ODABRANA POGLAVLJA OPTIMIZACIJE
SEMINARSKI RAD

PROBLEM MAKSIMALNOG POKRIVANJA SA PRIORITETNIM ODABIROM RESURSA

Student:
Anđela Donević
1023/2018

Nastavnici :
prof. dr Zorica Stanimirović
dr Stefan Mišković

Apstrakt

U radu je razmatrana verzija problema maksimalnog pokrivanja (Maximal covering location problem, MCLP). Rešava se problem uspostavljanja p novih ustanova, tako da se maksimizuje ukupna potražnja kupaca. Pritom, postojeće ustanove moraju ostati uspostavljene, a kupci na osnovu ličnih preferenci biraju koju će ustanovu posetiti . Problem je rešavan pomoću dve metaheurističke metode: Optimizacijom rojem čestica i Tabu pretraživanjem. Razmatrana je hibridizacija prethodno pomenutih metoda. Sve metode su testirane na generisanim instancama, a dobijeni rezultati su poređeni sa rezultatima CPLEX egzaktong rešavača.

Ključne reči: Problem maksimalnog pokrivanja, Optimizacija rojem čestica, Tabu pretraga

Opis problema

Posmatramo problem gde je potrebno uspostaviti p novih ustanova u gradu gde neki broj ustanova istog tipa već postoji. Cilj je da što veća ukupna potražnja bude zadovoljena. Kupci su slobodni da biraju koju će ustanovu posetiti. Njihov odabir može zavisiti od velikog broja različitih faktora kao što su starost, prosek primanja, obrazovanje i slično. Rezultat odabira kupca ne mora biti ustanova koja je njemu najbliža, ali smatra se da postoji najveće rastojanje R koje je kupac spreman da pređe kako bi došao do ustanove. Ukoliko je rastojanje između kupca K i ustanove U manje od R reći ćemo da je ustanova U moguća za kupca K, odnosno kupac K moguć za ustanovu U. Dalje, svaki kupac će imati listu lokacija ustanova poređanih po prioritetu posećivanja, a razmatraju se samo one ustanove koje su za datog kupca moguće.
Problem je formulisan kao problem lineanrog programiranja sa dva nivoa (Bilevel Maximal Covering Location Problem,) u daljem tekstu BMCLP. Prvi nivo koji odgovara problemu uspostavljanja ustanova i drugi nivo koji odgovara problemu pridruživanja kupaca ustanovama.

Matematička formulacija problema

U ovom delu razmatramo formulaciju problema BMCLP i njegovu preformulaciju koja ima samo jedan nivo (Single Level Maximal Covering Location Problem) u daljem teksu SLMCLP. Problemi BMCLP i SLMCLP imaju isto optimalno rešenje (Díaz et all, 2017).

Prvo ćemo definisati promenljive i parametre potrebne za formulaciju ovih problema.
Promenljive odlučivanja su:

$$x_{ij} = \begin{cases} 1, & \text{kupac } j \text{ je pridružen ustanovi uspostavljenoj na lokaciji } i \\ 0, & \text{inače} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{ustanova je uspostavljena na lokaciji } i \\ 0, & \text{inače} \end{cases}$$

gde je $i \in I, j \in J$.
Ostali parametri:

· **R** - maksimalno rastojanje koje je kupac spreman da pređe kako bi došao do ustanove

· **p** - broj novih ustanova koje treba da se uspostave na nekim od potencijalnih lokacija

· $\mathbf{I}_1$ - skup potencijalnih lokacija za uspostavljanje ustanova

· $\mathbf{I}_2$ - skup lokacija postojećih ustanova

· **I** - $I = I_1 \cup I_2$ skup svih lokacija za ustanove

· $\mathbf{J}_1$ - skup svih korisnika koji nisu opsluženi postojećim ustanovama

· $\mathbf{J}_2$ - skup svih korisnika koji su opsluženi postojećim ustanovama

· **J** - $J = J_1 \cup J_2$ skup svih korisnika

· $\mathbf{D}_j$ - Potražnja kupca j

· $\mathbf{g}_{ij}$ - koeficijent prioriteta kupca j ka ustanovi i

Kupac j je moguć za ustanovu i ukoliko je rastojanje između kupca j i ustanove i manje od R. Ustanova i je moguća za kupca j ukoliko je rastojanje između kupca j i ustanove i manje od R. Primetimo da ako je ustanova i moguća za kupca j onda je i kupac j moguć za ustanovu i.

Za svaku ustanovu razmatramo samo one kupce koji su za nju mogući. Slično, za svakog kupca razmatramo samo one ustanove koje su za njega moguće. Zato uvodimo skupove:

· **I(j)** skup svih ustanova i koje su moguće za kupca j, $j \in J$

· **J(i)** skup svih kupaca j koji su mogući za ustanovu i, $i \in I$

Na osnovu prethodno definisanih promenljivih, parametara i pomoćnih skupova problem linearnog programiranja dobija sledeći oblik:

BLMCLP:

$$\max_{x,y} \sum_{i \in I_1} \sum_{j \in J(i))} D_j x_{ij} \tag{1}$$

$$y_i = 1, \quad \forall i \in I_2 \tag{2}$$

$$\sum_{i \in I_1} y_i = p \tag{3}$$

$$y_i \in \{0, 1\} \quad \forall i \in I \tag{4}$$

Gde je x rešenje problema:

$$\max_{x} \sum_{i \in I} \sum_{j \in J(i))} g_{ij} x_{ij} \tag{5}$$

$$\sum_{i \in I(j)} x_{ij} = 1, \quad \forall j \in J_2 \tag{6}$$

$$x_{ij} \leq y_i, \quad \forall i \in I, j \in J(i) \tag{7}$$

$$\sum_{i \in I(j)} x_{ij} \leq 1, \quad \forall j \in J_1 \tag{8}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, j \in J \tag{9}$$

Problem uspostavljanja ustanova (prvi nivo): Funkcija cilja predstavlja ukupnu potražnju pokrivenu novouspostavljenim ustanovama.

Ograničenja predstavljaju sledeće:

(2) Garantuje da će postojeće ustanove ostati uspostavljene na svojim lokacijama

(3) Biće uspostavljeno tačno još p novih ustanova pored već postojećih

(4) Promenljiva odlučivanja uzima binarne vrednosti

Problem pridruživanja kupaca (drugi nivo): Funkcija cilja predstavlja zbir koeficijenta preferenci svih kupaca

(6) Potražnja koja je bila pokrivena postojećim ustanovama ostaće pokrivena i nakon uspostavljanja novih ustanova. Može biti pokrivena i novouspostavljenim ustanovama

(7) Garantuje da potražnja korisnika može biti pokrivena samo ustanovom koja je za njega moguća

(8) Neopslužen korisnik ostaje neopslužen ukoliko ne postoji ustanova koja je za njega moguća

(9) Promenljiva odlučivanja uzima binarne vrednosti

Pronalaženje rešenja problema u ovom obliku je teško, pa se prelazi na jednonivoski problem (SLMCLP) (Díaz et al, 2017). Prelazak se postiže dodavanjem ograničenja:

$$\sum_{s=k+1}^{|I(j)|} x_{i_s j} + y_{i_k} \leq 1, \forall j \in J, k \in \{1, ..., |I(j)| - 1\}, \tag{10}$$

gde je su $i_1, i_2, ..., i_{|I(j)|} \in I(j)$ takvi da važi sledeći poredak koeficijenata prioriteta:

$$g_{i_1,j} > g_{i_2,j} > ... > g_{i_{|I(j)|},j}$$

Ovo ograničenje garantuje da će kupac od svih ustanova koje su za njega moguće izabrati onu za koju ima najveći koeficijent prioriteta.
Dobija se problem u obliku

SLMCLP:

$$\max_{x,y} \sum_{i \in I_1} \sum_{j \in J(i))} D_j x_{ij} \tag{11}$$

$$\sum_{i \in I_1} y_i = p \tag{12}$$

$$y_i = 1, \quad \forall i \in I_2 \tag{13}$$

$$\sum_{i \in I(j)} x_{ij} = 1, \quad \forall j \in J_2 \tag{14}$$

$$\sum_{i \in I(j)} x_{ij} \leq 1, \quad \forall j \in J_1 \tag{15}$$

$$x_{ij} \leq y_i, \quad \forall i \in I, j \in J(i) \tag{16}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, j \in J \tag{17}$$

$$y_i \in \{0, 1\} \quad \forall i \in I \tag{18}$$

Načini rešavanja

Konstruisane su dve metaheurstičke metode, Tabu pretraga (eng. *Tabu search*), Optimizacija rojem čestica (eng. *Particle swarm optimization*) i njihova hibridizacija. Reprezentacija rešenja i način računanja funkcije cilja su isti u svim algoritmima koji su korišćeni.

Reprezentacija rešenja i funkcija cilja

Rešenje je predstavljeno nizom bitova dužine $|I|$. Jedinica na poziciji $i \in I$ označava da je na i-toj lokaciji uspostavljena ustanova. Među prvim $|I_1|$ bitova ima tačno p jedinica, to jest među svim potencijalim lokacijama uspostavljeno je tačno p ustanova. Preostalih $|I_2|$ bitova uvek je jednako jedinici, to jest postojeće ustanove uvek ostaju uspostavljene.

Na osnovu tako kodiranog rešenja vrši se alokacija kupaca korišćenjem pohlepnog (greedy) algoritma. Svaki kupac se dodeljuje uspostavljenoj i njemu dopustivoj ustanovi sa najvećom preferencom. Funkcija cilja računa se kao suma potražnje alociranih kupaca.

Optimizacija rojem čestica

Optimizacija rojem čestica pripada grupi populacionih metaheuristika i zasniva se na ponašanju jedinki u okviru populacije. Ova metaheuristika dozvoljava interakciju između jedinki populacije i time je omogućena razmena informacija. Svaka čestica odgovara jednom dopustivom rešenju. U svakoj iteraciji čestice pokušavaju da istovremeno poprave svoju poziciju koristeći svoje iskustvo (kognitivni aspekt) i iskustvo drugih čestica (socijalni aspekt). Za i-tu česticu roja u k-toj iteraciji poznato je sledeće:

- Trenutno rešenje $Y_i^{[k]}$ - trenutna pozicija čestice
- Brzina $v_i^{[k]}$ - pravac u kom bi se čestica kretala da nema uticaja roja
- Najbolja pozicija čestice p_i

U toku izvršavanja pamti se i najbolje rešenje celog roja (p_g).

Za početak, potrebno je generisati roj inicijalizacijom svih čestica u njemu (metod $initialize\_particle$ u pseudokodu, Slika 1.). Inicijalizacija čestice podrazumeva inicijalizaciju pozicije i inicijalizaciju brzine. Početna pozicija čestice se dobija tako što se na slučajan način od prvih $|I_1|$ bitova bira tačno p koji će imati vrednost jedan. Početna brzina je vektor dimenzije n, gde je n broj čestica u roju. Svaka koordinata u početnom vektoru brzine je slučajno odabrana vrednost iz intervala $[-V_{max}, V_{max}]$.
U svakoj iteraciji prolazi se kroz sve čestice u roju i realizuju se sledeći koraci:

1. Evaluacija čestice
2. Ažuriranje brzine čestice
3. Ažuriranje pozicije čestice

Evaluacija čestice podrazumeva računanje funkcije cilja (fitnes funkcije) za poziciju date čestice (metod $evaluate$ u pseudokodu, Slika 1). Ukoliko je pozicija date čestice bolja od globalno najbolje pozicije u smislu vrednosti funkcije cilja, vrši se ažuriranje najbolje globalne pozicije.

Ažuriranje brzine čestice (metod $update\_velocity$ u pseudokodu, Slika 1) podrazumeva računanje nove brzine pomoću formule:

$$v_i^{[k+1]} = \omega \cdot v_i^{[k]} + c_1\rho_1(p_i - Y_i^{[k]}) + c_2\rho_2(p_g - Y_i^{[k]}),$$

gde je ω parametar inercije (*inertia weight parameter*) koji kontroliše uticaj prethodih brzina čestice na trenutnu. ρ_1 i ρ_2 su slučajno izabrane vrednosti iz intervala $[0, 1]$. Parametar C_1 je koeficijent kontrole kognitivnog učenja, odnosno kontroliše koliko će čestica učiti iz svog iskustva. Parametar C_2 je koeficijent kontrole socijalnog učenja. Pomoću njega se reguliše globalni uticaj roja na učenje pojedinačne čestice. Neka od koodrinata tako dobijenog vektora brzine v_i^j može imati vrednost koja nije u intervalu $[-V_{max}, V_{max}]$ pa se vrši sledeća korekcija:

$$v_i^j = V_{max} \cdot \frac{v_i^j}{||v_i^j||}$$

Ovom korekcijom se ne menja pravac vektora v_i^j.

Ažuriranje pozicije čestice podrazumeva kretanje u dopustivom prostoru i prelazak u novo dopustivo rešenje na osnovu sračunate brzine (metod $update\_postition$ u pseudokodu, Slika 1). Bit Y_i^j, $j \in \{1, ..., |I_1|\}$ u novom rešenju $\hat{Y}_i$ biće jedan ako je sučajno generisana vrednost u manja od vrednosti sigmoidne funkcije $\sigma(v_i^j)$. Sigmoidna funkcija je definisana kao:

$$\sigma(x) = \frac{1}{1 + exp(-x)}$$

Značajna je jer ima vrednosti u intervalu $[0, 1]$ i pogodna je za modelovanje verovatnoće. Moguće je da ovako dobijeno rešenje $\hat{Y}_i$ ne pripada dopustivom skupu, to jest ima ili više ili manje od p uspostavljenih ustanova. U skladu sa time potrebno je izvršiti korekciju rešenja $\hat{Y}_i$ tako da ono postane dopustivo i ima tačno p uspostavljenih ustanova.

Algorithm 2: Particle swarm optimization

```
Input: num_particles, maxiter
Output: val_best
for i ∈ {j, ...num_particles} do
    swarm[j] = initialize_particle()
end
for i ∈ {0, ..., maxiter} do
    for j ∈ {1, ...num_particles} do
        val = evaluate(swarm[j])
        update_velocity(swarm[j])
        update_position(swarm[j])
        if val > val_best then
            val_best = val
        end
    end
end
return val_best
```

Slika 1: Optimizacija rojem čestica - pseudokod

Tabu pretraga

Tabu pretraga pripada grupi S-metaheuristika i zasnovana je na lokalnom pretraživanju. Uvodi se skup zabranjenih poteza T (*tabu skup*), koji sigurno ne donose poboljšanje prilikom pretrage. U svakoj iteraciji se proverava da li novo rešenje pripada tabu skupu. Ukoliko rešenje pripada tabu skupu prelazi se na sledeću iteraciju, čime se sprečava povratak pretrage u loše rešenje.

Prvi korak u izvršavanju ovog algoritma je inicijalizacija početnog rešenja. Inicijalizacija se vrši tako što se na slučajan način od prvih $|I_1|$ bitova bira tačno p koji će imati vrednost jedan. Time je slučajnim odabirom uspostavljano p ustanova.

U svakoj iteraciji formira se novo dopustivo rešenje (metod $new\_feasible$ u pseudokodu, Slika 2). Razmatrana su dva načina formiranja novog dopustivog rešenja: razmena tačno dva bita i razmena k bitova. Razmena dva bita se vrši tako što je od svih pozicija gde su bitovi jednaki jedinici slučajno odabere pozicija i. Na sličan način se među svim nula bitovima

odabere pozicija j. Zatim, bit na poziciji i postaje nula, a bit na poziciji j postaje jedinica. Razmena k bitova se vrši tako što se među svim nula bitovima slučajno odaberu k koji će postati jednice. Takođe, među bitovima koji imaju vrednost jedan, slučajno se biraju k koji postaju nule. Razmena k bitova kao metoda generisanja novog rešenja se koristi kada nakon $\beta * maxiter$ iteracija nije došlo do poboljšanja rešenja, gde je $maxiter$ maksimalni broj iteracija a $\beta \in (0, 1)$. U suprotnom se novo dopustivo rešenje dobija razmenom dva bita.

Pri svakom generisanju novog rešenja dobija se i kod koji će služiti za formiranje tabu skupa. S obzirom da algoritam zasnovan na lokalnoj pretrazi kod se formira samo prilikom generisanja novog rešenja razmenom tačno dva bita. Ako se menjaju bitovi na pozicijama i i j kod će imati vrednost (i, j). Intuicija iza ovoga je da ako razmena bitova i i j ne doprinese poboljšanju vrednosti funkcije cilja u n-toj iteraciji, neće doprineti ni u m-toj, gde $n < m$, pa se ovo može proglasiti zabranjenim potezom. Nakon generisanja novog rešenja i koda, kod se dodaje u tabu skup, ukoliko se u njemu već ne nalazi. Za tabu skup definisana je maksimalna dužina, koja će zavisiti od dimenzije instance. Ukoliko je tabu skup pun, izbacuje se prvi element iz liste, a novi kod se dodaje na kraj liste.

Algorithm 1: Tabu search

```
Input: maxiter
Output: val_best
Y_0 ← initialzie(), i ← 0, j ← 0
Y_current ← Y_0, val_current ← f(Y_current)
Y_best ← Y_current, val_best ← val_current
tabu_len ← ρ * |I_1|^2
k ← max([p/4], 1)
while i < maxiter do
    if j > β * maxiter then
        Y_new ← new_feasible(Y_current, k)
    else
        Y_new, code ← new_feasible(Y_current, 1)
    end
    if code ∉ tabu_set then
        i ← i + 1
        if val_new > val_current then
            j ← 0
            Y_current ← Y_new
            val_current ← val_new
            if val_best < val_new then
                Y_best ← Y_new
                val_best ← val_new
            end
        else
            j ← j + 1
        end
    else
        if |tabu_set| < tabu_len then
            add code to tabu_set
        else
            drop first element in tabu_set
            add code to tabu_set
        end
    end
end
return val_best
```

Slika 2: Tabu pretraživanje - pseudokod

Hibridizacija

Hibridizacija dve opisane metaheuristike izvršena je tako što se rešenja Tabu pretraživanja koriste kao početna rešenja za Optimizaciju rojem čestica. Parametar α određuje koliko će čestica iz roja biti inicijalizovano Tabu pretragom (metod $initialize\_particle\_tabu$ u pseudokodu, Slika 3). Ostale čestice inicijalizuju se kao u standardnoj Optimizaciji rojem čestica (metod $initialize\_particle$ u pseudokodu, Slika 3).

Algorithm 3: Hibrid

Input: $num\_particles, maxiter$
Output: val_{best}
for $i \in \{j, ...[\alpha * num\_particles]\}$ **do**
  $swarm[j] = initialize\_particle\_tabu(\beta * maxiter)$
end
for $i \in \{[\alpha * num\_particles] + +, ..., num\_particles\}$ **do**
  $swarm[j] = initialize\_particle()$
end
for $i \in \{0, ..., maxiter\}$ **do**
  for $j \in \{1, ...num\_particles\}$ **do**
    $val = evaluate(swarm[j])$
    $update\_velocity(swarm[j])$
    $update\_position(swarm[j])$
    if $val > val_{best}$ **then**
      $val_{best} = val$
    end
  end
end
$return\ val_{best}$

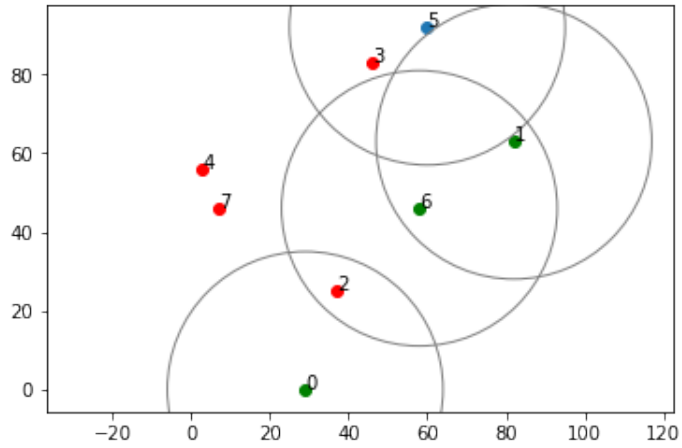
Slika 3: Hibridni algoritam - pseudokod

Instance

Instance su generisane u skladu sa preporukama iz literature (Díaz el al, 2017.)[1]. Radi jednostavnosti korišćene su samo celobrojne veličine.
Broj kupaca, broj ustanova, R i p biraju se proizvoljno tako da odgovaraju realnom problemu i koriste se kao parametri za generisanje ostalih podataka. Koordinate ustanova i kupaca birane su kao slučajne veličine iz kvadrata $[0, 100] \times [0, 100]$. Na osnovu broja kupaca i broja ustanova slučajno se biraju lokacije koje će odgovarati kupcima odnosno ustanovama. Definisane lokacije za ustanove dele se na lokacije već postojećih ustanova i lokacije potencijalnih ustanova. Podela se vrši tako da postojeće ustanove čine 33% ukupnog broja ustanova u slučaju malih instanci, odnosno 10% u slučaju srednjih i velikih instanci. Podela kupaca na opslužene i neopsluževe vrši se na osnovu rastojanja između kupaca i definisanih postojećih ustanova.
Koeficijenti prioriteta g_{ij} generisani su kao slučajne vrednosti iz trouglaste raspodele po preporuci iz (Cánovas et al, 2007) [2]. Na osnovu matrice $[g_{ij}]_{i,j}$ formira se matrica lokacija tako da je

u koloni $k, k \in \{0, |I| - 1\}$ dat niz lokacija ustanova sortiranih po prioritetima posećivanja g_{ik} kupca k. Potražnja svakog kupca je slučajna veličina iz normalne raspodele i pripada intervalu $[0, 100]$.

Prilikom implementacije algoritmama za generisanje instanci korišćen je programski jezik PYTHON, verzija 3.6. Kao rezultat izvršavanja ovih algoritama dobijaju se datoteke čiji su sadržaji generisane instance.



Slika 4: Grafički prikaz instance. Crveni kružići predstavljaju korisnike, plavi predstavljaju postojeće ustanove, a zeleni potencijalne nove ustanove. Poluprečnik svakog kruga je $R = 35$

Eksperimentalni rezultati i analiza

Generisano je 60 instanci za testiranje, koje su podeljene po veličini u tri grupe: instance malih(20), srednjih(20) i velikih(20) dimenzija. Za rešavanje problema linearnog programiranja egzaktinm rešavačom korišćen je ILOG CPLEX OPTIMIZATION STUDIO 12.8 u kombinaciji sa programskim jezikom C. Vreme izvršavanja ograničeno je na 2h. Za implementaciju metaheuristika korišćen je programski jezik PYTHON. Testiranje je vršeno na računaru sa *Intel Core i5-8265U* procesorom i 8 GB RAM memorije.

Sve vrednosti parametara koje su korišćene u implementiranim algoritmima utvrđene su eksperimentalnim putem (Tabela 1).

Za svaku instancu prikazano je optimalno rešenje pronađeno CPLEX rešavačem, kao i vreme izvršavanja. Svaka metahuristika pokrenuta je 15 puta za instance malih dimenzija (instance tipa S), 10 puta za instance srednjih dimenzija (instance tipa M) i 5 puta za instance velikih dimenzija (instance tipa L). Pritom su na nivou instance računate sledeće vrednosti:

- $best$ - Najbolje pronađeno rešenje
- t_{best} - Srednje početno vreme. Početno vreme je vreme za koje je algoritam prvi put došao do najboljeg rešenja.
- t_{tot} - Srednje ukupno vreme izvršavanja.
- $agap$ - Srednje odsutpanje dobijenog rešenja od optimalnog (u procentima)

Tabela 1: Vrednosti parametara koji su korišćeni u metodama

Metoda	Parametri	Vrednost parametra
Optimizacija rojem čestica	$maxiter$	50 za instance tipa S, 500 za instance tipa M, 1000 za instance tipa L
	n	50
	V_{max}	2000
	ω	10
	c_1	100
	c_2	100
Tabu pretraživanje	$maxiter$	10000 za instance tipa S, 30000 za instance tipa M, 70000 za instance tipa L
	β	0.2
Hibrid	$maxiter$	50 za instance tipa S, 500 za instance tipa M, 1000 za instance tipa L
	α	$\frac{1}{3}$
	β	0.2
	n	30
	V_{max}	200
	ω	2
	c_1	10
	c_2	10

- $agap_\sigma$ - Standardna devijacija odstupanja (u procentima)

Ukoliko je algoritam prilikom izvršavanja došao do optimalnog rešenja, među rezultatima umesto numeričke vrednosti stoji naznaka *opt*.

Instance malih dimenzija

Za instance malih dimenzija najbolje rezultate, u smislu najboljeg srednjeg vremena, dao je algoritam Tabu pretraživanja (Tabela 3). Za ovu grupu instanci Tabu pretraživanje pri svakom izvršavanju pronalazi optimalno rešenje. Izuzetak su instance $S18$ i $S20$ gde je pronađeno optimalno rešenje, ali je $agap$ i standardna devijacija različita od nule. Hibridni algoritam je za ovu grupu instanci dao najbolje rešenje u smislu optimalnosti rešenja, na šta ukazuju male vrednosti parametara $agap$ i $agap_\sigma$. Za instance $S1$-$S19$ on pri svakom izvršavanju pronalazi optimalno rešenje, a za instancu $S20$ ima najbolje vrednosti $agap$ i $agap_\sigma$ parametara u poređenju sa ostalim algoritmima (Tabela 2). Metod optimizacije rojem čestica za 18 instanci uvek pronalazi optimalno rešenje (Tabela 2). Za instance $S18$ i $S20$ je pronađeno optimalno rešenje sa vrednostima $agap$ i $agap_\sigma$ većim od vrednosti koje Tabu pretraga dala za te instance.

Instance srednjih dimenzija

Za instance srednjih dimenzija najbolje rezultate, u smislu optimalnosti pronađenog rešenja, dao je hibridni algoritam. Za instance $M1$ - $M11$ on pronalazi optimalno rešenje u nekom izvršavanju. Za skoro sve instance iz ove grupe $agap$ ima vrednosti manje od 1%, a $agap_\sigma$ vrednosti manje od 0.85%. Optimizacijom rojem čestica pronđena su optimalna rešenja za instance $M1$ - $M8$. Tabu pretraživanje ima najkraće vreme izvršavanja u okviru ove grupe, ali daje loše rezulate u smislu optimalnosti. Vrednosti $agap$ nalaze se u intervalu $(3.5, 15.5)$, pa se rešenja dobijena ovim algoritmom ne mogu smatrati dobrom aproksimacijom optimalnog rešenja. Ovo se objašnjava činjenicom da je Tabu pretraga zasnovana na lokalnoj pretrazi, pa daje lošije rezultate sa povećanjem dimenzije pretraživačkog prostora.

Instance velikih dimenzija

Svi razmatrani algoritmi za ovu grupu instanci imaju kraće vreme izvršavanja od vremena izvršavanja egzaktnog rešavača. Prosečno vreme izvršavanja PSO algoritma je 197 sekundi, TS algoritma 185 sekundi, dok je prosečno vreme izvršavanja hibridnog algoritma 215 sekundi. Kako je prosečno vreme izvršavanja egzaktnog rešavača 1095 sekundi, metaheuristike daju značajno vremensko poboljšanje uz minimalno povećanje $agap$ vrednosti.

Za instance velikih dimenzija hibridni algoritam daje najbolje rezultate, u smislu optimalnosti rešenja. Algoritam dostiže optimalno rešenje za 5 instanci. U većini slučaja $agap$ je manji od 0.75% osim kod instanci $L4$ i $L16$ gde malo veći od 1%. Vrednosti rešenja malo variraju od optimalnog na šta ukazuju male vrednosti $agap_\sigma$ parametra. Sva rešenja pronađena hibridnim algoritmom su bolja od najboljih rešenja koja daju druga dva algoritma. Optimizacija rojem čestica dostiže optimalno rešenje samo na jednoj instanci i u proseku ima duplo veće vrednosti $agap$ parametra od onih koje daje hibridni algoritam. U tom pogledu smatramo da hibridni algoritam ima bolje performanse od Optimizacije rojem čestica. Tabu pretraga daje loša rešenja u smislu optimalnosti, $agap$ uzima vrednosti iz intervala $(6, 16.1)$. Iako su najbolja rešenja

daleko od optimalnih, pri svakom izvršavanju rešenja malo variraju. Jedno objašnjenje ovih rezultata je to što Tabu pretraga sa povećanjem dimenzije teško izlazi iz lokalnog optimuma.

Zaključak

Na osnovu prikazanih rezultata zaključuje se da među predloženim metodama najbolje peformanse, u smislu optimalnosti rešenja, ima hibridni algoritam, a najgore Tabu pretraživanje. Hibridni algoritam je postigao bolja rešenja od pojedinačnih metaheuristika. Ipak, dobijeni rezultati ukazuju da je potrebna dodatna optimizacija predloženih algoritama da bi oni bili pogodni za praktično rešavanje razmatranog problema. Rezultati se mogu unaprediti dodatnim podešavanjem parametara i uvođenjem paralelizacije prilikom izvršavanja.

Literatura

[1] Cánovas L., García S., Labbé M., Marín A., 2007. A strengthened formulation for the simple plant location problem with order. *Operations Research Letters*, 35(2), 141-150

[2] Díaz J.A., Luna D., Camacho-Vallejo JF., Casas-Ramírez MS., 2017. GRASP and hybrid GRASP-Tabu heuristis to solve maximal covering location problem with customer preference ordering, *Expert Systems With Applications*, 82, 67-76

[3] Drakulić D., Takači A., Marić M.,2010. New model of maximal covering location problem with fuzzy conditions, *Computing and Informatics*, Vol. 35, 2016, 635–652

[4] Eberhart R.C., Yuhui S. 2001. Particle swarm optimization: developments, applications and resources. in Evolutionary Computation, *Evolutionary Computation, Proceedings of the 2001 Congress*

[5] Marić M., Stanimirović Z., Milenković N., Đenić A. 2014. Metaheuristic approaches to solving large-scale bilevel uncapacitated facility location problem with cilent' preferences, *Yugoslav Journal of Operations Research*, , Number 3, 361–378

Tabela 2: Rezultati izvršavanja PSO algoritma za instance malih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$\sigma(\%)$
S1	2	50	4	4	18	0,010	opt	0,001	0,260	0,000%	0,000%
S2	3	50	4	4	125	0,075	opt	0,001	0,312	0,000%	0,000%
S3	2	45	6	4	245	0,004	opt	0,002	0,294	0,000%	0,000%
S4	3	45	6	4	105	0,002	opt	0,002	0,316	0,000%	0,000%
S5	4	40	6	6	105	0,005	opt	0,001	0,364	0,000%	0,000%
S6	2	40	6	6	209	0,004	opt	0,002	0,302	0,000%	0,000%
S7	4	38	8	6	484	0,011	opt	0,002	0,350	0,000%	0,000%
S8	3	38	8	6	83	0,002	opt	0,002	0,317	0,000%	0,000%
S9	5	35	9	11	248	0,006	opt	0,002	0,402	0,000%	0,000%
S10	6	35	9	11	265	0,007	opt	0,002	0,437	0,000%	0,000%
S11	5	32	14	11	445	0,009	opt	0,002	0,417	0,000%	0,000%
S12	6	32	24	11	617	0,029	opt	0,002	0,451	0,000%	0,000%
S13	8	30	23	17	970	0,012	opt	0,002	0,507	0,000%	0,000%
S14	9	30	23	17	497	0,013	opt	0,002	0,530	0,000%	0,000%
S15	8	25	28	17	496	0,017	opt	0,003	0,528	0,000%	0,000%
S16	9	25	33	17	871	0,015	opt	0,003	0,605	0,000%	0,000%
S17	20	15	50	30	1656	0,01	opt	0,001	0,761	0,000%	0,000%
S18	7	15	50	30	1086	0,01	opt	0,402	0,711	1,406%	1,175%
S19	20	12	60	30	1971	0,01	opt	0,001	0,752	0,000%	0,000%
S20	10	12	70	30	1667	0,01	opt	0,498	0,725	3,507%	1,491%

Tabela 3: Rezultati izvršavanja TS algoritma za instance malih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
S1	2	50	4	4	18	0,01	opt	0,001	0,489	0,000%	0,000%
S2	3	50	4	4	125	0,08	opt	0,000	0,056	0,000%	0,000%
S3	2	45	6	4	245	0	opt	0,014	0,506	0,000%	0,000%
S4	3	45	6	4	105	0,002	opt	0,000	0,056	0,000%	0,000%
S5	4	40	6	6	105	0,005	opt	0,000	0,067	0,000%	0,000%
S6	2	40	6	6	209	0,004	opt	0,000	0,650	0,000%	0,000%
S7	4	38	8	6	484	0,011	opt	0,000	0,076	0,000%	0,000%
S8	3	38	8	6	83	0,002	opt	0,000	0,875	0,000%	0,000%
S9	5	35	9	11	248	0,006	opt	0,000	0,711	0,000%	0,000%
S10	6	35	9	11	265	0,007	opt	0,000	1,032	0,000%	0,000%
S11	5	32	14	11	445	0,009	opt	0,002	0,421	0,000%	0,000%
S12	6	32	24	11	617	0,029	opt	0,001	0,608	0,000%	0,000%
S13	8	30	23	17	970	0,012	opt	0,001	0,535	0,000%	0,000%
S14	9	30	23	17	497	0,013	opt	0,001	0,617	0,000%	0,000%
S15	8	25	28	17	496	0,017	opt	0,002	0,336	0,000%	0,000%
S16	9	25	33	17	871	0,015	opt	0,002	0,368	0,000%	0,000%
S17	20	15	50	30	1656	0,009	opt	0,000	0,265	0,000%	0,000%
S18	7	15	50	30	1086	0,012	opt	0,204	0,495	0,184%	0,368%
S19	20	12	60	30	1971	0,005	opt	0,000	0,255	0,000%	0,000%
S20	10	12	70	30	1667	0,010	opt	0,126	0,503	2,268%	1,375%

Tabela 4: Rezultati izvršavanja hibridnog algoritma za instance malih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
S1	2	50	4	4	18	0,010	opt	0,119	0,301	0,000%	0,000%
S2	3	50	4	4	125	0,075	opt	0,009	0,229	0,000%	0,000%
S3	2	45	6	4	245	0,004	opt	0,112	0,293	0,000%	0,000%
S4	3	45	6	4	105	0,002	opt	0,009	0,225	0,000%	0,000%
S5	4	40	6	6	105	0,005	opt	0,009	0,290	0,000%	0,000%
S6	2	40	6	6	209	0,004	opt	0,119	0,344	0,000%	0,000%
S7	4	38	8	6	484	0,011	opt	0,011	0,280	0,000%	0,000%
S8	3	38	8	6	83	0,002	opt	0,116	0,348	0,000%	0,000%
S9	5	35	9	11	248	0,006	opt	0,120	0,457	0,000%	0,000%
S10	6	35	9	11	265	0,007	opt	0,121	0,471	0,000%	0,000%
S11	5	32	14	11	445	0,009	opt	0,121	0,465	0,000%	0,000%
S12	6	32	24	11	617	0,029	opt	0,125	0,475	0,000%	0,000%
S13	8	30	23	17	970	0,012	opt	0,132	0,574	0,000%	0,000%
S14	9	30	23	17	497	0,013	opt	0,124	0,573	0,000%	0,000%
S15	8	25	28	17	496	0,017	opt	0,126	0,571	0,000%	0,000%
S16	9	25	33	17	871	0,015	opt	0,127	0,593	0,000%	0,000%
S17	20	15	50	30	1656	0,009	opt	0,032	0,790	0,000%	0,000%
S18	7	15	50	30	1086	0,012	opt	0,444	0,830	0,000%	0,000%
S19	20	12	60	30	1971	0,005	opt	0,034	0,784	0,000%	0,000%
S20	10	12	70	30	1667	0,010	opt	0,487	0,809	0,396%	0,563%

Tabela 5: Rezultati izvršavanja PSO algoritma za instance srednjih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
M1	6	80	400	50	15474	24,052	opt	12,269	24,762	0,119%	0,059%
M2	7	80	400	50	14809	21,269	opt	13,478	23,482	0,510%	0,478%
M3	6	75	400	50	15291	19,519	opt	15,355	25,150	0,201%	0,317%
M4	7	75	400	50	13474	20,474	opt	14,931	23,024	0,086%	0,084%
M5	10	60	600	75	24497	62,258	opt	29,061	36,679	2,286%	1,448%
M6	11	60	600	75	24511	55,143	opt	26,060	37,893	1,207%	0,683%
M7	10	55	600	75	25194	42,569	opt	27,069	36,971	1,746%	0,882%
M8	11	55	600	75	25947	38,244	opt	30,087	41,377	0,396%	0,449%
M9	13	40	800	100	32760	38,981	32720	69,609	88,164	0,896%	0,715%
M10	15	40	800	100	34579	36,707	34554	49,403	58,782	1,055%	0,497%
M11	13	35	800	100	30366	23,078	30329	39,536	47,314	1,081%	0,934%
M12	15	35	800	100	31953	26,542	31841	44,560	53,175	2,180%	1,710%
M13	20	30	1200	150	46793	65,425	46567	65,413	72,422	1,796%	0,743%
M14	22	30	1200	150	47746	67,961	46907	63,265	68,261	3,140%	1,002%
M15	20	25	1200	150	48870	39,490	48008	64,691	74,095	3,121%	1,139%
M16	22	25	1200	150	46185	45,390	45834	65,616	71,881	1,648%	0,513%
M17	27	22	1600	200	58213	57,582	57136	83,714	92,305	2,973%	0,666%
M18	30	22	1600	200	62866	66,435	62404	85,663	94,067	1,839%	0,890%
M19	27	20	1600	200	67129	32,778	66008	91,506	97,571	3,348%	0,966%
M20	30	20	1600	200	62275	34,862	61574	76,388	85,705	2,980%	1,463%

Tabela 6: Rezultati izvršavanja TS algoritma za instance srednjih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
M1	6	80	400	50	15474	24,052	15150	6,377	15,855	3,516%	1,042%
M2	7	80	400	50	14809	21,269	14150	11,768	14,081	5,129%	0,685%
M3	6	75	400	50	15291	19,519	15064	4,931	15,630	3,859%	1,595%
M4	7	75	400	50	13474	20,474	13030	5,033	13,453	4,456%	0,894%
M5	10	60	600	75	24497	62,258	21945	8,303	22,155	11,231%	0,541%
M6	11	60	600	75	24511	55,143	22401	8,278	22,037	9,694%	0,720%
M7	10	55	600	75	25194	42,569	23332	15,735	23,107	8,188%	0,597%
M8	11	55	600	75	25947	38,244	24876	10,771	23,886	6,098%	1,138%
M9	13	40	800	100	32760	38,981	29783	8,996	32,389	10,451%	1,051%
M10	15	40	800	100	34579	36,707	31774	25,227	33,001	9,438%	0,872%
M11	13	35	800	100	30366	23,078	27151	19,863	32,458	11,854%	0,944%
M12	15	35	800	100	31953	26,542	28921	15,655	30,341	10,458%	0,899%
M13	20	30	1200	150	46793	65,425	41106	31,557	47,533	14,452%	1,727%
M14	22	30	1200	150	47746	67,961	42125	31,455	49,905	12,446%	0,489%
M15	20	25	1200	150	48870	39,490	44992	30,550	51,426	9,043%	1,817%
M16	22	25	1200	150	46185	45,390	41071	40,629	47,491	14,693%	0,621%
M17	27	22	1600	200	58213	57,582	50441	34,384	65,177	13,964%	0,493%
M18	30	22	1600	200	62866	66,435	53293	14,823	63,501	15,461%	0,171%
M19	27	20	1600	200	67129	32,778	59321	45,162	70,156	12,906%	0,848%
M20	30	20	1600	200	62275	34,862	54063	21,796	63,499	14,715%	1,122%

Tabela 7: Rezultati izvršavanja hibridnog algoritma za instance srednjih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
M1	6	80	400	50	15474	24,052	opt	9,089	23,906	0,169%	0,089%
M2	7	80	400	50	14809	21,269	opt	9,583	22,742	0,630%	0,418%
M3	6	75	400	50	15291	19,519	opt	13,658	23,520	0,082%	0,165%
M4	7	75	400	50	13474	20,474	opt	10,080	22,466	0,037%	0,042%
M5	10	60	600	75	24497	62,258	opt	24,511	36,410	1,176%	0,825%
M6	11	60	600	75	24511	55,143	opt	21,686	37,565	0,380%	0,413%
M7	10	55	600	75	25194	42,569	opt	23,424	35,057	0,470%	0,811%
M8	11	55	600	75	25947	38,244	opt	22,684	38,190	0,267%	0,341%
M9	13	40	800	100	32760	38,981	opt	33,829	46,411	0,346%	0,309%
M10	15	40	800	100	34579	36,707	opt	41,196	56,618	0,390%	0,302%
M11	13	35	800	100	30366	23,078	opt	24,344	48,160	0,518%	0,517%
M12	15	35	800	100	31953	26,542	31903	35,222	48,796	0,986%	0,549%
M13	20	30	1200	150	46793	65,425	46748	64,150	75,915	0,721%	0,397%
M14	22	30	1200	150	47746	67,961	47684	65,416	74,267	1,182%	0,683%
M15	20	25	1200	150	48870	39,490	48814	64,383	80,583	0,806%	0,599%
M16	22	25	1200	150	46185	45,390	46078	69,154	79,114	0,914%	0,527%
M17	27	22	1600	200	58213	57,582	57937	105,013	116,197	1,194%	0,498%
M18	30	22	1600	200	62866	66,435	62687	92,340	99,131	0,715%	0,375%
M19	27	20	1600	200	67129	32,778	66470	102,109	107,952	1,847%	0,439%
M20	30	20	1600	200	62275	34,862	62021	90,543	98,442	0,848%	0,305%

Tabela 8: Rezultati izvršavanja PSO algoritma za instance velikih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
L1	10	80	1800	100	61856	1818,266	opt	76,211	141,165	0,602%	0,737%
L2	15	80	1800	100	71950	1918,949	71857	99,216	157,026	0,697%	0,478%
L3	10	75	1800	100	65956	1639,919	65557	135,181	179,948	0,970%	0,212%
L4	15	75	1800	100	75102	1483,934	74153	127,325	155,787	2,389%	0,702%
L5	20	60	1850	150	76957	2506,606	76458	145,783	168,119	1,298%	0,392%
L6	25	60	1850	150	78192	2681,494	77768	145,693	173,877	0,822%	0,253%
L7	20	55	1850	150	63741	1788,013	63535	104,899	151,737	0,532%	0,185%
L8	25	55	1850	150	79693	1922,553	78923	158,354	185,313	2,066%	0,787%
L9	22	40	2000	200	75199	1229,530	74167	157,927	196,609	1,917%	0,389%
L10	30	40	2000	200	84381	1356,538	83312	175,436	208,334	1,733%	0,377%
L11	27	35	2090	210	85363	895,208	85047	160,243	212,731	0,867%	0,522%
L12	33	35	2090	210	88163	787,937	87051	186,558	208,554	1,578%	0,421%
L13	40	30	2100	200	90917	350,619	90386	160,442	192,412	0,727%	0,110%
L14	45	30	2200	200	89757	391,602	89068	173,641	216,820	1,276%	0,395%
L15	40	25	2200	200	89398	168,058	88708	160,914	194,592	1,053%	0,242%
L16	45	25	2180	220	96394	196,305	95159	178,221	224,929	1,902%	0,546%
L17	55	22	2180	220	95691	151,105	94979	216,695	219,837	0,918%	0,091%
L18	60	22	2250	250	97095	163,501	96739	184,470	217,169	0,588%	0,176%
L19	55	20	2650	250	114180	186,539	112901	237,593	257,494	1,529%	0,333%
L20	60	20	2700	300	111719	256,959	110423	248,730	273,280	1,488%	0,180%

Tabela 9: Rezultati izvršavanja TS algoritma za instance velikih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
L1	10	80	1800	100	61856	1818,266	55125	79,838	163,712	11,026%	0,177%
L2	15	80	1800	100	71950	1918,949	68077	95,116	159,575	5,966%	0,714%
L3	10	75	1800	100	65956	1639,919	61697	25,321	197,670	6,579%	0,149%
L4	15	75	1800	100	75102	1483,934	67708	152,789	156,687	10,313%	0,382%
L5	20	60	1850	150	76957	2506,606	67863	40,711	178,998	11,986%	0,138%
L6	25	60	1850	150	78192	2681,494	70874	87,905	178,943	9,543%	0,225%
L7	20	55	1850	150	63741	1788,013	55682	113,850	180,835	13,396%	0,615%
L8	25	55	1850	150	79693	1922,553	69652	100,371	167,139	12,828%	0,280%
L9	22	40	2000	200	75199	1229,530	63601	149,201	203,296	16,026%	0,493%
L10	30	40	2000	200	84381	1356,538	71423	49,501	180,865	16,079%	0,885%
L11	27	35	2090	210	85363	895,208	73009	145,290	199,827	14,811%	0,415%
L12	33	35	2090	210	88163	787,937	77222	49,228	198,459	12,757%	0,425%
L13	40	30	2100	200	90917	350,619	80404	77,628	165,874	13,383%	1,486%
L14	45	30	2200	200	89757	391,602	80702	56,464	173,962	10,304%	0,264%
L15	40	25	2200	200	89398	168,058	78855	132,528	169,399	11,837%	0,054%
L16	45	25	2180	220	96394	196,305	82960	124,144	176,142	14,125%	0,231%
L17	55	22	2180	220	95691	151,105	86152	93,926	183,850	10,033%	0,053%
L18	60	22	2250	250	97095	163,501	86128	157,706	179,193	11,366%	0,058%
L19	55	20	2650	250	114180	186,539	102252	57,862	238,294	10,480%	0,027%
L20	60	20	2700	300	111719	256,959	98132	125,531	240,362	14,306%	0,307%

Tabela 10: Rezultati izvršavanja hibridnog algoritma za instance velikih dimenzija

Instanca	p	radius	$\|J\|$	$\|I\|$	sol	t(s)	best	$t_{best}(s)$	$t_{tot}(s)$	agap(%)	$agap_{\sigma}(\%)$
L1	10	80	1800	100	61856	1818,266	opt	83,553	155,618	0,035%	0,069%
L2	15	80	1800	100	71950	1918,949	71895	115,105	165,981	0,317%	0,160%
L3	10	75	1800	100	65956	1639,919	opt	128,318	163,115	0,360%	0,371%
L4	15	75	1800	100	75102	1483,934	opt	118,069	168,029	1,137%	0,808%
L5	20	60	1850	150	76957	2506,606	76936	169,258	201,022	0,347%	0,244%
L6	25	60	1850	150	78192	2681,494	78146	137,356	184,756	0,402%	0,418%
L7	20	55	1850	150	63741	1788,013	63671	116,655	165,369	0,509%	0,234%
L8	25	55	1850	150	79693	1922,553	79667	181,981	197,398	0,627%	0,572%
L9	22	40	2000	200	75199	1229,530	opt	205,699	234,060	0,699%	0,622%
L10	30	40	2000	200	84381	1356,538	84360	186,989	226,504	0,091%	0,092%
L11	27	35	2090	210	85363	895,208	85205	209,793	235,588	0,326%	0,119%
L12	33	35	2090	210	88163	787,937	88110	205,382	238,261	0,254%	0,178%
L13	40	30	2100	200	90917	350,619	opt	170,336	215,486	0,267%	0,261%
L14	45	30	2200	200	89757	391,602	89662	186,729	221,994	0,549%	0,312%
L15	40	25	2200	200	89398	168,058	89162	141,027	214,403	0,616%	0,244%
L16	45	25	2180	220	96394	196,305	95973	211,681	228,807	1,052%	0,390%
L17	55	22	2180	220	95691	151,105	95422	155,852	229,075	0,490%	0,179%
L18	60	22	2250	250	97095	163,501	96886	228,239	251,912	0,303%	0,052%
L19	55	20	2650	250	114180	186,539	113498	263,415	281,910	0,745%	0,106%
L20	60	20	2700	300	111719	256,959	111333	306,645	317,992	0,589%	0,180%