II MEĐUNARODNI SIMPOZIJUM "INDUSTRIJSKO INŽENJERSTVO"
SIE'98
2nd INTERNATIONAL SYMPOSIUM OF INDUSTRIAL ENGINEERING

B2. - 10.

Neke metode za rešavanje problema trgovačkog putnika pomoću genetskih algoritama

mr Jozef Kratica[1]
Ivana Ljubić[2]
Vesna Šešum[1]
mr Vladimir Filipović[3]

REZIME

U ovom radu je opisano nekoliko novijih pristupa rešavanju problema trgovačkog putnika pomoću genetskih algoritama. Uporedno su opisane osnovne karakteristike datih algoritama, kao i njihovi rezultati u praksi.

Ključne reči: *Genetski algoritmi, problem trgovačkog putnika, kombinatrona optimizacija*

1. UVOD

1.1 Formulacija problema

Neka je dat povezan graf of N čvorova, gde je poznata dužina svake od grana. Potrebno je naći najkraći put koji prolazi kroz sve čvorove grafa tačno jednom, i završava se u početnom čvoru.

Problem trgovačkog putnika (TSP) se može matematički formulisati na sledeći način, tako da je potrebno naći:

$$\min\left(d(p_N, p_1) + \sum_{i=1}^{N-1} d(p_i, p_{i+1}) \right) \qquad (1)$$

ako je **d(k,l)** rastojanje između čvorova k i l, ako postoji grana koja spaja date čvorove. U suprotnom se za rastojanje uzima vrednost $+\infty$. Permutacija p označava kojim se redosledom prolazi kroz čvorove, odnosno p_i označava čvor grafa koji obilazimo u i-tom koraku.

U ovom radu su opisane metode za rešavanje simetričnog TSP-a, gde je matrica rastojanja d simetrična, odnosno **d(k,l) = d(l,k)**.

1.2 Načini rešavanja datog problema

TSP je klasičan primer jednog NP-teškog problema, i predstavlja jedan od najčešće rešavanih problema iz te klase. Takvi problemi su vrlo značajni, jer nalaženje optimalnog rešenja zahteva pretragu po pretraživačkom prostoru koji raste eksponencijalno u zavisnosti od veličine problema. Zbog toga je od posebnog značaja razvoj heurističkih metoda, koje daju rešenje što bliže optimalnom, uz istovremeno, relativno kratko vreme izvršavanja.

Postoji izuzetno veliki broj metoda za rešavanje TSP-a, pa opis svih metoda izlazi izvan okvira ovog rada. U [7] je opisan istorijat i pregled date oblasti do sredine 80-tih godina, a u [2] i [5] se mogu naći neki od najnovijih rezultata.

2. METODE REŠAVANJA POMOĆU GA

Ubrzo posle pojave genetskih algoritama (početkom 70-tih) počela je i njihova intenzivna primena za rešavanje problema kombinatorne optimizacije, a posebno NP-teških problema. Već krajem 80-tih godina je u [3] zabeležen veći broj GA implementacija za rešavanje TSP. I pored primene raznih vrsta genetskih operatora (PMX, OX, ...), rezultati su lošiji u poređenju sa ostalim metodama (simulirano kaljenje, tabu pretraživanje, ...).

Veliki pomak u poboljšanju performansi GA za rešavanje TSP-a je nastao paralelnom implementacijom opisanom u radu [9]. Detaljan opis ove implementacije i kasnija poboljšanja date strategije se mogu videti u [5]. Mi ćemo se zadržati na nekoliko novijih GA implementacija.

[1] Katedra za matematiku, Mašinski fakultet u Beogradu, 27. marta 80
[2] Katedra za matematiku, Građevinski fakultet u Beogradu, Bulevar revolucije 73
[3] Matematički fakultet u Beogradu, Studentski trg 16

2.1 E-2, E-3 i MPX operatori

U radovima [1] i [8] je opisano nekoliko genetskih operatora sa obećavajućim rezultatima. Koristi se permutacijska reprezentacija čvorova grafa, odnosno jedinka se sastoji od niza rednih brojeva čvorova u permutaciji *p*. Prednost date reprezentacije se ogleda u kompaktnom zapisu, a mana je nepogodan, za primenu genetskog algoritma, način predstavljanja ivica grafa.

E-2 i E-3 operatori ukrštanja se u najvećoj meri baziraju na korišćenju ivica koje pripadaju genetskim kodovima oba roditelja. Formira se lista korišćenih ivica koje sadrže kodovi roditelja, i potomak se pri tome dobija takvom rekombinacijom, da mu što je moguće veći broj ivica pripada datoj listi.

U slučaju E-2 operatora (Edge recombination- 2) potomak se formira na sledeći način:

- Početni čvor biramo na slučajan način;
- Iz liste korišćenih ivica se izbacuju sve pojave početnog čvora, a kasnije i svakog novog dodatog čvora;
- Sledeće čvorove dobijamo pretragom liste korišćenih ivica prema sledećem prioritetu:
  1) Čvorovi čije se odgovarajuće ivice javljaju u oba roditelja,
  2) Čvorovi sa najmanjim brojem ivica iz liste,
  3) Ukoliko ne postoji nijedna ivica u listi, dati čvor je blokiran, pa se sledeći čvor bira na slučajan način;

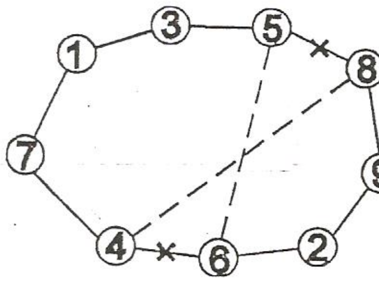
Primer 1. Neka su genetski kodovi roditelja **2 3 1 4 5** i **3 1 4 2 5**, tada je lista korišćenih ivica:
1: *3*,4,5 2: *5*,3,4 3: *1*, 2,5 4: *1*, 2,5 5: *2*,3,4
Ako za početni čvor izaberemo 2, sledeći je 5 jer se pojavljuje u oba roditelja. Sada ravnopravno učestvuju čvorovi 3 i 4. Ako izaberemo 3, sledeći čvorovi moraju biti redom 1 i 4 (jer su jedini preostali u listama za čvor 3 i čvor 1), pa je potomak **2 5 3 1 4**.

U slučaju operatora ukrštanja E-3 potomak se dobija na sličan način, pri čemu ako je čvor blokiran, pokušavamo da put trgovačkog putnika (permutaciju) proširimo na drugom kraju. Ukoliko su oba kraja blokirana, tek tada sledeći čvor biramo na slučajan način.

Na sličan način se formira potomak i kod MPX operatora (Maximal Preservative Crossover), tako da bude sličan roditeljima u smislu korišćenih ivica. MPX operator formira potomka direktnim kopiranjem segmenta genetskog koda prvog roditelja, a zatim popunjavanjem nedostajućih čvorova kopiranjem od drugog roditelja. Pri tome postoje pravila kako se izbegavaju čvorovi koji već postoje u genetskom kodu potomka.

Kod implementacije GA korišćenjem jednog od gorepomenutih operatora ukrštanja, hibridizacija se vrši korišćenjem *2-Opt* i *2-Popravi* heuristika. U slučaju *2-Opt* heuristike ispituje se mogućnost poboljšanja rešenja brisanjem 2 ivice i ponovnim spajanjem prekinutih delova pomoću nove 2 ivice. Ovo je moguće samo u slučaju simetričnog TSP, jer je potrebno promeniti orjentaciju jednog od delova pri ponovnom spajanju. Kod *2-Opt* heuristike se data procedura primenjuje na sve parove ivica, dok se u *2-Popravi* to izvodi samo za ivice u kojima su pri primeni E-2 odnosno E-3 operatora čvorovi bili blokirani, pa je izbor sledećeg čvora vršen na slučajan način. Kod obe heuristike su moguće podvarijante, tako da se primenjuju jednostruko ili višestruko. U slučaju višestruke primene, postupak se ponavlja za novodobijeni put, sve dok postoje takvi parovi ivica kod kojih je moguće poboljšanje.

Primer 2. Ako je put trgovačkog putnika *1 3 5 8 9 2 6 4 7* i brišemo ivice (*5*,*8*) i (*6*,*4*) tada će nove ivice biti (*5*,*6*) i (*8*,*4*), koje daju put *1 3 5 6 2 9 8 4 7* kao što možemo videti na slici 1.



Slika 1. Primer zamene kod *2-Opt* heuristike

Primena *2-Opt* heuristike je vremenski vrlo zahtevna operacija, pogotovo u slučaju višestruke primene. Zbog toga se u datoj implementaciji ([1] i [8]), jednostuko primenjuje *2-Opt* heuristika samo na poboljšavanje početne populacije generisane na slučajan način, a u kasnijim generacijama se poboljšavanje vrši višestrukom primenom *2-Popravi* heuristike.

U praksi je data implementacija dala zadovoljavajuće rezultate, tako da su za test primere sa nekoliko stotina čvorova (na primer Padberg-ov problem sa 532 čvora), u dostižnom vremenu dobijena ili optimalna rešenja ili suboptimalna rešenja bliska optimalnom. Primenom datih operatora (E-2, E-3 i MPX) na konkretnim primerima, najbolje rezultate su dali E-3 i MPX operatori. U nekim slučajevima je bio bolji E-3 a u drugim MPX, pa bi kombinovanje ta dva operatora možda dalo još bolje rezultate.

2.2 MX operator

U radu [4] se primenjuje potpuno novi pristup rešavanju TSP-a uz pomoć GA. Analizom prethodnih implementacija, u radu se predlaže da osnovni gradivni blokovi za TSP budu ivice, a ne fiksne pozicije čvorova. To se generalno obrazlaže činjenicom, da pomeranje čvora na određenu poziciju, bez korišćenja dodatnih informacija o ivicama, vrlo malo utiče na kvalitet rešenja. Iako se već i u ranijim radovima vidi tendencija što većeg korišćenja ivica, na račun čvorova ([1], [5] i [9]), u datom radu je prelaz potpun.

Iako se zbog male dužine zapisa i dalje formalno koristi permutacijska reprezentacija za memorisanje argumenata problema, stvarno se barata nad matričnom reprezentacijom ivica grafa. Vrednost elementa matrice je *1*, ako data ivica postoji u putu trgovačkog putnika (permutaciji), a u suprotnom je *0*. Primetimo da u svakoj vrsti odnosno koloni date matrice postoji tačno po jedna *1*, a ostalo su *0*. U tabeli 1. možemo videti primer matrične reprezentacije za put *1 5 3 4 2*.

Tabela 1. Primer matrične reprezentacije

	1	*2*	*3*	*4*	*5*
1	0	0	0	0	1
2	1	0	0	0	0
3	0	0	0	1	0
4	0	1	0	0	0
5	0	0	1	0	0

MX operator ukrštanja (Matrix Crossover) je sličan običnom jednopozicionom odnosno dvopozicionom operatoru ukrštanja (one-point crossover, two-point crossover), samo se barata nad kolonama matrice. Dakle, na slučajan način se bira jedna ili dve pozicije za ukrštanje. Tada se vrši razmena sadržaja kolona između datih pozicija. Na primer ako je pozicija za ukrštanje samo *3*, vrši se ukrštanje analogno jednopozicionom, pa se sadržaj 4. i 5. kolone razmenjuje izmežu roditelja. Ukoliko su date dve pozicije za ukrštanje, recimo *1* i *3*, tada se vrši razmena sadržaja 2. i 3. kolone.

Posle ovoga je moguće da postoje vrste u kojima ima po dve ili nema nijedna *1*. To razrešavamo pomeranjem jedne *1* iz vrsta gde ih je dve, u vrste gde ih nema nijedna. Pri tome forsiramo takva pomeranja koja čuvaju ivice, koje su postojale kod roditeljskih jedinki.

I pored toga što sada u svakoj vrsti i koloni imamo tačno jednu *1*, to ne garantuje da to predstavlja permutaciju, jer su mogući ciklusi. U tom slučaju raskidamo po jednu ivicu u svakom ciklusu, i povezujemo ih tako, da formiraju permutaciju, i da je što više novoformiranih ivica pripadalo roditeljskim jedinkama.

Primer 3. Matrica u tabeli 2. ne predstavlja permutaciju već samo dva odvojena ciklusa *3 2 5* i *1 4*. Tada je potrebno raskinuti jednu od veza *3-2*, *2-5* ili *5-3* i jednu od veza *1-4* ili *4-1*, i dodati dve veze između ciklusa recimo *2-1* i *4-5*. Tada smo dobili permutaciju *3 2 1 4 5*.

Tabela 2. Matrica koja ne predstavlja permutaciju

	1	*2*	*3*	*4*	*5*
1	0	0	0	1	0
2	0	0	0	0	1
3	0	0	0	1	0
4	1	0	0	0	0
5	0	0	1	0	0

U datom radu je primenjena turnirska selekcija, da bi se izbegli problemi preuranjene konvergencije, i sačuvala raznovrsnost genetskog materijala.

Mutacija se za razliku od ukrštanja primenjuje direktno na permutacijskoj, a ne na matričnoj reprezentaciji. Pri tome se na slučajan način biraju dve pozicije u permutaciji, i na slučajan način permutuju čvorovi između datih pozicija. Dužina mutiranog segmenta mora biti mala, da bi se izbegle prevelike promene u originalnim genetskim kodovima jedinki.

U svakoj generaciji se takođe vrši poboljšavanje svake od jedinki jednom od varijanti *2-Opt* heuristike.

U datom radu su takođe dobijeni vrlo dobri rezultati, tako da je, kao i kod metoda opisanih u prethodnom odeljku dobijeno rešenje optimalno ili blisko optimalnom, uz relativno kratko vreme izvršavanja. Najveći primer na kome je algoritam testiran je poznati Padberg-ov problem sa 318 čvorova. S obzirom na potencijal date metode, koja je u osnovnim delovima konceptualno potpuno nova, i drugačija od ostalih metoda, može se očekivati u budućnosti poboljšavanje date metode i kombinovanje sa drugim metodama rešavanja pomoću GA.

2.3 Poboljšani GX operator

U radu [6] je opisana jedno poboljšanje GX operatora ukrštanja (Greedy Crossover).

GX operator vrši ukrštanje, tako što se na slučajan način određuje jedan čvor koji postaje početni čvor za potomka. U svakom koraku se bira jedan od desnih suseda tekućeg čvora u oba roditelja. Ukoliko su susedni čvorovi u oba roditelja slobodni (neiskorišćeni u prethodnoj proceduri formiranja puta trgovačkog putnika), bira se onaj sa manjim rastojanjem. Ako je samo jedan sused slobodan, on postaje tekući čvor, a ako nema slobodnih suseda, bira se proizvoljni slobodni čvor.

U praksi je jedna od velikih mana ovog pristupa prilično veliki procenat tekućih čvorova koji nemaju slobodnih suseda, koji u nekim slučajevima dostiže i do 40%. To znači da je veliki deo genetskog koda potomaka generisan praktično na slučajan način, umesto da bude nasleđen od roditelja.

U modifikovanom GX operatoru se u potomku naizmenično dodaju čvorovi iz prvog i drugog roditelja. U svakom koraku se posmatraju levi i desni sused tekućeg čvora, ali samo kod jednog od roditelja (koji se naizmenično smenjuju). Kao i kod osnovne varijante GX operatora, ako su oba suseda slobodna bira se onaj sa manjim rastojanjem, ako je samo jedan slobodan on se bira, a u suprotnom bira se proizvoljni slobodni čvor. Primenom ove modifikacije se smanjuje procenat tekućih čvorova koji nemaju slobodnog suseda, što se rezultuje boljim performansama u odnosu na osnovni algoritam.

Protiv moguće preuranjene konvergencije i smanjene raznovrsnosti genetskog materijala primenjuje se poseban postupak. Kada posle neke generacije, jedinke u populaciji postanu suviše slične, odnosno više jedinki ima istu vrednost, jedna od datih sličnih jedinki se ostavlja, a kod ostalih se vrši potpuno ili delimično slučajno permutovanje čvorova. Na taj način se očuvava raznovrsnost genetskog materijala, uz istovremeni uslov da kvalitetne jedinke ostaju u populaciji.

U radu ([6]) su dati rezultati izvršavanja za Oliver-ov problem sa 30 čvorova i Eilon-ove probleme sa 50 i 75 čvorova. Iako su rezultati ove metode prilično lošiji u odnosu na metode opisane u prethodnim odeljcima, ideja je zanimljiva i postoji mogućnost primene u kombinaciji sa drugim metodama.

3. ZAKLJUČAK

Iako generalno GA po svojoj prirodi nisu baš najbolje prilagođeni za probleme kod kojih je bitan poredak (ordering problems), kojima pripada i TSP, opisane metode, daju vrlo dobre rezultate. Jedini je izuzetak poslednja opisana metoda, koja generalno daje slabije rezultate.

Zbog ograničenog obima, u ovom radu nisu opisane osnove GA (videti [3]), kao i pregled ostalih metoda za rešavanje TSP-a (videti [5]). I pored toga ovaj rad daje dobar pregled primene GA za rešavanje TSP-a, kao i neke od novijih rezultata u ovoj oblasti.

LITERATURA

[1] **Dzubera J., Whitley D.**, "Advanced Corelation Analysis of Operators for the Traveling Salesman Problem", In: *Parallel Problem Solving from Nature - PPSN III*, eds: Davidor Y., Schwefel H.P., Manner R., Springer-Verlag, pp. 68-77 (1994).

[2] **Fredman M.L., Johnson D.S., McGeoch L.A., Ostheimer G.**, "Data Structures for Traveling Salesman", *Journal of Algorithms*, Vol. 18, pp. 432-479 (1995).

[3] **Goldberg D.E.**, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Publ. Comp., Reading, Mass. (1989).

[4] **Homaifar A., Guan S., Liepins G.E.**, "A New Approach on the Traveling Salesman Problem by Genetic Algorithms", In: *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, Calif., pp. 460-466 (1993).

[5] **Johnson D.S., McGeoch L.A**, "The Traveling Salesman Problem: A Case Study in Local Optimization", In: *Local Search in Combinatorial Optimization*, eds: Aarts E.H.L. and Lenstra J.K., John Wiley & Sons Ltd., pp. 215-310 (1997).

[6] **Kureichick V.M., Melikhov A.N., Miagkikh V.V., Savelev O.V., Topchy A.P.**, "Some New Features in Genetic Solution of the Traveling Salesman Problem", In: *Proceedings of ACEDC '96*, University of Plymouth, UK (1996).

[7] **Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmous D.B.**, "The Traveling Salesman Problem", John Wiley & Sons (1985).

[8] **Mathias K., Whitley D.**, "Genetic operators, the fitness landscape and the traveling salesman problem", In: *Parallel Problem Solving from Nature - PPSN 2*, eds: Manner R. and Manderick B., North Holland - Elsevier, pp. 219-228 (1992).

[9] **Mühlenbein H., Gorges-Schleuter M., Krämer O.**, "Evolution algorithms in combinatorial optimization", *Parallel Computing*, Vol. 7, pp. 65-85 (1988).

Some Methods for Solving the Traveling Salesman Problem by Genetic Algorithms

ABSTRACT

In this paper are described several new methods for solving traveling salesman problem (TSP) by genetic algorithms (GAs). Comparative analysis of those methods are given, with analysis of their results in practice.

Keywords: *Genetic algorithms, traveling salesman problem, combinatorial optimization*