Преглед метода за рјешавање проблема максимално балансиране повезане партиције у графу

Драган Матић [1] Владимир Филиповић [2] Јозеф Кратица [3]

[1] Природно-математички факултет, Бања Лука

[2] Математички факултет, Београд

[3] Математички институт, Београд

Четврта математичка конференција Републике Српске
Требиње, 6-7. јуни 2014.

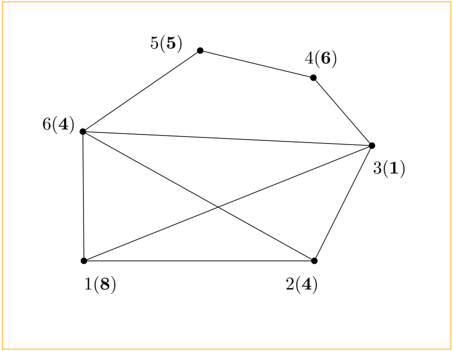
Дефиниција проблема

- Нека је $G = (V, E)$ повезан граф са скупом чворова $V = \{1, 2, ..., n\}$ и скупом грана E и нека су w_i тежине на чворовима.
- MBCP је одређивање подјеле скупа чворова V у два дисјунктна, непразна скупа V_1 и V_2, таква да важи:
  - подграфови графа G индуковани са V_1 и V_2 су повезани;
  - суме тежина чворова из V_1 и V_2 су по вриједности што ближе једна другој, тј. разлика између сума тежина је најмања могућа.

Формално, MBCP је проналажење партиције (V_1, V_2), такве да су пографови од G индуковани са V_1 and V_2 повезани, а вриједност $obj(V_1, V_2) = |w(V_1) - w(V_2)|$ је минимална.

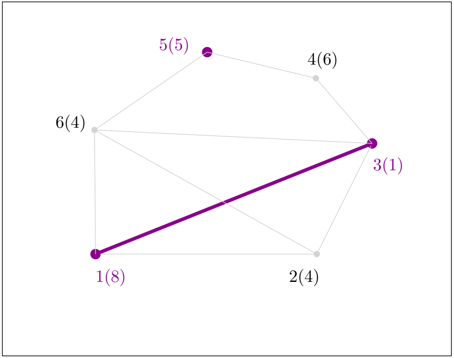
Примјер једноставног графа и рјешење за MBCP

Посматрајмо граф на слици.
Сума свих тежина је 28.
Идеална подјела би била када би подграфови имали суме тежина по 14.



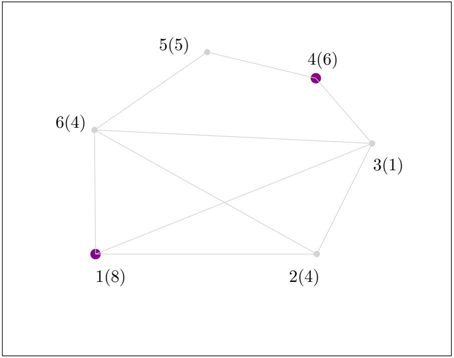
Примјер једноставног графа и рјешење за MBCP

Тада би подскуп који садржи чвор 1 садржавао још и чворове 3 и 5, или чвор 4.



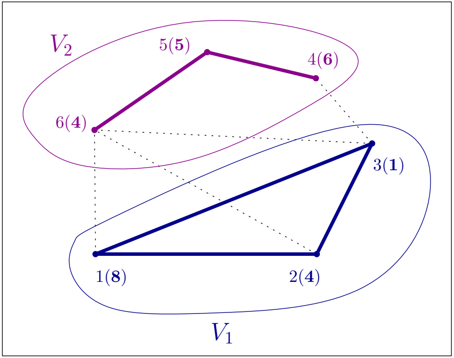
Примјер једноставног графа и рјешење за MBCP

У оба случаја, индукован подграф није повезан.



Примјер једноставног графа и рјешење за MBCP

Стога оптимално рјешење не може бити нула, већ најмање 2, јер је сума свих тежина паран број. Оптимално рјешење је
$V_1 = \{1, 2, 3\}$
$V_2 = \{4, 5, 6\}$
$|w(V_1) - w(V_2)| = 2$



Методе за рјешавање MBCP

Егзактна метода

- Метода мјешовитог цјелобројног линеарног програмирања (Матић, 2014.)

Хеуристичке методе

- BlockBalancing algorithm (Chlebiková, 1996.)
- Algorithm for partition refinement (Assunção и Furtado, 2008.)
- Генетски алгоритам (Ђурић, Кратица, Тошић и Филиповић, 2008.)
- Метода промјенљивих околина (Матић, 2014.)

Метода мјешовитог цјелобројног линеарног програмирања

Приказана у раду:
Matić, D. "A mixed integer linear programming model and variable neighborhood search for maximally balanced connected partition problem Applied Mathematics and Computation, 2014.

- Први MILP модел за рјешавање овог проблема
- Модел садржи полиномијалан број промјенљивих и ограничења

Метода мјешовитог цјелобројног линеарног програмирања

Полазне претпоставке

- Због саме природе функције циља, није тешко одредити њену линеарну формулацију у MILP моделу.
- Повезаност индукованих подграфова представља захтјеван задатак.
- Постоји могућност да се повезаност реализује преко покривајућег стабла.
- Уводимо концепт мрежног протока.
- Јавља се проблем што не знамо ком (под)стаблу би припадао који чвор.

Нашем графу придодајемо "нулти"чвор који је повезан са свим осталим чворовима.
Тај чвор је полазна тачка, из кога се пушта ток величине $|V|$, а на сваком преосталом чвору графа се оставља ток величине 1.

Метода мјешовитог цјелобројног линеарног програмирања

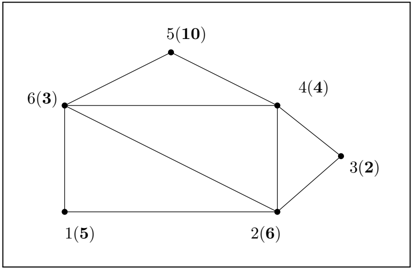
Полазне претпоставке

- Кроз гране се пушта и позитиван и негативан ток, а знак тока одређује оријентацију гране у покривајућем стаблу.
- Оријентација грана које су инцидентне са чвором 0 је јасна (све гране су усмјерене тако да је чвор 0 полазни).
- Апсолутна вриједност тока који излази из неког чвора је једнака броју чворова у подстаблу коме је тај чвор коријен.
- Уколико на датим подграфовима буде успјела конструкција одговарајућих покривајућих стабала, то би значило да је испуњен услов повезаности.

Прописани модел комбинује идеју мрежног тока и покривајућег стабла: стаблом контролишемо оријентацију грана, док модел тока искључује појаву контура.

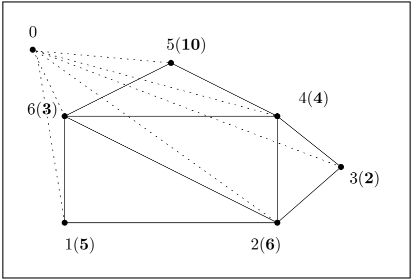
Примјер

Посматрајмо граф на слици.



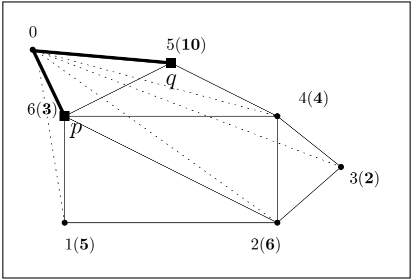
Примјер

Придодамо му чвор 0 који је повезан са свим чворовима графа.



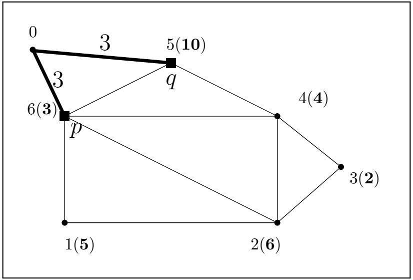
Примјер

У моделу ћемо подесити ограничења да је тај чвор 0 повезан са тачно два чвора (чворови p и q). Наравно, чворови p и q нису фиксни.



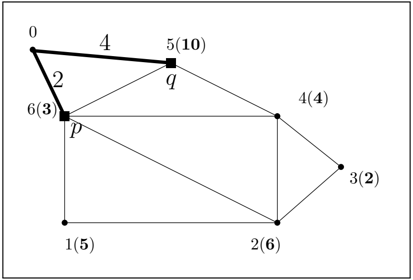
Примјер

Сада шаљемо проток. Ограничењима ћемо подесити да је укупан проток који полази из чвора 0 једнак броју чворова графа.



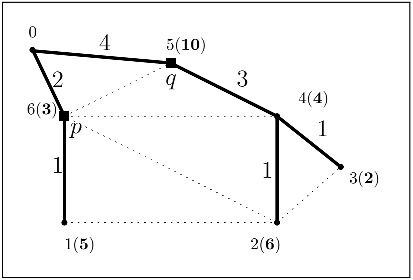
Примјер

Распоређени проток не мора обавезно да води ка оптималном рјешењу.



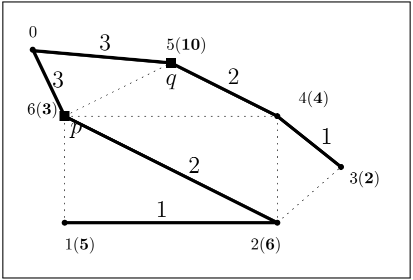
Примјер

У сваком чвору се оставља проток величине 1.



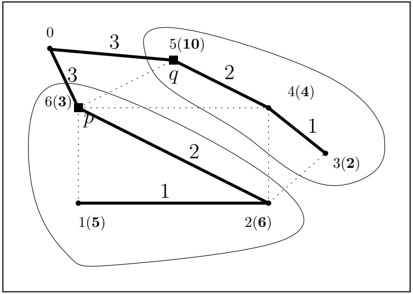
Примјер

Процес оптимизације ће одредити колики проток иде ка ком од изабраних чворова.



Примјер

Покривајућа стабла (подграфова) гарантују повезаност, док проток не дозвољава контуре.



Метода мјешовитог цјелобројног линеарног програмирања

- Прво уводимо чвор 0, који је повезан са свим чворовима из V.
- Проширени скуп чворова $\overline{V} = V \cup \{0\}$.
- Проширени скуп грана $\overline{E} = E \cup \partial E$, гдје је $\partial E = \{(0,i) : i \in V\}$.
- Суму свих тежина чворова из V означимо са $w_{sum} = \sum_{v_i \in V} w_i$.
- Пошто су G_1 и G_2 повезани, садрже покривајућа стабла, $T_1(V_1, E'_1)$ и $T_2(V_2, E'_2)$, респективно.
- Нека су p и q, $p \in V_1$ и $q \in V_2$ два чвора.
- Дефинишимо скупове $\overline{E}'_1 = E'_1 \cup \{(0,p)\}$ и $\overline{E}'_2 = E'_2 \cup \{(0,q)\}$
- И графове $T = (V, E'_1 \cup E'_2)$ и $\overline{T} = (\overline{V}, \overline{E}'_1 \cup \overline{E}'_2)$.

Метода мјешовитог цјелобројног линеарног програмирања

Сада уводимо промјенљиве за MILP модел

$$x_i = \begin{cases} 1, & i \in V_1 \\ 0, & i \in V_2 \end{cases},$$

$$y_e = \begin{cases} 1 & e \in \overline{E}'_1 \\ 0 & \text{иначе} \end{cases} \qquad z_e = \begin{cases} 1 & e \in \overline{E}'_2 \\ 0 & \text{иначе} \end{cases},$$

$$u_e \in [-n, n], e \in \overline{E}.$$

За сваку грану $e \in \overline{E}$, промјенљива u_e означава вриједност тока на грани e.

Формулација MILP модела за MBCP

$$\min \quad -w_{sum} + 2\sum_{i=1}^{n} x_i w_i, \tag{1}$$

уз ограничења

$$2\sum_{i=1}^{n} x_i w_i \geq w_{sum}, \tag{2}$$

$$y_e \leq \frac{1}{2}x_{i_e} + \frac{1}{2}x_{j_e}, \qquad e \in E, \tag{3}$$

$$z_e \leq 1 - \frac{1}{2}x_{i_e} - \frac{1}{2}x_{j_e}, \qquad e \in E, \tag{4}$$

$$y_e \leq x_{j_e}, \qquad e \in \partial E, \tag{5}$$

$$z_e \leq 1 - x_{j_e}, \qquad e \in \partial E, \tag{6}$$

Формулација MILP модела за MBCP

$$u_e \leq n \cdot y_e + n \cdot z_e, \qquad e \in \overline{E}, \tag{7}$$

$$u_e \geq -n \cdot y_e - n \cdot z_e, \qquad e \in \overline{E}, \tag{8}$$

$$\sum_{e:j_e=i} u_e - \sum_{e:i_e=i} u_e = 1, \qquad i \in V, \tag{9}$$

$$\sum_{e:i_e=0} u_e = n, \tag{10}$$

$$\sum_{e \in E} y_e + \sum_{e \in E} z_e = n - 2, \tag{11}$$

$$\sum_{e \in \partial E} y_e + \sum_{e \in \partial E} z_e = 2, \tag{12}$$

$$x_i, y_e, z_e \in \{0,1\},\ i \in V,\ e \in \overline{E}, \tag{13}$$

$$u_e \in [-n, n],\ e \in \overline{E}. \tag{14}$$

Метода мјешовитог цјелобројног линеарног програмирања

Функција циља $\mathrm{obj_{MILP}}$ се дефинише као

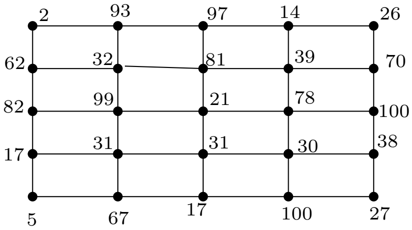
$$\mathrm{obj_{MILP}}(x,y,z,u) = -w_{sum} + 2\sum_{i=1}^{n} x_i w_i$$

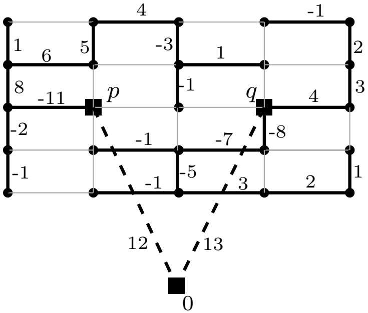
с обзиром на ограничења (2)-(14).

Теорема

Партиција $V^* = (V_1^*, V_2^*)$ је оптимална ако и само ако постоји оптимално рјешење (x,y,z,u) проблема (1)-(14).

Примјер





Апсолутна вриједност тока на свакој грани (i,j), која припада покривајућем стаблу, је једнака броју чворова у подстаблу са коријеном у чвору j.
Ако се израчунају суме тежина чворова по компонентама, види се да је разлика тих сума једнака $|630 - 629| = 1$.

Хеуристички методи - I алгоритам

"BlockBalancing алгоритам"

- Предложен у раду Janka Chlebíková, "Approximating the maximally balanced connected partition problem in graphs Information Processing Letters, 1996.
- У истом раду је доказано да је проблем NP тежак.
- За повезан граф $G = (V, E)$ са задатим тежинама на чворовима, посматрају се повезане партиције (V_1, V_2)
- Функција циља:

$$B^w(V_1, V_2) = \min(w(V_1), w(V_2))$$

Потребно је пронаћи:

$$\max B^w(G) = \max\{B^w(V_1, V_2)\}$$

- Друга варијанта функције циља:

$$B_0^w(V_1, V_2) = w(V_1) \cdot w(V_2)$$

"BlockBalancing алгоритам"

Улаз: Граф $G = (V, E)$, са тежинском функцијом w.

0. Поредај чворове графа тако да важи

$$w(v_1) \geqslant w(v_2) \geqslant w(v_3) \dots$$

   Иницијализуј $V_1 = \{v_1\}$, $V_2 = V \setminus V_1$.

1. Ако је $w(V_1) \geqslant \frac{1}{2} w(V)$ иди на корак 3. иначе на корак 2.
2. Нека је
   $V_0 = \{u \in V_2 | (V_1 \cup u, V_2 \setminus \{u\})$ је повезана партиција у $G\}$
   Изабери $u \in V_0$, такво да је $w(u) = \min_{v \in V_0} w(v)$.
   Ако је $w(u) < w(V) - 2w(V_1)$
   онда

   Постави $V_1 := V_1 \cup \{u\}$, $V_2 := V2 \setminus \{u\}$.

   Врати се на корак 1.

   иначе

   Иди на корак 3.
3. Резултат је добијена партиција (V_1, V_2)

"BlockBalancing алгоритам"

Теорема која прописује апроксимацију

Теорема: Предложени алгоритам у полиномијалном проналази повезану партицију (V_1, V_2) и важи:

- $\max B^w(G) \leqslant \frac{4}{3} B^w(V_1, V_2)$
- $\max B_0^w(G) \leqslant (8 - 4\sqrt{3}) B_0^w(V_1, V_2)$

Добија се апроксимација са фактором 1.072.

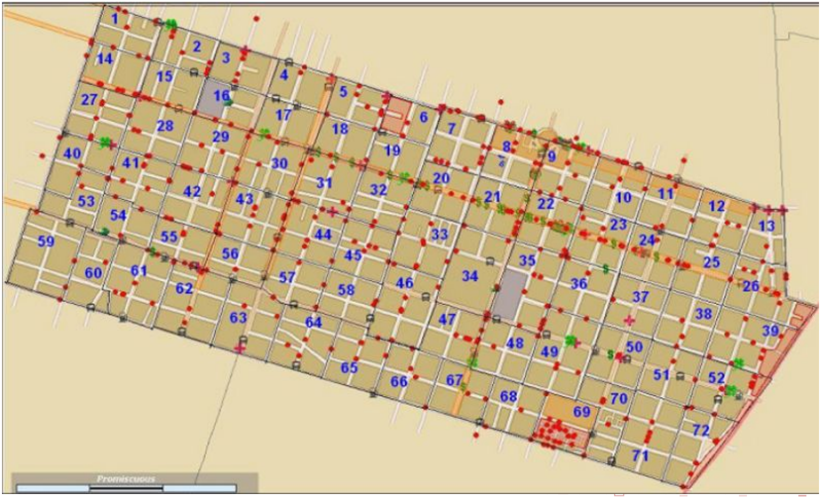
Хеуристички методи - II алгоритам

"Algorithm for partition refinement"

- Прописан у раду Assunção, Furtado, "A Heuristic Method for Balanced Graph Partitioning: An Application for the Demarcation of Preventive Police Patrol Areas Advances in Artificial Intelligence – IBERAMIA, Lecture Notes in Computer Science 2008.
- Разматра се проблем партиционисања на q, $q \geqslant 2$ партиција
- Мотивација за истраживање је одређивање подручја полицијских патрола у области "Aldeota"
- Као циљ се задаје да се одреди подјела града на подобласти, тако да полицијске патроле дјелују у тим подобластима, а да ниво криминала у тим подобластима буде уједначен

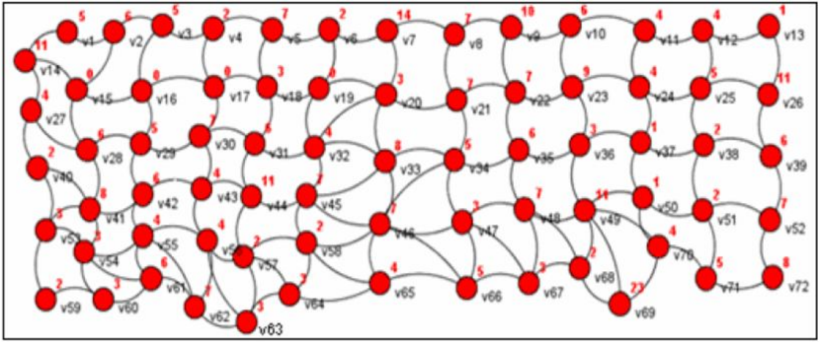
"Algorithm for partition refinement"

На основу мапе града формира се одговарајући граф



"Algorithm for partition refinement"

На основу мапе града формира се одговарајући граф.
За дијелове града се идентификује ниво криминала.



"Algorithm for partition refinement"

- Алгоритам се састоји из двије фазе:
  - Одређивање почетног рјешења, тј. почетне партиције
  - "Поправка рјешења"размјеном чворова између партиција
- За одређивање почетног рјешења користи се "Cartesian Parallel Nested Dissection algorithm прописан од стране Heath и Raghavan 1994.
- Предности овог приступа:
  - Алгоритам је погодан за графове код којих чворови одговарају географским координатама
  - Једноставан за имплементацију

"Algorithm for partition refinement"

Друга фаза

- "Фино партиционисање"је засновано на алгоритму из рада "An Efficient Heuristic Procedure for Partitioning Graphs Kernighan, Lin 1970.
- Основна идеја је да се врши размјена чворова између партиција, како би се боље расподијелиле тежине
- Чворови из партиција са већом тежином имају приоритет у размјењивању
- Процес се завршава када се размјеном више не може поправити баланс између партиција

"Algorithm for partition refinement ток алгоритма

Улаз: Иницијална партиција Partitions, добијена у првој фази

1. Одреди релацију сусједства између партиција
2. Поредај партиције у опадајућем поретку према тежинама
3. Постави индексТренутнеПартиције := 0;
4. Постави побољшање := false; тренутнаПартиција := null;
6. While (индексТренутнеПартиције $\leqslant$ Partitions.length) do
7.   тренутнаПартиција := Partitions[индексТренутнеПартиције];
8.   Поредај сусједне партиције у растућем редослиједу
9.   foreach сусједнаПартиција од тренутнаПартиција do
10.   побољшање = LOCALREFINEMENT (сусједнаПартиција, тренутнаПартиција);
12.   If (побољшање = false) then stop;
13.   If (побољшање = false) then индексТренутнеПартиције := индексТренутнеПартиције + 1;
15.   Else индексТренутнеПартиције := 0;
16.   Постави побољшање := false;
17.   UPDATEADJACENCYPARTITIONS (Partitions);
18.   Поредај партиције у опадајућем поретку према тежинама

Хеуристички методи - III алгоритам

"Генетски алгоритам"

- Предложен у раду B.Djuric, J. Kratica, D. Tosic, V. Filipovic, Solving the maximally balanced connected partition problem in graphs by using genetic algorithm, Computing and Informatics, 2008.
- Свака јединка се представља бинарним низом дужине $|V|$
- Елементи низа одговарају чворовима и указују на то којој компоненти дати чвор припада, тј. $i \in V_1$ ако $x_i = 1$ и $i \in V_2$ ако $x_i = 0$.
- Полазно рјешење се одређује на случајан начин.
- Могућа је појава недопустивих јединки (које одговарају неповезаним партицијама)
- Ту појаву рјешавамо тзв. казненом функцијом

Генетски алгоритам

Рачунање казнене функције

- Природно је да се у формулацију казнене функције укључи податак о броју компоненти повезаности.
- Ако су $nc(G_1)$ и $nc(G_2)$ бројеви компоненти повезаности од G_1, односно G_2, казнена функција f_{pen} се рачуна помоћу формуле

$$f_{pen} = (nc(G_1) + nc(G_2) - 2) * w_{max},$$

  гдје је w_{max} максимална тежина чвора у графу.
- За дато рјешење, алгоритам за сваку јединку рачуна вриједност функције циља помоћу формуле

$$obj(V_1, V_2) = |w(V_1) - w(V_2)| + f_{pen}.$$

- Примјењују се стандардни оператори једнопозиционог укрштања, модификовани оператор мутације и "фино градуисана турнамент селекција"

Хеуристички методи - IV алгоритам

Метода промјенљивих околина
Приказана у раду:
Matić, D. "A mixed integer linear programming model and variable neighborhood search for maximally balanced connected partition problem Applied Mathematics and Computation, 2014.

- Рјешење за дати граф $G=(V,E)$ је представљено као бинарни низ x дужине $|V|$.
- Елементи низа одговарају чворовима и указују на то којој компоненти дати чвор припада, тј. $i \in V_1$ ако $x_i = 1$ и $i \in V_2$ ако $x_i = 0$.
- Полазно рјешење се одређује на случајан начин.

Рачунање функције циља

Током процеса претраживања, јасно је да се оваквом репрезентацијом могу јавити недопустива рјешења, што значи да су један (или оба) подграфа индукована са V_1 и V_2 (G_1 и G_2 респективно) неповезани.

- Уводимо казнену функцију која се рачуна на исти начин као код генетског алгоритма
- Функција циља се такође рачуна на сличан начин, тј.

$$obj(V_1, V_2) = |w(V_1) - w(V_2)| + f_{pen}.$$

Процедура размрдавања и локално претраживање

- VNS користи стандардне процедуре shaking и local search
- Да би дефинисала k-та околина,
  - На случајан начин се бира неких k елемената из S.
  - Сваком изабраном елементу мијења се припадајућа компонента.
  - $k_{min} = 2$, $k_{max} = \min\{30, |S|/2\}$.
- Локално претраживање је засновано на размјени припадајућих компоненти за парове елемената из S.

Експериментални резултати

- За тестирање MILP формулације кориштена су два MILP рјешавача: CPLEX 12.1 и Gurobi 4.0.
- Два скупа инстанци: правоугаоне мреже - BR инстанце (16 инстанци), уведене у (Blum and Roli, 2003.) и случајно генерисане инстанце - DKTF инстанце (21 инстанца), уведене у (Djurić et al. 2008).
- Добијени резултати су поређени са резултатима који се срећу у литератури.

Експериментални резултати за BR инстанце

Инст.	$\lvert V \rvert$	$\lvert E \rvert$	opt	$t_{cplex}(s)$	$t_{gur}(s)$	GA	t_{GA}	VNS	$t_{VNS}(s)$
05x05a	25	40	1	11.26	1016.99	opt	0.36	opt	0.34
05x05b	25	40	1	56.41	536.27	opt	0.37	opt	0.33
05x06a	30	49	0	4.11	0.17	opt	0.41	opt	0.50
05x06b	30	49	1	101.58	7077.59	opt	0.43	opt	0.60
05x10a	50	85	1	866.49	2539.09	opt	0.72	opt	2.30
05x10b	50	85	0	N/A	N/A	opt	0.91	opt	2.47
05x20a	100	175	0	N/A	N/A	opt	1.73	opt	18.12
05x20b	100	175	1	N/A	N/A	opt	1.77	opt	20.81
07x07a	49	84	0	3139.73	2528.21	opt	0.804	opt	2.79
07x07b	49	84	1	1053.79	N/A	opt	0.743	opt	3.31
07x10a	70	123	1	N/A	1858.10	opt	1.249	opt	6.38
07x10b	70	123	0	6112.54	N/A	opt	1.186	opt	7.33
10x10a	100	180	1	N/A	N/A	opt	1.809	opt	22.40
10x10b	100	180	1	N/A	N/A	opt	1.633	opt	25.05
15x15a	225	420	0	N/A	N/A	opt	4.439	opt	195.49
15x15b	225	420	0	N/A	N/A	opt	4.669	opt	216.16

Експериментални резултати за DKTF инстанце

Inst.	|V|	|E|	opt	t_{cplex}	t_{gur}	GA	t_{GA}	VNS	t_{VNS}
rnd01	20	30	1.14274	500.13	187.51	opt	0.52	opt	0.159
rnd02	20	50	0.01965	1899.18	480.89	opt	0.49	opt	0.198
rnd03	20	100	0	1307.85	36.10	0.00626	0.68	0.0008	0.281
rnd04	30	50	0	n.v:0.1817	5510.0	0.00915	0.58	0.00241	0.522
rnd05	30	70	0	5361.1	2081.6	0.01036	0.65	0.00148	0.505
rnd06	30	200	0	n.v:0.0108	403.56	0.00029	0.93	0.00007	1.036
rnd07	50	70	-	n.v:0.0328	n.v:0.4176	0.01608	0.92	0.00104	3.282
rnd08	50	100	0	n.v:0	5188.6	0.00531	1.10	0.00005	2.386
rnd09	50	400	0	n.v:0.0484	5112.0	0.00024	1.92	0.00008	5.864
rnd10	70	100	-	n.v:0.2083	N/A	0.01328	1.40	0.00022	7.678
rnd11	70	200	-	n.v:0	N/A	0.00023	1.737	0.00001	8.140
rnd12	70	600	-	n.v:0.0356	N/A	0.00232	2.910	0.00004	13.202
rnd13	100	150	-	n.v:0	N/A	0.00859	2.032	0.00037	23.473
rnd14	100	300	-	n.v:0	N/A	0.00137	2.355	0.00001	22.984
rnd15	100	800	-	n.v:0.7812	N/A	0.00155	3.812	0.00001	46.39
rnd16	200	300	-	n.v:0	N/A	0.00574	4.972	0.00006	203.31
rnd17	200	600	-	n.v:0	N/A	0.00058	5.255	0	206.34
rnd18	200	1500	-	n.v:0	N/A	0.00011	7.380	0.00001	411.47
rnd19	300	500	-	n.v:0	N/A	0.00039	7.747	0.00001	692.21
rnd20	300	1000	-	n.v:0	N/A	0.00025	7.560	0.00001	609.02
rnd21	300	2000	-	n.v:0	N/A	0.00008	11.61	0	1256.5

Правци даљег рада

- Формулација мјешовитог цјелобројног линеарног програмирања за уопштени проблем (партиционисање графа на k, $k \geqslant 2$ партиција.
- Развој генетског алгоритма за рјешавање уопштеног проблема
- Развој методе промјенљивих околина за рјешавање уопштеног проблема
- Примјена развијених метода за рјешавање сличних проблема из практичног живота

Хвала за пажњу!

доц. др Драган Матић
руководилац Студијског програма за математику и информатику
Природно математички факултет
Бања Лука
email: matic.dragan@gmail.com