Rešavanje problema minimalnog k-trgovačkog putnika

Marica Bogićević, Marina Brkić

Januar 2019

Sažetak

Ovaj rad se bavi primenom genetskih algoritama na problem višestrukog trgovačkog putnika MTSP. U radu je implementiran genetski algoritam u programskom jeziku Python.

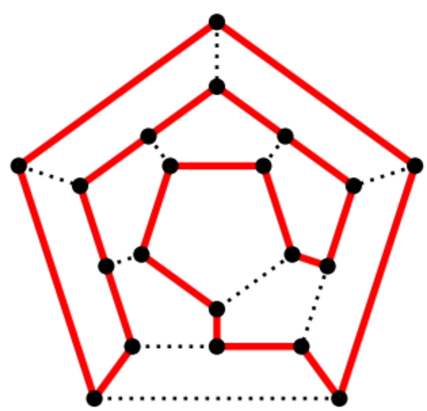
Sadržaj

1. Uvod

1.1 Problem trgovačkog putnika

1.1.1 Istorija

Problem trgovačkog putnika (eng. Travelling Salesman Problem – TSP) je klasičan problem u polju diskretne i kombinatorne optimizacije. Spada u grupu NP-teških problema a složenost mu je O(n!). Matematičke probleme slične njemu prvi je razmatrao Euler, koji se bavio pitanjem kako bi skakač na šahovskoj tabli posetio svih 64 mesta samo jednom. Početkom 20. veka matematičari William Rowan Hamilton i Thomas Kirkman su razmatrali probleme koji se svode na problem trgovačkog putnika. Hamilton je izmislio Ikozijansku igru u kojoj je cilj da se nadje zatvorena putanja ivice dodekaedra i da se svako njegovo teme pojavi tačno jednom u putanji. U današnjoj terminologiji to bi znacilo da treba da se nađe Hamiltonova putanja u grafu kome su čvorovi temena, a ivice su ivice dodekaedra.



Slika 1: Ikozijanska igra

Njegova opšta forma se pojavljuje 30-tih godina 20. veka. Pojam "trgovački putnik" prvi put je upotrebljen 1932. godine od strane Karl Mengera koji je u svom radu spomenuo brute-force algoritam i definisao TSP onako ga danas definišemo. Problem je vremenom postajao sve više popularan i izučavan od strane mnogih naučnika.

Problem trgovačkog putnika (eng. Travelling Salesman Problem – TSP) možemo neformalno da formulišemo na sledeći način: Dato je n gradova i poznate su sve udaljenosti između njih. Trgovački putnik treba da obiđe sve gradove i da se na kraju vrati u grad odakle je i krenuo, ali da pri tome razdaljina koju pređe bude najkraća. Formalna definicija problema trgovačkog putnika [Cormen et al., 2001] glasi:

Definicija 1.1 (Problem Trgovačkog putnika TSP) Ako je dat kompletan neusmeren težinki graf, sa nenegativnim celobrojnim težinama, naći Hamiltonovu putanju sa najmanjom težinom.

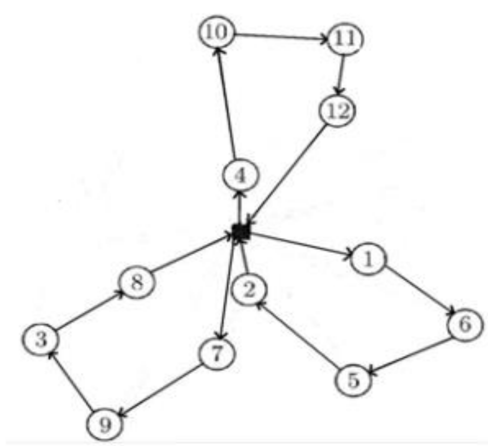
Osim optimizacijske varijante ovog problema, postoji i tzv. Forma problema odlučivanja:

Definicija 1.2 (TSP -odlučivanje) Ako je dat kompletan težinski graf i pozitivan realan broj *L*, treba odrediti da li postoji Hamiltonov put kraći od *L*.

Mi ćemo se u nastavku baviti problemom k-minimalnog trgovačkog putnik.
Problem višestrukog trgovačkog putnika (MTSP) uključuje raspoređivanje m>1 trgovaca da posete skup n>m čvorova tako da svaki čvor posete tačno jednom. Cilj je smanjiti ukupnu udaljenost koju su prešli svi prodavači. MTSP je primer kombinatorno optimizacionih problema, te ima mnoštvo primena, uglavnom u područjima usmjeravanja i raspoređivanja. U ovom radu, predložen je modifikovani hibridni metaheuristički algoritam pod nazivom GA2OPT za rešavanje MTSP-a. U ovom algoritmu, u prvoj fazi, MTSP je rešen modifikovanim genetskim algoritmom (GA) u svakoj iteraciji, a u drugoj fazi se koristi algoritam lokalnog pretraživanja 2-opt za poboljšanje rešenja za tu iteraciju.

2. Problem k-minimalnog trgovačkog putnika

Problem višestrukog trgovačkog putnika (MTSP) je generalizacija poznatog problema trgovačkog putnika (TSP), gde se može koristiti više od jednog trgovca u rešenju. Primeri MTSP-a: problem usmeravanja školskog autobusa, problem isporuke... Stoga je pronalaženje učinkovitog algoritma za MTSP važno i indicira na poboljšanje rešenja drugih složenih problema usmeravanja.
MTSP se može uopšteno definisati : Dato nam je n > 1 čvorova, neka su m trgovaca smešteni na jedan početni čvor. Preostali čvorovi koji će biti posećeni kasnije, nazivaju se posredni čvorovi. MTSP treba da pronadje putanje za sve m trgovce, koji svi počinju i završavaju se na početnom čvoru, tako da je svaki čvor, bez početnog, posećen tačno jednom i ukupni trošak poseta svim čvorovima je minimizovan.
U MTSP-u mora biti n čvorova podeljen u m tura, pri čemu je svaka tura rezultirala TSP-om za jednog prodavača. MTSP je teži od TSP jer zahteva dodeljivanje čvorova svakom prodavaču, kao i optimalno uređenje čvorova unutar svakog obilaska prodavača.



Slika 2: Primer rešenja MTSP-a

3. Algoritmi za rešavanje

Tehnike koje se koriste za rešavanje MTSP-a mogu biti kategorizovane u egzaktne, heurističke i metaheurističke algoritme.

Egzaktni pristupi za rešavanje MTSP-a se uspešno koristi samo za relativno male probleme, ali garantuju optimalnost na osnovu različitih tehnika. Ove tehnike primenjuju algoritme koji generišu i donju i gornju granicu na stvarnoj minimalnoj vrednosti instance problema. Ako se gornja i donja granica podudaraju, dokaz optimalnosti je postignut. U literaturi postoji mnogo studija koje su predlagale egzaktne algoritme za rešavanje MTSP-a. Ovi algoritmi su baziran na Lagranžovom algoritmu relaksacije, metodi grananja i rezanja (branch-and-cut method), itd. Iako je MTSP konceptualno jednostavan, teško je dobiti optimalno rešenje. Drugim rečima, kada se poveća veličina problema, egzaktne metode ne mogu ga rešiti. Dakle, heurističke ili metaheurističke metode su potrebne za njegovo rešavanje u razumnom vremenu s velikim veličinama. Neki od poznatih heurističkih algoritama su gravitational emulation search, lokalno pretraživanje, i Lin-Kernighan.

Novonastali algoritam u osnovi pokušava kombinovati osnovne heurističke metode u viši nivo s ciljem učinkovitijeg i delotvornijeg istraživanja prostora za pretraživanje u posljednjih 30 godina. Ove metode se danas obično nazivaju metaheuristike. Termin metaheuristika prvi put se pojavio u knjizi Glover, F. (1986). „Future Paths for Integer Programming and Links to Artificial Intelligence”. *Computers and Operations Research* . Sastoji se od dve grčke reči *„heuristic*“ koja proizilazi iz glagola *heuriskein* koji znači "pronaći", dok meta prefiks znači "izvan u gornjem nivou". Pre nego što je ovaj pojam široko usvojen, metaheuristike su se često nazivale moderne heuristike. Uopšteno, vrlo je bitno koristiti metaheurističke algoritme pri rešavanju složenih problema optimizacije. Pošto su metaheuristički pristupi vrlo učinkoviti u zaobilaženju lokalnog optimuma, oni su jedni od najboljih grupnih algoritama za rešavanje kombinatornih problema optimizacije. Zato se sve nedavne publikacije temelje na metaheurističkim pristupima kao što je genetski algoritam (GA) , memetički algoritam (MA), kolonija mrava (AS) i optimizacija rojeva (PSO) .

Dok se TSP smatra jednim od standardnih problema u operacionom istraživanju i u literaturi, MTSP još nije privukao veću pažnju. Dakle, u ovom radu se koristi modifikovani GA za rešavanje MTSP-a. Osim toga, 2-opt algoritam za lokalno pretraživanje se primjenjuje kako bi se povećala učinkovitost predloženog algoritma.

4. Matematički model

Neka je G (V , A) savršeni neusmereni povezani graf sa skupom čvorova V = {0,1, ...,} i skupom ivica A = {(i , j) : i, j ∈ V, i ≠ j}. Ako graf nije savršen, nedostatak neke ivice se zamjenjuje sa ivicom koja ima beskonačnu veličinu. Za prikazivanje celobrojnog linearnog programskog modela za MTSP, uvedene su sledeće promenljive:

n = broj čvorova za svaku instancu.
m = broj trgovaca koji se koristi za svaku instancu.
C = matrica troškova na grafu G je simetrična i tačna za nejednakost trougla. Znači da
C_{ij} = c_{ji} i C_{ij} +$C_{jk} \geq C_{ik}$ za svaki (i,j,k = 1, 2, ...,n) .

$$x_{ij} = \begin{cases} 1, akotrgovacdirektnoideodikaj \\ 0, inače \end{cases}$$

Dakle, jedna od zajedničkih formulacija celobrojnog programiranja za MTSP mogu se napisati na sledeći način:

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

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 2, \ldots, n \quad (2)$$

$$\sum_{i=1}^{n} x_{ij} = m \qquad j = 1 \quad (3)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 2, \ldots, n \quad (4)$$

$$\sum_{j=1}^{n} x_{ij} = m \qquad i = 1 \quad (5)$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1 \ (\phi \neq S \subset N = \{2, \ldots, n\}), |S| \geq 2 \quad (6)$$

$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1 \ (\phi \neq S \subset N = \{2, \ldots, n\}), |S| \geq 2 \quad (7)$$

$$x_{ij} \in \{0, 1\} \quad (8)$$

Objektivna funkcija (1) smanjuje ukupnu udaljenost proputovanih tura. Skupovi ograničenja (2) i (3) se staraju da trgovac poseti jednom svaki čvor i m puta početni čvor. Skupovi ograničenja (4) i (5) se staraju da trgovac ode iz svakog čvora jednom i da se vrati u početni čvor m puta. Ograničenja (6) i (7) su za izbegavanje prisutnosti pod-tura za svakog trgovca. Konačno, skup ograničenja (8) definiše binarna stanja promenljivih.

5. Genetski algoritami

5.1 Sta su genetski algoritmi i čemu služe

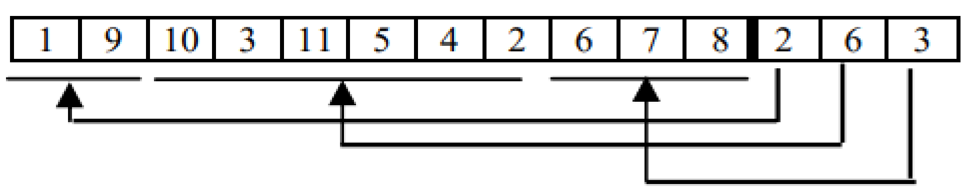
Genetski algoritmi (*Genetic Algorithms*, GA) su porodica algoritama koja je inspirisana Darvinovom teorijom evolucije. Tvorcem ove oblasti se smatra John Holland. Iako ne nalaze uvek optimalno rešenje, njihova prednost je što mogu da u razumnom vremenu nađu rešenje koje je svega 2-3% lošije od optimalnog.

Metaheuristički algoritmi, kao što su memetički algoritmi, optimizacija rojem čestica, tabu pretraga itd su uspešno primenjivani na mnoge teške optimizacione probleme koji uključuju problem trgovačkog putnika, problem usmeravanja vozila i ostale. U ovom poglavlju prvo je objašnjen Genetički algoritam (GA), a zatim je naš algoritam analiziran detaljnije.

Genetski algoritam je jedan od najstarijih metaheurističkih algoritama kojem su mnogo pažnje posvetili istraživači širom sveta. Algoritam se koristi za rešavanje kombinatorno optimizacionih problema zasnovanih na principima prirodne selekcije i genetike. GA krece od grupe inicijalnih rešenja koja se zovu inicijalna populacija. Zatim, koristi se fitnes funkcija koja predstavlja ocenu kvaliteta jedinke. Svaki put se izaberu dva rešenja iz populacije, koja se zovu roditelji, prema verovatnoći selekcije koja je proporcionalna njihovoj fitnes vrednosti. Roditelji se ukrštaju i dobijaju se dva nova rešenja u sledećoj generaciji. Nova rešenja će da zamene stara, ako imaju bolju fitnes funkciju.
Operacija mutacije se primenjuje na nova rešenja koja su zasnovana na verovatnoći mutacije.
Selekcija, ukrštanje i mutacija se ponavljaju da proizvedu nova rešenja dok veličina nove populacije ne bude jednaka staroj. Iteracija zatim kreće od nove populacije. Pošto nova rešenja imaju veću verovatnoću da budu izabrana za ukrštanje i prenose osobine roditelja, očekuje se da će nova generacija biti bolja od stare. Proces se nastavlja dok broj generacija ne stigne do n ili se kvalitet rešenja ne može lako poboljšati.

5.2 GA2OPT algoritam

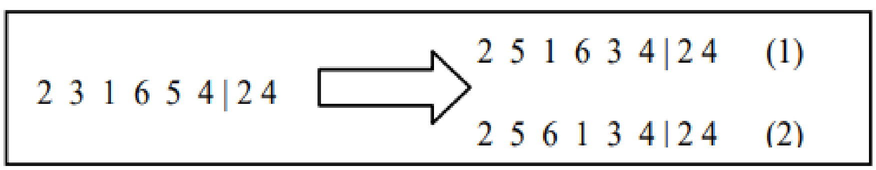
U GA2OPT algoritmu se samo jedna vrsta hromozoma obično koristi za rešavanje MTSP. Ova tehnika podrazumeva korišćenje jednog hromozoma dužine n + m i zove se tehnika "jednog hromozoma". U ovoj tehnici n čvorova su reprezentovani permutacijom celih brojeva od 1 do n. Ova permutacija je podeljena na m podtura umetanjem m negativnih celih brojeva od 1 do m koja predstavlja promenu od jednog trgovca do drugog. Primer ilustrovan na slici 3 pokazuje da bi prvi trgovac posetio čvorove 1 i 9 (tim redosledom), drugi trgovac bi posetio čvorove 10, 3, 11, 5, 4 i 2 (tim redosledom) i treći trgovac bi posetio 6, 7 i 8 čvor (tim redosledom).



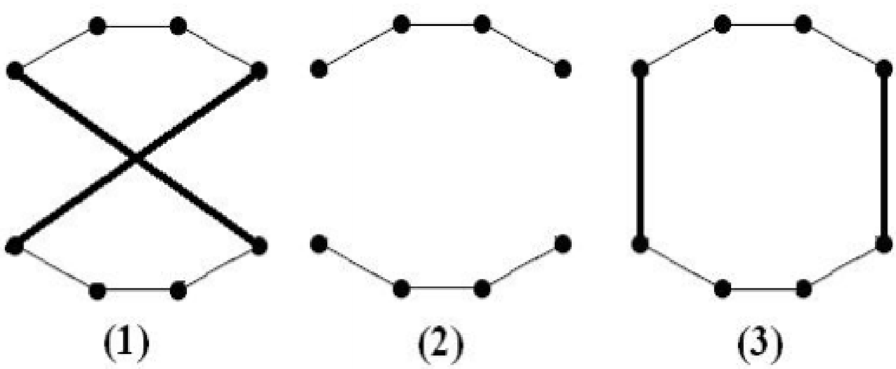
Slika 3: Reprezentacija hromozoma

Jedno od najboljih ukrštanja u smislu kvaliteta i brzine je uredjeno ukrštanje. Ovde je razmotren ovaj metod, koji se lako implementira. U ukrštanju se na slučajan način bira tačka prekida koja deli roditelja na dva podniza. Desni podniz roditelja je izabran. Nakon selekcije čvorova, proces je isti kao kod uredjenog ukrštanja. Jedina razlika je da umesto što se selektuje nekoliko slučajnih pozicija u roditelju sve pozicije desno od tačke koja je izabrana na slučajan način su selektovane. Očigledno je da ovaj metod dopušta samo generacije sa validnim stingovima.

U algoritmu se koriste dve mutacije. Ovi operatori slučajno biraju dve tačke u stringu i menjaju im mesta (Slika 4-1) ili obrće podniz izmedju ove dve tačke prekida. (Slika 4-2). Štavise, literatura o metaheuristici pokazuje da je obećavajući pristup za dobijanje kvalitetnih rešenja povezivanje lokalne pretrage kao što je 2-opt algoritam kada se traži bolje rešenje u poređenju sa prethodnim iteracijama. U suštini, verovatnoća pronalaženja boljeg rešenja koje je približno dobrom rešenju je relativno velika u ovakvim situacijama. 2-opt heuristika pokušava da unapredi rutu tako što zamenjuje dve nesusedne ivice sa druge dve ivice. (Slika 5). Treba zapamtiti da postoji nekoliko
ruta za povezivanje čvorova i kreiranje nove ture. Dakle, jedinstvena tura će biti prihvaćena
samo ako, prvo, ograničenja nisu narušena, drugo, nova tura predstavlja bolju vrednost za problem nego prethodno rešenje. Ovaj proces se ponavlja dok redukcija rute ne bude više moguća.



Slika 4. Mutacija



Slika 5: 2-opt algorita

Pseudo-kod za GA2OPT algoritam:

```
procedure GA2OPT algorithm
        S:= none;                                   // S is population of solutions //
        n= the number of nodes;
        if n is even then nn:=n else nn:=n+1;       // the number of chromosomes //
        l:= int[n/10] ;                             // l is the number of used mutations in each iteration //
        s* is the random solution;                  // s* is the best solution found yet //
        f* is value of s* ;                         // f* is the best value found yet //
            for i:= 1 to n do                       // main cycle //
                for j:= 1 to nn do
                    Construct a solution s_i as Fig. 2;
                    Find value of the s_i and call it f(s_i) ;
                    S=S∪s_i ;
                end
                for j := 1 to nn do
                    if j is odd then
                        begin
                        do crossover for s_j and s_{j+1} as Fig. 3 and called them s*_i and s*_{i+1} ;
                        if values of s*_j and s*_{j+1} are better than s_j and s_{j+1} respectively, then replaced them;
                        end
                    end
                end
                select l number chromosomes from S and do mutation for them;
                if new solutions based on mutations are better than before, then replace new solutions and their values;
                find the best solution and value of S and called s*_i and f(s*_i)
                    if f(s*_i) < f* then
                    begin
                        apply 2-opt local search to s*_i ;
                        f* := f(s*_i) ;
                        s* = s*_i ;
                    end // save the best so far solution //
        end
          show s* and f*
end // procedure //
```

6. Rezultati i tehnički detalji

Nekoliko rezultata je upoređeno i predstavljeno u ovom delu. Algoritam je kodiran u Pythonu i implementiran na virtuelnoj mašini sa operativnim sistemom Ubuntu. Algoritam će se zaustaviti nakon n iteracija. Da bi se otkrila varijabilnost performansi GA2OPT od jednog do drugog pokretanja, za svaku instancu sa različitim brojevima biće 10 pokretanja. U ovim testovima, efikasnost i performanse GA2OPT algoritma su upoređene sa nekim od najboljih implementiranih tehnika, kao što su modifikovani genetski algoritam (MGA) i Modifikovana kolonija mrava (MACO). Ovi algoritmi su primenjeni na nekoliko instanci iz TSP problema dostupne na TSPLIB uključujući Pr76, Pr152, Pr226,Pr299, Pr439 and Pr1002. Tabela 1 ilustruje karakteristike 6 instanci problema. Ovi problemi su u opsegu od n=76 do n=1002 čvora. Koristi se Euklidsko rastojanje i poredi se sa realnim brojevima. Kolone 2-6 pokazuju problem veličine n, broj trgovaca m, maksimalni broj gradova koji trgovac može da poseti l, broj pokretanja za svaku instancu tb, broj iteracija nakon kojih će predloženi algoritam da se zaustavi ako ne bude poboljšanja T. Takodje, da bi se videle performanse metode, najbolja rešenja koja su objavljena u literaturi i na mreži BSK, prikazana su u kolini 7.

The characteristics of the six problem instances

BKS	T	tb	l	M	n	Instance
157444	76	10	20	5	76	Pr76
127839	152	10	40	5	152	Pr152
166827	226	10	50	5	226	Pr226
82106	299	10	70	5	299	Pr299
161955	439	10	100	5	439	Pr439
382198	1002	10	220	5	1002	Pr1002

Tabela 1

U Tabeli 2 su upoređeni GA2OPT algoritam sa objavljenim podacima.

Poslednja kolona opisuje različite instance, dok kolone 4-5 prikazuju dva dobro poznata i najbolja objavljena rezultata dobijena korišćenjem metaheurističkih algoritama. Kolona 3 se odnosi na najbolji rezultat od predloženih metoda za ove instance. Kolona 2 predstavlja najbolji rezultat objavljen u literaturi i na internetu za ove instance. Kolona 1 prikazuje mean gap vrednosti. Gap je definisan kao procenat devijacije najboljeg poznatog rešenja iz literature. Drugim rečima, gap je jednak 100[c(s**)-c(s*)]/c(s*), gde je s** najbolje rešenja algoritma za ponuđene instance, a s* je najbolje rešenje za iste istance sa interneta. Rezultat poređenja pokazuje da GA2OPT ima sposobnost da pobegne od tački lokalnog optimuma i nađe najbolje rešenje za većinu instanci. Pored toga, predloženi algoritam daje bolja rešenja od MGA i MACO za neke od instanci. Preciznije, rezultat ovog upoređivanja pokazuje da predloženi algoritam ima lošija rešenja od MGA in Pr152 i bolja rešenja od MGA u drugim programima od Pr76 do Pr1002.

Gap	BKS	GA2OPT	MACO[11]	MGA [18]	instance
+0.00	157444	157440	178597	157444	Pr76
-0.01	127839	127852	130953	127839	Pr152
+0.00	166827	166817	167646	166827	Pr226
+0.01	82106	82095	82106	82176	Pr299
-0.01	161955	162150	161955	173839	Pr439
+0.00	382198	382185	382198	427269	Pr1002

Tabela 2

Rezultat pokazuje da iako MACO daje bolje rezultate od predloženog algoritma za instancu pod imenom Pr439, ovaj algoritam ne može da održi optimalna rešenja za druge i daje lošije rešenje od predloženog algoritama. Računarski eksperimenti još pokazuju da u opštem slučaju, predloženi algoritam daje bolje rezultate u poređenju sa druga dva, uključujući MGA i MACO algoritme u terminima kvaliteta rešenja.

Implementirajući GA2OPT algoritam došli smo do sledećih rezultata:

- za veliki broj gradova (pr. 1002) smo dobili graf koji nije razumljiv i vreme izvršavanja je duze od sat vremena (što je i bilo očekivano da će dosta sporije raditi)
- za manji broj gradova (pr. 76 i 5 putnika) se pokazao relativno dobro i završio je pretragu najkraćeg puta u solidnom vremenu

7. Zaključak

U ovom radu, hibridni algoritam koji kombinuje modifikovani GA i 2-opt lokalna pretraga je predložen za rešavanje MTSP. GA2OPT je učinkovitiji od modifikovane optimizacije kolonije mrava i modifikovanog genetskog algoritma. Za velike probleme, posebno, ovaj algoritam pribegava boljim rešenjima u poređenju sa prethodnim algoritmom. Takođe se čini da će kombinacija predloženog algoritma sa drugim metaheurističkim algoritmima uključujući simulativno žarenje, optimizaciju kolonija mrava, Tabu pretraživanje, itd. dati bolje rezultate. Osim toga, upotreba ovog predloženog algoritma za sve verzije problema preusmeravanja vozila se predlaže za buduća istraživanja.

Literatura

D. Applegate, R. Bixby, V. ChvÃ¡tal, and W. Cook. Implementing the dantzigfulkerson-johnson algorithm for large traveling salesman problems. Mathematical Programming, 97:91_153, 2003. ISSN 0025-5610.

T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. Introduction to Algorithms. McGraw-Hill Higher Education, 2nd edition, 2001. ISBN 0070131511.

G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale travelingsalesman problem. Journal of the operations research society of America, pages 393_410, 1954.

J. Holland. Adaptation in natural and arti_cial systems: an introductory analysis with applications to biology, control, and arti_cial intelligence. MIT press, 1992.

S. Lin and B. Kernighan. An e_ective heuristic algorithm for the travelingsalesman problem. Operations research, 21(2):498_516, 1973.