



ODABRANA POGLAVLJA OPTIMIZACIJE

DOMAĆI ZADATAK [6]

PROBLEM MAKSIMALNOG POKRIVANJA SA
PRIORITETNIM ODABIROM RESURSA

Student:

Anđela Donević
1023/2018

Nastavnici :

prof. dr Zorica Stanimirović
dr Stefan Mišković

Opis problema

U domaćem zadatku biće razmatrana verzija problema maksimalnog pokrivanja (Maximal covering location problem, MCLP). 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 može biti formulisan kao problem linearnog programiranja sa dva nivoa (Bilevel Maximal Covering Location Problem, BMCLP). Prvi nivo koji odgovara problemu uspostavljanja ustanova i drugi nivo koji odgovara problemu pridruživanja kupaca ustanovama.

Matematička formulacija problema

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

Prvo ćemo definisati promenljive i parametre potrebne za formulaciju ovih problema.

Promenljive odlučivanja $i \in I, j \in J$:

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

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

Ostali parametri:

- \mathbf{R} maksimalno rastojanje koje je kupac spreman da pređe kako bi došao do ustanove
- \mathbf{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
- \mathbf{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
- \mathbf{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:

- $\mathbf{I(j)}$ skup svih ustanova i koje su moguće za kupca j , $j \in J$
- $\mathbf{J(i)}$ 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:

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

$$y_i = 1, \quad \forall i \in I_2 \quad (2)$$

$$\sum_{i \in I_1} y_i = p \quad (3)$$

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

Gde je x rešenje problema:

$$\max_x \sum_{i \in I} \sum_{j \in J(i)} g_{ij} x_{ij} \quad (5)$$

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

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

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

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

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:

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

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

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

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:

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

$$\sum_{i \in I_1} y_i = p \quad (12)$$

$$y_i = 1, \quad \forall i \in I_2 \quad (13)$$

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

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

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

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

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

Instance

Instance su generisane u skladu sa preporukama iz literature (Díaz et 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 neopslužene 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 algoritama za generisanje instanci korišćen je programski jezik PYTHON. Kao rezultat izvršavanja ovih algoritama dobijaju se datoteke čiji su sadržaji generisane instance.

Primer instance

```
1 35
2 1
3 3
4 1
5 3
6 1
7
8 29 0
9 82 63
10 37 25
11 46 83
12 3 56
13 60 92
14 58 46
15 7 46
16
```

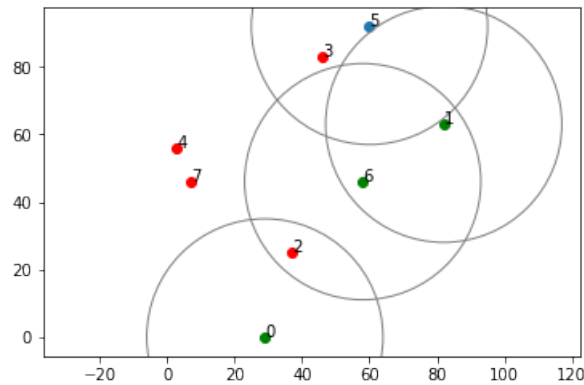
```

17 0 0 0 3
18 2 2 2 2
19 1 3 3 1
20 3 1 1 0
21
22 48 35 3 24
23
24 2 4 7 3
25 0 1 6 5

```

Prikazana je instanca *dat_1.txt*.

U redu 1 dato je maksimalno rastojanje R koje su kupci spremni da pređu. U redu 2 nalazi se broj novih ustanova p koje treba uspostaviti. U redovima 3 – 6 zadaje se broj elemenata skupa J_1 , broj elemenata skupa J_2 , broj elemenata skupa I_1 , broj elemenata skupa I_2 respektivno. Zatim se zadaju koordinate kupaca i ustanova (redovi 8 – 15). Podaci dati u redovima 17 – 20 predstavljaju matricu lokacija gde kolona k odgovara k – tom kupcu. U koloni k , $k \in \{0, |I| - 1\}$ je dat niz lokacija ustanova sortiranih po prioritetima posećivanja kupca k . Sledeći red (red 22) sardži potražnju kupaca. U redu 24 nalaze se lokacije kupaca. U prvom delu nalaze se lokacije kupca koji nisu opsluženi postojećim ustanovama (J_1), zatim lokacije opsluženih kupaca (J_2). Poslednji red sadrži lokacije postojećih i potencijalnih ustanova. U prvom delu se nalaze lokacije potencijalnih ustanova (I_1), a u drugom lokacije postojećih ustanova (I_2).



Slika 1: Grafički prikaz instance. Crveni kružići predstavljaju korisnike, plavi predstavljaju postojeće ustanove, a zeleni potencijalne nove ustanove. Poluprečnik svakog kruga je $R = 35$

Rezultati

Generisano je 60 instanci za testiranje i jedna ručno pravljen (*dat_1.txt*) koja je korišćena za prikaz problema (Slika 1) . Instance su podeljene po veličini u tri grupe: instance malih(21), srednjih(20) i velikih(20) dimenzija. Za rešavanje problema linearnog programiranja nad ovim instancama korišćen je ILOG CPLEX OPTIMIZATION STUDIO 12.8 u kombinaciji sa programskim jezikom C. Vreme izvršavanja ograničeno je na 2h.

Rezultati za instance malih, srednjih i velikih dimenzija u kojima je korišćena standardna vrednost za *relative MIP gap tolerance* dati su u sledećim tabelama:

Tabela 1: Rezultati nad instancama malih dimenzija, *relative MIP gap tolerance* = $1e-4$

Instanca	p	radius	$ J_1 $	$ J_2 $	$ I_1 $	$ I_2 $	rezultat	vreme (s)	iteracija	cvorovi	gap
dat_1.txt	1	35	3	1	3	1	48	0,001781	0	0	0
mclp_S1.txt	2	50	0	4	3	1	18	0,009843	0	0	0
mclp_S2.txt	3	50	1	3	3	1	125	0,075242	0	0	0
mclp_S3.txt	2	45	1	5	3	1	245	0,004248	0	0	0
mclp_S4.txt	3	45	2	4	3	1	105	0,001763	0	0	0
mclp_S5.txt	4	40	0	6	4	2	105	0,004773	0	0	0
mclp_S6.txt	2	40	4	2	4	2	209	0,004287	5	0	0
mclp_S7.txt	4	38	3	5	4	2	484	0,011281	0	0	0
mclp_S8.txt	3	38	5	3	4	2	83	0,002051	0	0	0
mclp_S9.txt	5	35	4	5	7	4	248	0,005984	0	0	0
mclp_S10.txt	6	35	1	8	7	4	265	0,007465	0	0	0
mclp_S11.txt	5	32	3	11	7	4	445	0,009172	5	0	0
mclp_S12.txt	6	32	12	12	7	4	617	0,029017	5	0	0
mclp_S13.txt	8	30	11	12	11	6	970	0,011931	34	0	0
mclp_S14.txt	9	30	6	17	11	6	497	0,012994	8	0	0
mclp_S15.txt	8	25	8	20	11	6	496	0,01738	14	0	0
mclp_S16.txt	9	25	16	17	11	6	871	0,014572	15	0	0
mclp_S17.txt	20	15	23	27	20	10	1656	0,009241	0	0	0
mclp_S18.txt	7	15	28	22	20	10	1086	0,012403	29	0	0
mclp_S19.txt	20	12	48	12	20	10	1971	0,005454	0	0	0
mclp_S20.txt	10	12	51	19	20	10	1667	0,010226	36	0	0

Tabela 2: Rezultati nad instancama srednjih dimenzija, *relative MIP gap tolerance* = $1e-4$

Instanca	p	radius	$ J_1 $	$ J_2 $	$ I_1 $	$ I_2 $	rezultat	vreme (s)	iteracija	cvorovi	gap
mclp_M1.txt	6	80	0	400	45	5	15474	24.05242	4299	0	0
mclp_M2.txt	7	80	0	400	45	5	14809	21.269244	2157	0	0
mclp_M3.txt	6	75	0	400	45	5	15291	19.518548	2237	0	0
mclp_M4.txt	7	75	0	400	45	5	13474	20.474093	3126	3	0
mclp_M5.txt	10	60	0	600	67	8	24497	62.258355	3230	0	0
mclp_M6.txt	11	60	10	590	67	8	24511	55.142537	4493	0	0
mclp_M7.txt	10	55	0	600	67	8	25194	42.569424	4275	0	0
mclp_M8.txt	11	55	0	600	67	8	25947	38.244486	4796	0	0
mclp_M9.txt	13	40	15	785	90	10	32760	38.980687	6656	0	0
mclp_M10.txt	15	40	5	795	90	10	34579	36.706881	6302	0	0
mclp_M11.txt	13	35	37	763	90	10	30366	23.078012	5106	0	0
mclp_M12.txt	15	35	64	736	90	10	31953	26.54172	7276	0	0
mclp_M13.txt	20	30	20	1180	135	15	46793	65.425173	6849	0	0
mclp_M14.txt	22	30	1	1199	135	15	47746	67.960984	11350	0	0
mclp_M15.txt	20	25	316	884	135	15	48870	39.490235	11571	0	0
mclp_M16.txt	22	25	245	955	135	15	46185	45.390189	13433	4	0
mclp_M17.txt	27	22	181	1419	180	20	58213	57.582019	20798	0	0
mclp_M18.txt	30	22	203	1397	180	20	62866	66.434618	19438	15	0.000049
mclp_M19.txt	27	20	282	1318	180	20	67129	32.77755	18532	0	0
mclp_M20.txt	30	20	140	1460	180	20	62275	34.861663	14063	0	0

Tabela 3: Rezultati nad instancama velikih dimenzija, *relative MIP gap tolerance* = $1e-4$

Instanca	p	radius	$ J_1 $	$ J_2 $	$ I_1 $	$ I_2 $	rezultat	vreme (s)	iteracija	cvorovi	gap
mcp_L1.txt	10	80	0	1800	90	10	61856	1818.265867	11157	0	0
mcp_L2.txt	15	80	0	1800	90	10	71950	1918.949482	38796	19	0
mcp_L3.txt	10	75	0	1800	90	10	65956	1639.918904	65651	24	0.000096
mcp_L4.txt	15	75	0	1800	90	10	75102	1483.934291	16708	0	0
mcp_L5.txt	20	60	0	1850	135	15	76957	2506.605742	12151	0	0
mcp_L6.txt	25	60	0	1850	135	15	78192	2681.494262	45392	16	0
mcp_L7.txt	20	55	0	1850	135	15	63741	1788.013033	10516	0	0
mcp_L8.txt	25	55	0	1850	135	15	79693	1922.552651	12150	0	0
mcp_L9.txt	22	40	0	2000	180	20	75199	1229.529904	15627	0	0
mcp_L10.txt	30	40	0	2000	180	20	84381	1356.538403	23520	0	0
mcp_L11.txt	27	35	0	2090	189	21	85363	895.207844	21560	18	0.000076
mcp_L12.txt	33	35	0	2090	189	21	88163	787.936995	12538	0	0
mcp_L13.txt	40	30	0	2100	180	20	90917	350.619322	7836	0	0
mcp_L14.txt	45	30	40	2160	180	20	89757	391.601622	15432	4	0.000045
mcp_L15.txt	40	25	4	2196	180	20	89398	168.058196	11271	3	0
mcp_L16.txt	45	25	1	2179	198	22	96394	196.305305	11402	0	0
mcp_L17.txt	55	22	363	1917	198	22	95691	151.105479	20923	9	0.000051
mcp_L18.txt	60	22	189	2061	225	25	97095	163.501212	14473	0	0
mcp_L19.txt	55	20	176	2574	225	25	114180	186.539462	50640	81	0.000026
mcp_L20.txt	60	20	33	2667	270	30	111719	256.958919	34947	17	0.000081

Tabela 4: *relative MIP gap tolerance = 0.0*

Instanca	p	radius	$ J_1 $	$ J_2 $	$ I_1 $	$ I_2 $	rezultat	vreme (s)	iteracija	cvorovi	gap
mclp_M18.txt	30	22	203	1397	180	20	62866	61.768315	19449	16	0
mclp_L3.txt	10	75	0	1800	90	10	65956	1976.476381	65662	25	0
mclp_L11.txt	27	35	0	2090	189	21	85363	1156.785863	21573	19	0
mclp_L14.txt	45	30	40	2160	180	20	89757	367.860179	15436	5	0
mclp_L17.txt	55	22	363	1917	198	22	95691	140.397397	21006	11	0
mclp_L19.txt	55	20	176	2574	225	25	114180	179.561716	51250	85	0
mclp_L20.txt	60	20	33	2667	270	30	111719	259.085839	35384	21	0

Nad instancama kod kojih je u rezultatu dobijena *gap* vrednost veća od nule a vreme izvršavanja manje od $2h$, vršeno je novo testiranje. Parametar *relative MIP gap tolerance* postavljen je na 0.0. Rezultati su predstavljeni u Tabeli 4.

Analiza rezultata

Prosečno vreme izvršavanja programa na instancama manjih dimenzija je 0,006 sekundi, na srednjim 40,938 sekundi i 1094,682 sekundi na velikim. Ovo nam govori da vreme izvršavanja programa raste sa povećavanjem broja kupaca i broja ustanova, što je i očekivano.

Povećavanjem parametra R raste vrednost funkcije cilja i vreme izvršavanja programa.

Ako je p mnogo manje od $|I|$ onda se uglavnom sa povećanjem p povećava vreme izvršavanja programa.

Promena parametra *relative MIP gap tolerance* sa standardne vrednosti na 0.0 utiče samo na povećanje broja čvorova dok optimalno rešenje ostaje isto.

Izvorni kod, izvršna verzija programa i instance nalaze se na [linku](#).

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 heuristics to solve maximal covering

location problem with customer preference ordering, *Expert Systems With Applications* 82, 67-76