Univerzitet u Beogradu

Matematički fakultet

Miloš Jordanski

Rešavanje problema uspostavljanja uslužnih objekata primenom heurističkih metoda

master rad

Beograd

2014.

Mentor: **dr Miroslav Marić**
Matematički fakultet, Beograd

Članovi komisije: **dr Dušan Tošić**
Matematički fakultet, Beograd

dr Zorica Stanimirović
Matematički fakultet, Beograd

dr Miroslav Marić
Matematički fakultet, Beograd

Datum polaganja: _____

Sadržaj

1. Uvod 1
2. Heurističke metode 3
2.1. Kombinatorna optimizacija 3
2.2. Heuristike 4
3. Problem uspostavljanja uslužnih objekata 5
3.1. Opis problema 5
3.2. Matematička formulacija problema 6
4. Iterativno lokalno pretraživanje 8
4.1. Lokalno pretraživanje 8
4.2. Iterativno lokalno pretraživanje 9
4.3. Implementacija iterativnog lokalnog pretraživanja 10
5. Heuristika zasnovana na roju čestica 12
5.1. Implementacija heuristike zasnovane na roju čestica 14
6. Heuristika simuliranog kaljenja 15
6.1. Implementacija heuristike simuliranog kaljenja 16
7. Hibridni algoritmi 17
7.1. Hibridni algoritam PSO-ILS 17
7.2. Hibridni algoritam PSO-SA 17
8. Rezultati 18
8.1. Opis instanci 18
8.2. Rezultati testiranja 18
9. Testiranje parametara 29
9.1. Testiranje parametara heuristike simuliranog kaljenja 29
9.1.1. Parametar α 29
9.1.2. Parametri $Tmax$ i $Tmin$ 30
9.2. Testiranje parametara heuristike zasnovane na roju čestica 30
9.2.1. Parametar broj čestica 31
9.2.2. Parametri $vmin, vmax$ 31
9.2.3. Parametar inercija 32

9.2.4. Parametri kognitivnog i socijalnog učenja 33
9.3. Testiranje parametara iterativnog lokalnog pretraživanja 33
9.3.4 Parametar kriterijum zaustavljanja 33
10. Zaključak 35
11. Literatura 36

1. Uvod

Termin optimizacija ima široku upotrebu u različitim oblastima kao što su prirodne, tehničke, društvene, ekonomske nauke. U matematici izraz optimizacija se odnosi na proučavanje problema u kojima se traži maksimum ili minimum neke funkcije. Postoje optimizacioni problemi koji se ne mogu rešiti egzaktnim metodama u realnom vremenu, pa se za njihovo rešavanje koriste druge metode.

Mnogi realni problemi u praksi su optimizacioni problemi i imaju osobinu da je relativno lako naći neko dopustivo rešenje, ali je veoma teško naći najbolje moguće rešenje, odnosno optimalno rešenje. Sve veći interes da se problemi reše na najbolji mogući način doveo je do razvijanja različitih metoda za rešavanje optimizacionih problema. Posedovanje najboljeg mogućeg rešenja, ili njemu bliskog rešenja može da dovede do npr. uštede novca, energije, vremena, itd. Na primer, menadžer investicionog fonda mora da izabere investicije koje investitorima omogućavaju najveći mogući procenat povraćaja novca, a u isto vreme svode rizik od gubitka na prihvatljivo nizak nivo.

Problem uspostavljanja uslužnih objekata sa više nivoa na osnovu afiniteta korisnika - FLSDP (Facility location and scale decision problem with customer preference), koji je opisan u ovom radu, je optimizacioni problem. U FLSDP-u problemu treba da se odredi koje lokacije iz skupa potencijalnih lokacija uslužnih objekata treba uspostaviti, ali i koji korisnici treba da budu usluženi u kojim objektima kako bi se maksimizovala isplativost uspostavljenih objekata [22]. Da bi se uspostavio uslužni objekat, potrebno je da potencijalni profit objekta budi veći od unapred zadatog minimalnog profita, dok se pri uparivanju korisnika i uspostavljenih uslužnih objekata vodi računa o afinitetima korisnika. Na ovaj problem se nailazi prilikom izgradnje ugostiteljskih objekata.

FLSDP problem je NP-težak, pa su algoritmi za nalaženje optimalnog rešenja na većim instancama, zbog njihove neefikasnosti, u praksi neprimenljivi. Zato se za rešavanje ovog problema koriste heurističke metode. U ovom radu su predložene sledeće heuristike: heuristika zasnovana na roju čestica - PSO (Particle swarm optimization), iterativno lokalno pretraživanje - ILS (Iterated local search) i heuristika simuliranog kaljenja - SA (Simulated annealing). U cilju pronalaženja što boljih rešenja, predloženi su i hibridni algoritmi PSO-ILS i PSO-SA.

FLSDP problem spada u grupu problema maksimalnog pokrivanja korisnika - MCLP (Maximal covering location problem) i problema uspostavljanja uslužnih objekata - FLP (Facility location problem). MCLP je optimizacioni problem u kojem treba da se odredi koje lokacije iz skupa potencijalnih lokacija uslužnih objekata treba uspostaviti kako bi što više korisnika bilo opsluženo [5, 7, 11, 18, 19, 26, 39, 41]. Korisnik je opslužen ako mu je najbliži uspostavljeni uslužni centar dovoljno blizu. Na MCLP problem se nailazi prilikom izgradnje bolnica, vatrogasnih stanica, škola. Na primer, potrebno je izgraditi škole tako da sva deca imaju dostupnu školu na rastojanju koje je manje od unapred zadatog rastojanja. FLP problem je optimizacioni problem u kojem treba da

se odredi koje lokacije iz skupa potencijalnih lokacija uslužnih objekata treba uspostaviti kako bi se maksimizovala isplativost izgrađenih objekata [8, 30, 33]. Na FLP problem se nailazi prilikom izgradnje bolnica, vatrogasnih stanica, škola, ugostiteljskih objekat, skladišta. Na primer, potrebno je izgraditi skladišta tako da se minimizuju troškovi prevoza između skladišta i ugostiteljskih objekata.

Rad se sastoji od 11 poglavlja. U prvom poglavlju dat je kratak uvod u problem. U drugom poglavlju je predstavljen problem kombinatorne optimizacije, kao i načini za njegovo rešavanje. U trećem poglavlju je opisan problem i matematička formulacije problema uspostavljanja uslužnih objekata. U četvrtom, petom i šestom poglavlju su opisane heuristike ILS, PSO i SA, respektivno, kao i način na koji su implementirani za rešavanje ovog problema. Hibridni algoritmi PSO-ILS i PSO-SA, kao i način na koji su oni implementirani za rešavanje ovog problema, opisani su u sedmom poglavlju. U osmom poglavlju su predstavljeni i analizirani rezultati testiranja prethodno opisanih algoritama. Rezultati testiranja parametara heuristika SA, PSO i ILS su izloženi u devetom poglavlju. Zaključna razmatranja su izložena u desetom poglavlju, gde je dat osvrt na postignute rezultate, kao i ideje za dalja proširenja i unapređenja razmatranih algoritma. U poslednjem poglavlju je navedena literatura koja je korišćena pri izradi ovog rada.

2. Heurističke metode

2.1. Kombinatorna optimizacija

Kombinatorna optimizacija je matematička disciplina u kojoj se proučavaju problemi nalaženja ekstremnih vrednosti funkcije na najviše prebrojivom skupu. Ima široku upotrebu u: telekomunikacijama, računarskim mrežama, softverskom inženjerstvu, veštačkoj inteligenciji.

Kombinatornom optimizacijom rešava se problem sledećeg oblika. Dat je najviše prebrojiv (konačan ili prebrojivo beskonačan) skup S i funkcija $f: S \to R$. Naći minimum funkcije *f* na skupu S, odnosno odrediti

$$\min_{x \in S} f(x).$$

Skup *S* se naziva dopustiv skup, a funkcija *f* funkcija cilja. Za tačku $x \in S$ kaže da je dopustivo rešenje problema [12, 13].

Analogno, ovaj problem se može definisati kao problem traženja maksimuma funkcije $f: S \to R$ na najviše dopustivom skupu X, jer važi jednakost:

$$\max_{x \in S} f(x) = -\min_{x \in S}(-f(x)).$$

Ako je $X \subseteq \mathbb{Z}^n$, reč je o celobrojnom programiranju, a sva dopustiva rešenja su oblika $x = (x_1, x_2, \dots, x_n)$, gde $x_i \in \mathbb{Z}, 1 \leq i \leq n$ [32]. Specijalan slučaj celobrojnog programiranje je problem linearnog celobrojnog programiranja, koji je oblika:

$$\min_{x \in S} c^T x, \qquad S = \{x \in \mathbb{Z}^n \mid Ax \leq b\},$$

gde su *c* i *b* n-dimenzioni celobrojni vektori, a *A* celobrojna matrica $m \times n$ [40]. Problem linearnog programiranja je uopštenje prethodnog problema, kada skup S sadrži realne vektore:

$$\min_{x \in S} c^T x, \qquad S = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

Specijalan slučaj problema celobrojnog programiranja je problem binarnog programiranja (0-1 programiranje), koje je oblika:

$$\min_{x \in S} c^T x, \qquad S = \{x \in \{0,1\}^n \mid Ax \leq b\}.$$

Problem ranca, problem trgovačkog putnika, problem raspoređivanja poslova, problem uspostavljanja uslužnih objekata su samo neki od problema koji se rešavaju metodama kombinatorne optimizacije [15, 43].

2.2. Heuristike

Problemi kombinatorne optimizacije su uglavnom NP-teški problemi, pa je vreme izvršavanja algoritama, koji nalaze optimalno rešenje, nedopustivo dugo [12]. Zato se za rešavanje ovakvih problema koriste heuristike.

Pod heuristikom se podrazumeva tehnika za rešavanje problema kojom se traži dobro rešenje, za relativno kratko vreme. Heuristika ne daje nikakvu informaciju o tome koliko je dobijeno rešenje blisko optimalnom rešenju problema. Cilj korisnika je da heuristikom brzo dođe do rešenja koje je „dovoljno“ dobro za problem koji se rešava. Heuristike su u širokoj upotrebi za rešavanje optimizacionih problema, naročito za velike instance. Heuristike primenjene na probleme velikih dimenzija daju za kratko vreme dobra rešenja, koja su često optimalna iako se optimalnost ne može dokazati.

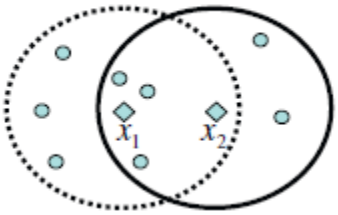
Heuristike se mogu klasifikovati na više načina. One se uglavnom dele na: heuristike zasnovane na poboljšanju jednog rešenja i heuristike zasnovane na poboljšanju populacije rešenja. Korisnik heuristike zasnovane na poboljšanju jednog rešenja radi isključivo sa jednim dopustivim rešenjem, dok korisnik heuristike zasnovane na poboljšanju populacije rešenja radi sa skupom dopustivih rešenje koje nastoji da poboljša. Primeri heuristika zasnovanih na poboljšanju jednog rešenja su: lokalno pretraživanja (Local search) [1], iterativno lokalno pretraživanje (Iterated local search) [20, 36], tabu pretraga (Tabu search) [6], heuristika simuliranog kaljenja (Simulated annealing) [10, 25, 44], metoda promenljivih okolina (Variable neighborhood search) [31]. Primeri heuristika zasnovanih na poboljšanju populacije rešenja su: genetski algoritam (Genethic algorithms) [27, 42], heuristika zasnovana na roju čestica (Particle swarm optimizitaion) [24], optimizacija kolonijom pčela (Bee colony optimization) [37], mravlje kolonije (Ant colony optimization) [4].

3. Problem uspostavljanja uslužnih objekata

3.1. Opis problema

U FLSDP problemu potrebno je odrediti lokacije, iz skupa ponuđenih lokacija, na kojima treba izgraditi uslužne objekte, tako da ukupna isplativost uspostavljenih objekata bude što veća. Jedan primer ovog problema može se opisati slikom 1. Neka su ponuđena dva uslužna objekta x1 i x2. Jedan od njih može biti otvoren zbog veličine budžeta. Afinitet za uslužni centar x1 je 0.5, a za x2 0.9. Zahtevana isplativost za otvaranje uslužnog centra je 4, gde se isplativost može definisati kao proizvod afiniteta i broja kupaca koji obuhvata uslužni centar. Krug na slici 1 pokazuje pokrivenost svakog od kandidata x1 i x2, odnosno da objekat x1 može opslužiti 6, a objekat x2 5 potrošača. Dok je x1 objekat favorizovan na osnovu blizine, afiniteti potrošača favorizuju x2, jer je isplativost objekta x2 (4.5) veća od isplativosti x1 (3). Kako je zahtevana isplativost 4, samo objekat x2 može da se otvori.

Svaki od uslužnih objekata može biti drugačijeg nivoa. Nivo objekta određuje usluge koje pruža taj objekat. Objekat nivoa 1 može da pruži uslugu tipa 1, objekat nivoa 2 može da pruži usluge tipa 1 i tipa 2. Korisnici biraju objekte na osnovu afiniteta i tipa usluge koju zahtevaju.



Slika 1. Primer FLSDP

Ugostiteljske firme, kao što su Starbucks ili McDonalds, žele da otvore nove uslužne objekte na osnovu rezultata istraživanja tržišta koje uključuje pristupačnost, afinitete prema njihovim proizvodima, itd. Pomoću ovog problema moguće je odrediti gde treba izgraditi uslužne objekte i koje veličine oni treba da budu.

Jung Man Lee i Young Hoom Lee su predložili rešavanje ovog problema pomoću Lagranžove relaksacije i CPLEX rešavača [22]. Prvo se Lagranžovom relaksacijom dobije izmenjeni problem koji se nakon toga podeli na dva potproblema koji se rešavaju CPLEX rešavačem. Primenom ove metode dobijenu su veoma zadovoljavajući rezultati.

3.2. Matematička formulacija problema

Matematički model ovog problema je definisan na sledeći način:

Indeksi:
i : indeks potencijalnih uslužnih objekata (*i=1,…,I*)
j : indeks čvora potrošača (*j=1,…,J*)
k : indeks vrste usluge (*k=1,…,K*)
s : indeks nivoa uslužnog objekta (*s=1,…,S*)

Promenljive:

$$x_{ijk}=\begin{cases}1, ako\ je\ potrošač\ u\ čvoru\ j\ uslužen\ uslugom\ k\ u\ objektu\ na\ lokaciji\ i\\ 0, inače\end{cases}$$

$$y_{is}=\begin{cases}1, ako\ je\ objekat\ nivoa\ s\ otvoren\ na\ lokaciji\ i\\ 0, inače\end{cases}$$

Podaci:

p_{ij}: afinitet čvora potrošača j prema uslužnom objektu i

d_{jk}: broj ljudi koji traži uslugu k u čvoru j

t_{isk}: kapacitet usluge k u objektu nivoa s na lokaciji i

b_{is}: cena uspostavljanja novog objekta nivoa s na lokaciji i

V: ukupan budžet za uspostavljanje svih objekata

MCR_s: minimalna zahtevana isplativost koja je potrebna da bi se uspostavio objekat nivoa s

$$a_{ij}=\begin{cases}1, ako\ je\ potrošaču\ u\ čvoru\ j\ dostupan\ objekat\ na\ lokaciji\ i\\ 0, inače\end{cases}$$

Maksimizovati:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} p_{ij}d_{jk}x_{ijk} \qquad (1)$$

Pod uslovima:

$$\sum_{j=1}^{J} d_{jk}x_{ijk} \le \sum_{s=k}^{S} t_{isk}y_{is}\,, \qquad \forall i,k \qquad (2)$$

$$x_{ijk} \le a_{ij}\,, \qquad \forall i,j,k \qquad (3)$$

$$\sum_{i=1}^{I} x_{ijk} \le 1\,, \qquad \forall j,k \qquad (4)$$

$$MCR_S y_{ij} \le \sum_{j=1}^{J}\sum_{k=1}^{s} p_{ij}d_{jk}a_{ij}\,, \qquad \forall i,s \qquad (5)$$

$$\sum_{i=1}^{I}\sum_{s=1}^{S} b_{is}y_{is} \le V \qquad (6)$$

$$\sum_{s=1}^{S} y_{is} \le 1\,, \qquad \forall i \qquad (7)$$

$$y_{is} \in \{0,1\}\,, \qquad \forall i,s \qquad (8)$$

$$x_{ijk} \in \{0,1\}, \qquad \forall i,j,k \qquad (9)$$

Cilj je maksimizovati ukupnu isplativost uspostavljenih uslužnih objekata (1). Uslov (2) ograničava broj potrošača koji mogu biti usluženi jednom uslugom u jednom objektu. Kako potrošač može biti uslužen samo u objektima koji su mu dostupni, naveden je i uslov (3). Uslov (4) obezbeđuje da potrošač može biti uslužen određenom uslugom samo u jednom objektu. Da bi se uspostavio novi objekat nekog nivoa, potencijalna isplativost mora biti veća od zadate minimalne vrednosti za taj nivo, što obezbeđuje uslov (5). Uslov (6) obezbeđuje da ukupna cena izgradnje svih uspostavljenih objekata ne bude veća od predviđenog budžeta. Uslov (7) obezbeđuje da samo objekat jednog nivoa može biti otvoren na jednoj lokaciji. Na kraju, uslovi (8) i (9) ukazuju na binarnu prirodu promenljivih y_{is} i x_{ijk}.

4. Iterativno lokalno pretraživanje

4.1. Lokalno pretraživanje

Lokalno pretraživanje je heuristički pristup rešavanja problema u kojem se nastoji pronaći najbolje rešenje problema smanjujući prostor pretraživanja [1, 20]. Često se primenjuje u kontinualnoj i diskretnoj optimizaciji.

Svakom elementu x iz prostora dopustivih rešenja S pridružuje se neki podskup dopustivog prostora rešenja $N(x) \subset S$, koji se naziva okolina elementa x. Elementi $y \in N(x)$ nazivaju se susedima elementa x.

Za početno rešenje bira se proizvoljna tačka iz prostora dopustivih rešenja. U svakoj iteraciji se pretražuje okolina tekućeg rešenja i kada se pronađe sused koji je po nekom kriterijumu bolji od tekućeg rešenja, ono se proglašava tekućim rešenjem. Ukoliko se takav sused ne pronađe, pretraživanje staje i za aproksimaciju optimalnog rešenja uzima se ono za koje je vrednost funkcije cilja najmanja (ako se rešava problem minimizacija). Lokalno pretraživanje može se prikazati sledećim pseudokodom:

Algoritam 1. Lokalno pretraživanje

***Inicijalizacija**: izabrati početno rešenje $x_0 \in S$;*

$x^*=x_0$; $f^*=f(x_0)$;

***Iterativni korak:** n=1,2,3…*

1. U okolini $N(x_n)$ trenutnog rešenja naći sledeće rešenje x_{n+1} na osnovu nekog kriterijuma;

/* ako je u pitanju problem minimizacije */

2. Ako je $f(x_{n+1})<f^*$ tada je

$f^*=f(x_{n+1})$;

$x^*=x_{n+1}$;

***Kraj:** Ako kriterijum izbora nije zadovoljen ni za jednog suseda iz okoline $N(x_n)$ ili je zadovoljen neki drugi kriteriju zaustavljanja*

X* se uzima za aproksimaciju optimalnog rešenja

Osnovni nedostatak lokalnog pretraživanja je što se obično zaustavlja prilikom nailaska na prvi lokalni minimum koji ne mora biti globalni minimum. Uspeh lokalnog pretraživanja dosta zavisi od početnog rešenja, strukture okoline i načina pretraživanja okoline. Prilikom definisanja okoline treba se truditi da sledeći uslovi budu zadovoljeni:

- Okolina svake tačke treba da bude simetrična
- Okolina ne treba da bude suviše velika, ni suviše mala
- Polazeći od proizvoljne tačke prostora S, nizom uzastopnih pomaka može se doći do bilo koje druge tačke ovog prostora

4.2. Iterativno lokalno pretraživanje

Iterativno lokalno pretraživanje - ILS predstavlja poboljšanje lokalnog pretraživanja. Poboljšanje se postiže uvođenjem perturbacije koja predstavlja značajnu promenu tekućeg rešenja. ILS heuristika se sastoji iz nekoliko koraka i može se prikazati sledećim pseudokodom:

Algoritam 2. ILS heuristika

***Inicijalizacija**: izabrati početno rešenje $s_0 \in S$;*

s=lokalno pretraživanje(s_0)*

Repeat

s' = perturbacija (s, istorija pretraživanja);*

s' = lokalno pretraživanje (s');*

s = kriterijum prihvatanja rešenja (s*, s*', memorija pretraživanja);*

Until *Kriterijum zaustavljanja*

Perturbacija vodi u udaljene regione pretraživačkog prostora i vrši se na osnovu tekućeg rešenja i istorije pretraživanja, tj. prethodnog iskustva. Kriterijum prihvatanja rešenja obično prihvata lokalni optimum. Za kriterijum zaustavljanja obično se uzima unapred određen broj iteraciji ili ponavljanje najboljeg rešenja određen broj puta, ili kombinacija prethodna dva kriterijuma [36].

Parametri ILS heuristike su:

- Kriterijum zaustavljanja
- Način definisanja okoline rešenja
- Način definisanja perturbacije

4.3. Implementacija iterativnog lokalnog pretraživanja

Za rešavanje ovog problema korišćeno je binarno kodiranje, s obzirom da su nepoznate promenljive binarne. Rešenje je kodirano binarnim nizom dužine *I*S*, koji predstavlja promenljive y_{is}, gde je *I* broj potencijalnih uslužnih objekata, a *S* maksimalan nivo jednog objekta. Okolina rešenja $(a_1, \dots, a_I)$, gde si a_i S-torke binarnih brojeva, predstavljaja sva dopustiva rešenja koja se mogu dobiti zamenom a_k i a_l, za neko *k,l*ϵ{*1,…, I*}. Funkcija prilagođenosti predstavlja aproksimaciju funkcije cilja i na osnovu nje se određuje kvalitet rešenja. Računanje funkcije prilagođenosti se odvija u tri koraka:

Korak 1. Za svaki čvor korisnika *j* niz potencijalnih uslužnih objekata se sortira u opadajućem poretku prema afinitetima

Korak 2. Svakom čvoru korisnika pridružuje se uspostavljeni uslužni objekat koji mu je dostupan i prvi slobodan iz liste prethodno sortiranih objekata

Korak 3. Dodaje se isplativost tih korisnika funkciji prilagođenosti (isplativost = afinitet*broj korisnika u čvoru)

Perturbacija rešenja, koja je implementirana u ovom radu, može se opisati sledećim pseudo kodom:

Algoritam 3. Način implementacije perturbacije rešenja

1) *brojač=0; ind=true;*

/* izlazi se iz petlje kada se nađe perturbacija rešenja, ili nakon I neuspelih pokušaja */
2) *while(brojač<I and ind==true)*
   a) *Na slučajan način izabrati indeks i*ϵ{*2,…,I-1*}*.*

   b) *If(i==2 or i==I-1)*
      /* ako je izabrana druga ili predposlednja lokacija u kodu; slika 2 prikazuje primer perturbacije u slučaju da je izabrana druga lokacija, dok slika 3 prikazuje primer perturbacije u slučaju da je izabrana predposlednja lokacija */
      then: Ako je redosled u kodu aib, gde su a,i,b S-torke binarnih brojeva, tada se pokušava zamena b sa a, tj. dobija se bia. Ako je to dopustivo rešenje: ind=false;

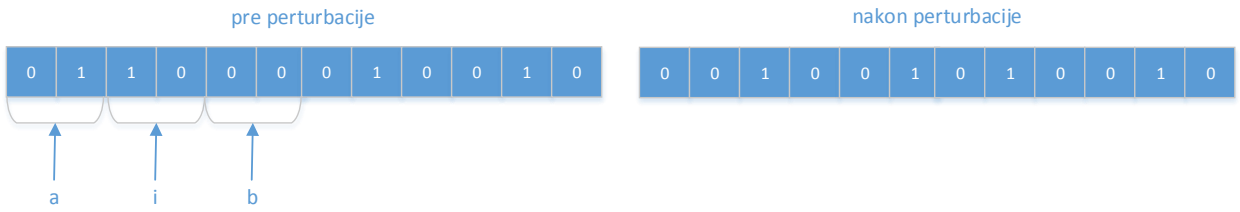
      /* ako je izabrana lokacija koja u kodu ima četiri susedne lokacije; slika 4 prikazuje jednu ovakvu perturbaciju */
      else: Ako je redosled u kodu abicd, gde su a, b, i , c, d S-torke binarnih brojeva, tada se pokušava zamena a sa d i b sa c. Ako bar jedna od ovih zamena uspe, tj. ako se bar u jednom slučaju dobije dopustivo rešenje: ind=false;

   c) *brojač ++;*

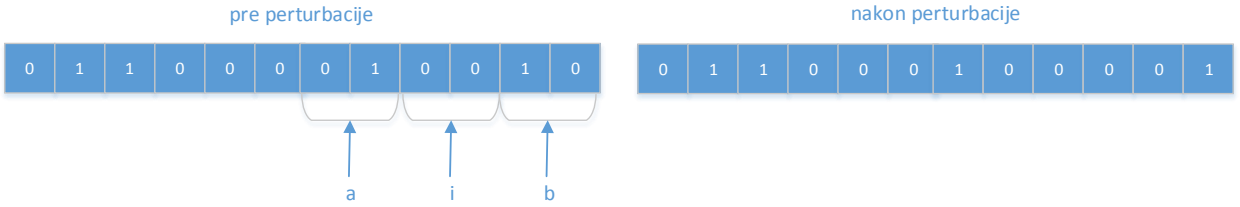
/* ako je bilo I neuspelih pokušaja */

3) If (brojač==I)

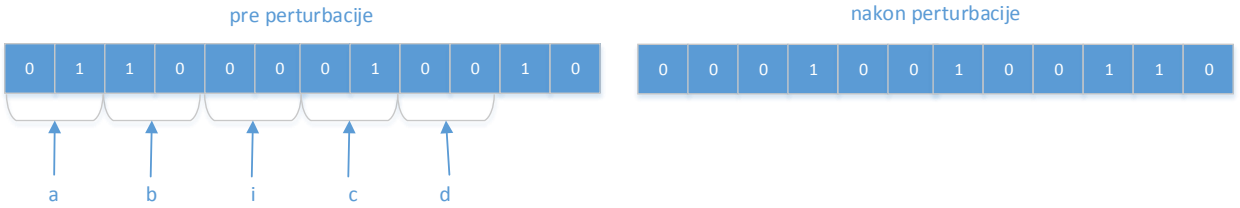
a) Na slučajan način odrediti rešenje;



Slika 2. Primer perturbacije: i=2



Slika 3. Primer perturbacije: i=I-1



Slika 4. Primer perturbacije: i=3

Prihvata se prvo rešenje iz okoline, koje je bolje od trenutnog rešenja. Na osnovu rezultata testiranja parametra kriterijuma zaustavljanja, koje je kasnije opisano u sekciji 9.3.4, algoritam se završava nakon I*J/4 iteracija ili ako se najbolje rešenje nije menjalo I*J/8 iteracija. Na kraju se poziva CPLEX rešavač koji računa funkciju cilja za rešenje koje ima najveću funkciju prilagođenosti. Rezultat CPLEX-a se uzima za aproksimaciju optimalnog rešenja.

5. Heuristika zasnovana na roju čestica

Heuristika zasnovana na roju čestica - PSO pripada kategoriji algoritama koji su inspirisani inteligencijom grupe. Tvorci ovog algoritma su Russel Ebenhart i James Kennedy [24]. Prema analogiji sa jatom ptica, optimizacija se zasniva na grupi čestica, odnosno ptica, koje lete po dopustivom prostoru u potrazi za najboljom lokacijom. Kretanje svake čestice je diktirano brzinom koja se neprestano menja u potrazi za što boljim rešenjem. Svaka čestica predstavlja jedno rešenje problema, i kreće se po dopustivom prostoru rešenja koristeći svoje iskustvo, ali i iskustvo drugih čestica. Sve čestice istovremeno pokušavaju da poboljšaju svoje pozicije. Svaka čestica sadrži sledeće elemente:

- x_i – trenutna pozicija čestice (trenutno rešenje)
- v_i – brzina, odnosno pravac u kojem bi se čestica kretala bez drugih uticaja, $v_i \epsilon [v_{min}, v_{max}]$
- p_i – najbolja pozicija čestice do sada
- p_g – najbolje rešenje celog roja ili najbolje rešenje okoline čestice roja

Postoje dve strategije za promenu brzine čestice: korišćenje iskustva celog roja ili korišćenje iskustva neke okoline čestice. Prilikom korišćenja iskustva okoline čestice, okolina čestice se može definisati na različite načine.

PSO heuristika može se opisati sledećim pseudokodom:

Algoritam 4. PSO heuristika

---

1) *Na slučajan način generisati čestice i=1,2,…, n koje čine roj čestica. Svakoj čestici pridružiti na slučajan način trenutnu poziciju i brzinu*
2) *Repeat*
   - *i) Izračunati* $f(x_i)$*;*
   - *ii) Za sve čestice i uraditi:*

     (a) $v_i(t) = promeni\ brzinu\big(v_i(t-1)\big)$;

     /*ako je brzina manja od minimalne dozvoljene, brzina postaje minimalna */
     (b) $if(v_i(t) < v_{min})\ then\ v_i(t) = v_{min}$ *;*

     /* ako je brzina veća od maksimalne dozvoljene, brzina postaje maksimalna */
     (c) $if(v_i(t) > v_{max})\ then\ v_i(t) = v_{max}$ *;*

     /* dodeliti novu poziciju čestici */
     (d) $x_i(t) = x_i(t-1) + v_i(t-1)$*;*

     /* ako je trenutno rešenje bolje od do sada najboljeg rešenja i-te čestice, menja se najbolje rešenje i-te čestice */
     (e) $if\big(f(x_i) < f(p_i)\big)\ then\ p_i = x_i$; *(problem minimizacije)*

     /* ako je trenutno rešenje bolje od do sada najboljeg rešenja, menja se najbolje rešenje */

(f) $if\left(f(x_i) < f(g_i)\right)\ then\ g_i = x_i$; *(problem minimizacije)*

3) *Until Kriterijum zaustavljanja*

Brzina određuje pravac i intenzitet u kojem će se čestica kretati. Parametri v_{min}, v_{max} predstavljaju minimalnu i maksimalnu brzinu čestice, odnosno njen intenzitet. Promena brzine čestice se vrši na sledeći način:

$$v_i(t)=w*v_i(t\text{-}1) + \rho_1*C_1*(p_i - x_i(t\text{-}1)) + \rho_2*C_2*(p_g - x_i(t\text{-}1))$$

ρ_1, ρ_2 –slučajno izabrane konstante iz [0,1]

C_1 - koeficijent kognitivnog učenja (uticaj iskustva čestice)

C_2 – koeficijent socijalnog učenja (iskustvo celog roja ili okoline)

w – parametar inercije koji kontroliše uticaj prethodnih brzina čestice na trenutno, $w \in [0,1]$

Za veće vrednosti w diversifikuje se pretraživanje, a za manje vrednosti w intenzivira se lokalno pretraživanje. Van den Bergh je pokazao da odnos između parametra inercije i koeficijenata socijalnog i kognitivnog učenja mora da zadovoljava sledeću relaciju:

$$\frac{C_1+C_2}{2} - 1 < w\ ,$$

inače čestice u roju mogu da imaju divergentno ciklino ponašanje [45].

Postoje i drugi načini za promenu brzine čestica [35]. Jedan od načina je:

$$v_i(t)=\chi*(\ v_i(t\text{-}1) + \rho_1*C_1*(p_i - x_i(t\text{-}1)) + \rho_2*C_2*(p_g - x_i(t\text{-}1))),$$

gde χ predstavlja koeficijent suženja, koji treba da zadovoljava sledeći uslov:

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4 * \varphi}\right|}, \qquad gde\ je\ \varphi = C_1 + C_2, \varphi > 4.$$

Parametri PSO heuristike su:

- Kriterijum zaustavljanja: broj iteracija, najbolje rešenje nije promenjeno nakon određenog broja iteracija, ili kombinacija ova dva kriterijuma
- Koeficijent kognitivnog učenja
- Koeficijent socijalnog učenja
- Parametar inercije
- Parametri v_{min}, v_{max}

Svi ovi parametri mogu da utiču na rešenje heuristike. Rezultati testiranja nekih parametara PSO heuristike biće kasnije prikazani u sekciji 9.1.

5.1. Implementacija heuristike zasnovane na roju čestica

Pri implementaciji PSO heuristike rešenje je kodirano na isti način kao i pri implementaciji iterativnog lokalnog pretraživanja. Funkcija prilagođenosti se i ovde koristi da bi se odredio kvalitet rešenja i računa se na identičan način kao i kod iterativnog lokalnog pretraživanja. S obzirom da je rešenje binarno kodirano, korišćena je funkcija $Sigmoid(v)$: $[v_{min},\ v_{min}] \rightarrow (0,1)$ koja služi da bi normalizovala brzinu [23]:

$$Sigmoid(v) = \frac{1}{1 + e^{-v}}$$

Prilikom implementacije korišćeno je iskustvo celog roja, a za promenu brzine čestice uzeta je sledeća formula:

$$v_i(t)=w*v_i(t-1) + \rho_1*C_1*(p_i - x_i(t-1)) + \rho_2*C_2*(p_g - x_i(t-1))$$

Nakon izvršenja petlje poziva se CPLEX rešavač koji računa funkciju cilja za rešenje koje ima najveću funkciju prilagođenosti. Rezultat CPLEX-a se uzima za aproksimaciju optimalnog rešenja. Na osnovu rezultata testiranja parametara, koje će biti kasnije opisano u sekciji 9.2, izabrani si sledeći parametri: broj čestica je 25, 35 i 40 u prvoj, drugoj i trećoj grupi instanci, respektivno, $w=0.529$, $C_1 = 1.49445$, $C_2 = 1.49445$, $v_{min} = -10$, $v_{max} = 10$. Predložene vrednosti parametara korišćene su i pri rešavanju drugih problema PSO heuristikom [14, 17].

6. Heuristika simuliranog kaljenja

Heuristika simulirano kaljenje - SA je jedna od heuristika koja se zasniva na principu lokalnog pretraživanja [10, 25, 44]. Ime je dobila zbog analogije sa procesom kaljenja nekog rastopljenog materijala do dostizanja čvrstog stanja. To se postiže oponašanjem učinka temperature u kretanju čestica kroz stanja različitih energija i postepenom smanjivanju te temperature. Ako se unutrašnja energija smanjuje, menja se i prihvata nova konfiguracija atom, dok se u slučaju povećanja energije, ta energija prihvata sa određenom verovatnoćom prema Bolcmanovom termodinamičkom zakonu. Od početne konfiguracije atoma, ovaj proces se ponavlja, pri čemu se postepeno smanjuje energija. Kada materijal dostigne stanje minimalne energije, on očvrsne. Kod heuristike koja se zasniva na ovom principu trenutnoj konfiguraciji atoma odgovara jedno dopustivo rešenje, unutrašnjoj energiji odgovara vrednost funkcije cilja, dok promeni energetskog stanja odgovara prelaz s prethodnog na naredno rešenje.

SA je iterativna metoda pretraživanja koja u svakoj iteraciji na slučajan način bira jedno rešenje iz okoline tekućeg rešenja. Ako je to rešenje bolje, onda ono postaje novo tekuće rešenje. Ako je novo rešenje lošije od tekućeg, ono ipak postaje tekuće rešenje, ali sa određenom verovatnoćom. Verovatnoća prihvatanja lošijeg rešenja vremenom opada i obično zavisi od parametra koji se naziva temperatura, što daje sličnost sa procesom kaljenja čelika. U početku je ta verovatnoća velika, pa će u cilju prevazilaženja lokalnog optimuma lošije rešenje biti prihvaćeno. Pred kraj izvršavanja algoritma verovatnoća prihvatanja lošijeg rešenja je jako mala i to se verovatno neće ni desiti, jer se smatra da je optimum dosegnut ili se nalazi blizu najboljeg posećenog rešenja, pa se izbegava pogoršanje tekućeg rešenja. Početno rešenje se bira na slučajan način. SA heuristika može se opisati sledećim pseudokodom:

Algoritam 5. SA heuristika

1. $s = s_0$, na slučajan način izgenerisati početno rešenje;
2. $T = T_{max}$, početna temperatura;
3. $Repeat$
   a. $Repeat$ (pri fiksiranoj temeraturi)
      i. $s' = lokalna\ pretraga(s)$;
      ii. $\Delta E = f(s') - f(s)$;
      iii. $if(\Delta E \leq 0)$

      /* prihvata se bolje rešenje */

      $then\ \ s = s'$;

      /* prihvata se lošije rešenje sa određenom verovatnoćom */

      else prihvati s' sa verovatnoćom $e^{\frac{-\Delta E}{T}}$;

   Until Kriterijum zaustavljanja

   $b.\quad T = update(T)$

$Until\ \ T > T_{min}$

Parametri SA su:

- T_{min}, T_{max}
- Kriterijum zaustavljanja
- Funkcija *update (T)* koja smanjuje temperaturu T

T_{min} i T_{max} predstavljaju minimalnu i maksimalnu temperaturu i trebalo bi da zavise od veličine funkcije cilja. Za kriterijum zaustavljanja uzima se broj iteracija koji treba da zavisi od veličine prostora pretraživanja i od veličine okoline definisane u okviru lokalnog pretraživanja. Broj iteracija može da varira u zavisnosti od temperature. Obično se broj iteracija povećava kako opada temperatura. Funkcija *update(T)* definisati na različite načine, kao na primer:

- $T_{k+1} = T_k - const$
- $T_{k+1} = \alpha * T_k\ ,\ \ 0 < \alpha < 1$
- $T_{k+1} = \ \alpha^k * T_k, 0 < \alpha < 1.$

6.1. Implementacija heuristike simuliranog kaljenja

Pri implementaciji SA heuristike rešenje je kodirano na isti način kao i pri implementaciji iterativnog lokalnog pretraživanja. Koristi se funkcija prilagođenosti da bi se odredio kvalitet rešenja i računa se na identičan način kao i kod iterativnog lokalnog pretraživanja. Okoline rešenja takođe se definiše isto kao i kod implementacije iterativnog lokalnog pretraživanja.

Nakon testiranja parametara, ustanovljeno je da sledeći parametri daju najbolja rešenja za problem uspostavljanja uslužnih objekata: $T_{max} = 70,\ T_{min} = 10,\ T_{k+1} = 0.95 * T_k$ (odnosno α=0.95). Predložene vrednosti parametara korišćene su i pri rešavanju drugih problema SA heuristikom [16, 21, 38]. Kriterijum zaustavljanja pri jednoj temperaturi je određen broj iteracija koji se povećava za I/5 pri svakoj promeni temperature. Broj iteraciji pri temperaturi T_{max}, odnosno početan broj iteracija je 25, 200 i 10*I u prvoj, drugoj i trećoj grupi instanci, respektivno.

7. Hibridni algoritmi

Primenom heuristika na različite probleme ispostavilo se da heuristike često daju dobra rešenja, ali ne i optimalna rešenja. Kombinovanjem drugih optimizacionih tehnika i heuristika može proizvesti novu efikasniju i bolju metodu za rešavanje velikih instanci. Hibridni algoritmi predstavljaju kombinaciju dve ili više heuristika u cilju postizanja boljih rezultata pri rešavanju nekog problema. Cilj hibridizacije je da dobre osobine dve ili više heuristika sjedine u jednu novu, složeniju i efikasniju metodu [2, 3, 9, 28, 29, 34]. Praktično bi svake dve heuristike mogle da se hibridizuju. U ovom radu predložena je hibridizacija heuristike zasnovane na roju čestica - PSO, iterativnog lokalnog pretraživanja - ILS i heuristike simuliranog kaljenja - SA. Ovde će biti predstavljena dva hibridna algoritma: hibridni algoritam PSO-ILS i hibridni algoritam PSO-SA.

7.1. Hibridni algoritam PSO-ILS

PSO i ILS mogu se hibridizovati na više načina. Jedan od načina je da se, nakon završetka PSO algoritma, pokrene i ILS algoritam čije je početno rešenje rezultat dobijen nakon PSO-a. Drugi način, koji je ovde implementiran, je da se PSO i ILS prepliću. U svakom koraku PSO algoritma, pokušava se poboljšanje 5 najboljih čestica pomoću ILS algoritma. Broj iteracija u ILS algoritmu je ograničen na 10. Program se zaustavlja ako najbolje rešenje nije promenjeno nakon I*I*I/4 iteracija. Ostali parametri, koji se odnose na PSO algoritam, su isti kao i pri implementaciji heuristike PSO.

7.2. Hibridni algoritam PSO-SA

Kao i u prethodnom hibridnom algoritmu, postoje više načina da se hibridizuju algoritmi PSO i SA. Ovde je predložen algoritam u kojem se, nakon pozivanja algoritma PSO, poziva algoritam SA, čije je početno rešenja rezultat PSO algoritma. Parametri koji se odnose na PSO algoritam su identični, osim što se algoritam završava ako najbolje rešenje nije promenjeno nakon 5*I*I iteracija. Parametri koji se odnose na SA algoritam su identični, osim što u ovom slučaju parametar α ima vrednost 0.9.

8. Rezultati

8.1. Opis instanci

Zbog nemogućnosti pronalaženja već kreiranih instanci za dati problem, implementiran je program za generisanje instanci. Sve instance sadrže dve vrste (nivoa) uslužnih objekata, mali i veliki objekti. Cena otvaranja malog objekta na nekoj lokaciji je slučajan broj između 100 i 200, dok je cena otvaranja velikog objekta slučajan broj između 300 i 400. Koordinate objekata i potrošača su slučajni brojevi od 0 do 100, dok je afinitet čvora potrošača prema objektu jednak *const / rastojanje do objekta*. Broj korisnika u jednom čvoru koji zahtevaju neku uslugu je slučajan broj između 1 i 5, dok je kapacitet jedne usluge u nekom objektu za male objekte slučajan broj između 350 i 550, a za velike slučajan broj između 650 i 850. Budžet za izgradnju svih objekata je 1000. Za generisanje slučajnih brojeva korišćena je funkcija *rand()*. Ostali podaci su parametri instanci. Instance su podeljene u 3 kategorije u kojima je broj čvorova korisnika jednak 100, 1000 i 2000.

U prvoj kategoriji:

- broj potencijalnih lokacija može da uzme vrednost 5, 7 ili 10
- (MCR_1, MCR_2) može da uzme vrednost (5,10), (7,15) ili (10,20)
- r može da uzme vrednost 20, 25 ili 30

U drugoj kategoriji:

- broj potencijalnih lokacija može da uzme vrednost 10, 20 ili 30
- (MCR_1, MCR_2) može da uzme vrednost (10,20), (20,40) ili (50,100)
- r može da uzme vrednost 20, 25 ili 30

U trećoj kategoriji:

- broj potencijalnih lokacija može da uzme vrednost 30, 50 ili 70
- (MCR_1, MCR_2) može da uzme vrednost (10,20), (20,40) ili (50,100)
- r može da uzme vrednost 20, 25 ili 30

8.2. Rezultati testiranja

Instance su prvo testirane korišćenjem IBM ILOG CPLEX 12.5 rešavača, kako bi se dobila optimalna rešenja, uz vremensko ograničenje od 30 min. Na svim instancama, iz prve i druge kategorije, CPLEX je uspeo da pronađe optimalno rešenja, dok na većini instanci iz treće kategoriji to nije bio slučaj. Svi prethodno opisani algoritmi pokretani su po 10 puta sa različitim početnim

vrednostima generatora slučajnih brojeva (seed-ovima) za svaku instancu kako bi se dobili što reprezentativniji rezultati i kako bi se ispitala stabilnost algoritma. U tabelama 1-6 su prikazani rezultati testiranja sve tri grupe instanci. Kolona *rezultat* kod heurističkih metoda predstavlja najbolje rešenje koje je algoritam dobio nakon 10 pokretanja, dok kolona *vreme* predstavlja prosečno vreme rada (u sekundama) algoritma u tih 10 pokretanja. Vrednost *agap* predstavlja srednje odstupanje (u procentima) rešenja od rešenja dobijenog CPLEX rešavačem i računa se po sledećoj formuli:

$$agap = \frac{1}{10} * \sum_{i=1}^{10} gap_i\,, \quad gde\ je\ gap_i = 100 * \frac{|result_i - cplex_{result}|}{cplex_{result}}$$

Ukoliko CPLEX nije uspeo da dobije optimalno rešenje za 30 min, tada *agap* predstavlja srednje odstupanje rešenja od najboljeg rešenja koje je dobijeno algoritmom i računa se po sledećoj formuli:

$$agap = \frac{1}{10} * \sum_{i=1}^{10} gap_i\,, \quad gde\ je\ gap_i = 100 * \frac{|result_i - result_{best}|}{result_{best}}$$

Vrednost *σ* predstavlja devijaciju dobijenih rešenja i računa se na sledeći način:

$$\sigma = \sqrt{\frac{1}{10} * \sum_{i}^{10} (gap_i - agap)^2}$$

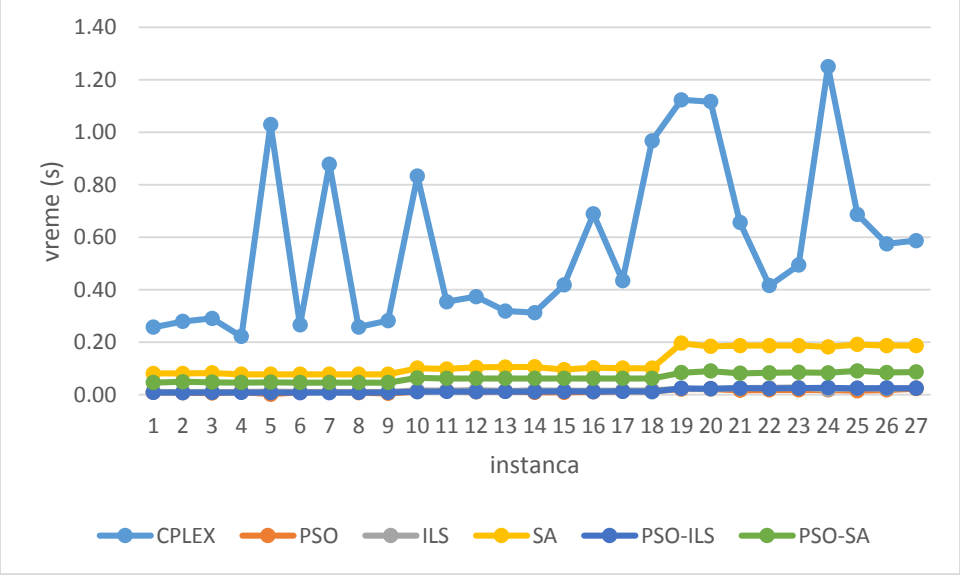
U tabelama 1 i 2 su prikazani rezultati testiranja svih algoritama na instancama iz prve kategorije. Na grafiku 1 i grafiku 2 može se videti grafički prikaz poređenja vremena izvršavanja i srednjih odstupanja predloženih algoritama na instancama iz prve kategorije. U ovoj grupi instanci algoritmi SA, PSO-ILS, PSO-SA daju optimalno rešenje na svim instancama, dok algoritmi PSO i ILS samo na jednoj instanci ne dostižu optimalno rešenja. Najbrži algoritam na ovoj grupi instanci je PSO, a nakon njega slede PSO-ILS, ILS, PSO-SA i SA, respektivno. Najstabilniji algoritam na ovoj grupi instanci je PSO-ILS, a nakon njega slede SA, PSO-SA, ILS i PSO, respektivno.

Tabela 1. Rezultati testiranja instanci iz prve grupe

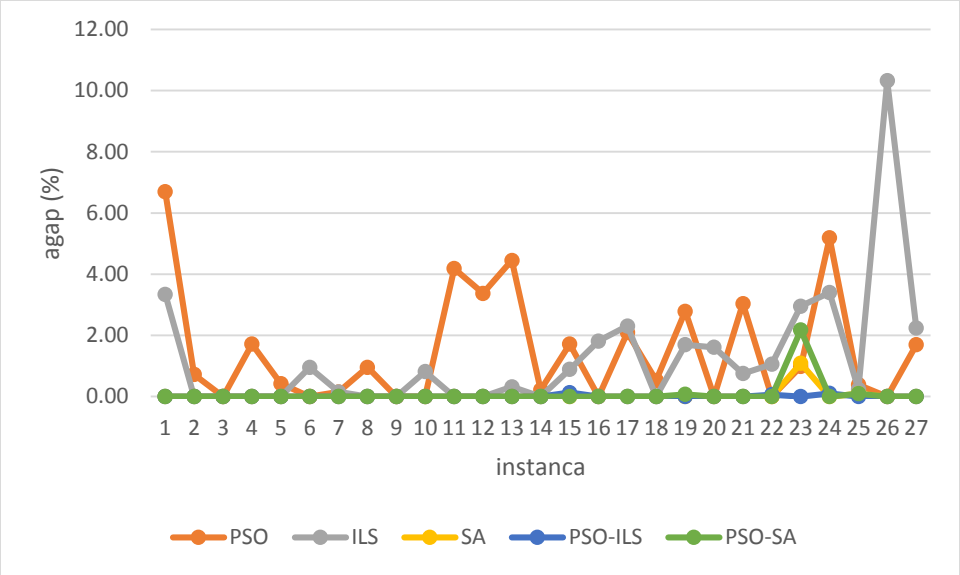
br.	naziv instance \ rezultati	CPLEX		PSO				ILS			
		rezultat	vreme	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ
1.	flsdp_5_100_5_10_20	240.09	0.26	**240.09**	0.01	6.69	3.35	**240.09**	0.01	3.34	4.09
2.	flsdp_5_100_5_10_25	284.11	0.28	**284.11**	0.01	0.72	1.45	**284.11**	0.01	0.00	0.00
3.	flsdp_5_100_5_10_30	190.92	0.29	**190.92**	0.01	0.00	0.00	**190.92**	0.01	0.00	0.00
4.	flsdp_5_100_7_15_20	171.84	0.22	**171.84**	0.01	1.71	3.57	**171.84**	0.01	0.00	0.00
5.	flsdp_5_100_7_15_25	166.81	1.03	**166.81**	0.00	0.42	0.42	166.81	0.01	0.00	0.00
6.	flsdp_5_100_7_15_30	255.32	0.27	**255.32**	0.01	0.00	0.00	**255.32**	0.01	0.95	1.46
7.	flsdp_5_100_10_20_20	191.03	0.88	**191.03**	0.01	0.15	0.23	**191.03**	0.01	0.15	0.23
8.	flsdp_5_100_10_20_25	158.61	0.26	**158.61**	0.01	0.96	1.47	**158.61**	0.01	0.00	0.00
9.	flsdp_5_100_10_20_30	205.81	0.28	**205.81**	0.00	0.00	0.00	205.81	0.01	0.00	0.00
10.	flsdp_7_100_5_10_20	165.16	0.83	**165.16**	0.01	0.00	0.00	165.16	0.02	0.82	1.63
11.	flsdp_7_100_5_10_25	286.86	0.35	**286.86**	0.02	4.18	1.51	**286.86**	0.02	0.00	0.00
12.	flsdp_7_100_5_10_30	253.43	0.37	**253.43**	0.01	3.37	1.69	253.43	0.02	0.00	0.00
13.	flsdp_7_100_7_15_20	195.32	0.32	**195.32**	0.01	4.45	2.13	**195.32**	0.01	0.32	0.64
14.	flsdp_7_100_7_15_25	199.13	0.31	**199.13**	0.01	0.22	0.67	199.13	0.02	0.00	0.00
15.	flsdp_7_100_7_15_30	221.76	0.42	**221.76**	0.01	1.71	2.50	221.76	0.02	0.89	0.76
16.	flsdp_7_100_10_20_20	193.40	0.69	**193.40**	0.01	0.00	0.00	193.40	0.02	1.81	2.77
17.	flsdp_7_100_10_20_25	239.13	0.44	**239.13**	0.01	2.09	1.28	239.13	0.02	2.30	1.81
18.	flsdp_7_100_10_20_30	306.31	0.97	**306.31**	0.01	0.53	0.73	306.31	0.02	0.00	0.00
19.	flsdp_10_100_5_10_20	183.77	1.12	182.43	0.02	2.79	2.48	**183.77**	0.02	1.69	2.33
20.	flsdp_10_100_5_10_25	259.27	1.12	**259.27**	0.02	0.00	0.00	**259.27**	0.02	1.61	1.78
21.	flsdp_10_100_5_10_30	299.70	0.66	**299.70**	0.02	3.04	1.68	299.70	0.03	0.75	1.32
22.	flsdp_10_100_7_15_20	194.80	0.42	**194.80**	0.02	0.00	0.00	194.80	0.03	1.06	1.50
23.	flsdp_10_100_7_15_25	300.23	0.49	**300.23**	0.02	0.98	2.96	300.23	0.03	2.95	1.40
24.	flsdp_10_100_7_15_30	260.54	1.25	**260.54**	0.02	5.19	1.86	**260.54**	0.02	3.40	1.90
25.	flsdp_10_100_10_20_20	213.15	0.69	**213.15**	0.02	0.39	1.16	213.15	0.03	0.24	0.39
26.	flsdp_10_100_10_20_25	318.93	0.58	**318.93**	0.02	0.00	0.00	312.27	0.02	10.32	3.31
27.	flsdp_10_100_10_20_30	344.15	0.59	**344.15**	0.02	1.70	3.94	344.15	0.03	2.24	1.76

Tabela 2. Rezultati testiranja instanci iz prve grupe

br.intance \ rezultati	SA				PSO-ILS				PSO-SA			
	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ
1.	240.09	0.08	0.00	0.00	**240.09**	0.01	0.00	0.00	240.09	0.05	0.00	0.00
2.	284.11	0.08	0.00	0.00	**284.11**	0.01	0.00	0.00	284.11	0.05	0.00	0.00
3.	190.92	0.08	0.00	0.00	**190.92**	0.01	0.00	0.00	190.92	0.05	0.00	0.00
4.	171.84	0.08	0.00	0.00	**171.84**	0.01	0.00	0.00	171.84	0.05	0.00	0.00
5.	166.81	0.08	0.00	0.00	**166.81**	0.01	0.00	0.00	166.81	0.05	0.00	0.00
6.	255.32	0.08	0.00	0.00	**255.32**	0.01	0.00	0.00	255.32	0.05	0.00	0.00
7.	191.03	0.08	0.00	0.00	**191.03**	0.01	0.00	0.00	191.03	0.05	0.00	0.00
8.	158.61	0.08	0.00	0.00	**158.61**	0.01	0.00	0.00	158.61	0.05	0.00	0.00
9.	205.81	0.08	0.00	0.00	**205.81**	0.01	0.00	0.00	205.81	0.05	0.00	0.00
10.	165.16	0.10	0.00	0.00	**165.16**	0.01	0.00	0.00	165.16	0.06	0.00	0.00
11.	286.86	0.10	0.00	0.00	**286.86**	0.01	0.00	0.00	286.86	0.06	0.00	0.00
12.	253.43	0.10	0.00	0.00	**253.43**	0.01	0.00	0.00	253.43	0.06	0.00	0.00
13.	195.32	0.11	0.00	0.00	**195.32**	0.01	0.00	0.00	195.32	0.06	0.00	0.00
14.	199.13	0.11	0.00	0.00	**199.13**	0.01	0.00	0.00	199.13	0.06	0.00	0.00
15.	221.76	0.10	0.00	0.00	**221.76**	0.01	0.13	0.40	221.76	0.06	0.00	0.00
16.	193.40	0.10	0.00	0.00	**193.40**	0.01	0.00	0.00	193.40	0.06	0.00	0.00
17.	239.13	0.10	0.00	0.00	**239.13**	0.01	0.00	0.00	239.13	0.06	0.00	0.00
18.	306.31	0.10	0.00	0.00	**306.31**	0.01	0.00	0.00	306.31	0.06	0.00	0.00
19.	183.77	0.20	0.00	0.00	**183.77**	0.02	0.00	0.00	183.77	0.08	0.07	0.22
20.	259.27	0.18	0.00	0.00	**259.27**	0.02	0.00	0.00	259.27	0.09	0.00	0.00
21.	299.70	0.19	0.00	0.00	**299.70**	0.02	0.00	0.00	299.70	0.08	0.00	0.00
22.	194.80	0.19	0.00	0.00	**194.80**	0.02	0.06	0.19	194.80	0.08	0.00	0.00
23.	300.23	0.19	1.09	1.33	**300.23**	0.02	0.00	0.00	300.23	0.09	2.18	1.09
24.	260.54	0.18	0.00	0.00	**260.54**	0.03	0.11	0.16	260.54	0.08	0.00	0.00
25.	213.15	0.19	0.10	0.29	**213.15**	0.02	0.00	0.00	213.15	0.09	0.10	0.29
26.	318.93	0.19	0.00	0.00	**318.93**	0.03	0.00	0.00	318.93	0.08	0.00	0.00
27.	344.15	0.19	0.00	0.00	**344.15**	0.02	0.00	0.00	344.15	0.09	0.00	0.00



Grafik 1. Prikaz vremena za instance iz prve grupe



Grafik 2. Prikaz srednjeg odstupanja za instance iz prve grupe

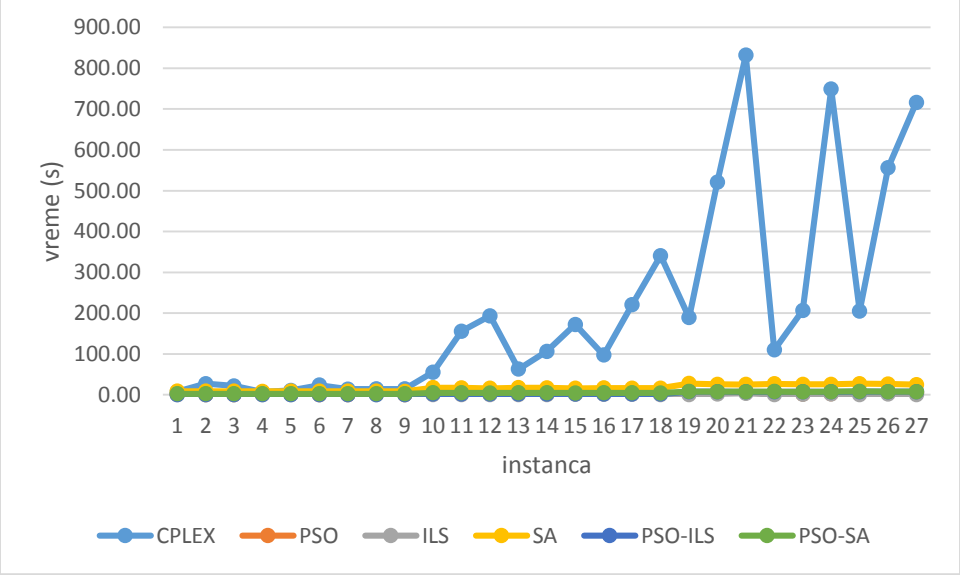
U tabelama 3 i 4 su prikazani rezultati testiranja svih algoritama na instancama iz druge grupe. Na grafiku 3 i grafiku 4 može se videti grafički prikaz poređenja vremena izvršavanja i srednjih odstupanja predloženih algoritama na instancama iz druge grupe. Algoritam PSO na većini instanci dostiže optimalno rešenje, dok algoritmi ILS i SA svuda dostiže optimalno rešenje, osim na dve instance. Hibridni algoritam PSO-ILS samo na tri instance ne dostiže optimalno rešenje, dok PSO-SA sa svim instancama iz druge grupe dostiže optimalno rešenje. Najbrži algoritam na ovoj grupi instanci je PSO, a nakon njega slede PSO-ILS, PSO-SA, ILS i SA, respektivno. Najstabilniji algoritam je SA, a nakon njega slede PSO-SA, ILS, PSO-ILS i PSO, respektivno.

Tabela 3. Rezultati testiranja instanci iz druge grupe

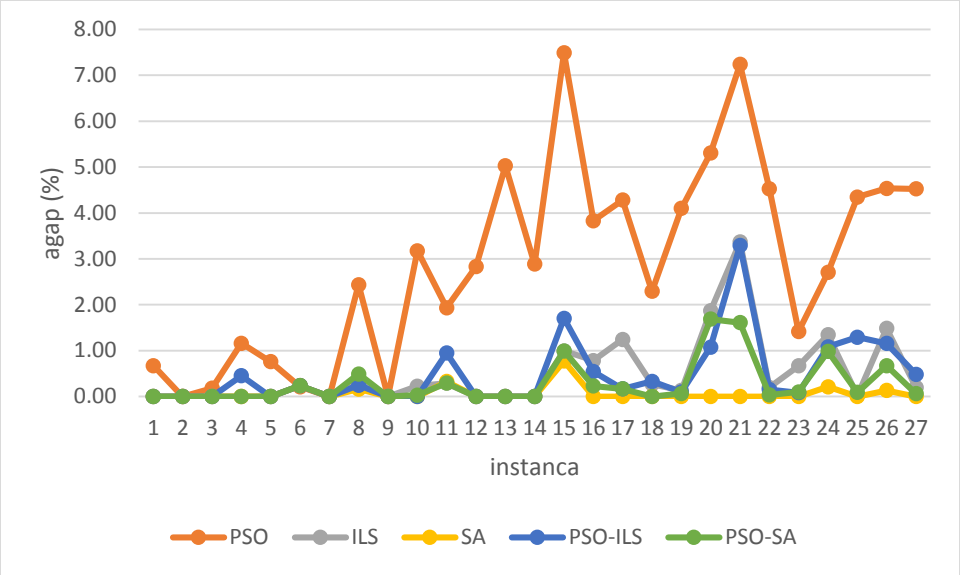
br.	naziv instance \ rezultati	CPLEX		PSO				ILS			
		rezultat	vreme	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ
1.	flsdp_10_1000_10_20_20	2011.79	7.33	**2011.79**	0.28	0.67	0.78	2011.79	1.36	0.00	0.00
2.	flsdp_10_1000_10_20_25	1987.14	26.89	**1987.14**	0.26	0.00	0.00	1987.14	1.45	0.00	0.00
3.	flsdp_10_1000_10_20_30	2717.81	21.61	**2717.81**	0.27	0.18	0.09	2717.81	1.36	0.00	0.00
4.	flsdp_10_1000_20_40_20	2125.38	6.63	**2125.38**	0.26	1.16	1.16	2125.38	1.18	0.00	0.00
5.	flsdp_10_1000_20_40_25	2099.45	10.41	**2099.45**	0.32	0.76	0.54	2099.45	1.32	0.00	0.00
6.	flsdp_10_1000_20_40_30	2555.79	23.58	**2555.79**	0.25	0.21	0.07	2549.76	1.33	0.24	0.00
7.	flsdp_10_1000_50_100_20	1837.68	13.79	**1837.68**	0.28	0.00	0.00	1837.68	1.18	0.00	0.00
8.	flsdp_10_1000_50_100_25	2137.57	14.09	**2137.57**	0.27	2.44	1.59	2137.57	1.39	0.33	0.40
9.	flsdp_10_1000_50_100_30	2425.40	13.91	**2425.40**	0.27	0.00	0.00	2425.40	1.47	0.00	0.00
10.	flsdp_20_1000_10_20_20	2136.76	55.25	**2136.76**	1.62	3.18	2.49	2136.76	5.25	0.23	0.38
11.	flsdp_20_1000_10_20_25	2361.83	155.57	**2361.81**	1.81	1.93	0.95	2361.81	4.58	0.30	0.10
12.	flsdp_20_1000_10_20_30	2837.75	193.33	**2837.75**	1.63	2.83	1.15	2837.75	5.26	0.00	0.00
13.	flsdp_20_1000_20_40_20	2133.28	62.72	**2133.28**	1.38	5.03	2.51	2133.28	5.90	0.00	0.00
14.	flsdp_20_1000_20_40_25	2515.27	106.13	**2515.27**	1.68	2.88	1.94	2515.27	6.18	0.00	0.00
15.	flsdp_20_1000_20_40_30	2632.00	172.21	2457.64	1.33	7.49	0.29	**2632.00**	5.36	0.99	0.33
16.	flsdp_20_1000_50_100_20	2153.43	97.23	2123.35	1.83	3.83	1.84	**2153.43**	5.63	0.79	0.45
17.	flsdp_20_1000_50_100_25	2575.87	220.65	**2575.87**	1.69	4.28	1.50	2575.87	4.57	1.24	1.11
18.	flsdp_20_1000_50_100_30	2682.99	340.56	**2682.99**	1.92	2.29	1.74	2682.99	5.14	0.27	0.33
19.	flsdp_30_1000_10_20_20	2097.95	188.91	**2097.95**	5.14	4.10	2.13	2097.95	11.99	0.13	0.15
20.	flsdp_30_1000_10_20_25	2601.18	520.51	2463.18	4.13	5.31	0.00	**2601.18**	10.87	1.87	0.96
21.	flsdp_30_1000_10_20_30	2737.47	831.91	**2737.47**	4.54	7.24	3.77	2737.47	10.75	3.37	1.92
22.	flsdp_30_1000_20_40_20	2249.40	109.49	2172.42	5.79	4.52	0.72	**2249.40**	11.41	0.20	0.17
23.	flsdp_30_1000_20_40_25	2546.77	206.65	2542.43	4.93	1.42	0.56	**2546.77**	12.63	0.67	0.52
24.	flsdp_30_1000_20_40_30	2712.82	748.67	2667.16	4.87	2.70	1.07	**2712.82**	12.23	1.35	0.97
25.	flsdp_30_1000_50_100_20	2142.68	204.87	**2142.68**	4.61	4.35	2.85	2142.68	11.98	0.02	0.02
26.	flsdp_30_1000_50_100_25	**2416.43**	556.11	2329.63	5.04	4.53	0.67	2386.09	11.39	1.48	0.13
27.	flsdp_30_1000_50_100_30	2744.59	716.24	2643.77	5.12	4.52	0.43	**2744.59**	12.11	0.20	0.07

Tabela 4. Rezultati testiranja instanci iz druge grupe

br.intance \ rezultati	SA				PSO-ILS				PSO-SA			
	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ
1.	2011.79	8.63	0.00	0.00	**2011.79**	0.17	0.00	0.00	2011.79	2.34	0.00	0.00
2.	1987.14	8.58	0.00	0.00	**1987.14**	0.17	0.00	0.00	1987.14	2.32	0.00	0.00
3.	2717.81	8.39	0.00	0.00	**2717.81**	0.17	0.00	0.00	2717.81	2.27	0.00	0.00
4.	2125.38	8.19	0.00	0.00	**2125.38**	0.17	0.46	0.91	2125.38	2.22	0.00	0.00
5.	2099.45	8.33	0.00	0.00	**2099.45**	0.17	0.00	0.00	2099.45	2.26	0.00	0.00
6.	2549.76	8.36	0.24	0.00	**2549.76**	0.19	0.24	0.00	2549.76	2.28	0.24	0.00
7.	1837.68	8.55	0.00	0.00	**1837.68**	0.20	0.00	0.00	1837.68	2.32	0.00	0.00
8.	2137.57	8.37	0.16	0.33	**2137.57**	0.18	0.25	0.38	2137.57	2.29	0.49	0.40
9.	2425.40	8.20	0.00	0.00	**2425.40**	0.22	0.00	0.00	2425.40	2.23	0.00	0.00
10.	2136.76	17.10	0.00	0.00	**2136.76**	1.57	0.00	0.00	2136.76	4.67	0.03	0.10
11.	2354.09	16.89	0.33	0.00	**2361.81**	1.82	0.95	0.91	2361.81	4.44	0.30	0.10
12.	2837.75	15.71	0.00	0.00	**2837.75**	1.55	0.00	0.00	2837.75	4.24	0.00	0.00
13.	2133.28	17.11	0.00	0.00	**2133.28**	1.62	0.00	0.00	2133.28	4.56	0.00	0.00
14.	2515.27	16.63	0.00	0.00	**2515.27**	1.62	0.00	0.00	2515.27	4.42	0.00	0.00
15.	2632.00	15.71	0.77	0.51	**2632.00**	1.59	1.70	1.26	2632.00	4.21	0.99	0.33
16.	2153.43	16.47	0.00	0.00	**2153.43**	1.55	0.55	0.36	2153.43	4.40	0.23	0.35
17.	2575.87	16.14	0.00	0.00	**2575.87**	1.56	0.17	0.50	2575.87	4.46	0.17	0.50
18.	2682.99	15.86	0.00	0.00	**2682.99**	1.63	0.33	0.65	2682.99	4.25	0.00	0.00
19.	2097.95	27.37	0.00	0.00	**2097.95**	7.40	0.10	0.23	2097.95	7.60	0.07	0.10
20.	2601.18	25.27	0.00	0.00	**2601.18**	7.18	1.08	1.08	2601.18	7.27	1.69	0.85
21.	2737.47	25.01	0.00	0.00	2710.42	7.01	3.29	1.20	**2737.47**	7.48	1.61	1.35
22.	2249.40	26.69	0.00	0.00	**2249.40**	7.13	0.15	0.10	2249.40	7.89	0.04	0.06
23.	2546.77	25.54	0.00	0.00	**2546.77**	7.40	0.09	0.09	2546.77	7.00	0.10	0.08
24.	2712.82	25.36	0.21	0.14	**2712.82**	7.12	1.08	0.54	2712.82	7.18	0.98	0.83
25.	2142.68	26.85	0.00	0.00	2119.80	7.34	1.29	0.19	**2142.68**	8.30	0.09	0.23
26.	2416.43	26.18	0.13	0.17	**2416.43**	7.20	1.16	0.67	2416.43	7.25	0.67	0.56
27.	2744.59	24.64	0.00	0.00	**2744.59**	7.27	0.48	0.53	2744.59	7.21	0.06	0.10



Grafik 3. Prikaz vremena za instance iz druge grupe



Grafik 4. Prikaz srednjeg odstupanja za instance iz druge grupe

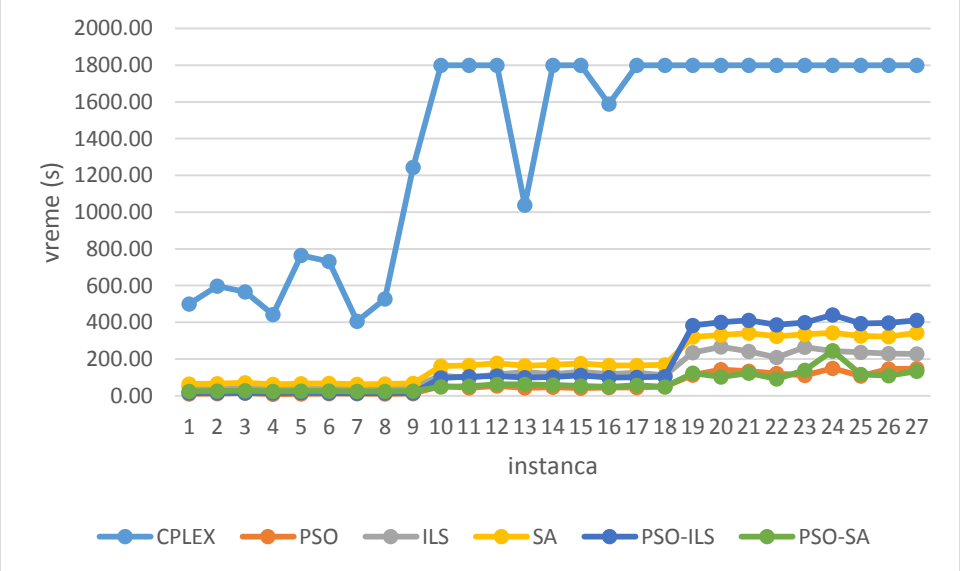
U tabelama 5 i 6 su prikazani rezultati testiranja svih algoritama na instancama iz treće grupe. Na grafiku 5 i grafiku 6 može se videti grafički prikaz poređenja vremena izvršavanja i srednjih odstupanja predloženih algoritama na instancama iz treće grupe. Na nekim instancama iz ove grupe CPLEX nije uspeo da pronađe optimalno rešenje za 30 min (vreme=t.l.). Algoritam PSO samo na nekoliko instanci dostiže iste vrednosti kao i CPLEX, dok na ostalim instancama daje lošija rešenja. Na instancama na kojima CPLEX ne pronalazi optimalno rešenje, algoritmi ILS, SA, PSO-ILS i PSO-SA pronalaze bolja rešenja. Algoritam ILS daje najbolje rezultate, a nakon njega slede PSO-SA, SA, PSO-ILS i PSO, respektivno. Najbrži algoritam je PSO, a nakon njega slede PSO-SA, ILS, SA i PSO-ILS, respektivno. Najstabilniji algoritam na ovoj grupi instanci je SA, a nakon njega slede PSO-SA, ILS, PSO-ILS i PSO, respektivno.

Tabela 5. Rezultati testiranja instanci iz treće grupe

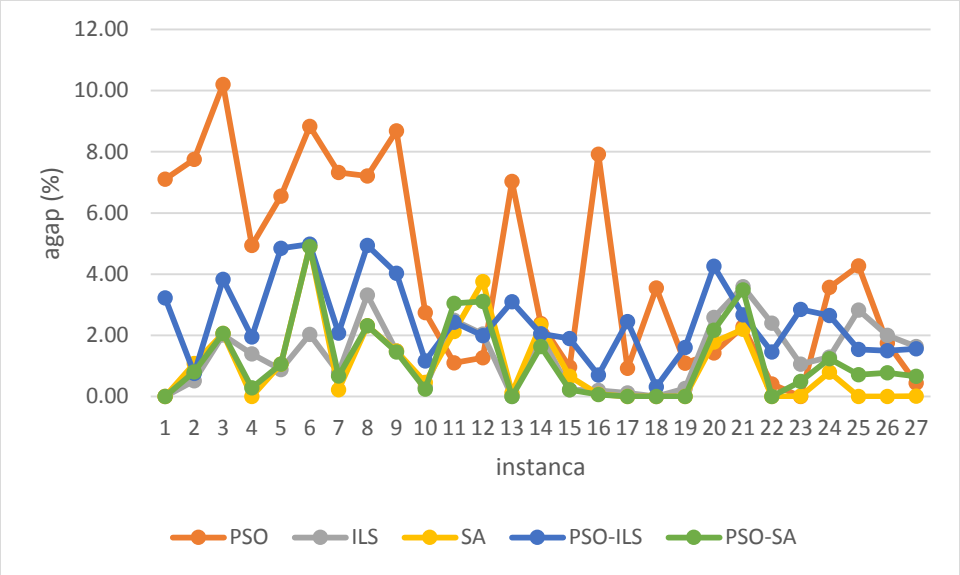
br.	naziv instance \ rezultati	CPLEX		PSO				ILS			
		rezultat	vreme	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ
1.	flsdp_30_2000_10_20_20	3942.18	499.05	**3942.18**	9.05	7.11	1.30	3942.18	47.76	0.00	0.00
2.	flsdp_30_2000_10_20_25	**4366.73**	597.71	4194.35	11.81	7.75	2.95	4360.00	48.21	0.52	0.38
3.	flsdp_30_2000_10_20_30	**4596.27**	566.00	4379.43	14.42	10.20	2.05	4526.08	51.65	2.01	0.16
4.	flsdp_30_2000_20_40_20	4114.72	441.46	**4114.72**	8.89	4.94	0.99	4114.72	44.48	1.40	1.38
5.	flsdp_30_2000_20_40_25	4399.59	765.03	4211.92	9.79	6.55	1.41	**4399.62**	42.98	0.88	0.88
6.	flsdp_30_2000_20_40_30	**4802.64**	732.07	4443.44	11.13	8.83	0.93	4802.62	47.76	2.03	0.95
7.	flsdp_30_2000_50_100_20	**4135.81**	406.17	3883.21	11.49	7.33	0.87	4135.76	39.07	0.78	0.46
8.	flsdp_30_2000_50_100_25	**4611.33**	526.89	4360.02	9.43	7.21	1.28	4504.40	42.24	3.32	0.96
9.	flsdp_30_2000_50_100_30	**4664.55**	1243.27	4410.86	10.81	8.69	1.28	4620.25	49.08	1.49	0.22
10.	flsdp_50_2000_10_20_20	4461.38	t.l.	4199.30	53.86	2.74	1.23	**4465.68**	109.01	0.31	0.23
11.	flsdp_50_2000_10_20_25	4342.53	t.l.	4275.14	42.60	1.10	0.47	**4513.75**	97.37	2.49	1.85
12.	flsdp_50_2000_10_20_30	4333.55	t.l.	4210.72	53.06	1.27	1.36	**4442.38**	116.87	2.05	1.56
13.	flsdp_50_2000_20_40_20	**4531.23**	1037.68	4246.23	43.35	7.04	0.76	4530.96	129.08	0.01	0.00
14.	flsdp_50_2000_20_40_25	4435.56	t.l.	4257.52	46.86	2.39	1.88	**4495.89**	118.20	2.04	1.16
15.	flsdp_50_2000_20_40_30	**4499.15**	t.l.	4172.88	41.02	0.96	0.48	4497.26	130.18	0.22	0.34
16.	flsdp_50_2000_50_100_20	4465.70	1588.60	4259.80	44.61	7.92	1.17	**4465.70**	118.26	0.21	0.17
17.	flsdp_50_2000_50_100_25	4568.32	t.l.	4356.23	44.89	0.93	0.65	**4641.94**	125.46	0.12	0.26
18.	flsdp_50_2000_50_100_30	**4501.78**	t.l.	4472.08	49.36	3.55	1.87	4488.06	111.28	0.00	0.00
19.	flsdp_70_2000_10_20_20	4417.95	t.l.	4192.59	112.45	1.09	0.55	**4499.04**	233.87	0.27	0.28
20.	flsdp_70_2000_10_20_25	4500.72	t.l.	4319.30	142.34	1.42	1.74	**4753.27**	266.71	2.59	1.00
21.	flsdp_70_2000_10_20_30	4527.78	t.l.	4492.48	132.85	2.29	1.25	**4701.01**	242.42	3.59	1.32
22.	flsdp_70_2000_20_40_20	4257.98	t.l.	3988.89	121.98	0.41	0.18	**4513.80**	209.32	2.40	0.81
23.	flsdp_70_2000_20_40_25	4537.19	t.l.	4372.12	110.62	0.00	0.00	**4725.70**	264.52	1.06	0.58
24.	flsdp_70_2000_20_40_30	4268.87	t.l.	4268.87	148.50	3.57	1.82	**4336.01**	242.84	1.29	1.12
25.	flsdp_70_2000_50_100_20	4596.97	t.l.	4337.47	107.72	4.27	1.45	**4605.35**	237.06	2.83	1.55
26.	flsdp_70_2000_50_100_25	4595.11	t.l.	4367.32	145.60	1.75	1.07	**4731.54**	228.91	2.00	0.86
27.	flsdp_70_2000_50_100_30	4481.78	t.l.	4435.80	147.74	0.44	0.98	**4590.38**	228.40	1.64	0.94

Tabela 6. Rezultati testiranja instanci iz treće grupe

br.intance \ rezultati	SA				PSO-ILS				PSO-SA			
	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ	rezultat	vreme	agap	σ
1.	3942.18	63.36	0.00	0.00	**3942.18**	15.40	3.22	1.51	3942.18	23.59	0.00	0.00
2.	4319.44	64.84	1.08	0.00	**4360.00**	14.75	0.75	0.39	4360.00	25.03	0.80	0.43
3.	4501.58	70.52	2.06	0.00	**4596.26**	15.91	3.83	2.45	4501.58	26.61	2.06	0.00
4.	4114.72	62.18	0.00	0.00	**4114.72**	15.41	1.95	1.53	4114.72	23.34	0.29	0.45
5.	4399.62	65.87	1.06	0.86	**4399.62**	16.63	4.84	2.95	4399.62	26.27	1.06	0.86
6.	**4726.71**	67.88	4.91	1.11	4640.89	15.23	4.98	0.94	4640.89	24.99	4.90	0.77
7.	4135.76	61.58	0.23	0.45	4108.84	13.49	2.07	0.95	**4135.76**	23.40	0.67	0.55
8.	4504.40	63.47	2.32	0.00	4467.20	14.29	4.93	1.26	**4504.40**	23.99	2.32	0.00
9.	4620.25	67.25	1.49	0.22	4608.83	15.28	4.03	1.31	**4620.25**	25.46	1.45	0.23
10.	4465.68	160.96	0.46	0.21	4460.13	96.56	1.16	0.72	**4465.68**	47.46	0.24	0.24
11.	4513.75	165.14	2.13	1.39	4385.61	104.16	2.43	0.90	**4513.75**	51.37	3.04	1.87
12.	4442.38	176.59	3.76	1.37	4295.34	107.82	1.99	1.26	**4442.38**	62.43	3.11	1.51
13.	4530.96	164.09	0.12	0.34	4462.44	99.33	3.10	0.75	**4530.96**	61.33	0.01	0.00
14.	4469.86	168.64	2.34	1.16	4398.94	102.16	2.06	0.76	**4469.86**	58.90	1.64	1.11
15.	4497.26	176.23	0.66	0.22	4497.26	110.85	1.89	1.01	**4497.26**	53.16	0.22	0.34
16.	4462.42	164.33	0.07	0.00	4465.70	98.33	0.71	0.56	**4465.70**	47.70	0.07	0.02
17.	4697.79	164.84	0.00	0.00	4636.53	100.88	2.45	1.34	**4697.79**	57.76	0.00	0.00
18.	4488.06	168.94	0.00	0.00	4488.06	104.22	0.32	0.44	**4488.06**	48.41	0.00	0.00
19.	4499.04	319.57	0.00	0.00	**4502.46**	382.32	1.60	0.68	4499.04	123.96	0.00	0.00
20.	4680.01	331.35	1.78	0.66	4753.27	399.48	4.26	2.01	**4753.27**	103.00	2.17	1.14
21.	4701.01	341.51	2.20	1.30	4620.85	410.37	2.67	1.29	**4701.01**	122.99	3.49	1.28
22.	4513.80	323.11	0.00	0.00	4374.11	385.61	1.46	0.95	**4513.80**	91.58	0.00	0.00
23.	4734.79	333.19	0.02	0.06	4650.84	398.83	2.85	1.49	**4734.79**	138.11	0.50	0.44
24.	**4330.35**	343.04	0.80	0.56	4330.35	440.35	2.65	1.48	4326.67	245.34	1.23	0.89
25.	4605.35	324.47	0.00	0.00	4438.85	393.59	1.54	0.52	**4605.35**	116.75	0.72	0.88
26.	4731.54	322.08	0.00	0.00	4564.92	395.99	1.50	0.85	**4731.54**	108.77	0.78	0.56
27.	4547.42	343.50	0.01	0.02	4444.76	410.14	1.57	0.69	**4561.40**	134.29	0.66	0.39



Grafik 5. Prikaz vremena za instance iz treće grupe



Grafik 6. Prikaz srednjeg odstupanja za instance iz treće grupe

9. Testiranje parametara

U ovom delu biće reči o rezultatima dobijenim promenom različitih parametara opisanih heuristika. Testiranje parametara je veoma značajno jer dobra kombinacija parametara neke heuristike može da ubrza konvergenciju heuristike. Razmatrane su različite vrednosti parametara i testirane na jednom podskupu instanci. Da bi se dobili što reprezentativniji rezultati, svi algoritmi su pokrenuti 5 puta sa različitim početnim vrednostima generatora slučajnih brojeva (seed-ovima) za svaku instancu. U tabelama 7-13 predstavljeni su rezultati testiranja parametara. Kolona *rezultat* predstavlja najbolje rešenje koje je algoritam dobio nakon 5 pokretanja, dok kolona *vreme* predstavlja prosečno vreme (u sekundama) rada algoritma u 5 pokretanja. Na osnovu rezultata testiranja predložene su vrednosti parametara za problem uspostavljanja uslužnih objekata. Da bi se došlo do predloženih parametara, napravljeni su test primeri sa instancama iz različitih grupa instanci, odnosno iz svake od grupa uzete su po tri instance.

9.1. Testiranje parametara heuristike simuliranog kaljenja

Promena parametara u heuristici može da ubrza njenu konvergenciju. Kako bi se dobili što bolji rezultati, testirani su sledeći parametri:

- Parametar α
- Parametri T_{max} i T_{min}

Prilikom testiranja ovih parametara broj iteracija u okviru svake temperature se povećava za *I/5* prilikom svakog smanjenja temperature. Broj iteracija u okviru najviše temperature, odnosno početan broj iteracija, zavisi od veličine instance. U prvoj grupi instanci broj iteracija je 25, u drugoj 200, a u trećoj *10*I*.

9.1.1. Parametar α

Parametar α određuje kojom brzinom se smanjuje temperatura. Za manje vrednosti parametara temperatura se brže smanjuje, pa je i vreme izvršavanja kraće, dok se za veće vrednosti parametara temperatura sporije smanjuje, pa je i vreme izvršavanja duže. Prilikom testiranja ovog parametra ostali parametri su fiksirani: $T_{max} = 10$, $T_{min} = -10$. Rezultati testiranja parametra α prikazani su u tabeli 7:

Tabela 7. Testiranje parametra α

naziv instance \ α	0.85		0.9		0.95		0.99	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.916	0.015	190.916	0.031	190.916	0.078	190.916	1.123
flsdp_7_100_7_15_25	199.126	0.031	199.126	0.046	199.126	0.109	199.126	1.45
flsdp_10_100_10_20_20	211.083	0.046	212.254	0.078	213.148	0.187	213.148	3.214
flsdp_10_1000_20_40_20	2125.38	2.121	2125.38	3.463	2125.38	7.378	2125.38	59.892
flsdp_20_1000_50_100_25	2575.87	3.884	2575.87	6.427	2575.87	14.71	2575.87	152.443
flsdp_30_1000_10_20_30	2693.46	5.694	2737.47	9.594	2737.47	22.978	2737.47	280.644
flsdp_30_2000_50_100_20	4089.45	15.974	4135.76	26.769	4135.76	61.573	4135.76	665.762
flsdp_50_2000_10_20_30	4245.11	46.02	4253.68	76.533	4295.34	177.59	4295.34	1926.59
flsdp_70_2000_20_40_25	4721.73	85.129	4734.79	144.612	4734.79	328.754	4734.79	3640.59

Na osnovu dobijenih rezultata za vrednost ovog parametra uzeto je α=0.95.

9.1.2. Parametri T_{max} i T_{min}

Parametri T_{max} i T_{min} određuju minimalnu i maksimalnu temperaturu. Prilikom testiranja ovih parametara, parametar α je fiksiran na vrednost 0.95. Rezultati testiranja parametara T_{max} i T_{min} prikazani su u tabeli 8:

Tabela 8. Testiranje parametara T_{max} i T_{min}

naziv instance \ (Tmax, Tmin)	(60, 10)		(60, 20)		(70, 10)		(70, 20)		(80, 10)		(80, 20)	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.9	0.1	190.9	0.0	190.9	0.1	190.9	0.1	190.9	0.1	190.9	0.1
flsdp_7_100_7_15_25	199.1	0.1	199.1	0.1	199.1	0.1	199.1	0.1	199.1	0.1	199.1	0.1
flsdp_10_100_10_20_20	213.1	0.2	213.1	0.1	213.1	0.2	212.3	0.1	212.3	0.2	213.1	0.1
flsdp_10_1000_20_40_20	2125.4	6.7	2125.4	4.0	2125.4	7.3	2125.4	4.6	2125.4	8.0	2125.4	5.2
flsdp_20_1000_50_100_25	2575.9	13.3	2575.9	7.6	2575.9	14.8	2575.9	8.9	2575.9	16.2	2575.9	10.2
flsdp_30_1000_10_20_30	2655.7	20.7	2640.5	11.7	2737.5	23.1	2611.6	13.7	2737.5	25.4	2651.2	15.6
flsdp_30_2000_50_100_20	4089.5	55.4	4089.5	32.1	4135.8	62.4	4135.8	36.8	4135.8	67.9	4089.5	42.3
flsdp_50_2000_10_20_30	4199.6	159.3	4253.7	91.4	4295.5	177.0	4221.8	105.7	4295.3	195.5	4284.7	121.4
flsdp_70_2000_20_40_25	4600.0	300.6	4725.7	171.5	4734.8	334.7	4734.8	198.9	4734.8	369.0	4734.8	230.1

Na osnovu rezultata testiranja za vrednost ovih parametara uzeto je $T_{max} = 70$ i $T_{min} = 10$.

9.2. Testiranje parametara heuristike zasnovane na roju čestica

Rezultat heuristike zasnovane na roju čestica zavisi od puno parametara. U cilju postizanja što boljih rezultata testiranu su sledeći parametri:

- Broj čestica u roju
- v_{min}, v_{max}

- Parametar inercije
- Koeficijenti kognitivnog i socijalnog učenja

Parametar kriterijum zaustavljanja nije testiran. Program se zaustavlja ako najbolje rešenje nije promenjeno nakon *6*I*I* iteracija, gde *I* predstavlja broj potencijalnih uslužnih objekata.

9.2.1. Parametar broj čestica

U opštem slučaju parametar broj čestica zavisi od problema koji se rešava, pa se mora dobiti eksperimentalno. Prilikom testiranja ovog parametra, ostali parametri ove heuristike su fiksirani: $v_{min} = -10,\quad v_{max} = 10,$ *w=0.529,* $C_1 = 1.49445,\ C_2 = 1.49445.$ Rezultati testiranja parametra broj čestica u roji su prikazani u tabeli 9:

Tabela 9. Testiranje parametra broj čestica

naziv instance \ broj čestica	25		30		35		40	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.916	0	190.916	0.015	190.916	0.015	190.916	0.015
flsdp_7_100_7_15_25	199.58	0	199.58	0.031	199.58	0.015	199.58	0.015
flsdp_10_100_10_20_20	213.148	0.031	213.148	0.046	213.148	0.046	213.148	0.046
flsdp_10_1000_20_40_20	2077	0.296	2077	0.28	2125.38	0.421	2125.38	0.234
flsdp_20_1000_50_100_25	2420.84	0.811	2421.87	0.858	2451.32	0.811	2451.32	2.106
flsdp_30_1000_10_20_30	2475.29	5.085	2475.29	3.822	2505.86	4.243	2505.86	7.082
flsdp_30_2000_50_100_20	3730.33	6.084	3749.17	5.132	3780.66	6.24	3922.7	9.937
flsdp_50_2000_10_20_30	3736.41	34.663	4027.06	44.787	4040.37	29.78	4210.72	37.986
flsdp_70_2000_20_40_25	4372.12	62.228	4372.12	59.966	4372.12	142.849	4372.12	80.683

Različit broj čestica daje najbolje rezultate na različitim grupama instanci. Zato je za broj čestica predloženo: 25 čestica za prvu grupu instanci, 35 za drugu i 40 za treću grupu.

9.2.2. Parametri v_{min}, v_{max}

Brzina određuje pravac i intenzitet u kojem će se čestica kretati. Parametri v_{min}, v_{max} predstavljaju minimalnu i maksimalnu brzinu čestice, odnosno njen intenzitet. U opštem slučaju ovaj parametar zavisi od problema koji se rešava, pa se mora odrediti eksperimentalno. Prilikom testiranja ovog parametra ostali parametri ove heuristike su fiksirani: broj čestica je 25, 35 i 40 u zavisnosti od grupe instance, *w=0.529,* $C_1 = 1.49445,\ C_2 = 1.49445.$ Rezultati testiranja parametara v_{min} i v_{max} prikazani su u tabeli 10:

Tabela 10. Testiranje parametara v_{min}, v_{max}

naziv instance \ $[v_{min}, v_{max}]$	[-4, 4]		[-6, 6]		[-8, 8]		[-10, 10]	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.916	0.01	190.916	0.01	179.953	0	190.916	0.014
flsdp_7_100_7_15_25	199.126	0.01	181.27	0.01	199.126	0.01	199.126	0.021
flsdp_10_100_10_20_20	189.634	0.02	194.43	0.02	196.215	0.02	205.786	0.036
flsdp_10_1000_20_40_20	2032.31	0.15	2016.48	0.15	1983.34	0.15	2077	0.268
flsdp_20_1000_50_100_25	2356.77	1.24	2304.09	1.25	2360.82	1.11	2413.58	1.39
flsdp_30_1000_10_20_30	2323.41	2.25	2366.15	2.67	2366.15	3.97	2430.51	3.825
flsdp_30_2000_50_100_20	3517.25	4.79	3596.72	5.93	3596.72	8.95	3576.67	10.964
flsdp_50_2000_10_20_30	3696.92	35.93	3698.58	34.82	3696.92	29.18	4062.5	55.345
flsdp_70_2000_20_40_25	4277.3	169.57	4026.47	90.02	4026.47	55.13	4277.3	183.821

Na osnovu dobijenih rezultata za vrednost parametara uzete su: $v_{min} = -10$ i $v_{min} = 10$.

9.2.3. Parametar inercija

Parametar inercije kontroliše uticaj prethodne brzine na narednu brzinu. Za veće vrednosti parametra inercije, veći je uticaj prethodne brzine. Parametar inercije predstavlja i balans između diversifikacije i intenziviranje lokalnog pretraživanja. Za veće vrednosti parametra inercije pretraživanje se nastavlja u drugim delovima pretraživačkog prostora u odnosu na prethodnu poziciju. Ako je vrednost parametra inercije mala, tada se intenzivira lokalno pretraživanje. Prilikom testiranja ovog parametra, ostali parametri su fiksirani: broj čestica je 25, 35 i 40 u zavisnosti od grupe instance, $v_{min} = -10$, $v_{max} = 10$, $C_1 = 1.49445$, $C_2 = 1.49445$. Rezultati testiranja parametra inercija prikazani su u tabeli 11:

Tabela 11. Testiranje parametra inercije

naziv instance \par. inercije	0.529		0.6		0.7		0.8	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.916	0.015	190.916	0.015	190.916	0	190.916	0
flsdp_7_100_7_15_25	199.126	0	194.684	0.015	199.126	0.015	199.126	0.015
flsdp_10_100_10_20_20	213.148	0.046	205.047	0.015	199.58	0.015	196.215	0.031
flsdp_10_1000_20_40_20	2125.38	0.234	1957.97	0.156	2004.47	0.14	2077	0.156
flsdp_20_1000_50_100_25	2307.89	1.216	2307.89	1.107	2374.36	0.764	2356.77	0.936
flsdp_30_1000_10_20_30	2366.15	2.34	2366.15	3.354	2383.33	2.262	2269.59	3.978
flsdp_30_2000_50_100_20	3705.84	4.664	3596.72	9.11	3682.69	4.617	3596.72	14.352
flsdp_50_2000_10_20_30	4210.72	21.699	3788.09	30.248	3696.92	55.021	4200.56	19.874
flsdp_70_2000_20_40_25	4277.3	72.087	4277.3	92.258	4277.3	89.434	4054.15	60.84

Na osnovu dobijenih rezultata za vrednost parametra inercija predloženo je $w=0.529$.

9.2.4. Parametri kognitivnog i socijalnog učenja

Parametri kognitivnog i socijalnog učenja određuju smer čestice. Parametar kognitivnog učenja određuje afinitet čestice da ide ka svom do sada najbolje postignutom rešenju, dok parametar socijalnog učenja određuje afinitet čestice da ide ka do sada najboljem postignutom rešenju svih čestica. Prilikom testiranje ovog parametra ostali parametri su fiksirani: broj čestica je 25, 35 i 40 u zavisnosti od grupe instance, $v_{min} = -10$, $v_{max} = 10$, *w=0.529.* Rezultati testiranja parametara kognitivnog i socijalnog učenja prikazani su u tabeli 12:

Tabela 12. Testiranje parametara kognitivnog i socijalnog učenja

naziv instance \ (c1, c2)	(1.25, 1.25)		(1.25, 1.49445)		(1.49445, 1.25)		(1.49445,1.49445)	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.916	0.015	179.953	0	179.953	0.008	190.916	0
flsdp_7_100_7_15_25	199.126	0.015	159.128	0.015	162.939	0.012	199.126	0.015
flsdp_10_100_10_20_20	195.072	0.016	189.634	0.015	189.634	0.02	198.369	0.015
flsdp_10_1000_20_40_20	2016.97	0.14	1972.12	0.156	1957.67	0.153	2125.38	0.156
flsdp_20_1000_50_100_25	2209.11	0.795	2356.77	0.951	2209.11	0.791	2392.49	0.873
flsdp_30_1000_10_20_30	2475.29	2.246	2287.38	2.839	2325.69	4.029	2366.15	3.837
flsdp_30_2000_50_100_20	3596.72	5.99	3596.72	7.3	3583.32	4.606	3707.92	4.617
flsdp_50_2000_10_20_30	4210.72	19.89	3696.92	38.032	3565.83	20.188	3698.58	23.634
flsdp_70_2000_20_40_25	4026.47	65.488	4026.47	88.436	4026.47	83.934	4277.3	80.761

Na osnovu dobijenih rezultata predložene su sledeće vrednosti parametara kognitivnog i socijalnog učenja: $C_1 = 1.49445$, $C_2 = 1.49445$.

9.3. Testiranje parametara iterativnog lokalnog pretraživanja

U ovom radu je testiran samo parametar kriterijum zaustavljanja, dok način definisanja okoline i perturbacije rešenja nije menjan.

9.3.4 Parametar kriterijum zaustavljanja

Parametar kriterijum zaustavljanja je veoma bitan i od njega dosta zavisi rezultat heuristike. Algoritam se završava nakon određenog broja iteracija ili ako se najbolje rešenja nije menjalo određen broj iteracija. Tako npr. (200, 100) znači da se algoritam zaustavlja nakon 200 iteracija, ili ako se najbolje rešenje nije menjalo 100 iteracija. Rezultati testiranja parametra kriterijum zaustavljanja prikazani su u tabeli 13:

Tabela 13. Testiranje parametra kriterijum zaustavljanja

naziv instance \ kriterijum	(200, 100)		(1000, 500)		(5000, 2500)		(25000, 12500)	
	rezultat	vreme	rezultat	vreme	rezultat	vreme	rezultat	vreme
flsdp_5_100_5_10_30	190.916	0.015	190.916	0.031	190.916	0.14	190.916	0.624
flsdp_7_100_7_15_25	199.126	0.031	199.126	0.046	199.126	0.218	199.126	0.639
flsdp_10_100_10_20_20	213.148	0.015	213.148	0.062	213.148	0.265	213.148	0.873
flsdp_10_1000_20_40_20	2058.07	0.124	2125.38	0.593	2125.38	3.229	2125.38	9.5
flsdp_20_1000_50_100_25	2420.84	0.234	2457	1.076	2575.87	3.478	2575.87	28.407
flsdp_30_1000_10_20_30	2444.44	0.546	2551.98	1.779	2737.47	8.174	2737.47	28.953
flsdp_30_2000_50_100_20	3840.32	1.014	4023.68	2.543	4079.26	11.668	4135.76	64.131
flsdp_50_2000_10_20_30	3992.17	1.653	4073.29	5.553	4204.69	17.004	4367.12	151.601
flsdp_70_2000_20_40_25	4253.29	2.246	4268.9	6.411	4684.95	28.704	4734.79	149.838

Različiti kriterijumi daju najbolje rezultate na različitim grupama instanci. Zato je za kriterijum zaustavljanja predloženo da se algoritam zaustavlja nakon *I*J/4* iteracija ili ako se najbolje rešenje nije menjalo *I*J/8*.

10. Zaključak

U ovom radu je opisan problem uspostavljanja uslužnih objekata sa više nivoa na osnovu afiniteta korisnika, FLSDP problem i predstavljeni načini za rešavanje pomenutog problema. FLSDP je NP-težak problem pa egzaktne metode mogu dati optimalna rešenja u razumnom vremenu, samo na malim instancama. Zato su u ovom radu predložene heurističke metode za rešavanje ovog problema. Implementirane su tri heurističke metode: heuristika zasnovana na roju čestica - PSO, iterativno lokalno pretraživanje - ILS i heuristika simuliranog kaljenja - SA. U cilju objedinjavanja dobrih osobina heurističkih metoda u novu metodu, predložena su i dva hibridna algoritma PSO-ILS i PSO-SA. Hibridizacija algoritama je urađena na različite načine. Kod PSO-ILS hibridnog algoritma PSO i ILS se prepliću, dok se kod PSO-SA hibridnog algoritma SA nadovezuje na PSO algoritam.

U radu su testirani i parametri heuristika PSO, ILS i SA u cilju postizanja što boljih rešenja. Prikazani su rezultati testiranja parametara i predloženi optimalni parametri za rešavanje ovog problema. Dobar odabir parametara je veoma uticao na kvalitet dobijenih rešenja.

Algoritmi su testirani na tri grupe instanci. Generalno, heuristike su se pokazale veoma uspešnim pri rešavanju ovog problema. Na prvoj i drugoj grupi instanci skoro svi algoritmi dostižu optimalna rešenja na svim instancama iz grupe. Na instancama iz treće, odnosno na najvećim instancama, algoritmi ILS, SA, PSO-ILS i PSO-SA daju odlične rezultate, odnosno dosta bolje rezultate nego CPLEX za 30 min. Svi predloženi algoritmi pronalaze optimalna rešenja dosta brže od CPLEX-a. Nakon analize rezultata ne može se precizno utvrditi koja heuristika daje najbolje rezultate na svim instancama. ILS i PSO-SA algoritmi daju najbolja rešenja na instancama na kojima CPLEX nije uspeo da pronađe optimalno rešenje.

I na ovom problemu hibridizacija heuristika se pokazala veoma uspešnom, naročito hibridizacija PSO-SA. PSO algoritam pronalazi dobro početno rešenje za SA algoritam, što doprinosi efikasnosti PSO-SA algoritma.

Doprinos ovog rada ogleda se u činjenici da se prvi put predloženi algoritmi koristi za rešavanje FLSCP problema. Dalje unapređenje rada i istraživanja moguće je realizovati na neki od sledećih načina:

- paralelizacija PSO algoritma
- korišćenje CPLEX-a za računanje funkcije cilja
- implementacija nekih drugih heurističkih metoda i hibridnih algoritama.

11. Literatura

[1] Aarts E., Lenstra J. K., "Local Search in Combinatorial Optimization", *John Wiley & Sons* (1997).

[2] Abdinnour H., "A hybrid heuristic for the uncapacitated hub location problem", *European Journal of Operational Research* (1998), 106: 489-499.

[3] Acampora G., Loia V., Salerno S., Viriello A., "A hybrid evolutionary approach for solving the ontology alignment problem", *International Journal of Intelligent Systems* (2012), 27 (3): 189-216.

[4] Alayl K., Engin O., Alper D., "Using ant colony optimization to solve hybrid flow shop scheduling problems", *The international journal of advanced manufacturing technology* (2007), 35 (5-6): 541-550.

[5] Arakaki R. G. I., Lorena L. A. N., "A constructive genetic algorithm for the maximal covering location problem", *Proceedings of the 4th metaheuristics international conference* (2001), 13–17.

[6] Battiti R., Tecchiolli G., "The reactive tabu search", *ORSA journal on computing* (1994), 6 (2): 126-140.

[7] Berman O., Krass D., "The generalized maximal covering location problem", *Computers & Operations Research* (2002), 29: 563–581.

[8] Brandeau M. L., Chiu S. S., "An overview of representative problems in local research", *Management Sience* (1989), 645-674.

[9] Chen J., "A hybrid heuristic for the uncapacitated single allocation hub location problem", *OMEGA International Journal of Management Science* (2007), 35: 211-220.

[10] Chibante R., "Simulated Annealing Theory with Applications", *Published by Sclyo* (2010).

[11] Church R. L., ReVelle C., "The maximal covering location problem", *Papers of Regional Scence Association* (1974), 32, 101-118.

[12] Cook W. J., Cunningham W. H., Pulleyblank W. R., Schrijver A., "Combinatorial Optimization", *A Wiley-Interscience Publication* (1997), First edition.

[13] Cvetković D., Čangrilović M., Kovačević-Vujičić V., Dugošija Đ., Simić S., Vuleta J., "Kombinatorna optimizacija: Matematička teorija i algoritmi", *Društvo operacionih istraživača Jugoslavije* (1996).

[14] Das S., Abrahim A., Konar A., "Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives", *Department of Electronics and Telecommunication Engineering, Jadavpur University*.

[15] Davendra D., "Traveling Salesman Problem, Theory and Application", *InTech* (2010).

[16] Entezari R., "Some Simulated Annealing Applications And Card Shuffling", *Department of Statistics – University of Toronto* (2012).

[17] Dubinsky C., Millwater H. R., "Allocation of Testing Resources in Statistical Simulations Using Particla Swarm Optimization", *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum And Aerospace Exposition* (2011).

[18] Galvao R. D., Espejo L. G. A., Boffey B., "A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem", *European Journal of Operational Research* (2000), 124: 377–389.

[19] Galvao R. D., ReVelle C., "A Lagrangean heuristic for the maximal covering location problem", *European Journal of Operational Research* (1996), 88: 114–123.

[20] Glover F., Gary A. K., "Handbook of Metaheuristics", *Kluwer Academic Publishers* (2003).

[21] Guo J. Q., Zheng L., "A modified simulated annealing algorithm for estimating solute transport parameters in stream from tracer experiment data", *Environmental Modelling & Software 20* (2005), 811–815.

[22] Jung M. L., Young H. L., "Facility location and scale decision problem with customer preference", *Computers & Industrial Engineering* (2012), Volume 63, Issue 1, 184-191.

[23] Kennedy J., Eberhart R. C., "A discrete binary version of the particle swarm algorithm", *Proceedings of the IEEE International Conference on Systems* (1997), 4104-4108.

[24] Kennedy J., Eberhart R. C., "Particle Swarm Optimization", *Proceedings of IEEE Internationl Conference on Neural Networks* (1995), 1942-1948.

[25] Kirkpatrick S., Gelatt C. D., Vecchi M. P., "Optimization by Simulated Annealing", *Science 220* (1983).

[26] Lorena L. A. N., Pereira M. A., "A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition", *International Journal of Industrial Engineering* (2002), 9 (1): 57–67.

[27] Marić M., "An Efficient Genetic Algorithm For Solving The Multi-Level Uncapacitated Facility Location Problem", *Computing and Informatics* (2010), 29 (2): 183-201.

[28] Marić M., Stanimirović Z., Stanojević P., "An efficient memetic algorithm for the uncapacitated single allocation hub location problem", *Soft Computing* (2013).

[29] Marić M., Stanimirović Z., Božović S., "Hybrid metaheuristic method for determinig locations for long-term health care facilities", *Annals of Operations Research*, DIO: 10.1007/s10479-013-1313-8.

[30] Mazzola J. B., Neebe A. W., "Lagrangian relaxation based solution procedures for a multiproduct capacitated facility location problem with choice of facility type", *European Journal of Operations Research* (1999), 115: 285–299.

[31] Mladenović N., Hansen P., "Variable neighborhood search", *Computers & Operations Research* (1997), 24 (11): 1097-1100.

[32] Nemhauser G. L., Wolsey L. A., "Integer and combinatorial optimization", *Wiley* (1988).

[33] Nozick L. K., "The fixed charge facility location problem with coverage restrictions", *Transportation Research Part* (2001), 37: 281–296.

[34] Ong Y. S., Lim M. H., Zhu N., Wong K. W., "Classification of Adaptive Memetic Algorithms: A Comparative Study", *IEEE Transactions on Systems Man and Cybernetics* (2006), Part B. 36 (1): 141.

[35] Parsopoulos K. E., Vrahatis M. N., "Particle Swarm Optimization: Advances and Applications", *Information Science Reference* (2010).

[36] Paquete L., Stutzle T., "An Experimental Investigation of Iterated Local Search for Coloring Graphs*", Darmstadt University of Technology, Computer Science Departmant*.

[37] Pham D. T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M., "The Bees Algorithm", *Technical Note, Manufacturing Engineering Centre* (2005).

[38] Qin J., Ni L., Shi F., "Combined Simulated Annealing Algorithm for the Discrete Facility Location Problem", *The Scientific World Journal* (2012).

[39] ReVelle C., Scholssberg M., Williams J., "Solving the maximal covering location problem with heuristic concentration", *Computers & Operations Research* (2008), 35(2): 427–435.

[40] Schrijver A., "Theory of linear and integer programming", *John Wiley and Sons* (1998).

[41] Senne E. L. F., Pereira M. A., Lorena L. A. N., "A decomposition heuristic for the maximal covering location problem", *Advances in Operations Research* (2010), 1–12.

[42] Stanimirović Z., Marić M., Božović, S., Stanojević P., "An Efficient Evolutionary Algorithm for Locating Long-Term Care Facilities", *INFORMATION TECHNOLOGY AND CONTROL* (2012), 41 (1): 77-89.

[43] Taillard E., "Benchmarks for basic scheduling problems", *European Journal of Operational Research* (1993).

[44] Tan C. M., "Simulated Annealing", *In-Tech* (2006).

[45] Van den Bergh F., "An Analysis of Particle Swarm Optimizers, PhD Thesis", *Department of Computer Science, University of Pretoria, South Africa (2002)*.