

МЕТОД ВЕТВЕЙ И ОТСЕЧЕНИЙ
ДЛЯ ЗАДАЧИ РАЗБИЕНИЯ НА КЛИКИ^{*)}

Р. Ю. Симанчёв^{1,2,a}, И. В. Уразова^{2,b}, Ю. А. Кочетов^{3,c}

¹Омский научный центр СО РАН,
пр. Карла Маркса, 15, 644024 Омск, Россия

²Омский гос. университет им. Ф. М. Достоевского,
пр. Мира, 55А, 644077 Омск, Россия

³Институт математики им. С. Л. Соболева СО РАН,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

E-mail: ^aosiman@rambler.ru, ^burazovainn@mail.ru, ^cjkochet@math.nsc.ru

Аннотация. Проводится численное исследование метода ветвей и отсечений, адаптированного для решения задачи разбиения на клики (CPR). Эта задача заключается в нахождении в полном рёберно-взвешенном графе семейства попарно не пересекающихся клик минимального общего веса. Рассматриваются два частных случая задачи CPR. Первый известен как задача агрегации бинарных отношений (ABRP), а второй — как задача аппроксимации графа (GAR). Для известного ранее класса фасетных неравенств многогранника задачи разработан алгоритм отсечения, включающий два новых основных элемента: поиск решения с заданной гарантированной точностью и процедуру локального поиска для решения задачи идентификации неравенств. Предложенный алгоритм отсечения используется для построения нижних оценок в методе ветвей и отсечений. Для поиска верхних оценок точного решения используются специальные эвристики. Проведён вычислительный эксперимент на случайно сгенерированных графах. Наш метод позволил найти оптимальное решение для ранее изученных экземпляров ABRP и новых задач большой размерности. Задача GAR оказалась более сложным в вычислительном отношении частным случаем задачи CPR. Кроме того, для нашего алгоритма выделены простые и сложные классы GAR. Табл. 5, рис. 1, библиогр. 32.

Ключевые слова: метод ветвей и отсечений, фасетное неравенство, локальный поиск.

^{*)}Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 18-07-00599).

Введение

Пусть $K_n = (V, E)$ — полный неориентированный граф без петель и кратных рёбер, $|V| = n$. Остовный подграф $H \subset K_n$ называется *M-графом*, если каждая его компонента связности является кликой или одновершинным графом. Множество всех *M*-графов в K_n обозначим через $\mu(V)$. Полагаем, что на множестве E рёбер графа K_n определён аддитивный вещественный функционал $c: E \rightarrow R$. Задача разбиения на клики (clique partitioning problem, CPP) заключается в нахождении *M*-графа, минимизирующего функционал c на множестве $\mu(V)$. Она является важной задачей кластерного анализа, возникающей при необходимости разбиения множества объектов на непересекающиеся группы и оптимизации той или иной меры межгрупповых отношений. В общем случае задача NP-трудна [1]. В этой связи в последнее время появилось много работ, посвящённых разработке приближённых методов её решения, включая эвристики (см., например, [2, 3] и др.). Высокая универсальность CPP приводит к широкому разнообразию реальных приложений, включая агрегацию бинарных отношений (aggregation of binary relations, ABRP) [1, 4], групповые технологии (group technology) [5, 6], обнаружение сообществ (community detection) [7].

Нас будут интересовать полиэдральные методы решения задачи. Для всякого подграфа $D \subseteq K_n$ через VD и ED будем обозначать его множества вершин и рёбер соответственно. С графом K_n свяжем евклидово пространство R^E размерности $\frac{n^2-n}{2}$, поставив в соответствие каждому ребру ось координат в R^E . Это пространство может рассматриваться как пространство вектор-столбцов с компонентами, индексированными элементами из E . Если $x \in R^E$ и $S \subseteq E$, то через $x(S)$ обозначим линейную форму $\sum_{e \in S} x_e$. Вектором инцидентий произвольного остовного графа $D \subseteq K_n$ называется вектор $x^D \in R^E$ с компонентами $x_e^D = 1$ при $e \in ED$ и $x_e^D = 0$ при $e \notin ED$. Последнее правило, очевидно, задаёт взаимно однозначное соответствие между множеством всех остовных подграфов графа K_n и множеством вершин единичного куба в R^E . В этих терминах определим многогранник *M*-графов или, что то же самое, многогранник задачи разбиения на клики следующим образом:

$$P_n = \text{conv}\{x^H \in R^E \mid H \in \mu(V)\},$$

а саму задачу разбиения на клики — так:

$$\min\{\rho(x) = c^T x \mid x \in P_n\}.$$

В полиэдральной постановке упомянутые выше частные случаи CPP имеют один и тот же многогранник и отличаются структурой целевых

функций. Как показывают вычислительные эксперименты [8, 9], посвящённые точному решению задачи разбиения на клики, структура вектора весов s существенно влияет на размерность доступной для решения задачи.

В настоящее время метод ветвей и отсечений является самой мощной основой для полиэдральных алгоритмов. Свойства и структура граней многогранника играют решающую роль в этом методе. Для СРР первые результаты в этом направлении были получены в [9, 10], где приведены формулировка целочисленного линейного программирования и первые классы фасетных неравенств: неравенства треугольника (triangle inequalities), 2-дольные неравенства (2-partition inequalities), 2-chorded cycle и 2-chorded wheel неравенства. Позднее в [6, 11, 12] были обнаружены новые опорные и фасетные неравенства и исследована вычислительная сложность соответствующих задач отделения (separation problem) [6, 13]. Сравнение нижних оценок для СРР, полученных с помощью некоторых фасетных неравенств, представлено в [6].

В настоящей работе мы рассматриваем два типа целевой функции. Первый является полиэдральной формализацией задачи агрегации бинарных отношений, а второй известен как задача аппроксимации графа. Рассмотрим эти целевые функции более подробно.

Задача агрегации бинарных отношений (ABRP). При заданных множестве n объектов V , p характеристиках и $(n \times p)$ -матрице $D = (d_{ik})$, где d_{ik} представляет свойство объекта i по отношению к характеристике k , найти «лучшее» разбиение набора объектов на непересекающиеся классы «однородных» объектов. В [9] каждая характеристика трактуется как бинарное отношение эквивалентности R_k на множестве V . Соответственно под «лучшим» понимается отношение эквивалентности R^* на V , которое минимизирует величину $\sum_{k=1}^p |R \triangle R_k|$, где

$$R \triangle R_k = \{(i, j) \mid (i, j) \in R \text{ и } (i, j) \notin R_k \text{ или } (i, j) \notin R \text{ и } (i, j) \in R_k\}.$$

Всякое отношение эквивалентности на множестве объектов можно понимать как M -граф $H \subset K_n$. Следовательно, $|R \triangle R_k| = |EH_k| + x^H(E\overline{H}_k) - x^H(EH_k)$, где H и H_k — M -графы, определяющие отношения эквивалентности R и R_k , $k = 1, \dots, p$, соответственно. Таким образом, в [9] целевая функция определяется как сумма вида

$$\rho_D(H) = \sum_{k=1}^p |R \triangle R_k| = \sum_{k=1}^p (|EH_k| + x^H(E\overline{H}_k) - x^H(EH_k)).$$

Отсюда получается полиэдральная постановка задачи агрегации бинарных отношений

$$\min \left\{ \rho_D(x) = c_0 + \sum_{e \in E} c_e x_e \mid x \in P_n \right\}, \quad (1)$$

где константа c_0 и веса c_e определяются равенствами

$$c_0 = \sum_{e \in E} |\{k \in \{1, \dots, p\} \mid e \in EH_k\}|,$$

$$c_e = p - 2|\{k \in \{1, \dots, p\} \mid e \in EH_k\}|.$$

Задача является NP-трудной [1]. В упомянутой работе [9] получены точные решения большого числа реальных прикладных задач типа АВРР. Среди них максимальными по размерности стали задачи, данные для которых взяты из голосований государств-членов ООН по резолюциям, принятым на 39-й сессии Генеральной Ассамблеи ООН в 1984 г. по вопросам в области прав человека ($n = 158$, $p = 3$), отношения к ядерному оружию ($n = 158$, $p = 15$) и помощи палестинским беженцам на Ближнем Востоке ($n = 158$, $p = 9$). Были также рассмотрены задачи со случайными данными, но в них число объектов не превышало 34, а число характеристик не превышало 13.

Задача аппроксимации графа (GAP). Пусть $G \subset K_n$ — некоторый априори заданный остовный подграф в K_n . Задачей аппроксимации графа G называется задача поиска M -графа H , минимизирующего на множестве $\mu(V)$ функционал

$$\rho_G(H) = |EG \cup EH| - |EG \cap EH|.$$

В терминах пространства R^E задача аппроксимации графа имеет вид

$$\min \{ \rho_G(x) = |EG| + x(E\bar{G}) - x(EG) \mid x \in P_n \}. \quad (2)$$

Впервые задача аппроксимации графа была сформулирована в [14]. Позже в ряде работ были найдены нетривиальные классы графов, на которых задача полиномиально разрешима [15, 16]. В [17] задача аппроксимации графа рассмотрена как частный случай задачи кластеризации деревьев и доказана её NP-трудность. Систематическое изучение задачи аппроксимации графа началось в прошлом десятилетии, когда задача была переоткрыта под разными именами (Correlation clustering, Cluster editing) различными группами авторов [18–20]. В частности, была установлена NP-трудность различных её вариантов [18, 20, 21], а для их решения предложены первые приближённые алгоритмы с гарантированной оценкой точности [18, 22, 23]. Лучший из известных на сегодняшний день приближённый алгоритм для задачи аппроксимации графа гарантированно находит решение не более чем в 2,5 раза хуже оптимального [24, 25].

Задача аппроксимации графа допускает также обобщение, связанное с присвоением весов рёбрам аппроксимируемого графа (weighted cluster

editing). В этом случае целевая функция определяется как разность между суммой весов рёбер из $EG \setminus EH$ и суммой весов рёбер из $EH \setminus EG$. При этом в постановке допускается наличие рёбер, обязательно входящих в аппроксимирующий M -граф, и рёбер, которым это запрещено. В работе [8] предлагаются специальные процедуры сокращения размерности, использующие ограничение на значение целевой функции, с дальнейшим использованием процедур отсечения и ветвления. В [8] приводятся результаты экспериментов на реальных прикладных задачах и задачах, построенных с помощью специальной рандомизированной процедуры. Проведённый вычислительный эксперимент показывает, что сочетание методов сокращения размерности и полиэдральных методов позволило решить примеры, содержащие до тысячи вершин.

В настоящей работе разработан алгоритм ветвей и отсечений для задачи разбиения на клики. Отличительными особенностями алгоритма являются использование процедур локального поиска для решения задачи отделения и построения верхних оценок оптимума, а также поиск решения с заданной гарантированной точностью. Предложенный алгоритм апробирован на известных и новых экземплярах ABRP и GAP. Проведённый вычислительный эксперимент показал, что ABRP достаточно эффективно решается предложенным алгоритмом (получены точные решения для $n = 300$ без применения ветвления), в то время как при решении GAP возникают серьёзные вычислительные трудности. При этом обе задачи NP-трудны, имеют один и тот же многогранник и отличаются только структурой целевых функций. Таким образом, необходимы дополнительные исследования в этом направлении, чтобы понять влияние этого аспекта на полиэдральные методы. Кроме того, для случаев GAP эффективность подхода зависит от плотности графов. Случайно сгенерированные экземпляры с малой и высокой плотностью просты для метода. Экземпляры со средней плотностью являются наиболее сложными. Часть результатов данной работы представлена на Международную конференцию «Теория математической оптимизации и исследование операций» (MOTOR2019).

Статья организована следующим образом. В разд. 1 изложены основные понятия и формулировки известных результатов о полиэдральной структуре CPP. В разд. 2 приведена ЦПП-модель задачи идентификации (separation problem) для класса опорных неравенств, предложенных в [10, 12], и описан алгоритм локального поиска для её решения. В разд. 3 предложены эвристики для построения начального рекорда и улучшения верхних оценок. Разд. 4 посвящён описанию процедуры отсечения для решения CPP с заданной гарантированной оценкой точности. В разд. 5 эта процедура встраивается в алгоритм ветвей и отсечений как способ

получения нижних оценок оптимума. Результаты вычислительного эксперимента по анализу и сравнению предлагаемых процедур, а также выявлению свойств различных случаев CPP представлены в разд. 6.

1. Основные понятия и факты

Для изложения результатов статьи введём следующие обозначения и понятия. Для любого графа $D \subset K_n$ через VD и ED будем обозначать множества его вершин и рёбер соответственно. Для ребра $e \in E$ будем также использовать запись uv , где u и v — вершины из V , инцидентные ребру e . Каждое множество рёбер $S \subset E$ индуцирует в K_n некоторый подграф T , в котором $ET = S$ и VT — множество вершин, инцидентных рёбрам из S . Там, где не возникает двусмысленности, граф, индуцированный множеством рёбер S , будем обозначать через S . Для подграфов D и F из K_n положим

$$D \cup F = ED \cup EF, \quad D \cap F = ED \cap EF.$$

Множество $P \subset R^E$ называется *многогранником*, если оно является выпуклой оболочкой конечного числа точек. Линейное неравенство $a^T x \leq a_0$ ($a, x \in R^E$, $a \neq 0$, $a_0 \in R$, T — знак транспонирования) называется *правильным к многограннику P* , если оно не нарушается ни одной из точек многогранника. Правильное неравенство называется *опорным к многограннику P* , если существует по крайней мере одна точка из P , обращающая его в равенство. Всякое опорное к P неравенство $a^T x \leq a_0$ порождает множество $\{x \in P \mid a^T x = a_0\}$, которое называется *гранью многогранника P* . Максимальные по включению собственные грани многогранника называются *фасетами*. Ясно, что фасетой является та и только та грань, размерность которой на 1 меньше размерности самого многогранника. Опорное неравенство, порождающее фасету, назовём *фасетным*.

Полиэдром в R^E называется множество решений конечной системы линейных уравнений и неравенств с переменными x_e , $e \in E$, если оно ограничено. Согласно теореме Вейля — Минковского для всякого многогранника существует совпадающий с ним в теоретико-множественном смысле полиэдр, и наоборот.

В [10, 12] было показано, что $(0, 1)$ -вектор $x \in R^E$ является вектором инциденций M -графа, если и только если он удовлетворяет системе

$$\begin{aligned} -x_{uv} + x_{uw} + x_{vw} &\leq 1, \\ x_{uv} - x_{uw} + x_{vw} &\leq 1, \\ x_{uv} + x_{uw} - x_{vw} &\leq 1, \end{aligned} \tag{3}$$

где $u, v, w \in V$ — всевозможные тройки попарно различных вершин,

$$x_{uv} \geq 0 \quad \text{для всех } uv \in E. \quad (4)$$

Полиэдр, определяемый системой (3)–(4), обозначим через M_n . Таким образом, $P_n \subset M_n$, причём включение именно строгое (в [12] приведены примеры нецелочисленных вершин полиэдра M_n) и каждая целочисленная вершина M_n является вектором инцидентий некоторого M -графа. Следуя [9], ограничения вида (3) будем называть *треугольниками*.

В [10] представлены несколько классов неравенств, порождающих фасеты многогранника M -графов. В частности, фасетными являются все ограничения полиэдра M_n . В [10] также отмечено, что при проведении вычислительного эксперимента по решению задачи агрегации бинарных отношений методом отсечения достаточно было использования в качестве отсекающих плоскостей лишь 2-дольных неравенств (2-partition inequalities). Неравенства этого класса определяются следующим образом. Пусть K^1 и K^2 — две непустые вершинно непересекающиеся клики в K_n и T — полный двудольный граф с долями VK^1 и VK^2 . Неравенство вида

$$x(ET) - x(EK^1) - x(EK^2) \leq \min\{|VK^1|, |VK^2|\}$$

называется *2-дольным неравенством*. Оно является правильным относительно многогранника P_n и определяет его фасету тогда и только тогда, когда $|VK^1| \neq |VK^2|$.

В [12] введён и изучен новый класс опорных к P_n неравенств. Пусть $U = \{u_1, u_2, \dots, u_k\}$ и $W = \{v_1, v_2, \dots, v_p\}$ — непустые подмножества множества V , $U \cap W = \emptyset$, $k \geq 1$, $p \geq 2$. Через T_i , $i = 1, 2, \dots, k$, обозначим звезду в K_n с центром в вершине u_i и лучами $u_i v_j$, $j = 1, 2, \dots, p$. Через K_p обозначим клику на множестве вершин W . Положим $T = \bigcup_{i=1}^k T_i$. Граф $T \cup K_p$ называется *k-парашютом*. С этим графом свяжем неравенство

$$x(ET) - x(EK_p) \leq \frac{k^2 + k}{2}.$$

В [12] доказано, что такое неравенство, индуцированное k -парашютом $T \cup K_p$, является опорным к многограннику P_n тогда и только тогда, когда $p \geq k$, а фасетным — тогда и только тогда, когда $k = 1$. В этой связи особый интерес для нас будут представлять именно 1-парашюты. Кроме того, нетрудно заметить, что пересечением класса 2-дольных неравенств и класса k -парашютов является класс 1-парашютов.

Для заданного 1-парашюта $T \cup K_p$ вершину графа K_n , образующую множество U , будем иногда называть *парашютистом*, а клику K_p — *куполом 1-парашюта*.

Для задач комбинаторной оптимизации в полиэдральной постановке при разработке процедур отсечения, использующих опорные неравенства как отсекающие плоскости, на передний план выходит задача идентификации (separation problem) [6, 13, 26, 27]. В общем случае она заключается в следующем. Даны некоторый класс L неравенств, опорных к многограннику P , и точка $\bar{x} \in R^E$. Требуется в классе L найти неравенство, строго отделяющее точку \bar{x} от многогранника P , либо доказать, что в L такого неравенства нет. В [28] показано, что при $NP \neq co-NP$ для полиэдра общего вида и класса неравенств, определяющих все фасеты выпуклой оболочки его целочисленных точек, не существует полиномиального алгоритма решения задачи идентификации. В связи с этим целесообразно рассматривать задачу идентификации применительно к конкретным классам неравенств относительно конкретных задач. Например, идентификация кликовых фасет для многогранника симметричной задачи коммивояжёра полиномиально разрешима, а идентификация гребневых неравенств и неравенств деревьев клик для той же задачи NP-трудна [27].

Для многогранника CPP очевидно, что так как число попарно различных троек вершин в K_n полиномиально по n , задача идентификации треугольников полиномиально разрешима. В [9] для поиска 2-дольного неравенства, отсекающего заданную точку, использовалась эвристическая процедура, поскольку сложностной статус задачи идентификации 2-дольных неравенств на тот момент был не известен. В 2001 г. в [6] была показана NP-трудность задачи идентификации для этого класса. Однако важно иметь в виду следующее. Поскольку многогранник P_n целиком лежит в единичном кубе пространства R^E , отсекаемую точку можно сразу полагать лежащей в единичном кубе. Это несколько усиливает постановку задачи идентификации. Возможно следующее ещё большее усиление. Поиск неравенства, отсекающего текущий нецелочисленный оптимум, начинается, как правило, с класса неравенств с полиномиально разрешимой задачей идентификации. Поэтому постановка задачи идентификации для класса 2-дольных неравенств может быть усилена до следующей: найдётся ли 2-дольное неравенство, отсекающее точку, удовлетворяющую всем треугольникам? В [6] даны замечания, позволяющие судить о NP-трудности задачи именно в этой усиленной формулировке. В [13] дано новое строгое доказательство этого факта для класса 1-парашютов и, как следствие, 2-дольных неравенств и k -парашютов.

2. Идентификация 1-парашютов

Задача идентификации 1-парашютов относительно многогранника M -графов примет следующий вид. При заданной точке $\bar{x} \in R^E$, $0 \leq \bar{x} \leq 1$,

среди всех 1-парашютов $T \cup K_p$ графа K_n требуется найти такой, для которого $\sum_{e \in ET} \bar{x}_e - \sum_{e \in EK_p} \bar{x}_e > 1$. Эта задача может быть формализована в виде задачи целочисленного линейного программирования. Определим следующие булевы переменные:

$x_u \in \{0, 1\}$, $u \in V$, — выбор парашютиста;

$y_u \in \{0, 1\}$, $u \in V$, — выбор вершин купола;

$z_{uv} \in \{0, 1\}$, $u, v \in V$, — выбор рѳбер купола;

$t_{uv} \in \{0, 1\}$, $u, v \in V$, — выбор рѳбер между парашютистом и куполом.

Требуется максимизировать функцию

$$\frac{1}{2} \left(\sum_{u,v \in V} \bar{x}_{uv} t_{uv} - \sum_{u,v \in V} \bar{x}_{uv} z_{uv} \right) \quad (5)$$

при условиях

$$z_{uv} \geq y_u + y_v - 1, \quad u, v \in V, \quad (6)$$

$$z_{uv} \leq y_u, \quad z_{uv} \leq y_v, \quad u, v \in V, \quad (7)$$

$$t_{uv} \geq x_u + y_v - 1, \quad u, v \in V, \quad (8)$$

$$t_{uv} \leq x_u, \quad t_{uv} \leq y_v, \quad u, v \in V, \quad (9)$$

$$x_u + y_u \leq 1, \quad u \in V, \quad (10)$$

$$\sum_{u \in V} x_u = 1, \quad (11)$$

$$\sum_{u \in V} y_u \geq 1, \quad (12)$$

$$x_u, y_u, z_{uv}, t_{uv} \in \{0, 1\}, \quad u, v \in V. \quad (13)$$

Ограничения (6), (7) определяют рѳбра внутри купола и фактически гарантируют равенство $z_{uv} = y_u y_v$. Ограничения (8), (9) определяют рѳбра между куполом и парашютистом и гарантируют равенство $t_{uv} = x_u y_v$. Ограничения (10) запрещают пересечение парашютиста с куполом. Ограничение (11) требует выбора ровно одного парашютиста. Ограничение (12) гарантирует непустой купол. В действительности для решения задачи идентификации достаточно проверить на совместность систему, образованную ограничениями (6)–(13) и неравенством

$$\sum_{u,v \in V} \bar{x}_{uv} t_{uv} - \sum_{u,v \in V} \bar{x}_{uv} z_{uv} > 2,$$

что, в целом, не облегчает задачу.

Как уже говорилось, задачу идентификации целесообразно рассматривать применительно к конкретным классам неравенств относительно конкретных задач. При такой стратегии помимо точных процедур решения часто используются эвристики (см., например, [9,27]). Так как задача

идентификации 1-парашютов NP-трудна, для её решения мы разработали процедуру локального поиска.

Начальное решение для задачи (5)–(13) генерируем с помощью жадной эвристики и определяем окрестность Flip + Swar [29, 30]. Набор соседних решений состоит из всех возможных решений задачи, которые могут быть получены по двум правилам: 1) можем добавить или удалить одну вершину из множества вершин текущего 1-парашюта, 2) можем заменить одну вершину в текущем 1-парашюте. Это хорошо известная окрестность для задач размещения [30, 31]. Чтобы сократить время выполнения поиска, используем рандомизированную окрестность [32] с параметром p . Каждый элемент окрестности Flip + Swar входит в эту рандомизированную окрестность с вероятностью p независимо от других элементов.

Итак, рассматривается следующая задача. Дана точка $\bar{x} \in M_n \setminus P_n$. В графе K_n нужно найти такой 1-парашют $T_u \cup K_p$, $p \geq 3$, что выполняется неравенство

$$\bar{x}(ET_u) - \bar{x}(EK_p) > 1.$$

Полагая величины \bar{x}_e весами рёбер $e \in E$, а величину $\bar{x}(ET_u) - \bar{x}(EK_p)$ — весом 1-парашюта $T_u \cup K_p$, можем сформулировать рассматриваемую задачу идентификации как задачу поиска в полном рёберно-взвешенном графе 1-парашюта с весом больше 1.

Всякий 1-парашют $T_u \cup K_p$ является кликой порядка $p + 1$. В связи с этим, зафиксировав какое-либо множество вершин, можем простым перебором найти на этом множестве самый тяжёлый 1-парашют. Формально эта процедура выглядит так.

Процедура $\gamma(W)$, просматривающая 1-парашюты на фиксированном множестве вершин $W = \{u_1, u_2, \dots, u_l\} \subset V$, где $l > 3$. Положим $\gamma(W) = 0$, T и K — графы с пустыми множествами вершин и рёбер.

Шаг i ($i = 1, 2, \dots, l$). Пусть T_i — звезда с центром в вершине u_i и лучами $u_i u_j$, $j = 1, \dots, i - 1, i + 1, \dots, l$, и K_i — клика на вершинах $u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_l$. Если $\bar{x}(ET_i) - \bar{x}(EK_i) > \gamma(W)$, то полагаем $\gamma(W) := \bar{x}(ET_i) - \bar{x}(EK_i)$, $T = T_i$ и $K = K_i$. Иначе оставляем всё неизменным. Переходим на шаг $i + 1$.

Если в результате процедуры получим $\gamma(W) > 1$, то $T \cup K$ — искомый 1-парашют на множестве вершин W . В процедуре $\gamma(W)$ не обязательно просматривать все вершины u_1, u_2, \dots, u_l . Можно остановиться сразу, как только $\gamma(W) > 1$.

Вообще говоря, в результате применения процедуры γ к конкретному множеству W может получиться $\gamma(W) \leq 1$. Тогда имеет смысл перейти к другому W и повторить процедуру. Переход к новому W

также целесообразен и в случае $\gamma(W) > 1$, чтобы увеличить разность $\gamma(W) - 1$, которую можно в определённом смысле считать «глубиной» отсечения. Это приводит к процедуре локального поиска, которую основываем на следующих трёх способах перехода к новому W .

1) Множество W' получается из W удалением одной вершины $u \in W$, т. е. $W' \subset W$ и $|W'| = |W| - 1$. Такой переход будем обозначать так: $W' = \varphi_1(W, u)$.

2) Множество W' получается из W добавлением одной вершины $u' \notin W$, т. е. $W \subset W'$ и $|W'| = |W| + 1$. Такой переход будем обозначать так: $W' = \varphi_2(W, u')$.

3) Множество W' получается из W удалением одной вершины $u \in W$ и добавлением одной вершины $u' \notin W$, т. е. $|W'| = |W|$ и $|W \cup W'| - |W \cap W'| = 2$. Для обозначения такого перехода будем использовать запись $W' = \varphi_3(W, u, u')$.

К случайно выбранному множеству W применяется каждый из указанных способов в рандомизированном режиме. А именно, переход φ_1 осуществляется последовательным просмотром вершин множества W , вершина $u \in W$ выбирается с вероятностью p_1 , запоминается такое W_1 , что $\gamma(W_1) = \max\{\gamma(\varphi_1(W, u)), u \in W\}$; переход φ_2 начинается с просмотра множества $V \setminus W$, вершина u' выбирается с вероятностью p_2 , запоминается такое W_2 , что $\gamma(W_2) = \max\{\gamma(\varphi_2(W, u')), u' \notin W\}$; переход φ_3 заключается в просмотре множества $W \times (V \setminus W)$, пара (u, u') выбирается с вероятностью p_3 , запоминается W_3 , для которого $\gamma(W_3) = \max\{\gamma(\varphi_3(W, u, u')), u \in W, u' \notin W\}$. В качестве нового множества вершин $W' = \varphi(W)$ выбирается то из множеств W_1 , W_2 и W_3 , на котором значение функции $\gamma(W)$ наибольшее.

Описанная процедура является одной итерацией эвристики, используемой нами для решения задачи идентификации 1-парашютов. При построении отсекающих 1-парашютов итерации продолжают до тех пор, пока не прекратится заметный рост функции γ .

3. Верхние оценки (рекорд)

Будем искать решение задачи разбиения на клики с заданной гарантированной точностью, а именно, решение \bar{x} будем называть α -приближённым, если $\rho(\bar{x}) \leq (1 + \alpha)\rho(x^{\text{opt}})$, где x^{opt} — оптимальное решение задачи, α — неотрицательный скаляр. В частности, если $\alpha = 0$, то α -приближённое решение является точным. В этой связи нам понадобится некоторое допустимое решение, с которым будем сравнивать текущий оптимум. В настоящем разделе предлагается эвристика для построения этого допустимого решения, которую также можно рассматривать

как алгоритм локального поиска для решения задачи с заданной гарантированной точностью.

3.1. Начальный рекорд. В графе $K_n = (V, E)$ будем понимать M -граф как правильное разбиение $r = \{U_1, U_2, \dots, U_t\}$ множества V на подмножества $U_i \subseteq V$, т. е. $U_i \cap U_j = \emptyset$ при $i \neq j$ и $\bigcup_{i=1}^t U_i = V$. При этом допускается существование в разбиении одновёршинных множеств. M -граф получается объединением каждого множества U_i в клику. Разбиение r также будем считать M -графом. Целевая функция вычисляется как $\rho(r) = c^T x^r$.

Начальный M -граф для процедуры построения рекорда. Необходимо построить M -граф с насколько возможно меньшим значением функции ρ . Определим граф $G_c \subset K_n$ как остовный подграф, индуцированный множеством рёбер $EG_c = \{e \in E \mid c_e < 0\}$. Через $N_{G_c}(u)$ обозначаем множество всех вершин, смежных с u относительно графа G_c .

ШАГ 1. Выбираем произвольно вершину $u_1 \in V$. Полагаем

$$U_1 = \{u_1\} \cup N_{G_c}(u_1).$$

ШАГ i , $i \geq 2$. Рассмотрим попарно не пересекающиеся подмножества U_1, U_2, \dots, U_{i-1} множества V . Если $V \setminus \bigcup_{j=1}^{i-1} U_j = \emptyset$, то конец: разбиение $r = \{U_1, U_2, \dots, U_{i-1}\}$ — требуемый M -граф. В противном случае выбираем $u_i \in V \setminus \bigcup_{j=1}^{i-1} U_j$ и полагаем

$$U_i = \{u_i\} \cup \left(N_{G_c}(u_i) \setminus \bigcup_{j=1}^{i-1} U_j \right).$$

На выходе получаем M -граф r и вычисляем $\rho(r)$.

Эту процедуру проделываем несколько раз при разном выборе вершин u_1, u_2, \dots, u_t . В качестве результата берём разбиение с наименьшим значением ρ . Далее с помощью следующей эвристики предпринимается попытка улучшения полученного рекорда. В окончательном варианте алгоритма ветвей и отсечений эта эвристика также используется для улучшения текущего рекорда (см. разд. 5).

3.2. Эвристика для построения рекорда. Будем использовать четыре способа перехода к новому разбиению r' .

1) Деление одного из множеств разбиения на два. В этом случае используем обозначение: $r' = \xi_1(r)$. Этот способ работает лишь при $t < n$.

Случайным образом выбираем среди всех элементов разбиения, мощность которых больше единицы, одно множество, скажем U_i . Случайным образом разбиваем его на два непустых подмножества U'_i и U''_i . Получаем новое разбиение $r' = \{U_1, \dots, U_{i-1}, U'_i, U''_i, U_{i+1}, \dots, U_t\}$. Вычисляем $\rho(r')$. Число элементов разбиения становится равным $t + 1$.

2) Объединение двух множеств разбиения в одно. Используем обозначение: $r' = \xi_2(r)$. Этот способ работает лишь при $t > 1$. Среди всех пар множества $\{1, 2, \dots, t\}^2$ случайным образом выбираем одну пару, скажем (i, j) . Разбиение r' получается объединением множеств U_i и U_j в одно множество. Вычисляем $\rho(r')$. Число элементов разбиения становится равным $t - 1$.

3) Перенос вершины из одного множества в другое. Обозначение: $r' = \xi_3(r)$. Случайно выбираем пару $(i, j) \in \{1, 2, \dots, t\}^2$. Из i и j случайно выбираем одно, скажем i . В U_i случайно выбираем одну вершину, скажем u . Полагаем $U_i := U_i \setminus \{u\}$, $U_j := U_j \cup \{u\}$. Если новое U_i окажется пустым, то число элементов разбиения станет равным $t - 1$, если нет, то останется равным t . Вычисляем $\rho(r')$.

4) Обмен вершинами. Обозначение $r' = \xi_4(r)$. Случайно выбираем пару $(i, j) \in \{1, 2, \dots, t\}^2$. Случайно выбираем вершины $u \in U_i$ и $v \in U_j$. Полагаем $U_i := U_i \setminus \{u\} \cup \{v\}$, $U_j := U_j \setminus \{v\} \cup \{u\}$. Вычисляем $\rho(r')$.

Начальное разбиение строится с помощью процедуры, описанной в начале данного раздела. Итерация эвристики заключается в следующем. К текущему разбиению r в рандомизированном режиме применяется каждый из указанных способов перехода. Среди получившихся разбиений $r_1 = \xi_1(r)$, $r_2 = \xi_2(r)$, $r_3 = \xi_3(r)$ и $r_4 = \xi_4(r)$ выбирается то, для которого значение функции ρ наименьшее. Итерации продолжаются до тех пор, пока не прекратится заметное убывание функции $\rho(r)$.

3.3. Округления. В алгоритме ветвей и отсечений также будем использовать два способа округления точки $\bar{x} \in B^E$, где B^E — единичный куб в R^E , до M -графа.

Процедура R_1 . Обозначим через $T_{\bar{x}}$ подграф в K_n , порождённый множеством рёбер $ET_{\bar{x}} = \{e \in E \mid \bar{x}_e = 1\}$. Пусть $G_c \cap T_{\bar{x}} = T_1 \cup T_2 \cup \dots \cup T_l$, где T_i , $i = 1, 2, \dots, l$, — компоненты связности графа $G_c \cap T_{\bar{x}}$. Каждую компоненту достраиваем до клики. Полученный M -граф обозначим через $R_1(\bar{x})$.

Процедура R_2 такая же, как и процедура R_1 , с тем отличием, что вместо графа $T_{\bar{x}}$ берётся граф $S_{\bar{x}}$, порождённый множеством рёбер $ES_{\bar{x}} = \{e \in E \mid \bar{x}_e > 0\}$. Полученный M -граф обозначим через $R_2(\bar{x})$.

Отметим, что эти процедуры применимы как для нецелочисленной точки, так и для целочисленной, не являющейся вектором инциденций

M -графа. При этом если \bar{x} целочисленная, то $T_{\bar{x}} = S_{\bar{x}}$ и, как следствие, $R_1(\bar{x}) = R_2(\bar{x})$.

4. Алгоритм отсечения и нижние оценки

В настоящем разделе будет описана процедура отсечения для решения задачи разбиения на клики. Мы следуем работе [9], где в качестве начальной релаксации взят единичный куб в R^E . Это целесообразно, так как при достаточно больших n количество ограничений полиэдра M_n имеет порядок n^3 и, несмотря на полиномиальность, на этапе поиска текущего непрерывного оптимума задача ЛП на полиэдре M_n решается слишком долго. Данная процедура используется как самостоятельный метод решения задачи, так и для построения нижних оценок в алгоритме ветвей и отсечений, который будет описан в разд. 5.

Пусть у нас имеется задача $\min\{c^T x \mid x \in P_n\}$ и класс L неравенств, правильных относительно многогранника M -графов P_n .

Алгоритм ФС. На вход алгоритма подаются Q — начальное множество; $L_0 = \emptyset$ — список отсечений; скаляр $\alpha \geq 0$ — коэффициент приближения; $\mu = -K$, где K достаточно большое число; x^* — вектор инцидентий M -графа, полученного с помощью процедуры из п. 3.1, $\text{гес} = c^T x^*$. Пару $(\text{гес}, x^*)$ назовём рекордом. При решении возникающих в процедуре задач линейного программирования используется симплекс-метод.

Полагаем $Q = B^E$, где B^E — единичный куб в R^E , задаваемый ограничениями $0 \leq x \leq 1$.

0-я (начальная) ИТЕРАЦИЯ. Положим

$$B = Q \cap \{x \in R^E \mid \mu \leq c^T x \leq \text{гес}\},$$

решаем задачу ЛП $\min\{c^T x \mid x \in B\}$, пусть \bar{x} — её оптимум. Переходим на 1-ю итерацию.

i -я ИТЕРАЦИЯ ($i = 1, 2, \dots$).

ШАГ 1 (текущий оптимум). Если \bar{x} — нецелочисленная точка, то переходим на шаг 3. Иначе — на шаг 2.

ШАГ 2 (проверка на M -граф). Подставляем точку \bar{x} в треугольные неравенства (3). Если она не нарушает этих неравенств, то \bar{x} — ОПТИМАЛЬНЫЙ M -ГРАФ. Иначе — на шаг 3.

ШАГ 3 (обновление рекорда). Применяем к точке \bar{x} процедуры округления R_1 и R_2 , описанные в п. 3.2, и полагаем

$$\text{гес} := \min\{\text{гес}, c^T x^{R_1(\bar{x})}, c^T x^{R_2(\bar{x})}\},$$

а в качестве x^* берём одну из точек x^* , $x^{R_1(\bar{x})}$ и $x^{R_2(\bar{x})}$, соответствующую новому значению гес . Переходим на шаг 4.

ШАГ 4 (проверка на α -приближённость). Если $(1 + \alpha)c^T \bar{x} \geq \text{res}$, то x^* — α -ПРИБЛИЖЁННЫЙ m -ГРАФ. Иначе переходим на шаг 5.

ШАГ 5 (идентификация неравенств класса L). Среди неравенств класса L локальным поиском, описанным в разд. 2, ищем семейство L_i неравенств, отсекающих точку \bar{x} . Если таких неравенств не найдено, то ЗАДАЧА НЕ РЕШЕНА. Иначе переходим на шаг 6.

ШАГ 6 (формирование новой релаксации). Для обеспечения неубывания целевой функции выбираем в L_{i-1} подсемейство неравенств, каждое из которых обращается точкой \bar{x} в равенство. Обозначим это подсемейство через L_{i-1}^* . Полагаем $\mu := \lceil c^T \bar{x} \rceil$, формируем новый релаксационный полиэдр

$$B := Q \cap \{x \in R^E \mid \mu \leq c^T x \leq \text{res}\} \\ \cap \{x \in R^E \mid a^T x \leq a_0 \text{ для всех неравенств из } L_{i-1}^* \cup L_i\}.$$

В качестве \bar{x} берём оптимальное решение задачи ЛП $\min\{c^T x \mid x \in B\}$. Переходим на $(i + 1)$ -ю итерацию.

Уточним некоторые детали алгоритма.

Замечание 1. В вычислительном эксперименте (см. разд. 6) в качестве класса L бралось объединение классов треугольников и 1-парашютов.

Замечание 2. Как показал проведённый вычислительный эксперимент, на шаге 5 мы получаем, как правило, достаточно много отсечений и добавление их всех на шаге 6 в текущую релаксацию задачи приводит к слишком быстрому росту числа ограничений. В этой связи при проведении вычислительного эксперимента применялись две стратегии. В первой стратегии добавлялось не более q неравенств, отсекающих точку \bar{x} , с наибольшими значениями величин $h(\bar{x}, a) = a^T \bar{x} - 1$, во второй — не более q неравенств с наибольшими значениями величин $d(\bar{x}, a) = \frac{a^T \bar{x} - 1}{\|a\|}$. В вычислительном эксперименте сравнение этих двух стратегий проводилось на основе фиксации времени, выделенного на построение нижней оценки. Лучшей считалась та стратегия, которая за выделенное время давала наибольший рост целевой функции. Величина q является параметром алгоритма.

Замечание 3. По сути в шаге 6 заложена процедура отбрасывания лишних ограничений, основанная на отказе от ограничений, не участвующих в получении оптимального решения на предыдущей итерации.

Замечание 4. Описанный алгоритм не всегда даёт решение задачи (см. шаг 5). Однако, как видно из результатов вычислительного эксперимента, приведённых в [9], для решения задачи агрегации бинарных

отношений хватало лишь треугольников и 1-парашютов. Забегая вперёд, отметим, что для задачи аппроксимации графа столь радужной картины не наблюдалось. Как правило, либо алгоритм работал слишком долго, либо имел место выход на «ЗАДАЧА НЕ РЕШЕНА». В этом случае мы начинали ветвление, а текущее значение целевой функции использовали как нижнюю оценку оптимума. Этот этап описан в разд. 5.

5. Алгоритм ветвей и отсечений

В алгоритм ветвей и отсечений будет встроен описанный в разд. 4 алгоритм отсечения FC с небольшой модификацией, состоящей в удалении шагов 1 и 2. Эту модификацию будем обозначать через FC'.

При ветвлении с целью уменьшения требуемой памяти сохраняем лишь грани единичного куба, порождаемые координатами, по которым происходит ветвление, нижние оценки для этих граней и текущие оптимумы, на которых эти оценки достигаются. Таким образом, для хранения узлов ветвления достаточно завести массив Ψ , содержащий $n^2 - n + 1$ столбцов. Каждая строка этого массива будет устроена следующим образом: если грань куба определяется условиями $\{x \in B^E \mid x_e = 0 \text{ при } e \in E_0 \text{ и } x_e = 1 \text{ при } e \in E_1\}$, то соответствующая строка в столбцах с именами из E_0 и E_1 будет содержать 0 и 1 соответственно; столбцы с именами из $E \setminus (E_0 \cup E_1)$ помечаются как пустые; следующие $\frac{n^2-n}{2}$ столбцов отведены под координаты текущего оптимума \bar{x} ; последний столбец в этом массиве отведён для нижней оценки целевой функции на M -графах, принадлежащих данной грани, причём в предположении целочисленности коэффициентов целевой функции $c^T x$ все эти нижние оценки будут целыми числами. Грань куба, определяемую строкой массива, с текущим оптимумом \bar{x} и нижней оценкой μ будем обозначать через (B, \bar{x}, μ) . Зададим также характеристику T — границу времени, выделенного на построение нижней оценки.

Алгоритм BFC.

0-я (начальная) итерация. Эта итерация заключается в применении алгоритма FC при $Q = B^E$ в течение времени, не превышающем T . На выходе получим один из следующих случаев:

- 1) «ОПТИМАЛЬНЫЙ M -ГРАФ» (см. шаг 2 алгоритма FC),
- 2) « α -ПРИБЛИЖЁННЫЙ M -ГРАФ» (см. шаг 4 алгоритма FC),
- 3) «ЗАДАЧА НЕ РЕШЕНА» (см. шаг 5 алгоритма FC),
- 4) время работы алгоритма FC достигло величины T .

В первых двух случаях алгоритм заканчивает свою работу. В третьем и четвёртом случаях вносим в массив Ψ строку (B^E, \bar{x}, μ) , т. е. строку, у которой все клетки с именами из E пустые и \bar{x} — текущий оптимум. Переходим на 1-ю итерацию алгоритма BFC.

Перед k -й итерацией алгоритма ВФС имеем массив Ψ и рекорд $(\text{гес}, x^*)$.
 k -Я ИТЕРАЦИЯ.

ШАГ 1 (выбор очередного узла и ветвление). В массиве Ψ выбираем строку (B, \bar{x}, μ) с минимальным значением μ . Если \bar{x} — нецелочисленная точка, то пусть e — имя любой её дробной координаты, если \bar{x} целочисленная, то пусть e — имя любой её координаты, которая в строке (B, \bar{x}, μ) массива Ψ не фиксирована. Формируем две новых грани

$$B' = \{x \in B \mid x_e = \lfloor \bar{x}_e \rfloor\}, \quad B'' = \{x \in B \mid x_e = \lfloor \bar{x}_e \rfloor + 1 \pmod{2}\}.$$

Переходим на шаг 2.

ШАГ 2 (нижняя оценка для B'). Применяем алгоритм FC' к задаче $\min\{c^T x \mid x \in Q\}$ при $Q = B'$. Если на некоторой FC' -итерации получаем ответ « α -ПРИБЛИЖЁННЫЙ M -ГРАФ», то алгоритм ВФС заканчивает работу. Если получен ответ «ЗАДАЧА НЕ РЕШЕНА» или время работы алгоритма FC' достигло величины T , то строку (B', x', μ) , где x' — текущее решение, полученное алгоритмом FC' , вносим в массив Ψ ; если получен ответ «МНОЖЕСТВО ДОПУСТИМЫХ РЕШЕНИЙ ПУСТО», то не вносим. Переходим на шаг 3.

ШАГ 3 (нижняя оценка для B''). Этот шаг полностью повторяет шаг 2 с той разницей, что $Q = B''$ и осуществляется переход на шаг 4.

ШАГ 4 (фильтрация массива Ψ). Сравниваем строки массива Ψ по столбцу оценок с рекордом и удаляем из него те строки, для которых $\mu \geq \text{гес}$. Переходим на $(k+1)$ -ю итерацию.

6. Вычислительный эксперимент

Вычислительный эксперимент разбит на две части. Первая часть посвящена исследованию процедуры отсечения, вторая — решению задач методом ветвей и отсечений. Все эксперименты проводились на реальных экземплярах для ABRP, взятых из [9], и случайных тестовых экземплярах для GAP. Для GAP генерируем коэффициенты целевой функции следующим образом. Для заданного значения плотности β аппроксимируемого графа каждое ребро графа K_n имеет вес -1 с вероятностью β и вес $+1$ с вероятностью $1-\beta$. Как увидим позже, значение плотности имеет решающее значение для нашего точного метода. Для решения возникающих в алгоритме задач линейного программирования использовался пакет IBM ILOG CPLEX.

При исследовании **процедуры отсечения** выделялись два этапа.

ПЕРВЫЙ ЭТАП: подбор параметров локального поиска для задачи идентификации 1-парашютов (см. разд. 2). Параметрами локального поиска являются вероятности p_1 , p_2 и p_3 , с которыми осуществляются переходы к смежной клике (переходы φ_1 , φ_2 и φ_3 соответственно).

Мы проводили эксперимент, полагая $p_1 = p_2 = p_3 = p$. В качестве лучшего выбиралось то значение параметра p , при котором значение функции $\gamma(W)$ в среднем оказывалось наибольшим. Стабилизация роста функции $\gamma(W)$ наступала достаточно быстро (меньше секунды), поэтому временных ограничений на эту часть эксперимента не накладывалось. Для параметра p рассматривались значения 0,25, 0,4, 0,5, 0,6 и 0,75 при $n = 100$. Наибольшее значение функции γ при стабилизации наблюдалось при $p = 0,6$ и $p = 0,75$. На рис. 1 показан пример работы процедуры. По горизонтальной оси отложены итерации, по вертикальной — значения функции γ .

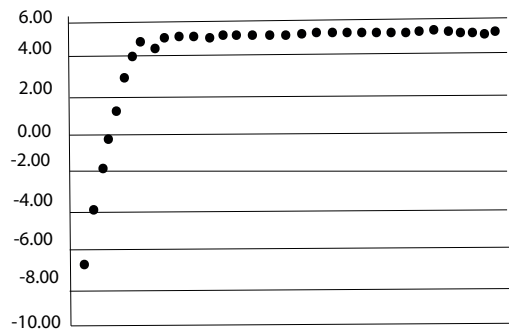


Рис. 1. Число вершин в последнем парашюте 15, итераций 100, $\gamma(W) = 5$

Кроме этого, выбор параметра p мы также оценивали с учётом целевой функции задачи. При тех же, что и на первом этапе, значениях p полученные 1-парашюты добавлялись по одному к ограничениям (3)–(4) со случайной целевой функцией вида $x(E\bar{G}) - x(EG)$ (задача GAP). Оценивалось значение целевой функции, полученное за 10 минут. Рассматривались задачи на числе вершин от 40 до 80, по 5 задач на каждую размерность. Результаты этой части эксперимента представлены в табл. 1. В первом столбце таблицы — имя задачи, во втором — число рёбер в графе G , в остальных — значения целевой функции при соответствующих p . В каждой строке жирным шрифтом выделены максимальные значения. Вновь чаще всего наибольший рост целевой функции наблюдается при $p = 0,6$ и $p = 0,75$.

Цель ВТОРОГО ЭТАПА описана в замечании 2 (разд. 4). На этом этапе мы ставим задачей не получение α -приближённого решения, а выбор комбинации стратегий, описанных в замечании 2. Для трёх различных значений параметра q , а именно $\frac{n}{10}$, $\frac{n}{5}$ и $\frac{n}{2}$, варьировались стратегии, определяемые величинами $h(\bar{x}, a) = a^T \bar{x} - 1$ и $d(\bar{x}, a) = \frac{a^T \bar{x} - 1}{\|a\|}$. Сравнение этих комбинаций проводилось на основе фиксации времени (10 минут),

Таблица 1

Выбор параметра p

Задача	$ EG $	$p_1 = p_2 = p_3 = p$				
		0,25	0,40	0,50	0,60	0,75
$f1 - 40$	385	216,64	216,84	216,89	218,49	218,23
$f2 - 40$	394	223,81	225,28	223,54	222,70	225,35
$f3 - 40$	400	227,33	225,82	225,86	228,16	226,55
$f4 - 40$	379	217,30	218,20	218,27	219,05	219,42
$f5 - 40$	400	227,38	225,50	225,14	225,78	227,38
$f1 - 50$	313	185,78	180,30	178,89	179,01	190,46
$f2 - 50$	306	177,15	175,05	174,32	177,01	177,19
$f3 - 50$	325	182,95	183,55	181,58	183,11	183,03
$f4 - 50$	312	167,25	178,94	178,63	178,49	178,97
$f5 - 50$	331	188,67	186,62	186,89	186,41	186,75
$f1 - 60$	447	252,13	246,32	252,85	249,32	252,05
$f2 - 60$	462	249,00	247,05	248,36	249,83	249,25
$f3 - 60$	460	247,75	247,13	245,92	245,83	245,17
$f4 - 60$	458	246,50	246,06	246,60	245,34	248,55
$f5 - 60$	430	233,19	232,75	234,46	234,34	232,31
$f1 - 70$	630	337,25	337,39	333,12	335,55	339,63
$f2 - 70$	605	320,29	320,30	321,50	320,50	321,00
$f3 - 70$	637	334,17	331,83	336,50	332,50	332,75
$f4 - 70$	578	307,56	309,75	306,25	311,25	306,65
$f5 - 70$	579	311,50	307,60	308,43	306,93	310,00
$f1 - 80$	778	404,00	407,20	404,75	402,50	405,50
$f2 - 80$	833	432,10	430,25	434,00	430,00	432,50
$f3 - 80$	785	411,50	408,50	410,00	413,14	411,00
$f4 - 80$	813	438,47	436,08	437,50	438,20	437,25
$f5 - 80$	821	427,08	427,90	428,03	429,00	425,83

лучшей считалась та комбинация, которая за выделенное время (одинаковое для всех) давала наибольшее значение целевой функции. В табл. 2 приведены результаты этой части эксперимента. Из табл. 2 видно, что более предпочтительной является комбинация $(q = \frac{n}{2}; d(\bar{x}, a))$.

Вторая часть эксперимента посвящена **алгоритму ветвей и отсечений**. Для всех задач использовались следующие параметры встроенных в алгоритм процедур: при решении задачи идентификации 1-парашютов были взяты параметры $p_1 = p_2 = p_3 = 0,6$; $q = \frac{n}{2}$; $T = 20$ минут; выбор добавляемых отсечений основывался на наибольших значениях величины $d(\bar{x}, a) = \frac{a^T \bar{x} - 1}{\|a\|}$. Такой выбор параметров алгоритма был основан на результатах первой части эксперимента.

Таблица 2

Определение стратегии выбора отсечений

Задача	$ EG $	$\frac{n}{10}$		$\frac{n}{5}$		$\frac{n}{2}$	
		f_h	f_d	f_h	f_d	f_h	f_d
$f1 - 40$	385	216,30	222,32	217,53	225,04	222,38	231,78
$f2 - 40$	394	227,50	229,33	228,45	240,59	234,69	244,23
$f3 - 40$	400	225,81	232,12	230,39	236,52	234,15	241,29
$f4 - 40$	379	217,27	224,61	220,41	230,36	225,44	234,73
$f5 - 40$	400	220,00	229,82	229,15	234,86	232,18	240,82
$f1 - 50$	313	173,99	186,14	175,42	185,23	177,04	186,43
$f2 - 50$	306	172,98	181,40	172,24	183,47	173,56	181,19
$f3 - 50$	325	178,70	184,22	181,25	188,81	183,58	187,48
$f4 - 50$	312	175,96	182,417	177,81	183,304	177,72	183,455
$f5 - 50$	331	183,25	194,493	185,44	195,26	184,40	192,363
$f1 - 60$	447	239,82	245,125	236,42	244,25	238,14	244,201
$f2 - 60$	462	247,50	253,688	243,99	255,339	243,39	253,01
$f3 - 60$	460	243,73	253,5	241,04	251,505	245,79	251,511
$f4 - 60$	458	243,99	254,075	244,91	252,352	245,75	251,785
$f5 - 60$	430	228,54	236,743	230,50	239,153	228,70	237,01
$f1 - 70$	630	325,00	331,15	325,33	334	325,00	334,227
$f2 - 70$	605	313,92	321,563	311,83	321,22	314,25	323
$f3 - 70$	637	328,50	337,75	329,50	333	327,00	335,45
$f4 - 70$	578	298,00	309,5	300,80	309,056	301,13	309,583
$f5 - 70$	579	302,33	312,667	303,17	308,636	302,50	307,559
$f1 - 80$	778	397,00	402	398,00	404	396,75	400
$f2 - 80$	833	429,00	428,278	426,00	430,833	427,00	428
$f3 - 80$	785	404,33	402,22	404,50	408,532	405,50	409,045
$f4 - 80$	813	416,50	416,56	416,00	417,253	417,00	421,05
$f5 - 80$	821	415,50	416,743	415,50	418,85	411,00	418,94

В работе [9] приведены точные решения большого количества задач агрегации бинарных отношений (т. е. задач разбиения на клики с целевой функцией вида $\rho_D(H)$ без константы c_0). Все данные в этих задачах имели реальное содержание и касались политики, бизнеса, социальных процессов. Алгоритм, которым решались данные задачи, был похож на алгоритм, описанный в настоящей работе. Отличие заключается в подходах к решению задачи идентификации 2-дольных неравенств, построения рекордов и сокращению числа ограничений. Максимальная размерность решённых в [9] задач была $n = 158$. Не вполне ясно, почему авторы не предприняли попытки решить задачи большей размерности, так как во всех задачах точное решение было получено за достаточно малое время, причём выход на этап ветвления не понадобился. В [9] также

решено некоторое количество задач типа ABRP со случайными данными, но их размерность не превышала $n = 34$, а число характеристик не превышало 13. Нам удалось решить все приведённые в [9] задачи, при этом характеристики работы алгоритма (число итераций, оптимальные значения) практически полностью совпали с [9]. Мы также рассмотрели несколько задач типа ABRP со случайными данными. Все они были решены без использования ветвления. Некоторые результаты по решению новых задач типа ABRP приведены в табл. 3.

Таблица 3

ABRP со случайными данными

n	Число характеристик	f_{opt}	Время	Число итераций
160	3	-7175	17 мин 13 с	5
160	4	-7732	23 мин 19 с	7
200	3	-7174	32 мин 51 с	5
200	4	-8572	25 мин 25 с	7
300	3	-7175	29 мин 16 с	5
300	4	-3209	119 мин 70 с	19

Дальнейшее исследование алгоритма ветвей и отсечений проводилось на задаче аппроксимации графа. Первый эксперимент этой части касался поиска α -приближённого решения задачи, второй — соотношению $+1$ и -1 в целевой функции. Результаты поиска α -приближённого решения приведены в табл. 4. Для каждой задачи выделялось время 3 часа. Кроме задач из табл. 1 и 2, было взято по пять задач размерности 20 и 30. Во втором столбце указана плотность аппроксимируемого графа, в третьем — указан параметр приближения α , а в четвёртом — содержатся результаты попытки решить задачу аппроксимации графа без ветвления, т. е. при снятом ограничении на параметр T (пустая ячейка означает, что задача не решена за 3 часа). В пятом столбце приведены результаты решения алгоритмом ветвей и отсечений, а в шестом — найденное α -приближённое значение целевой функции.

Кроме того, для задачи GAP исследовался вопрос зависимости работы нашего алгоритма от структуры целевой функции, а именно от соотношения отрицательных и положительных коэффициентов. Были взяты задачи с различной плотностью распределения -1 в целевой функции. Для всех задач искалось точное решение. Из табл. 4 видно, что за 3 часа решились задачи, у которых процент наличия -1 был либо больше семидесяти, либо меньше двадцати, причём эта картина наблюдалась практически вне зависимости от размерности задачи. В табл. 5

содержится следующее: n — число вершин в аппроксимируемом графе; плотность — доля -1 в целевой функции; f_{opt} — оптимальное значение целевой функции вида $\rho_G(H) - |EG|$; t — время решения задачи (пустая ячейка означает, что задача не решена за 3 часа); FC — число обращений к локальному поиску для решения задачи идентификации; BFC — число итераций ветвления.

Таблица 4

α -Приближённое решение GAP

Задача	Плотность	α	Алгоритм FC (с)	Алгоритм BFC (с)	α -прибли- жённое значение
$f1 - 20$	0,47	0,00	20,95	7,18	58
$f2 - 20$	0,56	0,00	17,55	9,55	65
$f3 - 20$	0,46	0,00	13,21	7,19	53
$f4 - 20$	0,52	0,00	9,21	9,3	59
$f5 - 20$	0,48	0,00	18,39	12,2	58
$f1 - 30$	0,53	0,00	2493,49	1785,3	155
$f2 - 30$	0,46	0,00	1318,22	652,29	138
$f3 - 30$	0,52	0,00	1157,13	693,32	151
$f4 - 30$	0,49	0,00	1818,56	1398,12	148
$f5 - 30$	0,51	0,00	1595,42	1271,09	152
$f1 - 40$	0,49	0,05		1540,13	330
$f2 - 40$	0,51	0,05		1600,08	327
$f3 - 40$	0,51	0,05		1671,02	330
$f4 - 40$	0,49	0,05		1586,45	315
$f5 - 40$	0,51	0,05		1652,34	321
$f1 - 50$	0,26	0,10		3534,12	267
$f2 - 50$	0,25	0,10		2976,31	263
$f3 - 50$	0,27	0,10		4985,07	279
$f4 - 50$	0,25	0,10		3109,43	269
$f5 - 50$	0,27	0,10		3988,16	276
$f1 - 60$	0,25	0,15		9720,38	398
$f2 - 60$	0,26	0,15		10681,14	396
$f3 - 60$	0,26	0,15		9307,46	407
$f4 - 60$	0,26	0,15		10142,09	407
$f5 - 60$	0,24	0,15		9223,27	366
$f1 - 70$	0,27	0,15		13980,09	565
$f2 - 70$	0,25	0,15		13320,54	548
$f3 - 70$	0,26	0,15		14400,05	562
$f4 - 70$	0,24	0,15		11520,31	517
$f5 - 70$	0,24	0,15		12180,24	519

Таблица 5

Соотношение ± 1 в GAP

n	Плотность	f_{opt}	t (с)	FC	BFC	n	Плотность	f_{opt}	t (с)	FC	BFC
50	0,95	-1093	4	0	3	60	0,3				
50	0,85	-909	8	0	5	60	0,2				
50	0,7	-483	35	2	16	60	0,15	-10	882	3	396
50	0,6					60	0,1	-17	798	7	355
50	0,5					60	0,05	-6	2840	31	220
50	0,4					70	0,95	-2133	20	0	4
50	0,3					70	0,85	-1715	73	0	13
50	0,2	-3	569	7	344	70	0,7	-903	347	0	26
50	0,15	-10	868	21	350	70	0,6				
50	0,1	-12	1896	54	419	70	0,5				
50	0,05	3,00	379	10	351	70	0,4				
60	0,95	-1590	12	0	4	70	0,3				
60	0,85	-1303	26	0	9	70	0,2				
60	0,7	-702	59	0	17	70	0,1	-5	4401	6	700
60	0,6					70	0,05	-22	2859	23	508
60	0,5					80	0,95	-3008	7	0	7
60	0,4					80	0,85	-2182	232	0	19

Заключение

В статье представлены результаты полиэдральной атаки на экземпляры задач ABRP и GAP. Разработан точный метод ветвей и отсечений, при котором наиболее сильные из найденных отсекающих плоскостей генерируются и собираются методом локального поиска. Наш метод позволил найти оптимальное решение для ранее изученных экземпляров ABRP и новых задач большой размерности. Для случаев GAP эффективность подхода зависит от плотности графов. Случайно сгенерированные экземпляры с малой и высокой плотностью просты для метода. Экземпляры со средней плотностью являются наиболее сложными. Кроме того, для $n > 30$ метод не может решить задачу GAP без ветвления и может сделать это для ABRP. При этом обе задачи NP-трудны, имеют один и тот же многогранник и отличаются только структурой целевых функций. Таким образом, необходимы дополнительные исследования в этом направлении, чтобы понять влияние этого аспекта на полиэдральные методы.

ЛИТЕРАТУРА

1. **Wakabayashi Y.** Aggregation of binary relations: Algorithmic and polyhedral investigations, Master Thesis. Univ. Augsburg, West Germany, 1986.
2. **Brimberg J., Janicijevic S., Mladenovic N., Urošević D.** Solving the clique partitioning problem as a maximally diverse grouping problem // *Optim. Lett.* 2017. Vol. 11. P. 1123–1135.
3. **Brusco M. J., Kohn H. F.** Clustering qualitative data based on binary equivalence relations: neighborhood search heuristics for the clique partitioning problem // *Psychometrika*. 2009. Vol. 74. P. 685–703.
4. **Marcotorchino F., Michaud P.** Heuristic approach to the similarity aggregation problem // *Methods Oper. Res.* 1981. Vol. 43. P. 395–404.
5. **Ham I., Hitomi K., Yoshida T.** Group technology: applications to production management. Dordrecht: Kluwer Acad. Publ., 1988.
6. **Oosten M., Rutten J. H. G. C., Spiessma F. C. R.** The clique partitioning problem: facets and patching facets // *J. Networks*. 2001. Vol. 38, No. 4. P. 209–226.
7. **Fortunato S.** Community detection in graphs // *Phys. Rep.* 2010. Vol. 486, No. 3. P. 75–174.
8. **Bocker S., Briesemeister S., Klau G. W.** Exact algorithms for cluster editing: evaluation and experiments // *Algorithmica*. 2011. Vol. 60. P. 316–334.
9. **Grotschel M., Wakabayashi Y.** A cutting plane algorithm for a clustering problem // *Math. Program. Ser. B*. 1989. Vol. 45. P. 59–96.
10. **Grotschel M., Wakabayashi Y.** Facets of the clique partitioning polytope // *Math. Program.* 1990. No. 47. P. 367–387.
11. **Grotschel M., Wakabayashi Y.** Composition of facets of the clique partitioning polytope // *Topics in Combinatorics and Graph Theory*. Heidelberg: Physica-Verl., 1990. P. 271–284.
12. **Симанчѳв Р. Ю., Уразова И. В.** О гранях многогранника задачи аппроксимации графов // *Дискрет. анализ и исслед. операций*. 2015. Т. 22, № 2. С. 86–101.
13. **Urazova I. V., Simanchev R. Yu.** Separation problem for k -parachutes // *Supp. Proc. DOOR 2016* (Vladivostok, Russia, Sept. 19–23, 2016) *CEUR-WS*, 2016. Vol. 1623. P. 109–114. <http://ceur-ws.org/Vol-1623/paperco16.pdf>
14. **Harary F.** On the notion of balance of a signed graph // *Mich. Math. J.* 1955. Vol. 2. P. 143–146.
15. **Фридман Г. III.** Одна задача аппроксимации графов // *Управляемые системы*. 1971. Вып. 8. С. 73–75.
16. **Zahn C. T.** Approximating symmetric relations by equivalence relations // *J. Soc. Ind. Appl. Math.* 1964. Vol. 12, No. 4. P. 840–847.
17. **Křivánek M., Morávek J.** NP-hard problems in hierarchical-tree clustering // *Acta Inform.* 1986. Vol. 23. P. 311–323.
18. **Bansal N., Blum A., Chawla S.** Correlation clustering // *Machine Learn.* 2004. Vol. 56. P. 89–113.

19. Ben-Dor A., Shamir R., Yakhimi Z. Clustering gene expression patterns // J. Comput. Biol. 1999. Vol. 6, No. 3–4. P. 281–297.
20. Shamir R., Sharan R., Tsur D. Cluster graph modification problems // Discrete Appl. Math. 2004. Vol. 144, No. 1–2. P. 173–182.
21. Агеев А. А., Ильев В. П., Кононов А. В., Талевнин А. С. Вычислительная сложность задачи аппроксимации графа // Дискрет. анализ и исслед. операций. 2006. Т. 13, № 1. С. 3–11.
22. Charikar M., Guruswami V., Wirth A. Clustering with qualitative information // J. Comput. Syst. Sci. 2005. Vol. 71, No. 3. P. 360–383.
23. Giotis I., Guruswami V. Correlation clustering with a fixed number of clusters // Theory Comput. 2006. Vol. 2, No. 1. P. 249–266.
24. Ailon N., Charikar M., Newman A. Aggregating inconsistent information: ranking and clustering // J. ACM. 2008. Vol. 55, No. 5. P. 1–27.
25. Van Zuylen A., Williamson D. P. Deterministic pivoting algorithms for constrained ranking and clustering problems // Math. Oper. Res. 2009. Vol. 34, No. 3. P. 594–620.
26. Симанчёв Р. Ю., Уразова И. В. Многогранник расписаний обслуживания идентичных требований параллельными приборами // Дискрет. анализ и исслед. операций. 2011. Т. 18, № 11. С. 85–97.
27. Padberg M. W., Rinaldi G. Facet identification for the symmetric traveling salesman polytope // Math. Program. 1990. Vol. 47. P. 219–257.
28. Karp R. M., Papadimitriou C. H. On linear characterizations of combinatorial optimization problems // SIAM J. Comput. 1982. Vol. 11. P. 620–632.
29. Alekseeva E., Kochetov Yu., Plyasunov A. Complexity of local search for the p -median problem // Eur. J. Oper. Res. 2008. No. 191. P. 736–752.
30. Iellamo S., Alekseeva E., Chen L., Coupechoux M., Kochetov Yu. Competitive location in cognitive radio networks // 4OR. 2015. Vol. 13, No. 1. P. 81–110.
31. Mladenović N., Brimberg J., Hansen P., Moreno-Pérez J. A. The p -median problem: a survey of metaheuristic approaches // Eur. J. Oper. Res. 2007. Vol. 179, No. 3. P. 927–939.
32. Diakova Z., Kochetov Yu. A double VNS heuristic for the facility location and pricing problem // Electron. Notes Discrete Math. 2012. No. 39. P. 29–34.

Симанчёв Руслан Юрьевич
Уразова Инна Владимировна
Кочетов Юрий Андреевич

Статья поступила
29 мая 2018 г.
После доработки —
4 июня 2019 г.
Принята к публикации
5 июня 2019 г.

THE BRANCH AND CUT METHOD
FOR THE CLIQUE PARTITIONING PROBLEM^{*)}

R. Yu. Simanchev^{1,2,a}, I. V. Urazova^{2,b}, and Yu. A. Kochetov^{3,c}

¹Omsk Scientific Center SB RAN,
15 Karl Marx Avenue, 644024 Omsk, Russia

²Dostoevsky Omsk State University,
55A Mir Avenue, 644077 Omsk, Russia

³Sobolev Institute of Mathematics,
4 Akad. Koptuyug Avenue, Novosibirsk, 630090, Russia

E-mail: ^aosiman@rambler.ru, ^burazovainn@mail.ru, ^cjkochet@math.nsc.ru

Abstract. A numerical study is carried out of the branch and cut method adapted for solving the clique partitioning problem (CPP). The problem is to find a family of pairwise disjoint cliques with minimum total weight in a complete edge-weighted graph. The two particular cases of the CPP are considered: The first is known as the aggregating binary relations problem (ABRP), and the second is the graph approximation problem (GAP). For the previously known class of facet inequalities of the polytope of the problem, the cutting-plane algorithm is developed. This algorithm includes the two new basic elements: finding a solution with given guaranteed accuracy and a local search procedure to solve the problem of inequality identification. The proposed cutting-plane algorithm is used to construct lower bounds in the branch and cut method. Some special heuristics are used to search upper bounds for the exact solution. We perform a numerical experiment on randomly generated graphs. Our method makes it possible to find an optimal solution for the previously studied cases of the ABRP and for new problems of large dimension. The GAP turns out to be a more complicated case of the CPP in the computational aspect. Moreover, some simple and difficult classes of the GAPs are identified for our algorithm. Tab. 5, illustr. 1, bibliogr. 32.

Keywords: branch and cut, facet inequality, local search.

^{*)}This research is supported by the Russian Foundation for Basic Research (Project 18-07-00599).

REFERENCES

1. **Y. Wakabayashi**, Aggregation of Binary Relations: Algorithmic and Polyhedral Investigations, *Master Thesis* (Universität Augsburg, West Germany, 1986).
2. **J. Brimberg, S. Janicijevic, S. Mladenovic, and D. Urošević**, Solving the clique partitioning problem as a maximally diverse grouping problem, *Optim. Lett.* **11**, 1123–1135 (2017).
3. **M. J. Brusco and H. F. Kohn**, Clustering qualitative data based on binary equivalence relations: Neighborhood search heuristics for the clique partitioning problem, *Psychometrika* **74**, 685–703 (2009).
4. **F. Marcotorchino and P. Michaud**, Heuristic approach to the similarity aggregation problem, *Methods Oper. Res.* **43**, 395–404 (1981).
5. **I. Ham, K. Hitomi, and T. Yoshida**, *Group Technology: Applications to Production Management* (Kluwer Acad. Publ., Dordrecht, 1988).
6. **M. Oosten, J. H. G. C. Rutten, and F. C. R. Spieksma**, The clique partitioning problem: Facets and patching facets, *J. Networks* **38** (4), 209–226 (2001).
7. **S. Fortunato**, Community detection in graphs, *Phys. Rep.* **486** (3), 75–174 (2010).
8. **S. Bocker, S. Briesemeister, and G. W. Klau**, Exact algorithms for cluster editing: Evaluation and experiments, *Algorithmica* **60**, 316–334 (2011).
9. **M. Grotschel and Y. Wakabayashi**, A cutting plane algorithm for a clustering problem, *Math. Program., Ser. B* **45**, 59–96 (1989).
10. **M. Grotschel and Y. Wakabayashi**, Facets of the clique partitioning polytope, *Math. Program.* **47**, 367–387 (1990).
11. **M. Grotschel and Y. Wakabayashi**, Composition of facets of the clique partitioning polytope, in *Topics in Combinatorics and Graph Theory*, (Physica-Verlag, Heidelberg, 1990), pp. 271–284.
12. **R. Yu. Simanchev and I. V. Urazova**, On the faces of the graph approximation problem polytope, *Diskretn. Anal. Issled. Oper.* **22** (2), 86–101 (2015) [Russian] [*J. Appl. Industr. Math.* **9** (2), 283–291 (2015)].
13. **I. V. Urazova and R. Yu. Simanchev**, Separation problem for k -parashutes, in *Supp. Proc. DOOR 2016, Vladivostok, Russia, September 19–23, 2016*, Vol. 1623 (Vladivostok, 2016), pp. 109–114. Available at <http://ceur-ws.org/Vol-1623/paperco16.pdf>.
14. **F. Harary**, On the notion of balance of a signed graph, *Michigan Math. J.* **2**, 143–146 (1955).
15. **G. Sh. Fridman**, A graph approximation problem, *Upravlyaemye Sistemy* **8**, 73–75 (1971) [Russian].
16. **C. T. Zahn**, Approximating symmetric relations by equivalence relations, *J. Soc. Indust. Appl. Math.* **12** (4), 840–847 (1964).
17. **M. Krivánek and J. Morávek**, NP-hard problems in hierarchical-tree clustering, *Acta Inform.* **23**, 311–323 (1986).

18. N. Bansal, A. Blum, and S. Chawla, Correlation clustering, *Machine Learning* **56**, 89–113 (2004).
19. A. Ben-Dor, R. Shamir, and Z. Yakhimi, Clustering gene expression patterns, *J. Comput. Biol.* **6** (3–4), 281–297 (1999).
20. R. Shamir, R. Sharan, and D. Tsur, Cluster graph modification problems, *Discrete Appl. Math.* **44** (1–2), 173–182 (2004).
21. A. A. Ageev, V. P. Il’ev, A. V. Kononov, and A. S. Talevnin, Computational complexity of the graph approximation problem, *Diskretn. Anal. Issled. Oper.* **13** (1), 3–11 (2006) [Russian] [*J. Appl. Indust. Math.* **1** (1), 1–8 (2007)].
22. M. Charikar, V. Guruswami, and A. Wirth, Clustering with qualitative information, *J. Comput. Syst. Sci.* **71** (3), 360–383 (2005).
23. I. Giotis and V. Guruswami, Correlation clustering with a fixed number of clusters, *Theory Comput.* **2** (1), 249–266 (2006).
24. N. Ailon, M. Charikar, and A. Newman, Aggregating inconsistent information: Ranking and clustering, *J. ACM* **55** (5), 1–27 (2008).
25. A. van Zuylen and D. P. Williamson, Deterministic pivoting algorithms for constrained ranking and clustering problems, *Math. Oper. Res.* **34** (3), 594–620 (2009).
26. R. Yu. Simanchev and I. V. Urazova, Scheduling unit-time jobs on the parallel processors polytope, *Diskretn. Anal. Issled. Oper.* **18** (11), 85–97 (2011) [Russian].
27. M. W. Padberg and G. Rinaldi, Facet identification for the symmetric traveling salesman polytope, *Math. Programming* **47**, 219–257 (1990).
28. R. M. Karp and C. H. Papadimitriou, On linear characterizations of combinatorial optimization problems, *SIAM J. Comput.* **11**, 620–632 (1982).
29. E. Alekseeva, Yu. Kochetov, and A. Plyasunov, Complexity of local search for the p -median problem, *Eur. J. Oper. Res.* **191**, 736–752 (2008).
30. S. Iellamo, E. Alekseeva, L. Chen, M. Coupechoux, and Yu. Kochetov, Competitive location in cognitive radio networks, *4OR* **13** (1), 81–110 (2015).
31. N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, The p -median problem: A survey of metaheuristic approaches, *Eur. J. Oper. Res.* **179** (3), 927–939 (2007).
32. Z. Diakova and Yu. Kochetov, A double VNS heuristic for the facility location and pricing problem, *Electron Notes Discrete Math.* **39**, 29–34 (2012).

Ruslan Yu. Simanchev
 Inna V. Urazova
 Yury A. Kochetov

Received May 29, 2018
 Revised June 4, 2019
 Accepted June 5, 2019