odjeljaka unutar rodoslova. Križanje se nastavlja na način da se u oba roditelja pronade sljedeći odjeljak na ruti nakon zadnjeg ubačenog odjeljka (odjeljak 8) koji je ubačen u potomka (slika 5).

Roditelj 1	(8	6	3	2	4	5	7)
Roditelj 2	(4	5	3	2	8	7	6)

Slika 5 Prikaz nastavka križanja

Figure 5 Representation of crossover continuation

Budući da je zadnji ubačeni odjeljak u dijete odjeljak 8, sljedeći odjeljak na ruti je odjeljak 6 iz roditelja 1 i odjeljak 7 iz roditelja 2 (slika 5). Između ta dva odjeljka odabire se onaj koji je bliže zadnje postavljenom odjeljku (odjeljku 8). To neka, npr., bude odjeljak 7 iz roditelja 2 (slika 5). Taj odjeljak ubacuje se u potomka i briše iz oba roditelja (slika 6).

Roditelj 1	(8	6	3	2	4	5	7)
Roditelj 2	(4	5	3	2	8	7	6)
Potomak	(8	7	-	-	-	-	-)

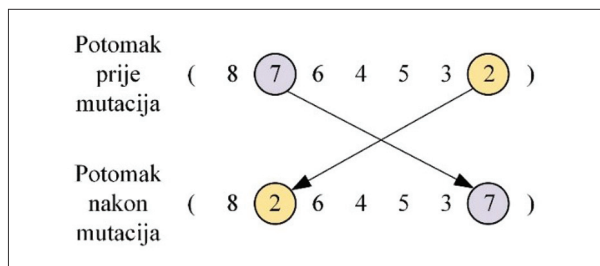
Slika 6 Prikaz postupka formiranja potomka

Figure 6 Representation of offspring forming procedure

Kada bi se odjeljak 7 (slika 5) već nalazio u potomku, onda bi odjeljak 6 (slika 5) bio upisan u potomka. Ako bi se oba odjeljka (6 i 7 sa slike 5) već nalazili u potomku, idući bi odjeljak u ruti bio odabran slučajnim odabirom od preostalih neubačenih odjeljaka. Zatim se ovaj postupak ponavlja, odnosno sljedeći kandidat za ubacivanje u potomka je odjeljak 6 budući da u oba roditelja u ovom koraku slijedi neposredno iza odjeljka 7 (slika 5). Taj postupak se ponavlja dok se ne ubace svi odjeljci.

Mutacija svakog kromosoma izvedena je operatorom mutacije zamjene gena (engl. *swap mutation operator*) [11], [12]. Za svaki kromosom

u populaciji generatorom pseudoslučajnih brojeva generira se broj u intervalu [0, 1). Ako je taj pseudoslučajni broj manji od vrijednosti parametra mutacije, dopušta se mutacija tog kromosoma. Operator mutacije zamjene gena radi tako da nasumično odabere dva gena, odnosno bilo koja dva odjeljka, i zamijeni njihova mjesta (slika 7).



Slika 7 Mutacija zamjenom gena

Figure 7 Exchange mutation

Dobrota svakog kromosoma izračunata je prema izrazu (2):

$$v = M - \sum_{i=2}^{n-1} d_{i,i+1} + d_{n,1} + d_{1,2} \tag{2}$$

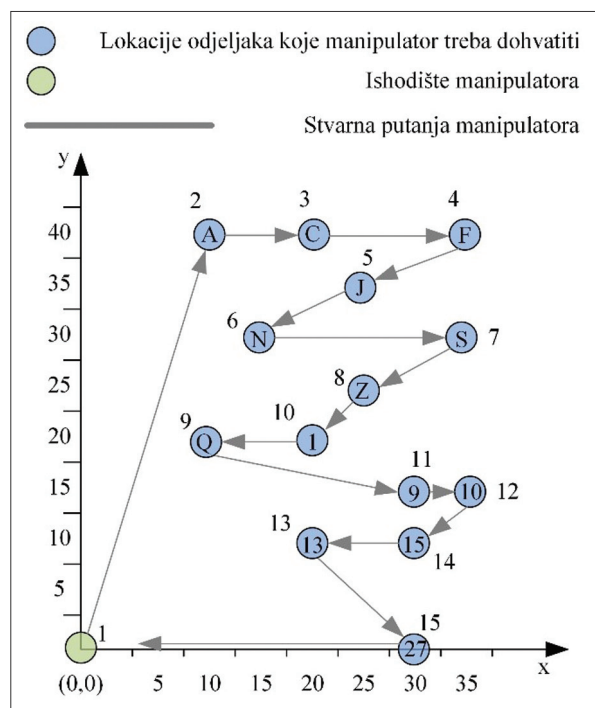
U izrazu (2) v predstavlja dobrotu jednog kromosoma (jedinke); n je broj odjeljaka (odjeljci 1,2,3,... n); M je umnožak broja odjeljaka uvećanog za 1 i najveće udaljenosti između odjeljka i početne pozicije (ishodišta manipulatora); $d_{i,i+1}$ su udaljenosti susjednih odjeljaka u ruti (odnosno u kromosomu), dok su $d_{n,1}$ i $d_{1,2}$ udaljenosti između prvog (odnosno zadnjeg) odjeljka u ruti (koji nije eksplicitni dio kromosoma) i njegovih susjeda u ruti. Parametar M osigurava da je dobrota uvijek pozitivan broj jer je zajamčeno veći ili jednak ukupnoj duljini rute između odjeljaka.

4. Rezultati istraživanja

4. Research results

Na slici 8 sivom bojom prikazana je stvarna putanja manipulatora i može se vidjeti da ona nije optimalna. U stvarnosti manipulator red po red sukcesivno dohvća proizvode, nakon čega se vraća u svoju početnu poziciju (ishodište koordinatnog sustava) i čeka sljedeći nalog. Prikazana ruta iznosi 253,38 jedinica i služiti će kao mjerilo performansi algoritma. GA implementiran je koristeći računalni program Matlab i programski kôd dostupan je kod autora. Pri tome nije korišten Global Optimization Toolbox Matlaba ili slični

paketi, nego je napisan vlastiti (izvorni) kôd u svrhu daljnjih istraživanja povezivanja GA (hibridizacije) s metodama lokalnog pretraživanja.



Slika 8 Prikaz stvarne rute manipulatora

Figure 8 Representation of real manipulator path

Performanse GA ovisne su o sljedećim parametrima: veličini populacije, vjerojatnosti mutacije, broju generacija i veličini prozora. Ti parametri mogu poprimiti širok raspon vrijednosti, što može značajno utjecati na rad i kvalitetu rješenja GA. Parametar veličine prozora fiksiran je na vrijednost 4. Utjecaj veličine prozora kao i ostalih parametara na rad GA može se pronaći u [13]. GA s velikim selekcijskim pritiskom ranije pronalaze rješenje, tj. brže konvergiraju nauštrb kvalitete dobivenog rješenja, jer brža konvergencija uzrokuje veću vjerojatnost zaglavljivanja u lokalnom optimumu. S druge strane, ako je selekcijski pritisak premali, vrijeme se troši na beskorisne iteracije jer je konvergencija u tom slučaju prespora [14].

Optimalne vrijednosti ostalih parametara utvrđene su eksperimentalno. Od veličine populacije očekuje se pozitivna korelacija s kvalitetom i brzinom konvergencije rješenja. Vjerojatnost mutacije ključni je parametar koji omogućuje izlaz GA iz lokalnog optimuma (sukladno s biološkim učinkom mutacije kao nužnim izvorom nastajanja novih svojstava). Prema tome, očekuje se značajan

učinak mutacije na kvalitetu rješenja, no taj učinak je složen. Broj generacija trebao bi pozitivno korelirati s kvalitetom rješenja, a negativno s brzinom izvođenja programa. U svrhu odabira najboljeg skupa parametara, slučajnim odabirom iz zadanog raspona (tablica 2) generirano je M skupova parametara. Za svaki skup parametara GA pokrenut je $N=100$ puta. Time je za svaki skup parametara dobiveno 100 najboljih vrijednosti dobrote. Analizom i usporedbom razdiobe vrijednosti dobrote

za svaki skup parametara dolazi se do zaključka o kvaliteti svakog promatranog skupa parametara.

Tablica 2 Zadani raspon parametara algoritma

Table 2 Default GA parameter range

	Veličina populacije	Vjerojatnost mutacije (p)	Broj generacija
min	10	0,01	10
max	100	0,5	1000

Tablica 3 Zadani raspon parametara algoritma

Table 3 Default GA parameter range

Skup parametara (M)	Veličina populacije	Vjerojatnost mutacije (p)	Broj generacija	Dobrota		
				Maksimalna vrijednost	Median	Aritmetička sredina (mean)
1	18	0,18	152	621,08	609,99	608,71
2	86	0,20	425	621,08	614,33	615,03
3	46	0,30	832	621,65	615,41	615,63
4	78	0,27	480	621,65	614,33	614,85
5	100	0,16	624	621,65	614,33	614,92
6	64	0,02	708	621,08	613,19	613,52
7	60	0,04	850	621,65	612,80	613,32
8	14	0,35	726	621,08	614,33	613,87
9	26	0,28	18	617,05	602,97	601,85
10	58	0,34	460	621,65	614,33	614,81
11	76	0,48	506	621,65	615,59	615,74
12	86	0,32	128	621,08	614,33	614,14
13	42	0,04	564	621,65	612,80	612,82
14	14	0,13	630	621,65	611,89	611,64
15	30	0,29	875	621,65	614,33	614,65
16	94	0,09	974	621,65	614,33	614,88
17	24	0,49	172	621,08	613,02	612,17
18	46	0,47	360	621,08	614,33	614,39
19	96	0,43	646	621,08	615,59	615,80
20	12	0,23	196	621,08	608,71	607,69
21	70	0,43	340	621,65	614,33	615,07
22	98	0,06	604	621,65	614,33	614,96
23	26	0,12	302	621,65	612,98	612,07
24	66	0,22	744	621,65	614,50	615,21
25	92	0,36	914	621,65	615,59	615,76
26	20	0,25	66	621,08	609,15	607,72
27	62	0,06	540	621,65	614,08	613,90
28	82	0,16	764	621,65	614,33	614,76
29	74	0,14	998	621,65	614,33	615,00
30	52	0,11	372	621,65	613,12	613,68
31	22	0,08	286	621,08	611,08	610,14
32	56	0,17	922	621,65	614,33	614,58
33	40	0,20	950	621,65	615,22	614,94
34	56	0,38	326	621,65	614,33	614,71
35	38	0,39	488	621,65	614,33	614,41
36	90	0,26	696	621,65	615,22	615,47
37	84	0,21	888	621,65	614,33	615,26
38	34	0,39	258	621,65	614,33	613,94
39	66	0,45	682	621,65	615,59	615,93
40	28	0,32	800	621,65	614,33	614,89
41	50	0,41	582	621,65	615,59	615,82
42	36	0,47	244	621,08	613,02	613,02
43	72	0,02	88	621,08	612,80	612,23
44	38	0,44	406	621,65	614,33	614,35
45	52	0,11	804	621,65	614,33	614,97
46	76	0,24	108	621,08	614,33	613,91
47	44	0,41	40	621,65	611,08	610,17
48	90	0,36	444	621,65	615,59	615,68
49	20	0,07	144	621,08	608,71	607,42
50	80	0,31	214	621,65	614,33	614,39

Od distribucije dobrote rješenja za najbolji skup parametara očekuje se najveći maksimum u odnosu na ostale distribucije, odnosno doseganje globalnog rješenja problema. Također se očekuje da što veći broj rješenja bude blizak maksimumu, odnosno da se za taj skup parametara konzistentno pronalazi kvalitetno rješenje. Da bi se to utvrdilo, uspoređuju se maksimumi i kvantili dobrote rješenja.

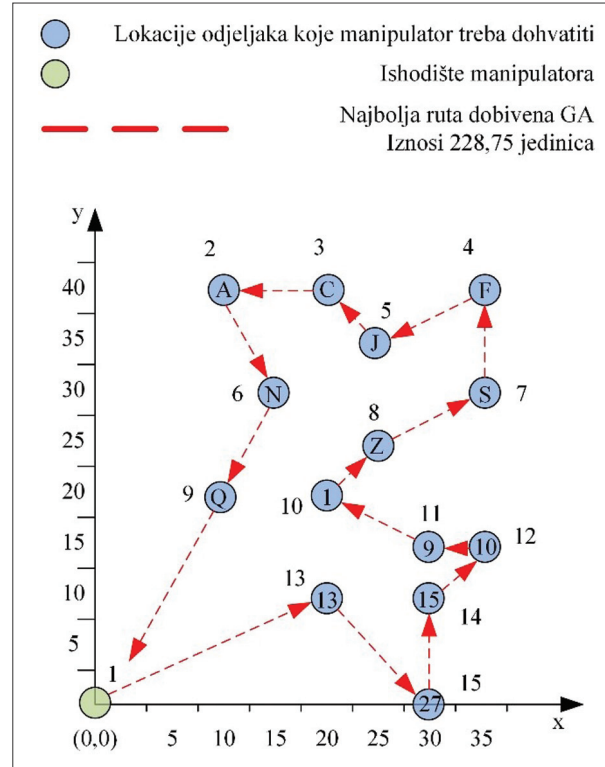
Na temelju dobivenih rezultata (tablica 3), odabrana su četiri najbolja skupa parametara: 25, 39, 41 i 48 (slika 9).

Na slici 9 prikazani su histogrami koji prikazuju razdiobu dobrote rješenja za četiri odabrana (najbolja) skupa parametara. Histogrami upućuju na zaključak da za te skupove parametara algoritam konzistentno pronalazi rješenja bliska optimalnom. Iz postupka odabira parametara vidljivo je da relativno visoka vjerojatnost mutacije (0,36 – 0,45) rezultira kvalitetnijim rješenjima. Također, utjecaj broja generacija na kvalitetu rješenja je pozitivan, s tim da se u ovisnosti o veličini problema smanjuje pozitivan utjecaj broja iteracija na kvalitetu rješenja (broj iteracija u rasponu od 444 - 914 davao je najbolje rezultate). Veličina populacije od 50 do 92 dala je najbolje rezultate, prema čemu je moguće zaključiti da je utjecaj veličine populacije također pozitivan, no ograničen veličinom problema.

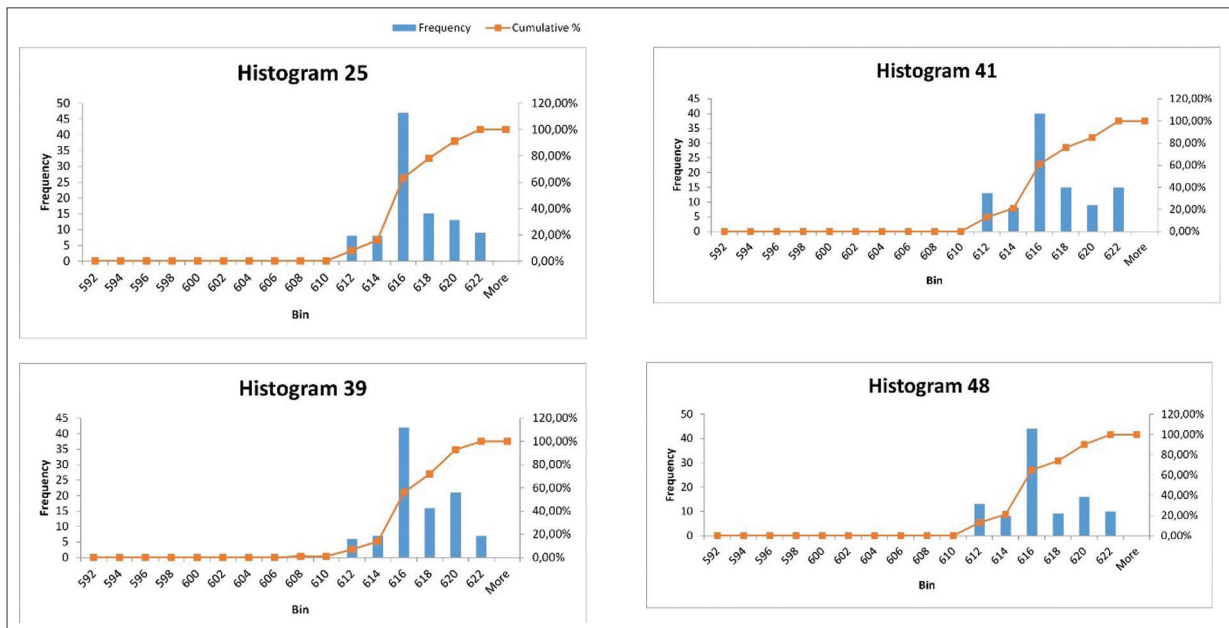
Za sva četiri skupa parametara dobiveno najbolje rješenje je ruta obilaska:

1 → 13 → 15 → 14 → 12 → 11 → 10 → 8
 → 7 → 4 → 5 → 3 → 2 → 6 → 9 → 1

duljine 228,75 jedinica (slika 10).



Slika 10 Prikaz najbolje rute dobivene GA
 Figure 10 Representation of best route obtained by GA



Slika 9 Prikaz histograma za četiri najbolja skupa parametara GA
 Figure 9 Histogram representation for the four best sets of GA parameters

5. Zaključak

5. Conclusion

Istraživanje je potvrdilo pretpostavku da predloženi GA (neovisno o parametrima), u realnom vremenu uspješno optimizira putanju manipulatora te redovito nalazi bolje rješenje, odnosno kraću putanju u usporedbi s postojećim modelom odabira putanje manipulatora. U promatranom je slučaju ušteda u duljini puta (što je proporcionalno utrošku vremena i energije) približno 10%.

Iz postupka odabira parametara vidljivo je da relativno visoka vjerojatnost mutacije (0,36 – 0,45) rezultira kvalitetnijim rješenjima. Također, utjecaj povećanog broja generacija na kvalitetu rješenja je pozitivan. I veličina populacije utječe pozitivno, no ograničena je veličinom zadanog problema.

Iako je postupkom odabira vrijednosti parametara uz višestruko pokretanje GA postignuto relativno često pronalaženje

rješenja bliskih optimalnom, uz proizvoljno odabrane parametre GA teško dolazi do globalnog optimuma. Rezultati sugeriraju da bi se za povećanje efikasnosti i konzistentnost pronalaženja dobrih rješenja, ovakav GA trebalo kombinirati (hibridizirati) nekom od metoda lokalnog pretraživanja prostora rješenja.

Istraživanje je pokazalo i da se optimizacija rute manipulatora ne može svesti samo na optimizaciju duljine puta. Za postizanje značajnijih ušteda potrebno je provesti optimizaciju cjelokupnog sustava dobave proizvoda iz odjeljaka.

Kako je problem trgovačkog putnika NP-potpun problem, a GA se pokazao kao prikladan za rješavanje toga problema, slijedi da je GA prikladan za rješavanje srodnih problema iz klase NP-potpunih problema, pod uvjetom da je moguća reprezentacija problema u skladu s GA.

Istraživanja vezana uz hibridizaciju i rješavanje problema s većim brojem lokacija u putanji manipulatora su u tijeku.

6. Reference

6. References

- [1] Bosančić, V.; Golemac, A.; Vojković, T.; Kako pomoći trgovačkom putniku; Osječki matematički list, Vol. 12., No. 2; ISSN: 1845-4607; e-ISSN: 1848-946X; pp. 139 – 149; 2013.
- [2] Held, M.; Hoffman, A.J.; Johnson, E.L.; Wolfe, P.; Aspects of the traveling salesman problem; IBM Journal of Research and Development, Vol 28., No. 4; pp. 476-486, 1984.
- [3] Čupić, M.; Prirodom inspirirani optimizacijski algoritmi; Skripta, Verzija: 1.0.7.; Fakultet elektrotehnike i računarstva, Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave; p.1.; Zagreb, 2010.
- [4] <http://160592857366.free.fr/joe/ebooks/ShareData/Heuristics%20for%20the%20Traveling%20Salesman%20Problem%20By%20Christian%20Nillson.pdf>; 30.10.2016.
- [5] Dvornik, J.; Numeričke, simboličke i heurističke metode; Građevinar, Vol. 55., No. 10; ISSN: 0350-2465; e-ISSN: 1333-9095; pp. 575 – 582; 2003.
- [6] Sivanandam, S.N.; Deepa, S.N.; Introduction to Genetic Algorithms; ISBN: 978-3-540-73189-4; Springer Berlin Heidelberg, 2008.
- [7] Goldberg D. E.; Genetic Algorithms in Search, Optimization & Machine Learning; Addison Wesley Publishing Company, Inc.; ISBN: 0-201-15767-5; Boston, 1989.
- [8] Matai, R.; Singh, S.; Mittal, M.L.; Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches; Traveling Salesman Problem, Theory and Applications; ISBN: 978-953-307-426-9; Chapter 1; Intech, Rijeka, 2010.
- [9] Golub, M.; Genetski algoritam, Prvi dio; Skripta, Verzija 2.3.; Fakultet elektrotehnike i računarstva, Zagreb, 2010.
- [10] Larrañaga, P.; Kuijperes, C.M.H.; Murga, R.H.; Inza, I.; Dizdarevic, S.; Genetic Algorithms for the Travelling Salesman Problem; Artificial Intelligence Review, Vol 13, No. 2; ISSN: 0269-2821; e-ISSN: 1573-7462; pp. 129-170; 1999.
- [11] Gen, M.; Cheng, R.; Genetic Algorithms and Engineering Design; ISBN: 978-0-471-12741-3; Ashikaga Institute of Technology, Ashikaga, Japan, 1997.

- [12] Banzhaf, W.; The “Molecular” Traveling Salesman; *Biological Cybernetics*, Vol. 64, No. 1; pp. 7-14; 1990.
- [13] Xie, H.; Zhang, Z.; Tuning Selection Pressure in Tournament Selection; <https://ecs.victoria.ac.nz/foswiki/pub/Main/Technical-ReportSeries/ECSTR09-10.pdf>
- [14] Golub, M.; Genetski algoritam, Drugi dio; Skripta, Verzija 2.2.; Fakultet elektrotehnike i računarstva, Zagreb, 2010.

AUTORI · AUTHORS



Hrvoje Rakić

Hrvoje Rakić rođen je 11. svibnja 1981. godine u Koprivnici, gdje je završio i prirodoslovno-matematičku gimnaziju. Diplomirao je na Fakultetu strojarstva i

brodogradnje Sveučilišta u Zagrebu, usmjerenje Zavarene konstrukcije, na temu „Tehnološka svojstva temeljnih radioničkih premaza“. Asistent je na Strojarskom odjelu Tehničkog veleučilišta u Zagrebu i izvodi nastavu na kolegijima Preddiplomskih stručnih studija mehatronike („Održavanje tehničkih sustava u mehatronici“, „Upravljanje proizvodnjom i projektima“ i „Matlab“) i elektrotehnike („Održavanje elektrotehničke opreme“) te Specijalističkog diplomskog stručnog studija informatike („Upravljanje i vođenje projekata“). Interni je auditor TVZ-a po ESG normi za visoko obrazovanje i po ISO 9001 normi te „SAP Certified Associate – Business Process Integration with SAP ERP 6.0 EHP6“ (TERP10). Član je tehničkog odbora Hrvatskog zavoda za norme (HZN) - TO 521: Usluge održavanja i Hrvatskog društva održavatelja (HDO).

Korespodencija

hrvoje.rakic@tvz.hr



Tomislav Brajković

Tomislav Brajković rođen je 28. ožujka 1983. godine u Zagrebu. Osnovnu školu braće Radića pohađao je u Kloštar Ivaniću, a nakon završetka Srednje škole Ivan Švear u Ivanić Gradu

upisuje Fakultet strojarstva i brodogradnje u Zagrebu. Diplomirao je 2009. godine, a naslov diplomskog rada bio je Utjecaj kemijske obrade na korozivnu postojanost Cr-Ni čelika. Trenutno je zaposlen u tvrtki KEL d.o.o. na poslovima nadzora i projektiranja.

Korespodencija

kel@zg.t-com.hr



Slobodan Obrenović

Slobodan Obrenović rođen je 27. prosinca 1976. godine u Zagrebu. Nakon završene XV. Gimnazije (International Baccalaureate) upisuje Fakultet Elektrotehnike i Računarstva.

Diplomirao je 2001. godine i od tada sudjeluje na projektima iz područja optimizacije, strojnog učenja, bioinformatike i robotike. Od 2015. godine radi na projektu analize i interpretacije podataka dobivenih sekvenciranjem DNA.

Korespodencija

slobodan@inovatus-usluge.hr