

Problem najkraćeg putovanja majstora

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Jovanović Nikola

Septembar 2022

Sažetak

Rešavanje NP teškog problema najkraćeg putovanja majstora (Minimum traveling repairman/minimum latency problem) tehnikom optimizacije - genetskim algoritmom uz dodatak lokalne pretrage.

Sadržaj

1	Uvod	2
1.1	Problem razmatranja	2
1.2	Praktična primena problema	2
2	Opis rešenja problema	2
2.1	Parametri	3
2.2	Poređenje	3
3	Eksperimentalni rezultati	3
4	Zaključak	7
	Literatura	8

1 Uvod

1.1 Problem razmatranja

Problem najkraćeg putovanja majstora (u literaturi poznat i kao problem sa minimalnim kašnjenjem), definisan u [1], pripada grupi NP-teških problema. Problem je zadat sledećim uslovima:

Graf $G = (V, E)$, početni čvor $r \in V$, dužine grana $l(e) \in R$ za svaku $e \in E$ koje zadovoljavaju nejednakost trougla ¹.

Rešenje: Obilazak grafa počevši iz čvora r , obilazeći svaki čvor iz G , npr. permutacija $\pi : [1..|V|] \rightarrow V$ tako da $\pi(1) = r$, gde π označava redosled obilaženja čvorova (računajući samo njihovo prvo obilaženje).

Mera:

$$\sum_{v \in V} d_{\pi}(r, v), \quad (1)$$

gde je $d_{\pi}(r, v)$ ukupno rastojanje pređeno u obilasku od r do prve posete čvora v .

Problem je do sada, jako uspešno, rešavan raznim tehnikama i algoritmima: genetskim uz mnogobrojne optimizacije, kvantnim itd.

1.2 Praktična primena problema

Problem najkraćeg putovanja majstora (PNPM) se može smatrati problemom putujućeg trgovca, ali se oni suštinski razlikuju.

PNPM nije samo teorijski, već je i značajni praktični optimizacioni problem zato što može da se primeni u mnogim oblastima (logistika, planiranje, rutiranje usmereno na korisnika ²).

Matematički, zadatak nam je da nađemo rutu koja minimizuje sume čekanja dolaska do svakog čvora, tako da ovaj problem može da bude posmatran kao problem usmeren na korisnika [2]

2 Opis rešenja problema

Za rešavanje ovog problema koristio sam poznatu tehniku optimizacije koja je zgodna za NP-teške probleme – genetski algoritam uz do-datak lokalne pretrage. Genetski algoritam je probabilistički algoritam pretraživanja zasnovan na mehanici prirodne selekcije i prirodne genetike [3]. U ovom slučaju:

- **Jedinka** je jedna permutacija π . Inicijalno, jedinka je nasumičan obilazak grafa. Veličina jedinke iznosi $|G|$
- **Fitness funkcija** je vreme čekanja opisano u izrazu 1 koje treba minimizovati. Jedna jedinka se smatra boljom od druge ukoliko je vrednost njene fitness funkcije manja.
- **Selekcija** je turnirska.
- **Ukrštanje** je jednopoziciono.
- **Mutacija** zasnovana na zameni (testirano je još 3 tipa mutacija).
- **Lokalna pretraga** je 2-opt.

¹ $||v_1|| + ||v_2|| \geq ||v_1 + v_2||$

²customer-centric routing

2.1 Parametri

Pošto ne možemo unapred znati najbolje parametre koji se koriste u genetskom algoritmu, vršio sam testiranje i došao do zaključka da su najbolji parametri, za koji je rezultat dovoljno dobar, a vreme izvršavanja nije preveliko za moje rešenje, sledeći:

- **Veličina populacije:** 400.
- **Broj iteracija:** 300.
- **Elitizam:** 60 jedinki.
- **Veličina turnira:** 7.
- **Verovatnoća mutacije:** 0.03.
- **Tip mutacije:** mutacija zasnovana na zameni.
- **Broj iteracija lokalne pretrage:** 5.

2.2 Poređenje

Da bih uporedio svoje dobijeno rešenje, kao referencu nisam koristio brute-force algoritam, što je za permutacije na bilo kojim graficima sa malo više čvorova nemoguće, već sam koristio najbolja poznata rešenja i algoritme iz literature.

Zaključak je da algoritam daje rezultate koji su blizu najboljim poznatim, a da vreme izvršavanja, iako duže, bude podnošljivo.

Za već navedene parametre, pokrenuo sam algoritam **10 puta** kako bih dobio prosek izvršavanja i tako ga uporedio sa dobro poznatim algoritmima. Za različite grafove, rešenja iz literature u poređenju sa mojim su data u nastavku.

3 Eksperimentalni rezultati

Ime grafa	Najbolje poznato rešenje	DBMEA			Moje rešenje sa lokalnom pretragom		
		Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)	Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)
att48	209320	209320	209320	0.48	768362	843217	6.52
berlin52	143721	143721	143721	0.73	182814	195961	6.71
dantzig42	12528	12528	12528	0.15	13555	14897	5.98
eil51	10178	10178	10178	0.75	11591	12760	6.82
eil101	27513	27513	27513	14	48729	52274	11.43
eil76	17976	17976	17976	1.45	24738	27723	9.12
lin105	603910	603910	603910	5.15	1287292	1544713	12.35
rat99	57986	57986	57986	9.99	99637	108284	11.43
st70	20557	20557	20557	1.55	29329	33820	8.74

Tabela 1: Upoređivanje rezultata izvršavanja DBMEA algoritma i mog algoritma sa lokalnom pretragom.

Ime grafa	Najbolje poznato rešenje	GILS-RVND			Moje rešenje sa lokalnom pretragom		
		Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)	Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)
att48	209320	209320	209320	0.32	768362	843217	6.52
berlin52	143721	143721	143721	0.46	182814	195961	6.71
dantzig42	12528	12528	12528	0.16	13555	14897	5.98
eil51	10178	10178	10178	0.49	11591	12760	6.82
eil101	27513	27513	27513	12.76	48729	52274	11.43
eil76	17976	17976	17976	2.64	24738	27723	9.12
lin105	603910	603910	603910	8.42	1287292	1544713	12.35
rat99	57986	57986	57986	11.27	99637	108284	11.43
st70	20557	20557	20557	1.65	29329	33820	8.74

Tabela 2: Upoređivanje rezultata izvršavanja GILS-RVND algoritma i mog algoritma sa lokalnom pretragom.

Ime grafa	Najbolje poznato rešenje	DBMEA			Moje rešenje bez lokalne pretrage		
		Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)	Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)
att48	209320	209320	209320	0.48	819836	896851	4.85
berlin52	143721	143721	143721	0.73	180936	202278	5.3
dantzig42	12528	12528	12528	0.15	13375	15686	4.4
eil51	10178	10178	10178	0.75	11805	13040	5.12
eil101	27513	27513	27513	14	48523	53792	9.68
eil76	17976	17976	17976	1.45	26449	28707	7.3
lin105	603910	603910	603910	5.15	1527988	1639254	10.05
rat99	57986	57986	57986	9.99	97534	107408	9.57
st70	20557	20557	20557	1.55	31692	36223	6.85

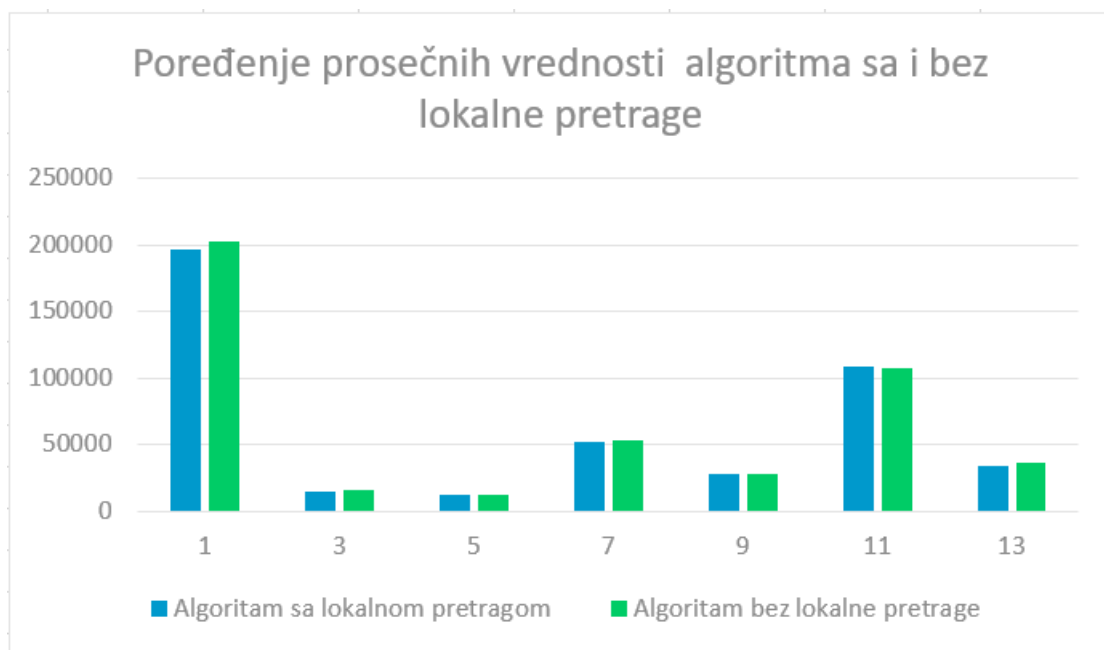
Tabela 3: Upoređivanje rezultata izvršavanja DBMEA algoritma i mog algoritma bez lokalne pretrage.

Ime grafa	Najbolje poznato rešenje	GILS-RVND			Moje rešenje bez lokalne pretrage		
		Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)	Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)
att48	209320	209320	209320	0.32	819836	896851	4.85
berlin52	143721	143721	143721	0.46	180936	202278	5.3
dantzig42	12528	12528	12528	0.16	13375	15686	4.4
eil51	10178	10178	10178	0.49	11805	13040	5.12
eil101	27513	27513	27513	12.76	48523	53792	9.68
eil76	17976	17976	17976	2.64	26449	28707	7.3
lin105	603910	603910	603910	8.42	1527988	1639254	10.05
rat99	57986	57986	57986	11.27	97534	107408	9.57
st70	20557	20557	20557	1.65	31692	36223	6.85

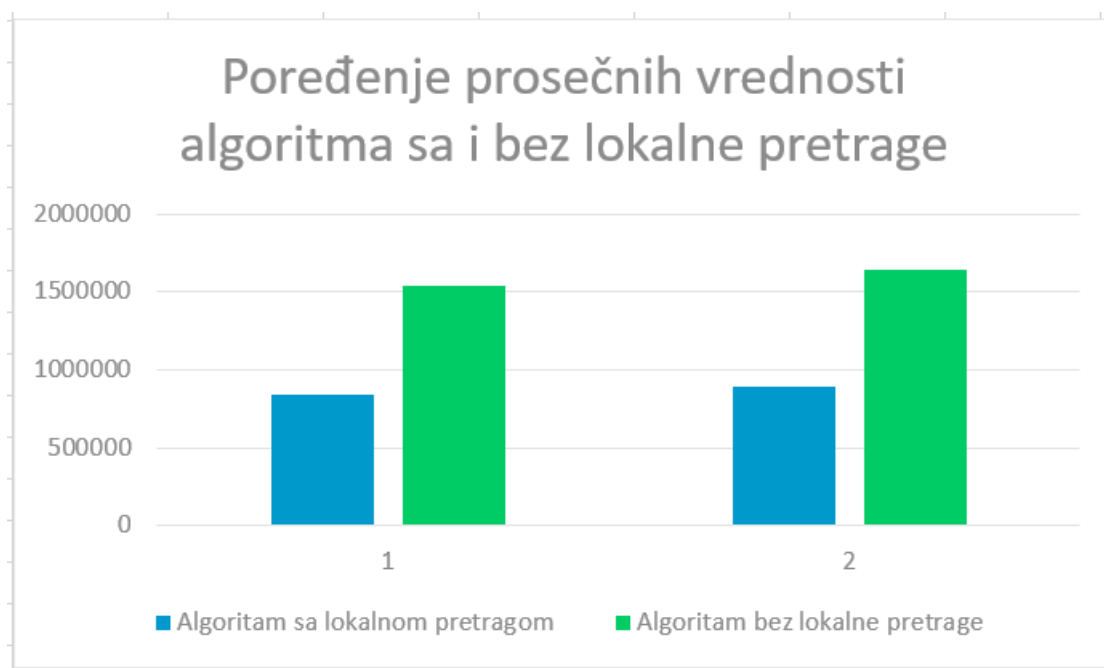
Tabela 4: Upoređivanje rezultata izvršavanja GILS-RVND algoritma i mog algoritma bez lokalne pretrage.

Ime grafa	Najbolje poznato rešenje	Moje rešenje sa lokalnom pretragom			Moje rešenje bez lokalne pretrage		
		Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)	Najbolja vrednost	Prosečna vrednost	Prosečno vreme (s)
att48	209320	768362	843217	6.52	819836	896851	4.85
berlin52	143721	182814	195961	6.71	180936	202278	5.3
dantzig42	12528	13555	14897	5.98	13375	15686	4.4
eil51	10178	11591	12760	6.82	11805	13040	5.12
eil101	27513	48729	52274	11.43	48523	53792	9.68
eil76	17976	24738	27723	9.12	26449	28707	7.3
lin105	603910	1287292	1544713	12.35	1527988	1639254	10.05
rat99	57986	99637	108284	11.43	97534	107408	9.57
st70	20557	29329	33820	8.74	31692	36223	6.85

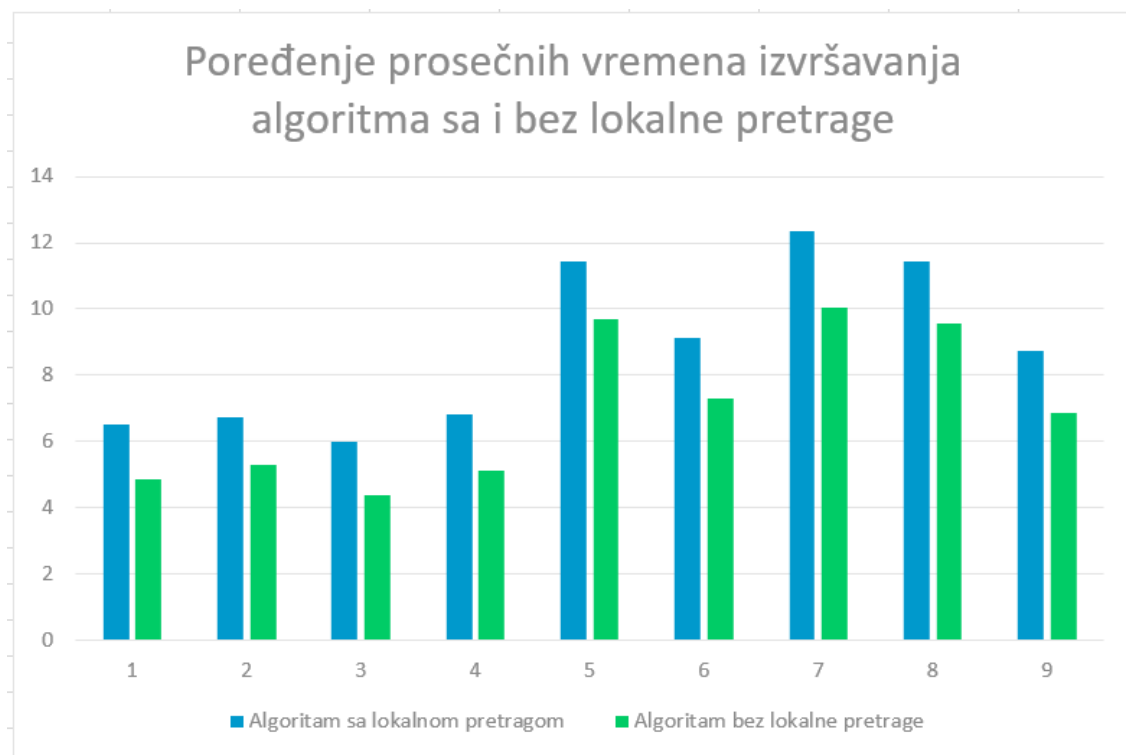
Tabela 5: Upoređivanje rezultata izvršavanja mog algoritma sa lokalnom pretragom i mog algoritma bez lokalne pretrage.



Slika 1: Poređenje prosečnih vrednosti algoritma sa i bez lokalne pretrage



Slika 2: Poređenje prosečnih vrednosti algoritma sa i bez lokalne pretrage



Slika 3: Poređenje prosečnih vremena izvršavanja algoritma sa i bez lokalne pretrage

4 Zaključak

Algoritam o kome je bilo reči u ovom radu za velike instance problema radi malo lošije, dok za manje instance pronalazi do 2 puta gori rezultat od najboljeg poznatog.

Što se vremena izvršavanja tiče, ono se može poprilično poboljšati u zavisnosti od potreba, a na blagu štetu vrednosti koja se dobije kao najbolje rešenje. Dalji napredak je moguć uz razne tehnike optimizacije koje su koristili i autori DBMEA algoritma [2] i uz dodatno ispitivanje najboljih kombinacija parametara.

Literatura

- [1] Viggo Kann *Minimum Traveling Repairman*. School of Electrical Engineering and Computer Science, 2000
- [2] Boldizsár Tüü-Szabó, Péter Földesi, László T. Kóczy *An Efficient Evolutionary Metaheuristic for the Traveling Repairman (Minimum Latency) Problem*. <https://www.atlantispress.com/>, 2020, 781-793
- [3] Manoj Kumar, Mohammad Husian, Naveen Upreti, Deepti Gupta *GENETIC ALGORITHM: REVIEW AND APPLICATION* International Journal of Information Technology and Knowledge Management, 2010, 1