

ANALIZA ALGORITAMA ROJA ČESTICA ZA PROBLEM RUTIRANJA ELEKTRIČNIH VOZILA

ANALYSIS OF PARTICLE SWARM ALGORITHMS FOR THE ELECTRIC VEHICLE ROUTING PROBLEM

Nevena Radešić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – Ovaj rad se bavi primjenom algoritma optimizacije rojem čestica (PSO) i njegovog unaprijeđenog oblika, generaliziranog algoritma optimizacije rojem čestica (GPSO), u rješavanju problema rutiranja električnih vozila (EVRP). Glavni cilj je poređenje performansi PSO i GPSO nad različitim reprezentacijama čestica, konkretno: vektora prioriteta i matrice susjedstva. Eksperimentalni rezultati pokazuju da matricna reprezentacija značajno nadmašuje vektorsku, i da GPSO algoritam postiže bolje performance od osnovnog PSO algoritma. Pored ovoga, u radu su objašnjena dva algoritma zasnovana na Belman-Fordovom algoritmu koji služe za podjelu kompletne rute na više manjih i određivanje optimalnih pozicija stanica za punjenje.

Ključne reči: EVRP, PSO, GPSO, vektor prioriteta, matrica susjedstva, Bellman-Ford algoritam

Abstract – This paper addresses the application of Particle Swarm Optimization (PSO) and its enhanced version, Generalized Particle Swarm Optimization (GPSO), in solving electric vehicle routing problem (EVRP). The main objective is to compare the performance of PSO and GPSO using different particle representations, specifically: priority vector and adjacency matrix. Experimental results show that the matrix representation significantly outperforms the vector representation, and the GPSO achieves better performance than the basic PSO. Additionally, the paper provides explanations of two algorithms based on the Bellman-Ford algorithm, which serve to divide the complete route into smaller ones and determine optimal positions for charging stations.

Keywords: EVRP, PSO, GPSO, Priority vector, Adjacency matrix, Bellman-Ford algorithm

1. UVOD

EVRP je složenija varijanta klasičnog problema rutiranja vozila (VRP). VRP se bavi optimizacijom ruta za vozila koja treba da opsluže skup geografski raširenih klijenata [1] sa poznatim zahtjevima, sa ciljem minimizacije ukupnih troškova prevoza.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoran Jeličić, red. prof.

EVRP proširuje ovaj koncept, uzimajući u obzir ograničenja dometa električnih vozila i potrebu za stanicama za punjenje, što značajno povećava kompleksnost problema.

2. DEFINICIJA PROBLEMA

Problem koji je rješavan se može predstaviti pomoću neusmjerenog grafa

$$G = (V, A)$$

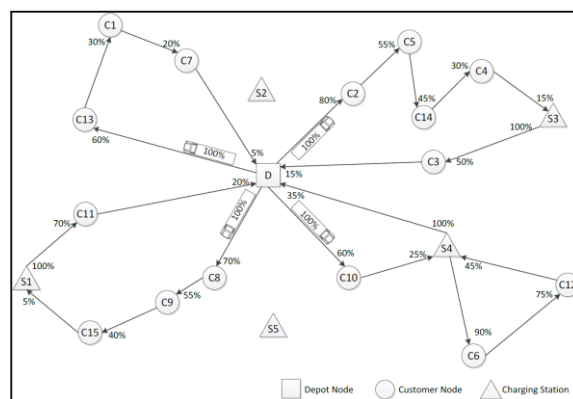
Gdje je V skup čvorova, a A skup grana. Čvorovi su klijenti, stanice za punjenje i skladište, a grane su veze između njih.

Skladište je jedinstveno i određeno je koordinatama u 2D prostoru. Svakog klijenta karakterišu potražnja i koodinate, dok su stanice određene koordinatama, tipom punjača i prosječnim vremenom čekanja.

Cilj je odrediti skup ruta za koje važe sledeća pravila:

- Svaka ruta počinje i završava se na skladištu.
- Svaki klijent mora biti uslužen tačno jednim električnim vozilom.
- Električna vozila mogu posjetiti stanicu za punjenje baterije između bilo koja dva klijenta.
- Nivo baterije električnog vozila mora uvijek biti između 20 i 80% kapaciteta baterije.
- Ukupno vrijeme putovanja je minimalno.

Primjer jednog rešenja za EVRP prikazan je na Slika 1 [2]:



Slika 1 – primjer rute za EVRP

3. ALGORITAM OPTIMIZACIJE ROJEM ČESTICA

Algoritam optimizacije rojem čestica (PSO) je stohastička optimizaciona metoda inspirisana socijalnim ponašanjem jata ptica ili riba.

Čestice predstavljaju skup potencijalnih rešenja i kreću se kroz prostor pretrage. Kretanje svake čestice određeno je njenom trenutnom pozicijom, prethodnom brzinom (komponenta inercije), ličnim najboljim iskustvom (kognitivna komponenta) i globalnim najboljim iskustvom cijelog roja (socijalna komponenta). Formalno zapisano, čestice se kreću na sledeći način:

$$v_i(t+1) = wv_i(t) + c_1r_1[\hat{x}_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)] \quad (1)$$

gdje je:

$v_i(t)$ – brzina i -te čestice u t -toj iteraciji

$x_i(t)$ – položaj i -te čestice u t -toj iteraciji

$\hat{x}_i(t)$ – najbolja pozicija koju je i -ta čestica dostigla u svojoj istoriji

$g(t)$ – najbolja pozicija koju je dostigao cijeli roj

w , c_1 i c_2 – parametri koji kontrolišu odnos između tri komponente kretanja čestica

r_1 i r_2 – slučajni brojevi iz intervala $[0, 1]$.

4. GENERALIZOVANI ALGORITAM OPTIMIZACIJE ROJEM ČESTICA

Generalizovani algoritam optimizacije rojem čestica (GPSO) predstavlja unapređenje klasičnog PSO algoritma.

GPSO se temelji na reprezentaciji kretanja čestica kao stohastičkog, diskretnog, linearnog sistema drugog reda:

$$x[k+1] + a_1 * x[k-1] + a_0 * x[k] = b_p * p[k] + b_b * g[k] \quad (2)$$

Ova formulacija omogućava uvođenje tri nova parametra: ρ , ζ i c , koji zamjenjuju tradicionalne PSO parametre (w , c_p , c_a):

- ρ kontroliše stabilnost i brzinu konvergencije. Vrijednosti bliže 1 podstiču širu eksploraciju, dok niže vrijednosti ubrzavaju konvergenciju.
- ζ određuje kako čestice osciluju oko atraktora (lične i globalno najbolje pozicije). Za vrijednosti bliske 1, čestice se kreću bez mnogo oscilacija, dok vrijednosti bliske -1 uzrokuju drastične oscilacije.
- c je težinski faktor koji služi da zamijeni kognitivni i socijalni koeficijent. Veće vrijednosti favorizuju ličnu, a manje globalnu najbolju poziciju.

Imajući u vidu semantiku parametara, jasno je da bi bilo pogodno i poželjno njihove vrijednosti mijenjati u toku izvršavanja. Na primjer, ρ se može linearno smanjivati od 0.95 do 0.6 [3]. ζ se može nasumično birati iz intervala $[-0.6, 0.4]$ [3], a c se može smanjivati od 0.8 do 0.2 [3].

Glavna prednost GPSO-a u odnosu na klasični PSO je jasnija interpretacija parametara u smislu dinamike čestica, što omogućava finiju kontrolu nad procesom optimizacije. Empirijske studije pokazuju da GPSO često nadmašuje klasični PSO i genetske algoritme na mnogim standardnim testovima optimizacije.

5. PREDLOŽENO REŠENJE

5.1. Dva pristupa rešavanju problema

Za rješavanje EVRP grubo gledano, postoje 2 grupe pristupa [4]:

- 1) *Cluster-first route-second* – najprije se klijenti grupišu u klustere koje opslužuje jedno vozilo, a onda se za svaki klaster traži optimalna ruta.
- 2) *Route-first cluster-second* – najprije se kreira jedna džinovska ruta (engl. giant route), a zatim se ona dijeli na više kraćih ruta koje zadovoljavaju ograničenja kapaciteta.

U ovom radu je implementiran *Route-first cluster-second* pristup.

5.2. Reprezentacija čestica

S obzirom na to da se u radu koristi *Route-first cluster-second* pristup, jasno je da kôd čestice treba da sadrži samo informaciju o redoslijedu klijenata. Za to su implementirala dva pristupa: vektor prioriteta i matrica susjedstva.

5.2.1. Vektor prioriteta

U ovoj reprezentaciji svakom klijentu se dodjeljuje prioritet. Veći prioritet znači da klijent treba da bude uslužen ranije. Na primjer, ako imamo 5 klijenata označenih brojevima od 1 do 5 i reprezentacija jedne čestice je:

$$[0.2, 0.5, 0.1, 0.3, 0.4],$$

onda taj vektor reprezentuje sledeći redoslijed klijenata:

$$[2, 5, 4, 1, 3].$$

Prednosti ovog pristupa su jednostavnost za implementaciju i razumijevanje, a glavni nedostatak je što se osim prioriteta, ne čuva nikakve druge informacije o ruti ili klijentima.

5.2.2. Matrica susjedstva

Pomenuti nedostatak ispravlja matrična reprezentacija. Svako polje (i, j) matrice označava vjerovatnoću/prioritet da se nakon klijenta i u ruti pojavi klijent j . Na primjer, Tabela 1 kodira česticu:

$$[\text{Складиште, клијент 3, клијент 2, клијент 1}]$$

Tabela 1 – primjer koda jedne čestice

(i, j)	skladište	klijent 1	klijent 2	klijent 3
skladište	0.0	0.1	0.2	0.7
klijent 1	0.3	0.0	0.5	0.2
klijent 2	0.4	0.3	0.0	0.3
klijent 3	0.6	0.1	0.3	0.0

5.3. Generisanje inicijalne populacije

Nakon što je određen način kodiranja čestica, za primjenu PSO algoritma potrebno je generisati inicijalnu populaciju čestica.

Inicijalna populacija za vektorsku reprezentaciju generisana je nasumično. Svaki element čestice je kreiran

kao nasumičan broj iz intervala (0, 1). Nakon toga su elementi vektora skalirani tako da je suma elemenata jednaka 1.

Za matricnu reprezentaciju nasumično generisanje nije dalo zadovoljavajuće rezultate, najvjerojatnije zbog prevelike dimenzionalnosti problema (n^2). Umjesto toga, odlučeno je da se iskoristi semantika matrice i dostupni podaci. Odnosno, da se svakom polju (i, j) dodijeli recipročna vrijednosti rastojanja između klijenata i i j . Na ovako kreiranu matricu dodaje se šum, da bi čestice bile što raznovrsnije. Konačno, vrijednosti u matrici su skalirane da bi se vrijednosti dovele u isti opseg.

5.4. Mutacije

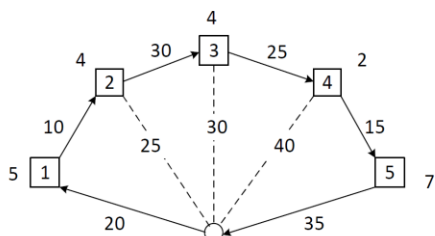
Mutacije predstavljaju slučajne izmjene u individualnim rešenjima populacije. Svrha mutacija je da prošire raznovrsnost postojećih rešenja i izbjegnju upadanje u lokalni optimum.

Za vektorsku reprezentaciju je implemenirana mutaciju zamjene (*swap*). Ova mutacija nasumično bira dvije pozicije u vektoru i mijenja njihove vrijednosti

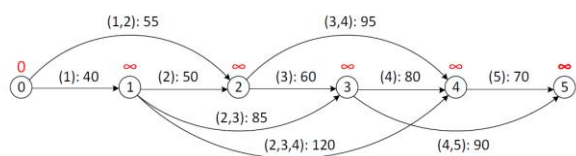
Za matricnu reprezentaciju smo primijenili mutaciju dodavanja šuma. Ovaj pristup dodaje mali, nasumični šum na elemente matrice, nakon čega se vrši renormalizacija.

5.5. Algoritam podjele

Korištenjem prethodno opisanih reprezentacija i pravila dobijena je ruta koja posjećuje sve klijente tačno jednom. Tako dobijenu rutu je potrebno podijeliti na više manjih. Problem takve podjele se može posmatrati kao problem najkraćeg puta u grafu ako se izvrše potrebne transformacije prikazane na Slika 2 i Slika 3 [4].



Slika 2 – prvi korak transformacije grafa



Slika 3 – drugi korak transformacije grafa

Ovako transformisan graf pogodan je za primjenu Belman-Fordovog algoritma koji pronalazi optimalnu podjelu.

5.6. Algoritam labeliranja

Posljednji korak u formiranju ruta je određivanje položaja stanica za punjenje. Za to je korišten algoritam labeliranja. Njegova ideja se zasniva na konceptu labele. Svaka labela čuva podatke o čvoru na koji se odnosi, nivou baterije, pređenom putu i prethodnoj labeli. Ova struktura omogućava algoritmu da "pamti" nekoliko

potencijalnih putanja do svakog čvora, uzimajući u obzir različite scenarije punjenja. Proces optimizacije se odvija kroz ekspanziju labela. Za svaki čvor, algoritam razmatra dvije mogućnosti: direktno kretanje do sljedećeg čvora ili odlazak do stanice za punjenje prije nastavka rute. Nova labela se dodaje na čvor samo ako ne postoji druga labela koja ju dominira (odnosi se na isti čvor, ali ima veći nivo baterije).

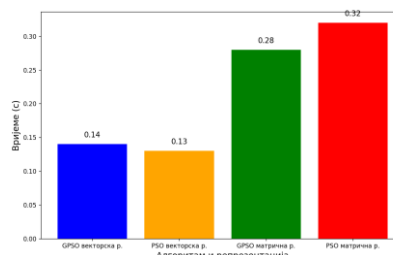
6. EKSPERIMENTALNI REZULTATI

Evaluacija predloženih metoda vršena je uz pomoć Šnajderovog skupa podataka [4]. On se sastoji iz dvije grupe instanci:

- Manje instance – do 15 klijenata;
- Veće instance – preko 100 klijenata

Prije testiranja bilo je potrebno podesiti vrijednosti parametara. To je urađeno nad jednim dijelom Šnajderovog skupa koji je izdvojen kao validacioni. Finalna testiranja vršena su nad drugim, testnim skupom.

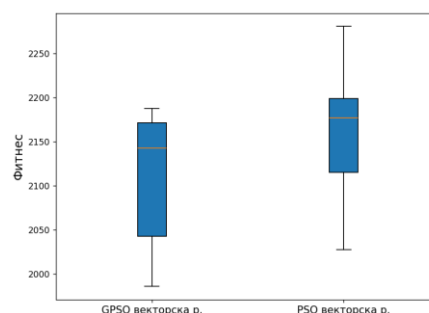
Nad manjim instancama sve metode su pronalazile optimalne rute. Razlika je bila samo u vremenu izvršavanja. Na Grafik 1 prikazano je prosječno vrijeme za 700 izvršavanja svakog pristupa.



Grafik 1 – vrijeme izvršavanja za manje setove

Očekivano, matricne reprezentacije su zbog svoje dimenzionalnosti imale duže vrijeme izvršavanja. PSO i GPSO algoritam imali su približno jednake performanse.

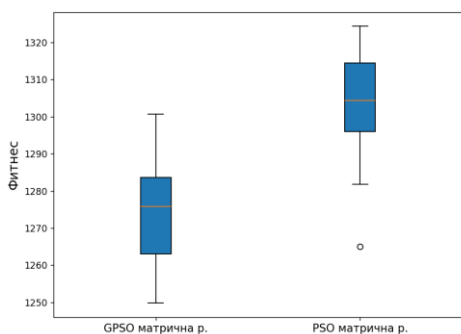
Stvarne razlike u performansama jasnije se vide na velikim instancama. Grafik 2 prikazuje boxplot vrijednost fitnes funkcije za PSO i GPSO nad vektorskom reprezentacijom čestica.



Grafik 2 – boxplot za vektorsku reprezentaciju

Možemo primijetiti da GPSO ima manji interkvartilni raspon u odnosu na PSO, što ukazuje na veću konzistentnost. Manja medijalna vrijednost ukazuje na bolju sposobnost pretrage GPSO algoritma.

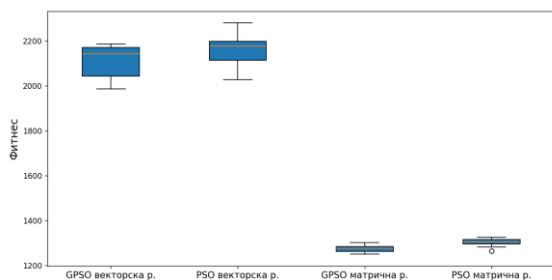
Grafik 3 prikazuje sličan grafikon, ali za matricnu reprezentaciju.



Grafik 3 – boxplot za matičnu reprezentaciju

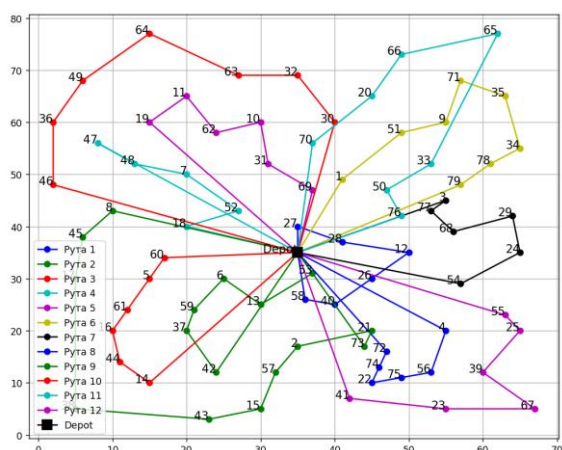
Ovdje vidimo da je razlika između algoritama još izraženija nego kod vektorske reprezentacije. PSO algoritam pokazuje veći raspon vrednosti, ima veću vrijednost medijane i outlier-e.

Ako oba boxplota smjestimo na isti grafik (Grafik 4), možemo primijetiti da matična reprezentacija generalno daje niže vrijednosti fitness funkcije. Ovo jasno pokazuje da je matična reprezentacija pogodnija za rješavanje EVRP.

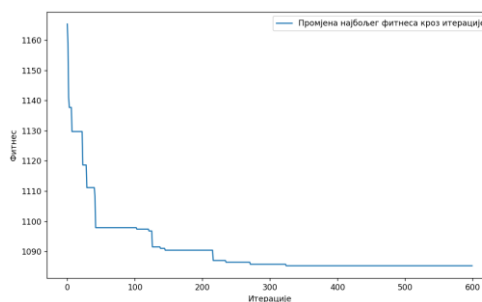


Grafik 4 – boxplotovi za sve pristupe

Slika 4 vizualizuje jedan primjer rute koju je pronašao GPSO algoritam sa matičnom reprezentacijom, dok Slika 5 prikazuje promjenu najboljeg fitnessa kroz iteracije. Iako nije pronađeno globalno najbolje rješenje, možemo reći da ovaj algoritam ima potencijal za rješavanje EVRP s obzirom na kompleksnost problema i skupa podataka.



Slika 4 – primjer rute



Slika 5 – promjena najboljeg fitnessa kroz iteracije

7. ZAKLJUČAK

U ovom radu fokus je bio na komparativnoj analizi PSO algoritma i njegove unapređene verzije GPSO nad dve reprezentacije čestica - vektora prioriteta i matrice susjedstva. Rezultati nedvosmisleno pokazuju da matična reprezentacija čestica nadmašuje vektorsku u kvalitetu dobijenih rešenja. Pored toga, eksperimenti su pokazali da GPSO algoritam postiže bolje performanse od osnovnog PSO algoritma.

8. LITERATURA

- [1] J. Ochelska-Mierzejewska, A. Poniszewska-Marańda, and W. Marańda, "Selected genetic algorithms for vehicle routing problem solving," **Electronics**, vol. 10, no. 24, p. 3147, Dec. 2021.
- [2] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," **Computers & Industrial Engineering**, vol. 161, p. 107650, Nov. 2021.
- [3] M. R. Rapaic, Z. Kanovic, Z. D. Jelcic, and D. Petrovacki, "Generalized PSO algorithm—An application to Lorenz system identification by means of neural-networks," in **Proc. 9th Symposium on Neural Network Applications in Electrical Engineering**, Sep. 2008, pp. 31-35.
- [4] C. Prins, N. Labadi, and M. Reghioi, "Tour splitting algorithms for vehicle routing problems," **International Journal of Production Research**, vol. 47, no. 2, pp. 507-535, Jan. 2009.
- [5] M. Schneider, A. Stenger, and D. Goetze, "The electric vehicle-routing problem with time windows and recharging stations," **Transportation Science**, vol. 48, no. 4, pp. 500-520, Nov. 2014.

Kratka biografija:



Nevena Radešić je rođena u Doboju 2000. godine. Diplomirala je 2023. godine na Fakultetu tehničkih nauka u Novom Sadu, na smjeru Softversko inženjerstvo i informacione tehnologije. Master rad iz oblasti Elektrotehničkog i računarskog inženjerstva, pod naslovom "Analiza algoritama roja čestica za problem rutiranja električnih vozila", odbranila je 2024. godine.