



Univerzitet u Beogradu  
Mašinski fakultet

## **Računarska inteligencija**

**Iterated Local Search**

**Lena Fodulović 4006/23**  
**Dušan Mirković 4007/23**

05.10.2025.

# Sadržaj

<b>1. Uvod</b>	<b>5</b>
<b>2. Opis rešenja</b>	<b>6</b>
2.1. Funkcija cilja . . . . .	6
2.2. Komponente ILS-s . . . . .	6
2.2.1. Lokalna pretraga . . . . .	7
2.2.2. Perturbacija . . . . .	7
2.2.3. Kriterijum prihvatanja rešenja . . . . .	7
<b>3. Rezultati eksperimenata</b>	<b>9</b>
3.1. Analiza rezultata za instancu kroA100 . . . . .	9
3.1.1. Analiza performansi za različite perturbacije . . . . .	9
3.1.2. Analiza performansi za različite kriterijume prihvatanja rešenja . . . . .	11
3.1.3. Zaključak za problem kroA100 . . . . .	13
3.2. Analiza rezultata za instancu d198 . . . . .	13
3.2.1. Analiza performansi za različite perturbacije . . . . .	13
3.2.2. Analiza performansi za različite kriterijume prihvatanja rešenja . . . . .	15
3.2.3. Zaključak za problem d198 . . . . .	17
3.3. Analiza rezultata za instancu lin318 . . . . .	17
3.3.1. Analiza performansi za različite perturbacije . . . . .	17
3.3.2. Analiza performansi za različite kriterijume prihvatanja rešenja . . . . .	19
3.3.3. Zaključak za problem lin318 . . . . .	21
3.4. Analiza rezultata za instancu pcb442 . . . . .	21
3.4.1. Analiza performansi za različite perturbacije . . . . .	21
3.4.2. Analiza performansi za različite kriterijume prihvatanja rešenja . . . . .	23
3.4.3. Zaključak za problem pcb442 . . . . .	25
3.5. Analiza rezultata za instancu rat783 . . . . .	25
3.5.1. Analiza performansi za različite perturbacije . . . . .	25
3.5.2. Analiza performansi za različite kriterijume prihvatanja rešenja . . . . .	26
3.5.3. Analiza performansi za različite perturbacije (t=300s) . . . . .	27
3.5.4. Analiza performansi za različite kriterijume prihvatanja rešenja (t=300s) . . .	29
3.5.5. Zaključak za problem rat783 . . . . .	31
3.6. Rezultati eksperimenta za različite TSP instance . . . . .	32
<b>4. Zaključak</b>	<b>33</b>

## Spisak tabela

3.1. Problem kroA100: prikaz komponenti ILS algoritma - 1 . . . . .	9
3.2. Problem kroA100: rezultati - 1 . . . . .	9
3.3. Problem kroA100: prikaz komponenti ILS algoritma - 2 . . . . .	11
3.4. Problem kroA100: rezultati - 2 . . . . .	11
3.5. Problem d198: prikaz komponenti ILS algoritma - 1 . . . . .	13
3.6. Problem d198: prikaz rezultata - 1 . . . . .	13
3.7. Problem d198: prikaz komponenti ILS algoritma - 2 . . . . .	15
3.8. Problem d198: prikaz rezultata - 2 . . . . .	15
3.9. Problem lin318: prikaz komponenti ILS algoritma - 1 . . . . .	17
3.10. Problem lin318: prikaz rezultata - 1 . . . . .	17
3.11. Problem lin318: prikaz komponenti ILS algoritma - 2 . . . . .	19
3.12. Problem lin318: prikaz rezultata - 2 . . . . .	19
3.13. Problem pcb442: prikaz komponenti ILS algoritma - 1 . . . . .	21
3.14. Problem pcb442: prikaz rezultata - 1 . . . . .	21
3.15. Problem pcb442: prikaz komponenti ILS algoritma - 2 . . . . .	23
3.16. Problem pcb442: prikaz rezultata - 2 . . . . .	23
3.17. Problem rat783: prikaz komponenti ILS algoritma - 1 . . . . .	25
3.18. Problem rat783: prikaz rezultata - 1 . . . . .	25
3.19. Problem rat783: prikaz komponenti ILS algoritma - 2 . . . . .	26
3.20. Problem rat783: prikaz rezultata - 2 . . . . .	26
3.21. Problem rat783: prikaz komponenti ILS algoritma - 3 . . . . .	27
3.22. Problem rat783: prikaz rezultata - 3 . . . . .	27
3.23. Problem rat783: prikaz komponenti ILS algoritma - 4 . . . . .	29
3.24. Problem rat783: prikaz rezultata - 4 . . . . .	29
3.25. Rezultati eksperimenta za različite TSP instance . . . . .	32

## Spisak slika

3.1. Problem kroA100: grafički prikaz rezultata - 1 . . . . .	10
3.2. Problem kroA100: grafički prikaz rezultata - 2 . . . . .	12
3.3. Problem d198: grafički prikaz rezultata - 1 . . . . .	14
3.4. Problem d198: grafički prikaz rezultata - 2 . . . . .	16
3.5. Problem lin318: grafički prikaz rezultata - 1 . . . . .	18
3.6. Problem lin318: grafički prikaz rezultata - 2 . . . . .	20
3.7. Problem pcb442: grafički prikaz rezultata - 1 . . . . .	22
3.8. Problem pcb442: grafički prikaz rezultata - 2 . . . . .	24
3.9. Problem rat783: grafički prikaz rezultata - 3 . . . . .	28
3.10. Problem rat783: grafički prikaz rezultata - 4 . . . . .	30

# 1. Uvod

**Iterativna lokalna pretraga** (engl. **Iterated Local Search, ILS**) predstavlja metaheuristički pristup za rešavanje kombinatornih optimizacionih problema. Osnovna ideja ovog algoritma jeste da ne pretražuje celokupan prostor rešenja, već da se fokusira na podskup rešenja koja su lokalno optimalna prema unapred definisanoj strategiji lokalne pretrage.

Nakon dobijanja početnog lokalnog optimuma, algoritam uvodi *perturbacije* kako bi izašao iz trenutne oblasti lokalnog optimuma, a zatim ponovo primenjuje *lokalnu pretragu*. Na taj način se vrši selektivno istraživanje prostora lokalnih optimuma, što povećava verovatnoću nalaženja kvalitetnijih rešenja u poređenju sa jednostavnim strategijama nasumičnog restartovanja. Na kraju svake iteracije, *kriterijum prihvatanja rešenja* odlučuje da li je novo rešenje bolje na osnovu definisane funkcije cilja.[1]

Na skripti 1.1 prikazan je pseudokod algoritma.

```
procedure Iterated Local Search
     $s_0$  = GenerateInitialSolution()
     $s^*$  = LocalSearch( $s_0$ )

    repeat
         $s'$  = Perturbation( $s^*$ ; history)
         $s^{*'} = \text{LocalSearch}(s')$ 
         $s^* = \text{AcceptanceCriterion}(s^*; s^{*'}; \text{history})$ 
    until termination_condition

return  $s^*$ 
```

*Skripta 1.1.* Pseudokod algoritma ILS

U ovom radu, algoritam ILS primenjen je na **Problem trgovačkog putnika** (engl. **Travelling Salesman Problem, TSP**). TSP je jedan od najpoznatijih kombinatornih optimizacionih problema, gde je za zadati skup gradova i udaljenosti između njih potrebno pronaći najkraći **Hamiltonov ciklus**, odnosno putanju koja polazi iz jednog grada, obilazi svaki grad tačno jednom i vraća se u početni.

TSP je NP-težak problem, što znači da se broj mogućih ruta povećava eksponencijalno sa porastom broja gradova. Upravo zbog te složenosti, pogodan je za primenu metaheuristika kao što je ILS.[1]

## 2. Opis rešenja

Cilj ovog rada je implementacija i eksperimentalno ispitivanje Iterativne lokalne pretrage na problemu trgovačkog putnika. Algoritam je implementiran u Python okruženju uz korišćenje biblioteke *tsplib95* za učitavanje standardnih TSP instanci. Rešenje je predstavljeno kao permutacija čvorova grafa, gde svaki element označava jedan grad u obilasku. Implementacija algoritma zasnovana je na postojećim pristupima iz literature [1], čime se reprodukuje metod za eksperimentalno ispitivanje efikasnosti algoritma.

Glavne komponente algoritma koje su istražene uključuju lokalnu pretragu, perturbaciju i kriterijum prihvatanja rešenja. Svako pokretanje algoritma je ograničeno na 180 sekundi, što omogućava standardizovano vreme izvršavanja i precizno poređenje različitih komponenti. Eksperimentalno poređenje omogućava procenu uticaja ovih komponenti na kvalitet rešenja. Dobijeni rezultati su upoređeni sa rezultatima iz literature, čime se potvrđuje tačnost implementacije i omogućava detaljna analiza efikasnosti različitih varijanti algoritma.

### 2.1. Funkcija cilja

Funkcija cilja 2.1 predstavlja ukupnu dužinu putanje koju trgovac mora da pređe, pri čemu svaki grad treba da se poseti tačno jednom i da se putanja završi u početnom gradu. Cilj algoritma je **minimizacija** ove funkcije, odnosno pronalaženje najkraće moguće putanje.

$$f(s) = \sum_{i=1}^n d(s_i, s_{i+1}), s_{n+1} = s_1. \quad (2.1)$$

gde je  $d(s_i, s_{i+1})$  udaljenost između uzastopnih gradova, a poslednji grad povezan je nazad sa prvim.

### 2.2. Komponente ILS-s

Pre nego što algoritam počne iterativnu pretragu, generiše se početno rešenje (initial solution), koje predstavlja nasumičnu permutaciju elemenata problema i služi kao polazna tačka za dalju optimizaciju. Komponente koje određuju ponašanje algoritma su *lokalna pretraga*, koja poboljšava trenutno rešenje pronalaženjem lokalnog optimuma, *perturbacija*, koja uvodi značajnije promene kako bi se izbeglo zaglavljivanje u lokalnim optimumima i *kriterijum prihvatanja rešenja*, koji definiše da li novo rešenje postaje polazna tačka za narednu iteraciju.

### 2.2.1. Lokalna pretraga

Lokalna pretraga (engl. Local Search) predstavlja ključnu komponentu ILS algoritma. Njena svrha je da poboljša dato rešenje pronalaženjem lokalnog minimuma u okolini trenutnog rešenja. Proces lokalne pretrage omogućava algoritmu da iterativno unapređuje početnu putanju i približava je optimalnom rešenju. Dodatna optimizacija izvršena je korišćenjem liste k najbližih suseda za svaki grad, čime se smanjuje broj razmatranih zamena i ubrzava proces pretrage.

U ovoj implementaciji za TSP korišćena je **2-opt** metoda, koja predstavlja jednostavan, ali efikasan način poboljšanja putanje. Metoda iterativno bira par grana u putanji i obrće segment između njih ako to dovodi do smanjenja ukupne dužine putanje. Ovaj postupak se ponavlja sve dok više nije moguće pronaći bolju zamenu u okolini trenutnog rešenja, čime se dolazi do lokalnog minimuma. 2-opt se u literaturi [1] navodi kao brz i efikasan oblik lokalne pretrage, dok se u istom radu opisuje drugačiji pristup za TSP, zasnovan na *Lin-Kernighan (LKH)* algoritmu.

### 2.2.2. Perturbacija

Perturbacija je komponenta ILS-a koja omogućava izlazak iz lokalnog minimuma i istraživanje novih oblasti prostora rešenja. U ovoj implementaciji korišćene su dve vrste perturbacije:

1. **Double bridge** – metoda koja odvaja četiri segmenta putanje i menja njihov raspored. Ova promena omogućava rešenju da se značajno modifikuje i napusti lokalni minimum, dok i dalje zadržava određene osobine prethodnog rešenja, čime se omogućava efektivna pretraga novih oblasti prostora rešenja.
2. **Segment shuffle** – metoda koja nasumično menja redosled elemenata u odabranom delu putanje. Ova perturbacija uvodi raznovrsnost u pretragu, ali u nešto manjoj meri nego Double bridge, što omogućava finije prilagođavanje i detaljnije istraživanje okolnih rešenja.

### 2.2.3. Kriterijum prihvatanja rešenja

Kriterijum prihvatanja rešenja određuje da li će novo dobijeno rešenje zameniti trenutno najbolje rešenje u toku iteracije ILS algoritma. U ovoj implementaciji korišćena su dva pristupa:

1. **Single improvement** - novo rešenje se prihvata samo ako je bolje od trenutnog, odnosno ukoliko smanjuje ukupnu dužinu putanje. Ovo je striktan kriterijum koji osigurava da se vrednost funkcije cilja nikada ne pogorša.
2. **Probabilistic acceptance** – novo rešenje se uvek prihvata ako je bolje od trenutnog. Ako je rešenje lošije, ono se i dalje može prihvatiti sa verovatnoćom koja zavisi od razlike u vredno-

stima funkcije cilja i trenutne temperature  $T$ , po principu sličnom metodi simuliranog žarenja (engl. Simulated Annealing). Ovaj pristup omogućava izlazak iz lokalnih minimuma i dodatnu eksploraciju prostora rešenja.



### 3. Rezultati eksperimenata

U ovom poglavlju predstavljeni su rezultati eksperimentalne primene algoritma ILS na problem TSP. Prikazani su rezultati testiranja algoritma na standardnim TSP instancama korišćenim u literaturi, kao i vizuelni prikaz rada algoritma pri kombinovanju različitih komponenti, što omogućava detaljno praćenje ponašanja algoritma.

Eksperimenti obuhvataju poređenje različitih komponenti ILS-a, kao i ocenu kvaliteta dobijenih rešenja u odnosu na optimalne i poznate rezultate iz literature. Takođe je prikazana efikasnost algoritma u smislu broja izvršenih iteracija unutar maksimalnog vremena od 180 sekundi, kao i uticaj primenjenih metoda lokalne pretrage, perturbacije i kriterijuma prihvatanja rešenja na konačne rezultate.

U nastavku analize razmatrani su sledeći standardni TSP problemi: **kroA100**, **d198**, **lin318**, **pcb442** i **rat783**, pri čemu sufiks u nazivu instance označava broj gradova koje putanja obuhvata.[2]

#### 3.1. Analiza rezultata za instancu kroA100

##### 3.1.1. Analiza performansi za različite perturbacije

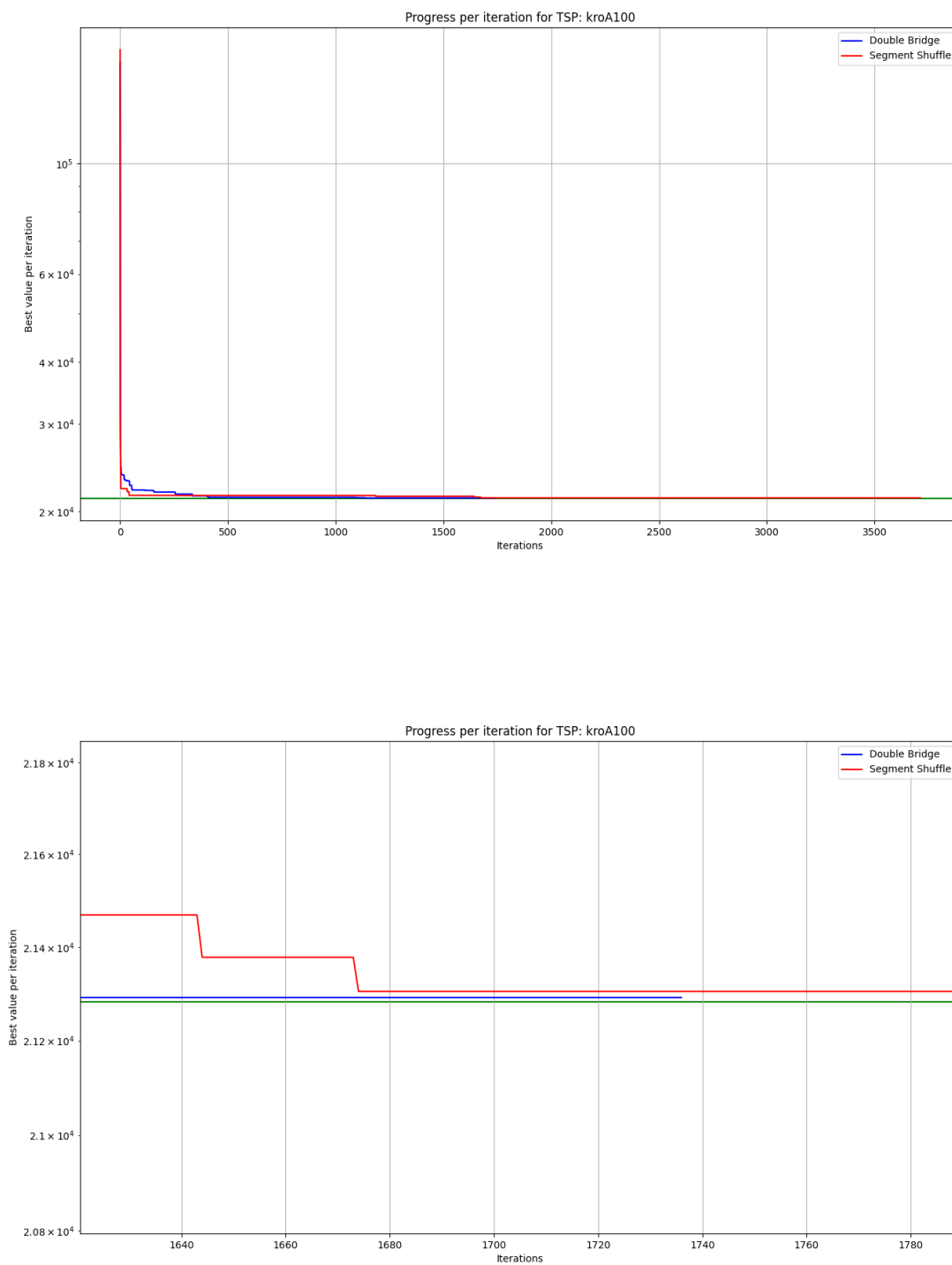
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.1 dobijeni su rezultati prikazani u tabeli 3.2. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature [3], koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

*Tabela 3.1.* Problem kroA100: prikaz komponenti ILS algoritma - 1

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Segment shuffle	Single improvement

*Tabela 3.2.* Problem kroA100: rezultati - 1

kroA100	Double bridge	Segment shuffle	Optimalno rešenje
Greška	0.05	0.11	0.0
Broj iteracija	33923	15404	56186



**Slika 3.1.** Problem kroA100: grafički prikaz rezultata - 1

Na prikazanim graficima 3.1 vidi se uporedno ponašanje algoritma pri korišćenju Double bridge i Segment shuffle perturbacija na instanci kroA100. Obe metode na početku brzo spuštaju vrednost

funkcije, ali Double bridge vrlo brzo dostiže gotovo optimalno rešenje i ostaje stabilan, dok Segment shuffle pravi nekoliko poboljšanja i postepeno se približava optimumu.

Zumirani prikaz potvrđuje da su završne vrednosti obe metode gotovo iste, ali Double bridge dolazi brže do optimuma. To potvrđuju i numerički rezultati, Double bridge je dostigao rešenje sa greškom od samo 0,05% nakon 33 923 iteracije, dok je Segment shuffle imao nešto veću grešku od 0,11% i manje iteracija (15404). Dakle, **Double bridge** je efikasniji i stabilniji na ovoj manjoj instanci, dok Segment shuffle pokazuje potencijal za povremena dodatna poboljšanja, ali uz nešto sporiju konvergenciju.

### 3.1.2. *Analiza performansi za različite kriterijume prihvatanja rešenja*

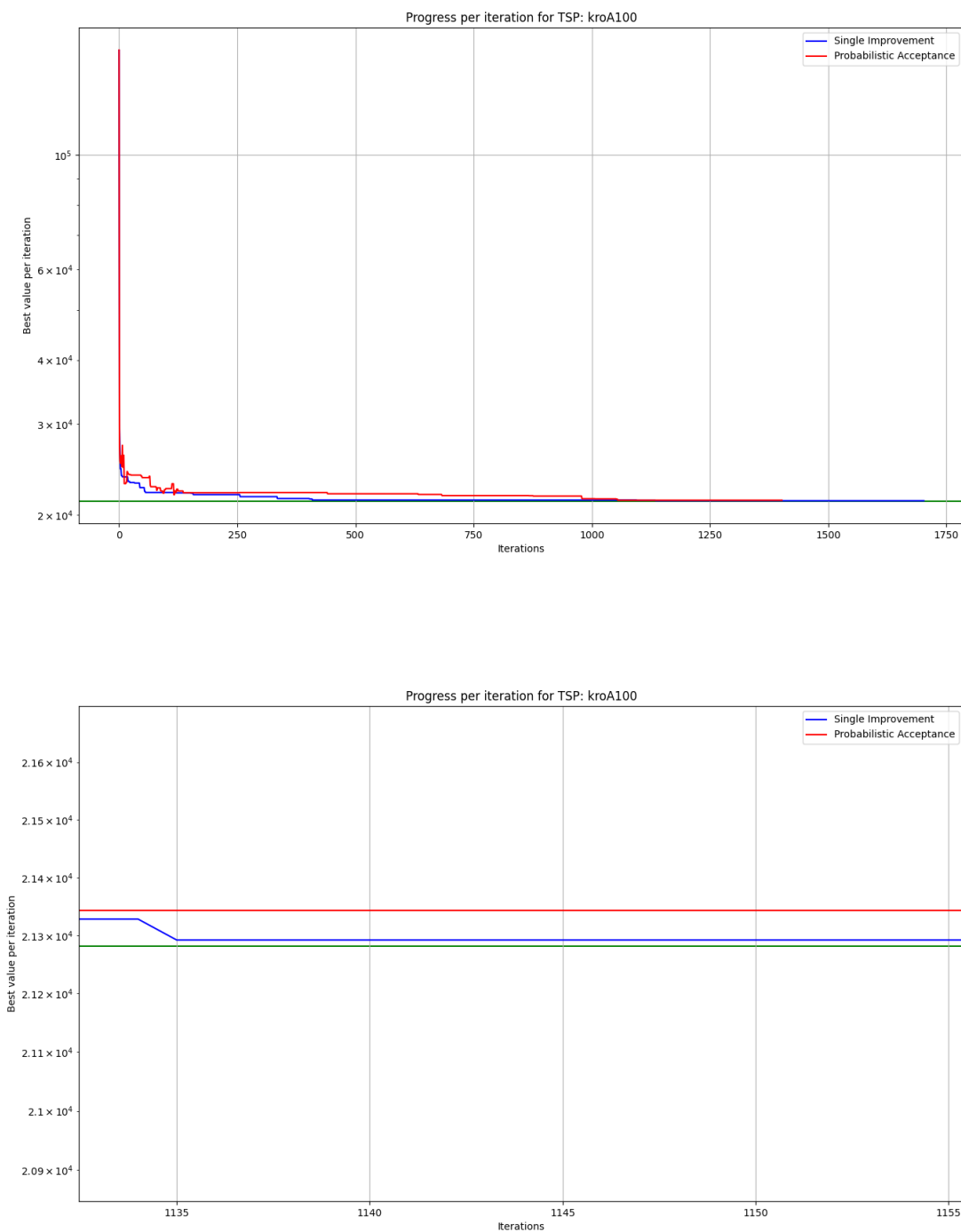
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.3 dobijeni su rezultati prikazani u tabeli 3.4. Obzirom na to da se Double bridge pokazao kao bolja strategija perturbacije, koristiće se u daljoj analizi.

**Tabela 3.3.** Problem kroA100: prikaz komponenti ILS algoritma - 2

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Double-bridge	Probabilistic acceptance

**Tabela 3.4.** Problem kroA100: rezultati - 2

kroA100	Single improvement	Probabilistic acceptance	Optimalno rešenje
Greška	0.05	0.29	0.0
Broj iteracija	33280	31613	56186



**Slika 3.2.** Problem kroA100: grafički prikaz rezultata - 2

Na prikazanim graficima 3.2 vidi se uporedno ponašanje algoritma pri korišćenju Single improvement i Probabilistic acceptance kriterijuma, pri čemu je perturbacija fiksirana na Double bridge jer je u prethodnim testovima pokazala bolje rezultate. Obe metode na početku brzo smanjuju dužinu putanje i algoritam se približava optimumu, ali Single improvement brže konvergira ka gotovo optimalnom

rešenju. Zimirani prikaz pokazuje da se plava linija (Single improvement) spušta niže i ranije od crvene (Probabilistic acceptance), što znači da češće zadržava poboljšanja i brže prihvata optimalnu putanju. Numerički rezultati potvrđuju zapažanja, Single improvement je dostigao rešenje sa greškom od samo 0,05% nakon 33280 iteracija, dok je Probabilistic acceptance imao nešto veću grešku od 0,29% uz 31613 iteracija. Dakle, na ovoj instanci kroA100 i uz Double bridge, Single improvement daje preciznije i stabilnije rezultate od probabilističkog pristupa.

### 3.1.3. Zaključak za problem kroA100

Za instancu **kroA100** može se zaključiti da algoritam, koristeći kombinaciju *Double bridge* perturbacije i *Single improvement* kriterijuma, brzo pronalazi gotovo optimalno rešenje.

## 3.2. Analiza rezultata za instancu d198

### 3.2.1. Analiza performansi za različite perturbacije

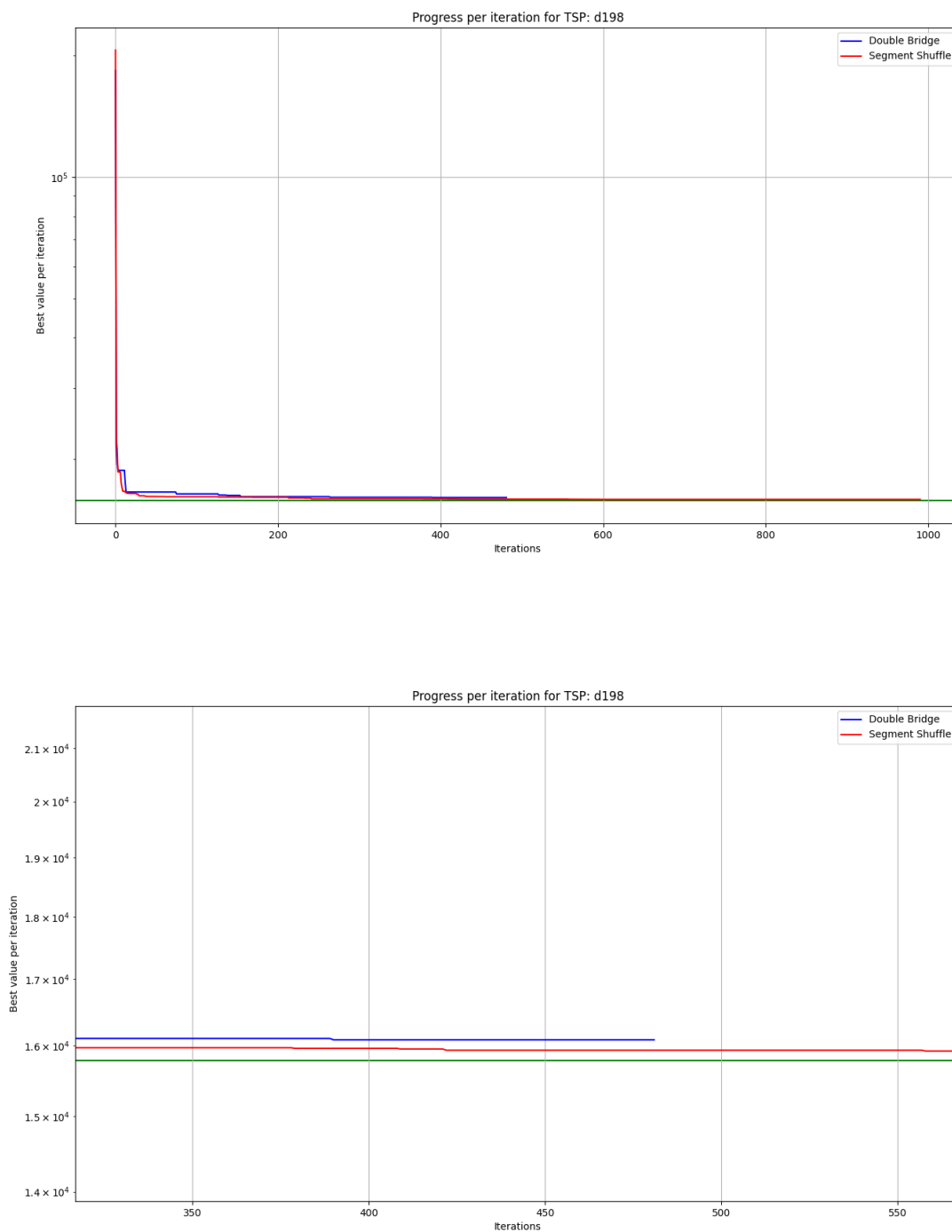
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.5 dobijeni su rezultati prikazani u tabeli 3.6. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature, koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

**Tabela 3.5.** Problem d198: prikaz komponenti ILS algoritma - 1

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Segment shuffle	Single improvement

**Tabela 3.6.** Problem d198: prikaz rezultata - 1

d198	Double bridge	Segment shuffle	Optimalno rešenje
Greška	1.93	0.82	0.0
Broj iteracija	14003	8288	36849



**Slika 3.3.** Problem d198: grafički prikaz rezultata - 1

Na prikazanim graficima 3.3 vidi se uporedno ponašanje algoritma pri korišćenju Double bridge i Segment shuffle perturbacija na instanci d198. Obe metode brzo smanjuju vrednost funkcije cilja u početnoj fazi, ali Double bridge postiže bolje rezultate, brže i stabilnije. Segment shuffle pokazuje malo sporiju konvergenciju i povremeno ostvaruje dodatna poboljšanja, ali ostaje iznad krive Double bridge

tokom većine iteracija. Zimirani prikaz potvrđuje da Double bridge postiže nižu konačnu vrednost ture i manju grešku. Takođe, numerički rezultati takođe to potvrđuju. Double bridge je dostigao grešku od 1,93% nakon 14003 iteracije, dok je Segment shuffle imao znatno manju grešku 0,82%, ali uz samo 8288 iteracija. Iako je Segment shuffle na kraju dao preciznije rešenje, Double bridge je brže dostigao dobro približenje optimumu i bio stabilniji tokom rada. Ovo sugerise da bi za brz pronalazak dobrog rešenja Double bridge bio pogodniji, dok **Segment shuffle** ima potencijal da dodatno poboljša rezultat ukoliko se algoritam izvršava duže.

### 3.2.2. *Analiza performansi za različite kriterijume prihvatanja rešenja*

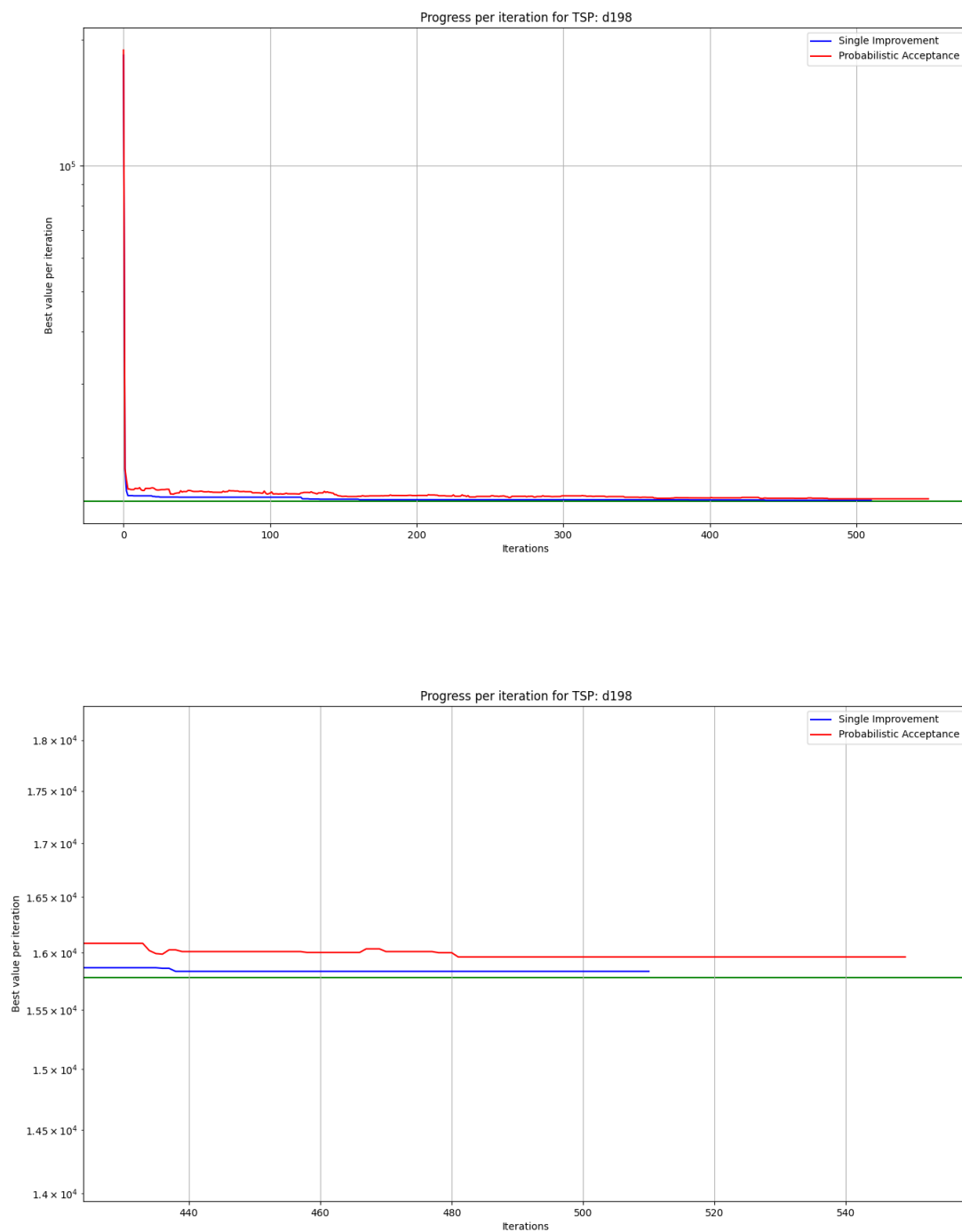
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.7 dobijeni su rezultati prikazani u tabeli 3.8. Obzirom na to da se Segment shuffle pokazao kao bolja strategija perturbacije, koristiće se u daljoj analizi.

**Tabela 3.7.** Problem d198: prikaz komponenti ILS algoritma - 2

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Segment shuffle	Single improvement
2-opt	Segment shuffle	Probabilistic acceptance

**Tabela 3.8.** Problem d198: prikaz rezultata - 2

d198	Single improvement	Probabilistic acceptance	Optimalno rešenje
Greška	0.34	1.14	0.0
Broj iteracija	5027	5656	36849



**Slika 3.4.** Problem d198: grafički prikaz rezultata - 2

Na prikazanim graficima 3.4 vidi se poređenje algoritma pri korišćenju Single improvement i Probabilistic acceptance pristupa na instanci d198. Obe metode brzo minimizuju vrednost funkcije cilja na početku, ali Single improvement dostiže nižu vrednost brže i ostaje stabilan, dok Probabilistic acceptance pravi više malih oscilacija i sporije se približava optimumu. Zumirani prikaz pokazuje da je Single



improvement uspeo da se spusti bliže optimalnoj vrednosti i zadrži je, dok Probabilistic acceptance ostaje nešto iznad i duže varira. To potvrđuju i numerički rezultati, Single improvement je postigao rešenje sa greškom od 0,34% posle 5 027 iteracija, dok je Probabilistic acceptance imao veću grešku od 1,14% uz 5 656 iteracija. Dakle, na ovoj instanci **Single improvement** pokazuje bržu i stabilniju konvergenciju i preciznije rešenje, dok Probabilistic acceptance zadržava nešto veću varijabilnost i slabiju tačnost.

### 3.2.3. *Zaključak za problem d198*

Za instancu d198 može se zaključiti da izbor komponenti ILS algoritma utiče na kvalitet i brzinu konvergencije funkcije cilja. Double bridge perturbacija pokazala je bržu i stabilniju konvergenciju, dok je Segment shuffle uz sporiji napredak uspeo da dodatno poboljša rešenje i postigne manju konačnu grešku. Sa stanovišta kriterijuma prihvatanja, Single improvement je bio efikasniji i dao preciznije i stabilnije rezultate, dok je Probabilistic acceptance imao veće oscilacije i slabiju tačnost. Dakle, kombinacija Double bridge i Single improvement pokazala je najbolji balans između brzine konvergencije i kvaliteta dobijenog rešenja, dok Segment shuffle ostaje dobra opcija kada je cilj potencijalno dodatno poboljšanje pri dužem vremenu izvođenja.

## 3.3. *Analiza rezultata za instancu lin318*

### 3.3.1. *Analiza performansi za različite perturbacije*

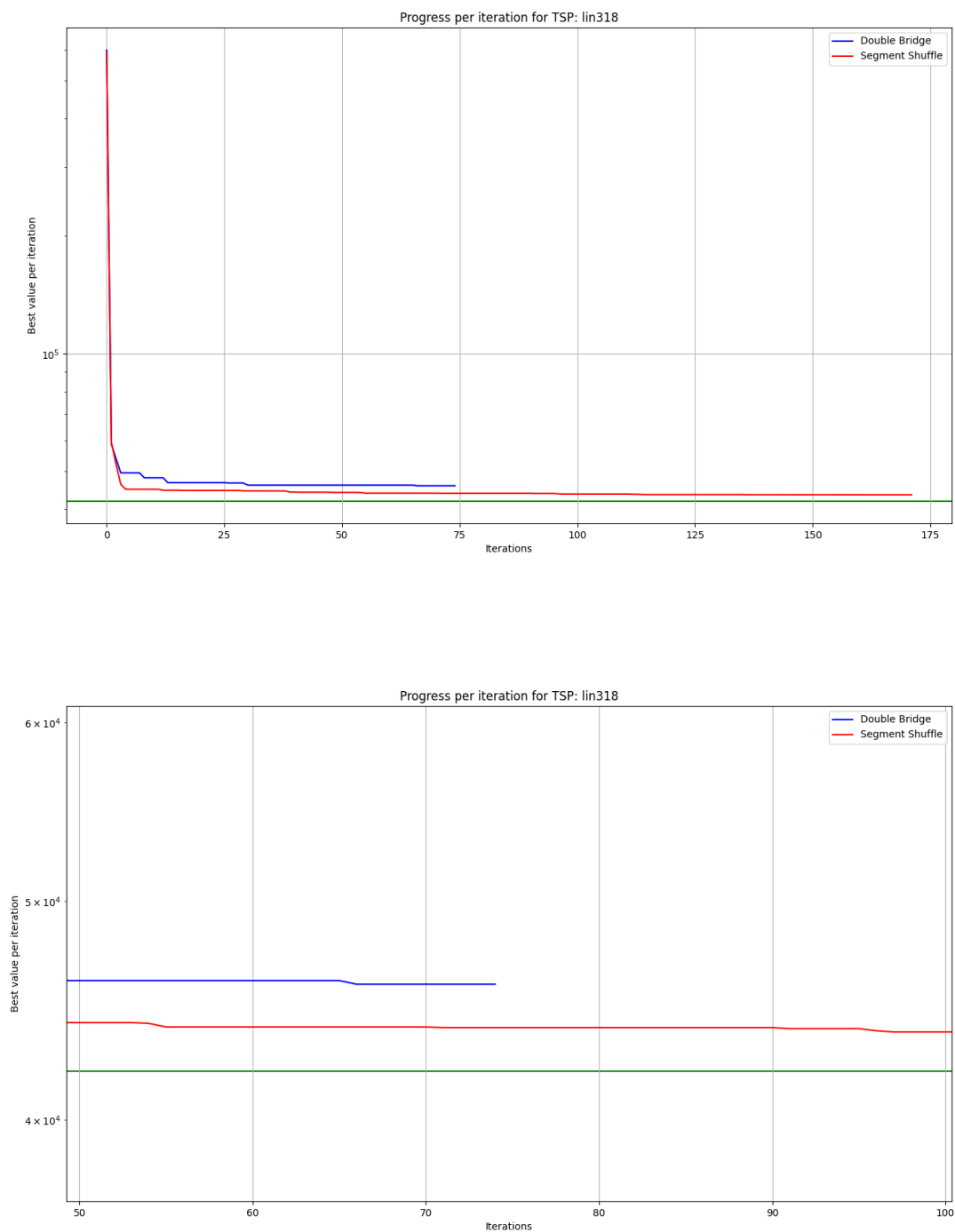
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.9 dobijeni su rezultati prikazani u tabeli 3.10. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature, koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

**Tabela 3.9.** Problem lin318: prikaz komponenti ILS algoritma - 1

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Segment shuffle	Single improvement

**Tabela 3.10.** Problem lin318: prikaz rezultata - 1

lin318	Double bridge	Segment shuffle	Optimalno rešenje
Greška	9.30	3.65	0.12
Broj iteracija	4950	4182	25540



**Slika 3.5.** Problem lin318: grafički prikaz rezultata - 1

Na prikazanim graficima 3.5 za instancu lin318 vidi se uporedno ponašanje algoritma pri korišćenju perturbacija Double bridge i Segment shuffle. Oba pristupa na početku vrlo brzo smanjuju vrednost putanje, ali se uočava da Segment shuffle brže pronalazi značajno bolje rešenje i nastavlja da se spušta ka optimumu, dok Double bridge ostaje na većoj vrednosti i nakon inicijalnog poboljšanja

stagnira. Zumirani prikaz potvrđuje da Segment shuffle uspeva da dodatno poboljša rešenje nakon nekoliko oscilacija, dok Double bridge ostaje u lokalnom minimumu. To potvrđuju i numerički rezultati, Segment shuffle je postigao manju grešku od 3,65% uz 4 182 iteracije, dok je Double bridge završio sa znatno većom greškom od 9,30% i uz malo više iteracija, 4 950. Dakle, na ovoj većoj instanci **Segment shuffle** pokazuje bolju sposobnost izlaska iz lokalnih minimuma i dolaska bliže optimumu, dok je Double bridge ovde manje efikasan.

### 3.3.2. *Analiza performansi za različite kriterijume prihvatanja rešenja*

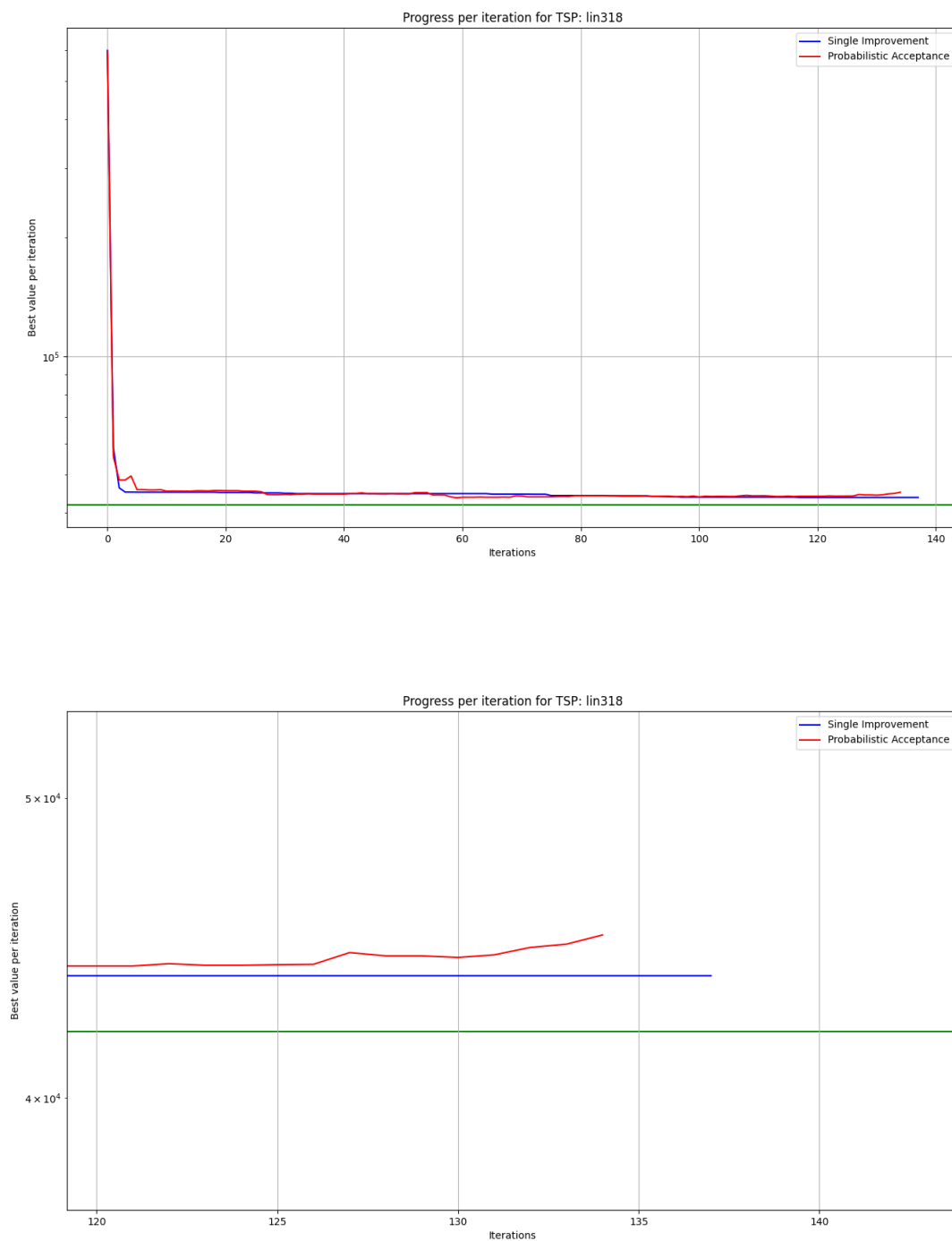
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.11 dobijeni su rezultati prikazani u tabeli 3.12. Obzirom na to da se Segment shuffle pokazao kao bolja strategija perturbacije, koristiće se u daljoj analizi.

**Tabela 3.11.** Problem lin318: prikaz komponenti ILS algoritma - 2

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Segment shuffle	Single improvement
2-opt	Segment shuffle	Probabilistic acceptance

**Tabela 3.12.** Problem lin318: prikaz rezultata - 2

lin318	Single improvement	Probabilistic acceptance	Optimalno rešenje
Greška	4.02	4.23	0.12
Broj iteracija	3749	4152	25540



**Slika 3.6.** Problem lin318: grafički prikaz rezultata - 2

Na prikazanim graficima 3.6 vidi se poređenje rada algoritma prilikom korišćenja Single improvement i Probabilistic acceptance kriterijuma na instanci lin318. Obe strategije brzo smanjuju vrednost funkcije cilja u početnim iteracijama, ali se ubrzo stabilizuju i prestaju da prave značajna poboljšanja. Single improvement pokazuje nešto stabilnije ponašanje, jer nakon brze inicijalne konvergencije do-

stiže rešenje koje ostaje konstantno bez oscilacija. S druge strane, Probabilistic acceptance se blago pogoršava i fluktuiru nakon početne faze, bez dodatnih unapređenja. Numerički rezultati potvrđuju ovaj utisak, Single improvement je došao do greške od 4.02% sa 3749 iteracija, dok je Probabilistic acceptance završio sa gotovo istom greškom (4.23%) uz nešto više iteracija (4152), ali bez boljeg krajnjeg rezultata. Dakle, na instanci lin318 **Single improvement** pokazuje stabilnije ponašanje i nešto bržu konvergenciju, dok Probabilistic acceptance ne donosi prednost i čak uvodi oscilacije koje mogu otežati finu optimizaciju.

### 3.3.3. Zaključak za problem lin318

Za instancu lin318 može se zaključiti da se povećanjem dimenzije problema menja i efikasnost perturbacija i kriterijuma prihvatanja. Segment shuffle se pokazao značajno uspešnijim od Double bridge perturbacije, jer je uspeo da izađe iz lokalnih minimuma i približi se optimumu, dok je Double bridge nakon brzog inicijalnog poboljšanja stagnirao i ostao sa većom greškom. Sa druge strane, poređenjem kriterijuma prihvatanja vidi se da Single improvement zadržava stabilniju i nešto bržu konvergenciju, dok Probabilistic acceptance ne donosi vidljive prednosti i čak uvodi oscilacije bez poboljšanja krajnjeg rešenja. Ovi rezultati sugerišu da za veće instance kao što je lin318 Segment shuffle bolje istražuje prostor rešenja, dok je Single improvement efikasniji i pouzdaniji izbor za prihvatanje rešenja.

## 3.4. Analiza rezultata za instancu pcb442

### 3.4.1. Analiza performansi za različite perturbacije

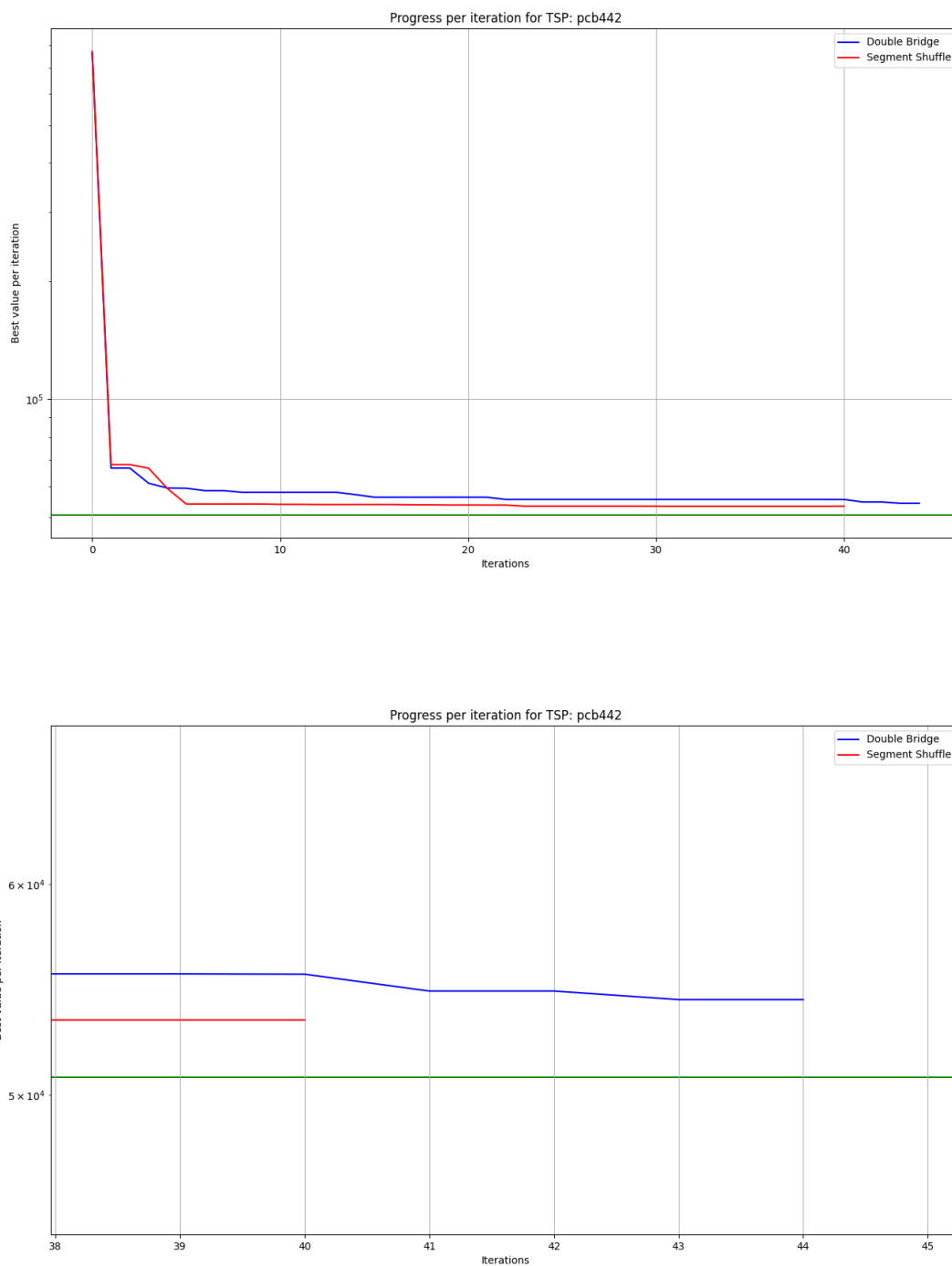
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.13 dobijeni su rezultati prikazani u tabeli 3.14. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature, koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

**Tabela 3.13.** Problem pcb442: prikaz komponenti ILS algoritma - 1

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Segment shuffle	Single improvement

**Tabela 3.14.** Problem pcb442: prikaz rezultata - 1

pcb442	Double bridge	Segment shuffle	Optimalno rešenje
Greška	6.95	5.09	0.11
Broj iteracija	3925	3279	40509



**Slika 3.7.** Problem pcb442: grafički prikaz rezultata - 1

Na prikazanim graficima 3.7 za instancu pcb442 uočava se da obe metode perturbacije, Double bridge i Segment shuffle, veoma brzo smanjuju vrednost funkcije cilja tokom početnih iteracija. Segment shuffle se u ranoj fazi brže približava optimalnom rešenju, dok Double bridge beleži nešto sporiji početni napredak, ali kasnije ostvaruje dodatna poboljšanja. Zumirani prikaz potvrđuje da je Segment

shuffle postigao bolji konačni rezultat, sa greškom od 5,09%, u poređenju sa 6,95% koliko je ostvario Double bridge, uz manji broj izvršenih iteracija 3279 naspram 3925. Dakle, za ovu instancu Segment shuffle pokazuje veću efikasnost i brže nalaženje boljih putanja, dok je Double bridge imao sporiju konvergenciju i nešto veću udaljenost od optimuma.

### 3.4.2. *Analiza performansi za različite kriterijume prihvatanja rešenja*

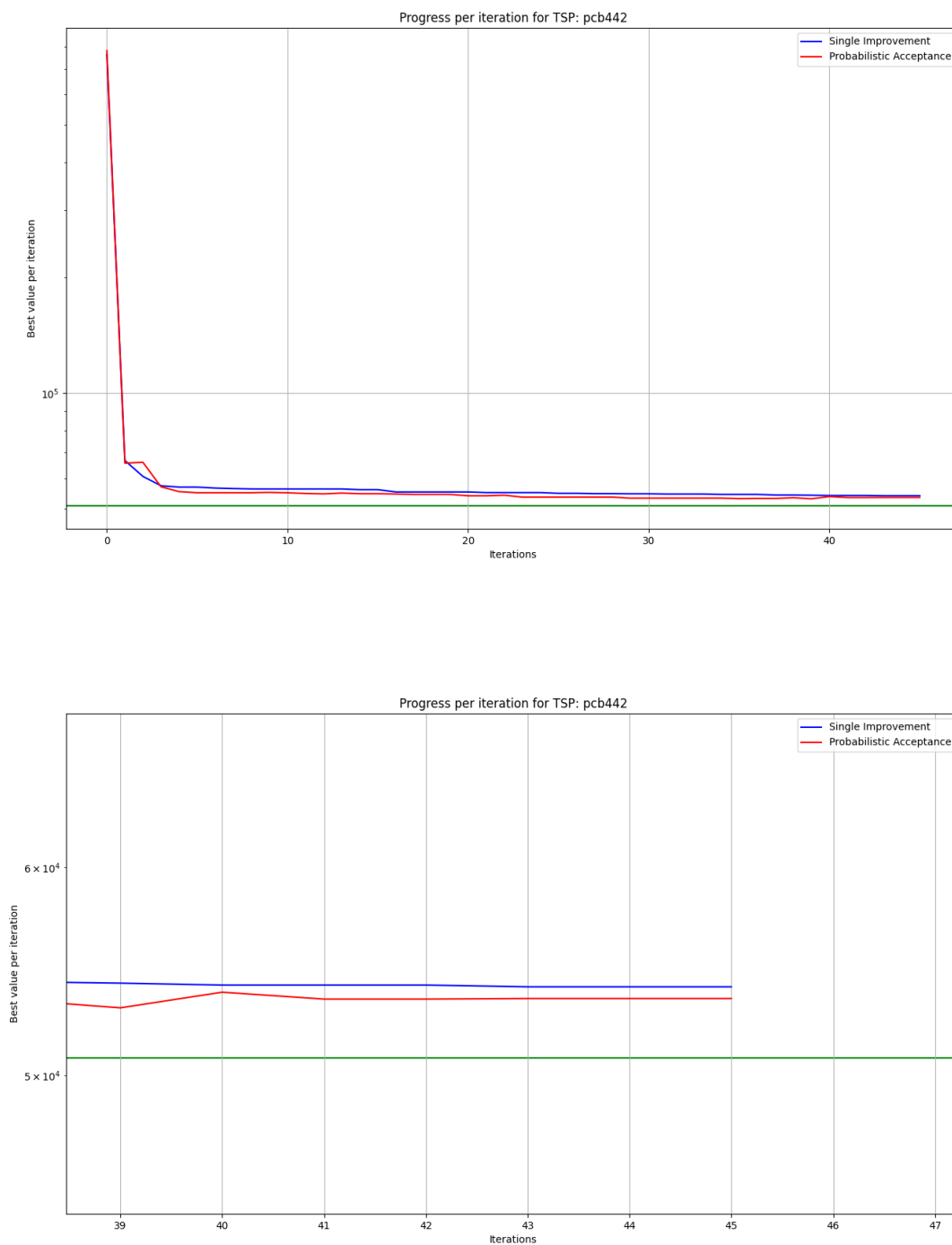
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.15 dobijeni su rezultati prikazani u tabeli 3.16. Obzirom na to da se Segment shuffle pokazao kao bolja strategija perturbacije, koristiće se u daljoj analizi.

**Tabela 3.15.** Problem pcb442: prikaz komponenti ILS algoritma - 2

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Segment shuffle	Single improvement
2-opt	Segment shuffle	Probabilistic acceptance

**Tabela 3.16.** Problem pcb442: prikaz rezultata - 2

pcb442	Single improvement	Probabilistic acceptance	Optimalno rešenje
Greška	6.42	4.47	0.11
Broj iteracija	3568	3807	40509



**Slika 3.8.** Problem pcb442: grafički prikaz rezultata - 2

Na prikazanim graficima 3.8 za instancu pcb442 prikazano je ponašanje algoritma prilikom poređenja Single improvement i Probabilistic acceptance kriterijuma prihvatanja. Obe strategije veoma brzo smanjuju vrednost funkcije cilja već u prvim iteracijama, nakon čega dolazi do stabilizacije i izostanka značajnijih poboljšanja. Zumirani prikaz pokazuje da Probabilistic acceptance dostiže nešto bolje



konačno rešenje od Single improvement kriterijuma, sa greškom od 4,47% naspram 6,42%. Takođe, Probabilistic acceptance održava niže vrednosti funkcije tokom vremena i pokazuje potencijal za dodatna poboljšanja, dok Single improvement brzo stagnira i ne ostvaruje nova poboljšanja. Numerički rezultati potvrđuju ovu razliku, Probabilistic acceptance postiže bolju tačnost uz sličan broj iteracija (3807 naspram 3568). Dakle, na ovoj većoj instanci, **Probabilistic acceptance** se pokazao efikasnijim, jer uspeva da pronađe rešenja bliža optimalnim.

### 3.4.3. Zaključak za problem pcb442

Za instancu pcb442 može se zaključiti da povećanjem složenosti problema Segment shuffle perturbacija pokazuje bolju sposobnost istraživanja prostora rešenja i pronalaženja kvalitetnijih putanja u odnosu na Double bridge, koji brže konvergira ali se često zaglavi u lokalnim minimumima. Kod kriterijuma prihvatanja, uočava se da Probabilistic acceptance donosi prednost u odnosu na Single improvement, jer omogućava povremeno prihvatanje lošijih rešenja i time pomaže izlazak iz lokalnih optimuma, što rezultira manjom konačnom greškom. Ovi rezultati ukazuju da za veće instance kao što je pcb442 kombinacija Segment shuffle perturbacije i Probabilistic acceptance kriterijuma pruža bolji balans između istraživanja i eksploatacije i dovodi do tačnijih rešenja.

## 3.5. Analiza rezultata za instancu rat783

### 3.5.1. Analiza performansi za različite perturbacije

Korišćenjem kombinacija komponenti prikazanih u tabeli 3.17 dobijeni su rezultati prikazani u tabeli 3.18. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature, koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

**Tabela 3.17.** Problem rat783: prikaz komponenti ILS algoritma - 1

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Segment shuffle	Single improvement

**Tabela 3.18.** Problem rat783: prikaz rezultata - 1

rat783	Double bridge	Segment shuffle	Optimalno rešenje
Greška	87.05	176.73	0.12
Broj iteracija	3839	3663	21937

Double bridge i Segment shuffle, inicijalno smanjuju vrednost funkcije cilja, ali vrlo brzo dolazi do zastoja u poboljšanjima. Razlog za to nije sama perturbacija, već činjenica da je zbog velike dimenzije

problema (783 grada) algoritam uspeo da izvrši praktično samo jednu iteraciju ILS algoritma pre nego što je istekao zadati vremenski limit od 180 sekundi. Posledično, nije došlo do faze intenzivnije lokalne pretrage niti do akumulacije poboljšanja kroz više iteracija, što je rezultovalo velikim greškama (87,05% za Double bridge i 176,73% za Segment shuffle) u poređenju sa poznatim optimumom. Dakle, dobijeni rezultati pre svega odražavaju ograničenje zadatog vremena izvođenja za ovako veliki problem, a ne nužno neefikasnost samih komponenti algoritma.

### 3.5.2. *Analiza performansi za različite kriterijume prihvatanja rešenja*

Korišćenjem kombinacija komponenti prikazanih u tabeli 3.19 dobijeni su rezultati prikazani u tabeli 3.20. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature, koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

**Tabela 3.19.** Problem rat783: prikaz komponenti ILS algoritma - 2

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Segment shuffle	Single improvement
2-opt	Segment shuffle	Probabilistic acceptance

**Tabela 3.20.** Problem rat783: prikaz rezultata - 2

rat783	Single improvement	Probabilistic acceptance	Optimalno rešenje
Greška	82.59	172.29	0.12
Broj iteracija	3853	3674	21937

Obe strategije prihvatanja rešenja, Single improvement i Probabilistic acceptance, uspevaju da smanje vrednost funkcije cilja u početnoj fazi, ali proces optimizacije se zaustavlja vrlo brzo. Razlog za ovakvo ponašanje nije nedostatak efikasnosti samih kriterijuma, već velika složenost instance sa 783 grada, algoritam je u predviđenom vremenskom ograničenju od 180 sekundi uspeo da izvrši praktično samo jednu potpunu iteraciju ILS ciklusa. To je dovelo do izostanka faze intenzivnije lokalne pretrage i kumulativnog poboljšanja kroz više iteracija. Numerički rezultati potvrđuju ovaj problem, Single improvement završava sa greškom od 82,59%, dok je Probabilistic acceptance još lošiji sa čak 172,29%. Dakle, visoke greške i dinamika konvergencije su posledica vremenskih ograničenja i kompleksnosti problema, a ne nužno neadekvatnosti kriterijuma prihvatanja.

Zbog navedenog ograničenja, vreme izvođenja algoritma je povećano sa 180 na 300 sekundi kako bi se omogućilo izvršavanje većeg broja lokalnih pretraga i dobila reprezentativnija procena performansi algoritma. Rezultati dobijeni nakon produženog vremena prikazani su u nastavku.

### 3.5.3. Analiza performansi za različite perturbacije ( $t=300s$ )

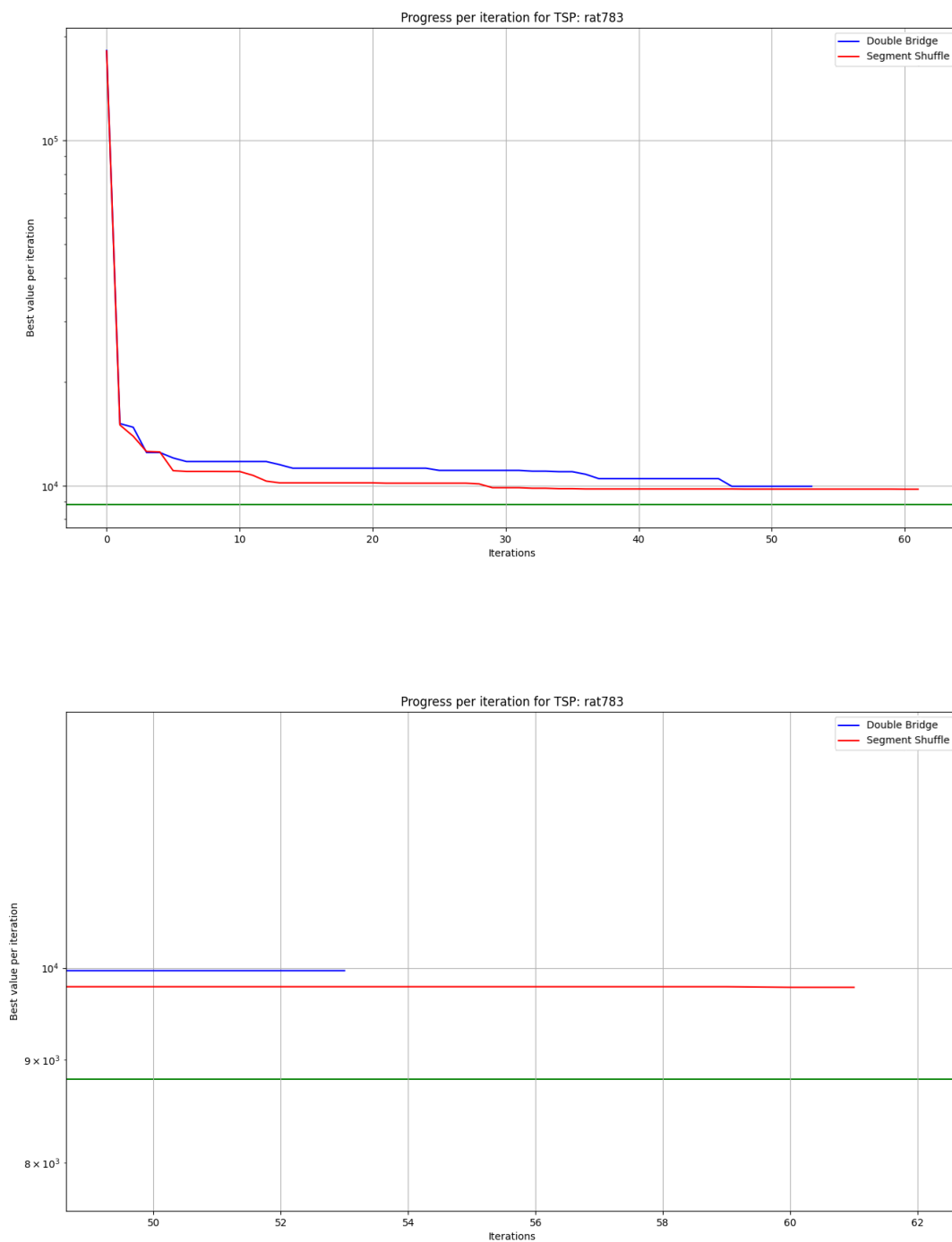
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.21 dobijeni su rezultati prikazani u tabeli 3.22. Poslednja kolona prikazuje optimalne vrednosti poznate iz literature, koje služe kao referenca za poređenje kvaliteta dobijenih rešenja.

**Tabela 3.21.** Problem rat783: prikaz komponenti ILS algoritma - 3

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Double-bridge	Single improvement
2-opt	Segment shuffle	Single improvement

**Tabela 3.22.** Problem rat783: prikaz rezultata - 3

rat783	Double bridge	Segment shuffle	Optimalno rešenje
Greška	13.21	11.06	0.12
Broj iteracija	6977	7061	21937



**Slika 3.9.** Problem rat783: grafički prikaz rezultata - 3

Na prikazanim graficima 3.9 za instancu rat783 vidi se da obe strategije perturbacije, Double bridge i Segment shuffle, uspešno smanjuju vrednost funkcije cilja u početnim iteracijama i nastavljaju da poboljšavaju rešenja kroz veći broj koraka zahvaljujući produženom vremenu izvođenja (300 sekundi). Za razliku od prethodnih eksperimenata sa kraćim vremenom, ovde algoritam uspeva da izvrši preko

7000 iteracija, što omogućava bolju lokalnu pretragu i postepeno približavanje optimumu. Segment shuffle pokazuje nešto bolju završnu tačnost, greška je smanjena na 11,06% u odnosu na 13,21% kod Double bridge pristupa, što se vidi i na grafiku. Obe metode su ipak ostale daleko iznad poznatog optimuma zbog same kompleksnosti problema, pri čemu se povećava trajanje svake 2-opt pretrage, pa i pored većeg broja iteracija nije moguće brzo postići nisku grešku. Ovo ukazuje da povećanje vremena poboljšava rezultate, ali se sa rastom dimenzije problema jasno oseća ograničenje same lokalne pretrage i rast njene računarske složenosti.

#### 3.5.4. *Analiza performansi za različite kriterijume prihvatanja rešenja ( $t=300s$ )*

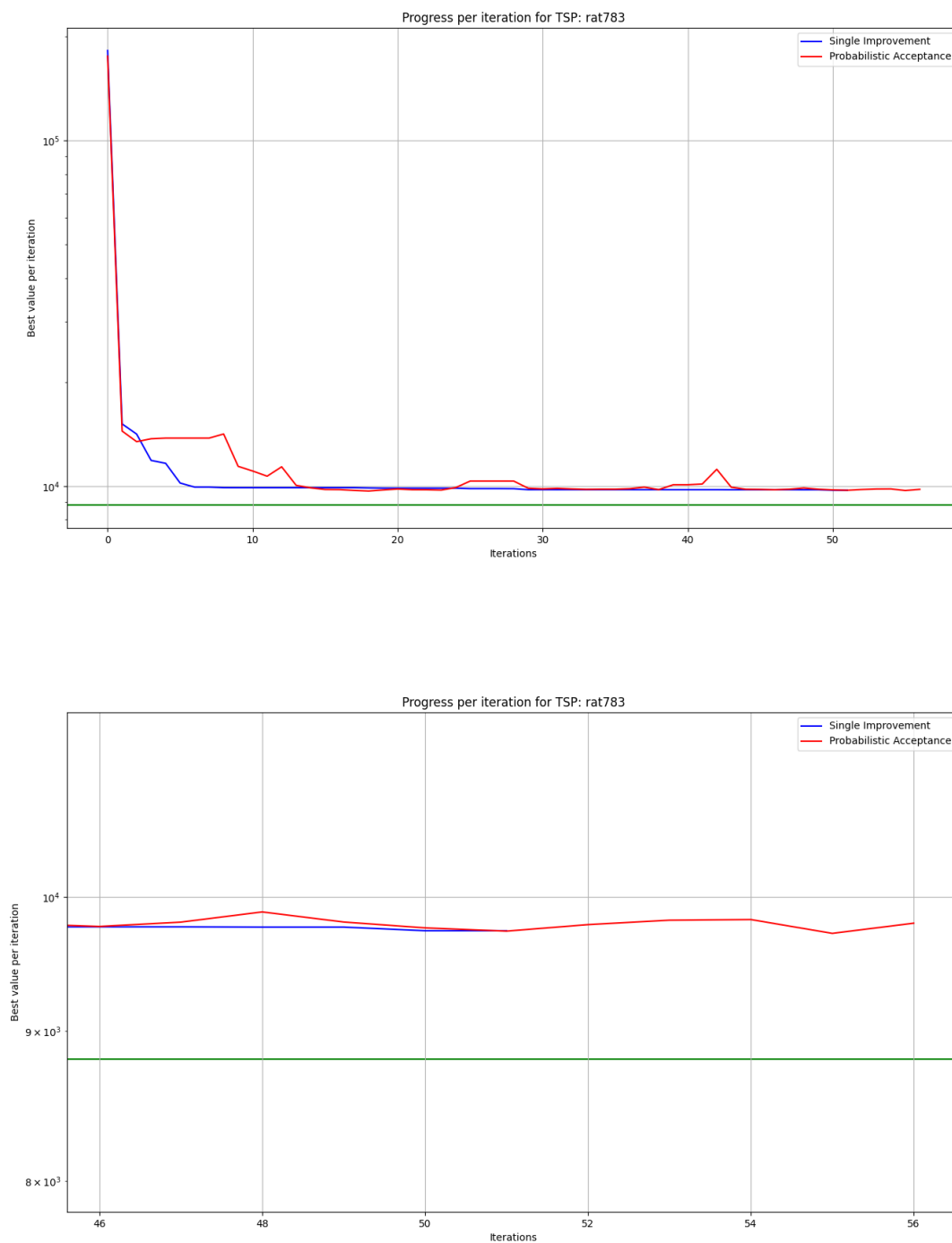
Korišćenjem kombinacija komponenti prikazanih u tabeli 3.23 dobijeni su rezultati prikazani u tabeli 3.24. Za ovaj eksperiment usvojena je Segment shuffle strategija perturbacije.

**Tabela 3.23.** Problem rat783: prikaz komponenti ILS algoritma - 4

Lokalna pretraga	Perturbacija	Kriterijum prihvatanja rešenja
2-opt	Segment shuffle	Single improvement
2-opt	Segment shuffle	Probabilistic acceptance

**Tabela 3.24.** Problem rat783: prikaz rezultata - 4

rat783	Single improvement	Probabilistic acceptance	Optimalno rešenje
Greška	10.58	9.99	0.12
Broj iteracija	6920	7309	21937



**Slika 3.10.** Problem rat783: grafički prikaz rezultata - 4

Na grafiku se vidi da obe strategije prihvatanja, Single improvement i Probabilistic acceptance, brzo smanjuju vrednost funkcije cilja u ranoj fazi i relativno brzo dostižu stabilizaciju. Međutim, Probabilistic acceptance pokazuje nešto veće oscilacije u srednjem delu pretrage, povremeno se udaljavajući od boljeg rešenja, dok Single improvement ostaje stabilniji i bez većih odstupanja. Na kraju, oba

pristupa dolaze do sličnih krajnjih vrednosti, ali Probabilistic acceptance dostiže nešto nižu grešku (9,99% naspram 10,58%) i ostvaruje više iteracija (7309 u odnosu na 6920) za isti vremenski okvir od 300 sekundi. Ovi rezultati ukazuju da na većim instancama kao što je rat783, Probabilistic acceptance, iako nešto varijabilniji tokom rada, može dodatno iskoristiti vreme za istraživanje i pronaći malo bolja rešenja, dok je Single improvement stabilniji, ali sa nešto slabijom konačnom tačnošću.

### 3.5.5. *Zaključak za problem rat783*

Na osnovu sprovedenih eksperimenata za instancu rat783 sa produženim vremenom izvođenja od 300 sekundi može se zaključiti da povećanje vremena omogućava algoritmu da izvede značajno veći broj iteracija i time postigne bolja rešenja nego u ranijim testovima sa kraćim trajanjem. Kod poređenja perturbacija, Segment shuffle je pokazao nešto malo tačnost u odnosu na Double bridge, ali su obe metode i dalje ostale relativno udaljene od poznatog optimuma zbog visoke složenosti problema i sporije 2-opt lokalne pretrage pri većem broju gradova. Slično tome, pri poređenju kriterijuma prihvatanja rešenja, Probabilistic acceptance je dao bolja krajnja rešenja i iskoristio dodatno vreme za istraživanje novih puteva, dok je Single improvement zadržao veću stabilnost ali i slabiju preciznost. Generalno, rezultati pokazuju da povećanje vremena poboljšava kvalitet rešenja, ali da kod velikih instanci složenost lokalne pretrage postaje ograničavajući faktor i sprečava algoritam da dostigne znatno niže greške.

### 3.6. Rezultati eksperimenta za različite TSP instance

U ovom delu rada prikazani su rezultati primene Iterated Local Search (ILS) algoritma na skupu od 20 TSP instanci različite složenosti. Za svaku instancu, algoritam je pokrenut jednom, koristeći kombinaciju komponenti koja se najbolje pokazala tokom prethodnih eksperimenata na odabranim primerima. U tabeli 3.25 su prikazani najbolji pronađeni rezultat (best fitness), poznati optimalni rezultat (optimal fitness), vreme izvršavanja u sekundama, broj iteracija algoritma i procenat greške u odnosu na optimalno rešenje.

**Tabela 3.25.** Rezultati eksperimenta za različite TSP instance

Problem name	Best fitness	Optimal fitness	Duration (s)	Iterations	Error (%)
kroA100	21282	21282	300	25928	0.00
d198	15927	15780	300	15373	0.93
lin318	43370	42029	300	11196	3.19
pcb442	52821	50778	300	9267	4.02
rat783	9724	8806	300	8275	10.42
pr1002	290233	259045	300	9008	12.04
pcb1173	63426	56892	300	8807	11.48
d1291	59344	50801	300	8943	16.82
fl1577	66037	22249	300	9293	196.81
pr2392	1630873	378032	300	12541	331.41
pcb3038	782303	137694	300	14547	468.15
fl3795	1418953	28772	300	11491	4831.71
rl5915	13858939	565530	300	18145	2350.61
gr666	326074	294358	300	7325	10.77
hk48	11461	11461	300	98164	0.00
brazil58	25395	25395	300	81540	0.00
a280	2639	2579	300	12406	2.33
ali535	222728	202310	300	8083	10.09
bayg29	1610	1610	300	174946	0.00
u2152	243174	64253	300	11157	278.46

Eksperimentalni rezultati pokazuju da Iterated Local Search dobro rešava probleme sa manjim i srednjim brojem gradova, ali mu tačnost značajno opada sa porastom dimenzije problema. Za manje instance poput kroA100, d198 i lin318, greška je ispod 5%, a u pojedinim slučajevima je rešenje optimalno (0% greške). Međutim, kod velikih instanci sa hiljadama čvorova, poput fl1577, pr2392, pcb3038, fl3795 i rl5915, greške drastično rastu i dostižu i više hiljada procenata, što ukazuje da algoritam u datom vremenskom ograničenju (300 s) ne uspeva da pronađe optimalna rešenja. Vreme izvršavanja je konstantno (300 s po instanci), dok broj iteracija varira u zavisnosti od složenosti grafa, od nekoliko hiljada do preko 98 000 (hk48). Generalno, ILS pokazuje stabilne performanse na manjim problemima, ali bez dodatne optimizacije i kompleksnijih lokalnih pretraga, skalabilnost ka veoma velikim TSP instancama je ograničena.



## 4. Zaključak

Rezultati eksperimenata na više standardnih TSP instanci različitih veličina ukazuju da lokalna pretraga *2-opt* predstavlja efikasan metod za brzo pronalaženje lokalnih optimuma, što potvrđuju stabilni rezultati kroz više pokretanja algoritma. Ipak, primećeno je da *2-opt* ima ograničene performanse kod većih instanci, gde sporije konvergira i često ostaje zarobljen u lokalnim minimumima, što ukazuje na potrebu za naprednijim strategijama u takvim slučajevima.

Poređenjem perturbacionih metoda, *Double bridge* uvodi veće promene i može pomoći pri izlasku iz jačih lokalnih minimuma, ali je *Segment shuffle* u većini slučajeva, uključujući i veće instance, davao bolja i stabilnija rešenja. Izuzetak je bila instanca sa 100 čvorova, gde je *Double bridge* bio nešto efikasniji.

Što se tiče kriterijuma prihvatanja, *Single improvement* strategija dovodi do konzistentnog poboljšanja rešenja, ali postoji rizik od preranog stagniranja. Uvođenje *Probabilistic acceptance* kriterijuma omogućilo je bolji balans između eksploracije i eksploatacije, čime je povećana verovatnoća pronalaženja boljih rešenja u kasnijim fazama pretrage.

Sveobuhvatno posmatrano, implementirani ILS algoritam pokazao je zadovoljavajuće performanse i stabilnost. Dobijeni rezultati su u skladu sa očekivanjima iz literature i potvrđuju efikasnost ovog metaheurističkog pristupa za rešavanje TSP problema.

*Mogući pravci daljeg istraživanja* obuhvataju primenu naprednijih heuristika za lokalnu pretragu, kao što je Lin–Kernighan algoritam, adaptivno podešavanje intenziteta perturbacija tokom izvršavanja, kao i uvođenje hibridnih kriterijuma prihvatanja koji bi dinamički balansirali između eksploatacije i eksploracije prostora rešenja. Pored toga, paralelizacija ključnih delova algoritma mogla bi značajno da ubrza izvođenje na većim instancama problema.

## Literatura

- [1] Helena R. Lourenço, Olivier C. Martin, Thomas Stützle, *Iterated Local Search*, Universitat Pompeu Fabra, Barcelona, Spain; Université Paris-Sud, Orsay, France; Darmstadt University of Technology, Darmstadt, Germany. Springer, Boston, MA, 2003. Dostupno na:  
<https://link.springer.com/book/10.1007/b101973>
- [2] TSPLIB95 – spisak instanci TSP problema koje se koriste u eksperimentima. Dostupno na:  
<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>
- [3] Optimalna rešenja za TSP instance. Dostupno na:  
<https://www.cse.unr.edu/~sushil/class/gas/TSP/STSP.html>