ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Март—апрель 2014. Том 21, № 2. С. 3–23

УДК 519.87

АЛГОРИТМ ВЕТВЕЙ И ГРАНИЦ ДЛЯ ЗАДАЧИ КОНКУРЕНТНОГО РАЗМЕЩЕНИЯ ПРЕДПРИЯТИЙ С ПРЕДПИСАННЫМ ВЫБОРОМ ПОСТАВЩИКОВ *)

В. Л. Береснев, А. А. Мельников

Аннотация. Изучается математическая модель, в которой две соперничающие стороны последовательно размещают свои предприятия, имея целью захватить потребителей и максимизировать прибыль. Модель представляется в виде задачи двухуровневого целочисленного программирования. В качестве оптимальных решений исследуемой задачи рассматриваются оптимальные некооперативные решения. Для отыскания приближённых и оптимальных решений задачи предлагается алгоритм ветвей и границ. Результаты вычислительного эксперимента показывают применимость алгоритма к решению индивидуальных задач малой и средней размерности.

Ключевые слова: двухуровневое программирование, оптимальное некооперативное решение, псевдобулева функция, метод ветвей и границ.

Введение

Рассматривается задача последовательного конкурентного размещения предприятий, являющаяся обобщением известной задачи размещения предприятий [3, 16]. В этой модели имеется две соперничающие стороны, которые последовательно открывают свои предприятия, стремясь захватить потребителей и максимизировать получаемую прибыль. Принятие решений соперничающими сторонами можно рассматривать как игру Штакельберга [18], а стороны, следуя терминологии этой игры, называть Лидером и Последователем. Математические задачи, получаемые в результате формализации этой игры, представляют собой задачи двухуровневого $(0,1)$-программирования [8–10, 13, 14] включающие задачу верхнего уровня (задачу Лидера) и задачу нижнего уровня (задачу

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 12–01–00077–a, 11–07–00474–a и 12–01–31090) и Министерства образования и науки Российской Федерации (соглашения 8227 и 8230).

Последователя). Эти задачи аналогичны задаче размещения предприятий с порядками [6, 7, 12], но их вид зависит от того, какие предположения используются при построении модели. Эти предположения касаются прежде всего принятого правила «захвата» потребителя одной из сторон, а также правил, используемых Лидером и Последователем при выборе открытого предприятия для обслуживания данного потребителя.

В задачах конкурентного размещения предприятий предполагается, что потребитель имеет собственные предпочтения, позволяющие ранжировать открытые предприятия. Потребителя захватывает та сторона, которая открывает наиболее предпочтительное для данного потребителя предприятие. Выбор поставщика для обслуживания потребителя, захваченного одной из сторон, в рассматриваемой ниже модели предписан. Им является наиболее предпочтительное для потребителя предприятие, открытое этой стороной. В [1, 11] исследуется задача конкурентного размещения предприятий в случае свободного выбора Последователем предприятия для обслуживания потребителя. В этой модели предполагается, что Последователь сам принимает решение о выборе предприятия, обслуживающего каждого из захваченных им потребителей.

В настоящей работе для рассматриваемой модели последовательного конкурентного размещения предприятий предлагается алгоритм ветвей и границ [3, 17] для поиска приближённых и оптимальных решений задачи. Важным элементом алгоритма является предложенный метод вычисления верхних границ для оптимальных значений рассматриваемых псевдобулевых функций. Идея метода применительно к частному случаю задачи изложена в [2]. В [5] для рассматриваемой задачи поиска оптимального некооперативного решения построен алгоритм вычисления верхней границы на всём множестве решений в предположении, что величины доходов являются монотонными. Аналогичные оценки построены в [1] для общего случая задачи последовательного конкурентного размещения предприятий, в которой Последователь использует правило свободного выбора предприятия для обслуживания захваченного потребителя. В [11] для последней задачи в случае поиска оптимального некооперативного решения построена верхняя граница для значений соответствующих псевдобулевых функций на подмножествах, заданных частичными $(0, 1)$-векторами. В нашей работе для рассматриваемой задачи поиска оптимального некооперативного решения предлагается алгоритм вычисления верхних границ для подмножеств решений, заданных частичными $(0, 1)$-векторами, в предположении, что величины доходов монотонны. При этом найдены новые свойства оптимальных решений,

использование которых позволяет существенно улучшить получаемые верхние границы.

Статья состоит из пяти разделов. В разд. 2 приводится формулировка задачи последовательного конкурентного размещения предприятий в виде задачи двухуровневого $(0,1)$-программирования. Вводится понятие оптимального некооперативного решения. Показано, что задача поиска такого решения сводится к задаче максимизации некоторых псевдобулевых функций. Разд. 3 посвящён описанию общей схемы алгоритма ветвей и границ для задачи максимизации псевдобулевых функций. Рассмотрен способ задания подмножеств $(0,1)$-векторов с использованием частичных $(0,1)$-векторов. В разд. 4 предлагается метод вычисления верхних границ для рассматриваемых псевдобулевых функций. В разд. 5 приводится алгоритм ветвей и границ для поиска оптимальных и приближённых (с априорной оценкой точности) некооперативных решений задачи последовательного конкурентного размещения предприятий. Обсуждаются результаты вычислительных экспериментов на тестовых примерах из библиотеки «Дискретные задачи размещения»[1)].

1. Задача последовательного конкурентного размещения предприятий

Задачу последовательного конкурентного размещения предприятий можно рассматривать как задачу Лидера в игре Штакельберга, в которой Лидер и Последователь последовательно размещают свои предприятия. Задача Лидера состоит в определении множества открываемых предприятий, максимизирующего получаемую прибыль при условии, что часть потребителей будет захвачена Последователем, который также стремится максимизировать свою прибыль. Как и в классической задаче размещения предприятий, для формальной записи задачи обозначим через $I = \{1, \ldots, m\}$ множество предприятий (возможных мест размещения предприятий), а через $J = \{1, \ldots, n\}$ — множество потребителей. Считаем, что предприятие $i \in I$ может быть открыто как Лидером, так и Последователем. Поэтому для любого $i \in I$ предполагаем, что заданные величины f_i и g_i равны фиксированным затратам на открытие предприятия i Лидером и Последователем соответственно. Если по каким-либо причинам Лидер или Последователь не может открыть предприятие i, то полагаем $f_i = +\infty$ или $g_i = +\infty$. Будем считать, что выбор открываемого предприятия для обслуживания потребителя $j \in J$ производится с учётом предпочтений потребителя j. Считаем, что пред-

[*1)] http://www.math.nsc.ru/AP/benchmarks/

почтения потребителя $j \in J$ задаются отношением линейного порядка $\succ_j$ на множестве I. Для i, $k \in I$ отношение $i \succ_j k$ означает, что из двух открытых предприятий i и k для потребителя $j \in J$ более предпочтительным является предприятие i. Отношение $i \succeq_j k$ означает, что либо $i \succ_j k$, либо $i = k$. Для любых $i \in I$, $j \in J$ через p_{ij} обозначим величину дохода, получаемого предприятием i, открытым Лидером или Последователем, при обслуживании потребителя j. Будем предполагать, что для всякого $j \in J$ доходы p_{ij}, $i \in I$, монотонны относительно порядка $\succ_j$, т. е. для любых $i, k \in I$, $i \succ_j k$, имеем $p_{ij} \geqslant p_{kj}$.

Пусть $I_0 \subset I$. Для всякого $j \in J$ обозначим через $i_j(I_0)$ элемент $i_0 \in I_0$ такой, что $i_0 \succeq_j i$ для любого $i \in I_0$. Если $I_0 = \{i \in I \mid w_i = 1\}$, где $w = (w_i)$, $i \in I$, — $(0,1)$-вектор, то для элемента $i_j(I_0)$ будем использовать также обозначение $i_j(w)$. Для $i \in I$ через $J_i(w)$ будем обозначать множество $\{j \in J \mid i \succeq_j i_j(w)\}$. Если $x = (x_i)$ и $z = (z_i)$ — два $(0,1)$-вектора, то через $x \vee z$ обозначим $(0,1)$-вектор $w = (w_i)$, где $w_i = \max(x_i, z_i)$, $i \in I$. Для определения стороны, захватывающей потребителя $j \in J$, принимается следующее правило. Пусть единичные компоненты $(0,1)$-вектора $x = (x_i)$, $i \in I$, означают предприятия, открытые Лидером, а единичные компоненты $(0,1)$-вектора $z = (z_i)$, $i \in I$, — предприятия, открытые Последователем. Полагаем, что потребитель $j \in J$ будет захвачен Лидером, если $i_j(x) \succeq_j i_j(z)$, и Последователем, если $i_j(z) \succ_j i_j(x)$. При выборе предприятия для обслуживания захваченного потребителя $j \in J$ считаем, что Лидер и Последователь используют правило *жёсткого выбора*, при котором таким предприятием будет $i_j(x)$, если потребитель захвачен Лидером, и $i_j(z)$, если потребитель захвачен Последователем. Использование этого правила моделирует ситуацию, в которой каждый потребитель самостоятельно выбирает обслуживающее его предприятие. Введём следующие переменные, аналогичные переменным классической задачи размещения предприятий:

x_i равна единице, если Лидер открывает предприятие $i \in I$, и нулю в противном случае;

x_{ij} равна единице, если предприятие $i \in I$, открытое Лидером, оказывается наиболее предпочтительным для потребителя $j \in J$ среди всех открытых Лидером предприятий, и нулю в противном случае;

z_i равна единице, если Последователь открывает предприятие $i \in I$, и нулю в противном случае;

z_{ij} равна единице, если предприятие $i \in I$, открытое Последователем, оказывается наиболее предпочтительным для потребителя $j \in J$ среди всех предприятий, открытых как Лидером, так и Последователем,

и нулю в противном случае.

С использованием введённых переменных задача последовательного конкурентного размещения предприятий формулируется как задача двухуровневого целочисленного программирования:

$$\max_{(x_i),(x_{ij})} \Big(-\sum_{i\in I} f_i x_i + \sum_{j\in J} \Big(\sum_{i\in I} p_{ij} x_{ij} \Big) \Big(1 - \sum_{i\in I} \widetilde{z}_{ij} \Big) \Big), \tag{1}$$

$$x_i + \sum_{k\in I | i \succ_j k} x_{kj} \leqslant 1, \quad i \in I,\ j \in J, \tag{2}$$

$$x_i \geqslant x_{ij}, \quad i \in I,\ j \in J, \tag{3}$$

$$x_i, x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J, \tag{4}$$

$((\widetilde{z}_i), (\widetilde{z}_{ij}))$ — оптимальное решение задачи (5)–(8):

$$\max_{(z_i),(z_{ij})} \Big(-\sum_{i\in I} g_i z_i + \sum_{j\in J} \sum_{i\in I} p_{ij} z_{ij} \Big), \tag{5}$$

$$x_i + z_i + \sum_{k\in I | i \succ_j k} z_{kj} \leqslant 1, \quad i \in I,\ j \in J, \tag{6}$$

$$z_i \geqslant z_{ij}, \quad i \in I,\ j \in J, \tag{7}$$

$$z_i, z_{ij} \in \{0,1\}, \quad i \in I,\ j \in J. \tag{8}$$

Как любая задача двухуровневого математического программирования задача (1)–(8) включает задачи верхнего уровня (1)–(4) и нижнего уровня (5)–(8). Задачу верхнего уровня будем обозначать через $\mathfrak{L}$, а задачу нижнего уровня — через $\mathfrak{F}$. Целевая функция (1) задачи $\mathfrak{L}$ выражает величину прибыли, получаемой Лидером с учётом стоимости открытых предприятий и потери части потребителей, захваченных Последователем. Условия (2)–(4) являются ограничениями задачи размещения с порядками. Неравенства (2) реализуют правило жёсткого выбора предприятия, открытого Лидером, для обслуживания потребителя. Эти же неравенства гарантируют, что для обслуживания каждого потребителя может быть выбрано только одно предприятие, открытое Лидером. Целевая функция (5) задачи $\mathfrak{F}$ выражает величину прибыли, получаемой Последователем. Неравенства (6) реализуют условия захвата потребителей Последователем при заданных предприятиях, открытых Лидером. В частности, эти условия показывают, что если предприятие открыто Лидером, то оно не может быть открыто Последователем. Остальные

условия задачи $\mathfrak{F}$ являются ограничениями классической задачи размещения предприятий. Для задачи (1)–(8) в целом будем использовать обозначение $(\mathfrak{L}, \mathfrak{F})$, а целевую функцию (1) задачи $\mathfrak{L}$ будем считать также целевой функцией задачи $(\mathfrak{L}, \mathfrak{F})$.

Допустимое решение $((x_i), (x_{ij}))$ задачи $\mathfrak{L}$ будем обозначать через X, а допустимое решение $((z_i), (z_{ij}))$ задачи $\mathfrak{F}$ — через Z. Пару $(X, \widetilde{Z})$, где X — допустимое решение задачи $\mathfrak{L}$, а $\widetilde{Z}$ — оптимальное решение задачи $\mathfrak{F}$, назовём *допустимым решением* задачи $(\mathfrak{L}, \mathfrak{F})$. Пусть $(X, \widetilde{Z})$, $X = ((x_i), (x_{ij}))$, $\widetilde{Z} = ((\widetilde{z}_i), (\widetilde{z}_{ij}))$, — допустимое решение задачи $(\mathfrak{L}, \mathfrak{F})$. Будем предполагать также, что если X — нулевое решение, то оптимальные решения $\widetilde{Z}$ задачи $\mathfrak{F}$ ненулевые. Поэтому далее допустимые решения $(X, \widetilde{Z})$ задачи $(\mathfrak{L}, \mathfrak{F})$ будем считать ненулевыми. Обозначим через $L(X, \widetilde{Z})$ значение целевой функции рассматриваемой задачи $(\mathfrak{L}, \mathfrak{F})$ на допустимом решении $(X, \widetilde{Z})$, а через $F(Z)$ — значение целевой функции задачи $\mathfrak{F}$ на допустимом решении Z.

Поскольку при некоторых допустимых решениях X задачи $\mathfrak{L}$ оптимальное решение $\widetilde{Z}$ задачи $\mathfrak{F}$ может быть выбрано не единственным образом, сформулируем некоторое правило, руководствуясь которым Последователь выбирает своё решение. Будем считать, что Последователь среди своих оптимальных решений выбирает наименее выгодное для Лидера. Допустимое решение $(X, \overline{Z})$ задачи $(\mathfrak{L}, \mathfrak{F})$ назовём *допустимым некооперативным решением* задачи $(\mathfrak{L}, \mathfrak{F})$, если $L(X, \overline{Z}) \leqslant L(X, \widetilde{Z})$ для всякого допустимого решения $(X, \widetilde{Z})$ задачи $(\mathfrak{L}, \mathfrak{F})$, а допустимое некооперативное решение $(X^*, \overline{Z}^*)$ назовём *оптимальным некооперативным решением*, если $L(X^*, \overline{Z}^*) \geqslant L(X, \overline{Z})$ для всякого допустимого некооперативного решения $(X, \overline{Z})$.

Заметим, что для любого допустимого решения X задачи $\mathfrak{L}$ соответствующее допустимое некооперативное решение $(X, \overline{Z})$ может быть построено алгоритмом, включающим два этапа. На этапе 1 при фиксированном решении X решается задача $\mathfrak{F}$ и вычисляется оптимальное значение F^* её целевой функции. На этапе 2 при фиксированном решении X решается следующая вспомогательная задача:

$$\max_{(z_i),(z_{ij})} \sum_{j \in J} \sum_{i \in I} p_{i_j(x)j} z_{ij}, \tag{9}$$

$$x_i + z_i + \sum_{k \in I | i \succ_j k} z_{kj} \leqslant 1, \quad i \in I,\ j \in J, \tag{10}$$

$$z_i \geqslant z_{ij}, \quad i \in I,\ j \in J, \tag{11}$$

$$z_i, z_{ij} \in \{0,1\}, \quad i \in I,\ j \in J, \tag{12}$$

$$-\sum_{i\in I} g_i z_i + \sum_{j\in J}\sum_{i\in I} p_{ij} z_{ij} \geqslant F^*. \tag{13}$$

Оптимальное решение $\overline{Z} = ((\overline{z}_i, \overline{z}_{ij}))$ этой задачи даёт искомое допустимое некооперативное решение $(X, \overline{Z})$ задачи $(\mathfrak{L}, \mathfrak{F})$. При этом величина $L(X, \overline{Z})$ будет одной и той же при любом оптимальном решении $\overline{Z}$ вспомогательной задачи (9)–(13). Заметим также, что само допустимое решение $X = ((x_i), (x_{ij}))$ задачи $\mathfrak{L}$ однозначно определяется $(0,1)$-вектором $x = (x_i)$. Поэтому любой $(0,1)$-вектор x однозначно определяет некоторое значение $L(X, \overline{Z})$ целевой функции рассматриваемой задачи на соответствующем допустимом некооперативном решении $(X, \overline{Z})$. Таким образом, получаем, что задачу поиска оптимального некооперативного решения задачи последовательного конкурентного размещения предприятий $(\mathfrak{L}, \mathfrak{F})$ можно представить как задачу максимизации некоторой псевдобулевой функции $f(x)$, $x \in B^m$. Эта функция задана неявным образом, и для вычисления её значения необходимо найти величину F^* — оптимальное значение целевой функции задачи нижнего уровня $\mathfrak{F}$, а затем — оптимальное решение вспомогательной задачи (9)–(13).

2. Метод ветвей и границ

Рассмотрим общую схему метода ветвей и границ с поиском в глубину для задачи максимизации псевдобулевой функции $f(x)$, $x \in B^m$. Оптимальное решение этой задачи обозначим через x^*. В соответствии с [3] будем предполагать, что заданы следующие функции, определённые на подмножествах $s \subseteq B^m$. Функцию $b(s)$, определяющую некоторое собственное подмножество множества s, будем называть *функцией ветвления*. Функцию $H(s)$ такую, что $H(s) \geqslant f(x)$ для всякого $x \in s$, будем называть *верхней границей*. Считаем, что $H(s) = f(x)$, если $s = \{x\}$. Рассмотрим также функцию $x(s)$, определяющую некоторое решение из множества s. Алгоритм, реализующий метод ветвей и границ с поиском в глубину, состоит из конечного числа однотипных шагов. На каждом шаге рассматриваются множество $D \subset B^m$, называемое *множеством неотброшенных решений*, подмножество $d \subset D$, называемое *множеством проверяемых решений*, и наилучшее известное на данном шаге решение $x^0 \in B^m$, называемое *рекордным решением*. На первом шаге D равно B^m, $d = B^m$, x^0 — произвольный элемент множества B^m. Пусть к очередному шагу имеются множества неотброшенных решений D и проверяемых решений $d \subset D$, а также рекордное решение x^0. Шаг начинается с вычисления значений функций $H(d)$ и $x(d)$. Если $f(x(d)) > f(x^0)$,

то полагается $x^0 = x(d)$. Шаг состоит в проверке множества d с целью выяснить, содержит ли оно решение лучше рекордного. Для этого проверяется справедливость неравенства $(1-\varepsilon)H(d) \leqslant f(x^0)$, где $\varepsilon \in [0,1)$ — параметр, задающий точность получаемого в результате работы алгоритма решения x^0. Если неравенство выполняется, то множество d отбрасывается и полагается $D = D \backslash d$. Если $D = \varnothing$, то алгоритм заканчивает работу, в противном случае полагается $d = b(D)$ и начинается следующий шаг. Если же указанное неравенство не выполняется, то полагается $d = b(d)$ и начинается следующий шаг. Алгоритм заканчивает работу через конечное число шагов и для полученного в результате его работы решения x^0 выполняется неравенство $(1-\varepsilon)f(x^*) \leqslant f(x^0)$.

Подмножества множества B^m удобно задавать с помощью так называемых частичных решений. Пусть $I = \{1, \dots, m\}$. Вектор $y \in \{0, 1, *\}^m$ назовём *частичным* $(0,1)$-*вектором*, или *частичным решением*. Частичное решение делит переменные псевдобулевой функции на переменные с заданным значением 0 или 1 и свободные. Для частичного $(0,1)$-вектора $y = (y_i),\ i \in I$, определим множества

$$I^0(y) = \{i \in I \mid y_i = 0\},\ I^1(y) = \{i \in I \mid y_i = 1\},\ I^*(y) = \{i \in I \mid y_i = *\}.$$

Вектор $x \in B^m$ назовём *продолжением* частичного решения y, если $I^0(y) \subseteq I^0(x)$ и $I^1(y) \subseteq I^1(x)$. Множество всех продолжений частичного решения y обозначается через $P(y)$. Частичное решение y назовём *упорядоченным*, если для него задан вектор порядка $\{i_1, \dots, i_q\} = I^0(y) \cup I^1(y)$, указывающий на очерёдность, в которой компонентам частичного решения присваивались значения 0 или 1. С упорядоченным частичным решением $y = (y_i)$ и вектором порядка $(i_1, \dots, i_q)$ помимо множества $P(y)$ можно связать ещё и некоторое множество $Q(y)$, $P(y) \subset Q(y)$. Чтобы определить множество $Q(y)$, для всякого k, $1 \leqslant k \leqslant q$, для которого $y_{i_k} = 1$, построим частичное решение $y(k) = (y_i(k)),\ i \in I$, такое, что

$$I^0(y(k)) = (I^0(y) \cap \{i_1, \dots, i_{k-1}\}) \cup \{i_k\},$$
$$I^1(y(k)) = I^1(y) \cap \{i_1, \dots, i_{k-1}\}.$$

Объединение множеств $P(y(k))$ для построенных частичных решений $y(k)$ и множества $P(y)$ образуют множество $Q(y)$. Полезность упорядоченного частичного решения состоит в том, что функция ветвления может быть определена таким образом, что на каждом шаге алгоритма ветвей и границ множества D и d будут задаваться некоторым упорядоченным частичным решением y и совпадать соответственно с множествами $Q(y)$ и $P(y)$ [3].

3. Верхняя граница и функция ветвления для задачи последовательного конкурентного размещения предприятий

Рассмотрим способ вычисления верхней границы $H(y)$ для исследуемой псевдобулевой функции $f(x)$ на подмножестве $P(y)$ продолжений частичного решения y. Для этого при фиксированном частичном решении $y = (y_i)$ рассмотрим задачу (1)–(4) с дополнительным ограничением

$$x_i = y_i, \quad i \in I^0(y) \cup I^1(y). \tag{14}$$

Задачу (1)–(4), (14) обозначим через $\mathfrak{L}(y)$, а задачу (1)–(4), (14), (5)–(8) — через $(\mathfrak{L}(y), \mathfrak{F})$. Задача $(\mathfrak{L}(y), \mathfrak{F})$ соостветствует задаче максимизации исследуемой псевдобулевой функции $f(x)$ на подмножестве $P(y)$, а верхняя оценка значения целевой функции задачи $(\mathfrak{L}(y), \mathfrak{F})$ на оптимальном некооперативном решении будет значением $H(y)$ искомой верхней границы на множестве $P(y)$. Напомним, что для всякого $j \in J$ доходы p_{ij}, $i \in I$, предполагаются монотонными относительно порядка $\succ_j$, т.е. $p_{ij} \geqslant p_{kj}$ для любых $i, k \in I$, $i \succ_j k$.

Укажем на некоторые свойства допустимых и оптимальных решений задачи $(\mathfrak{L}(y), \mathfrak{F})$. Для упрощения изложения рассмотрим множество $I \cup \{0\}$ и будем считать, что $p_{0j} = 0$ при любом $j \in J$ и $i \succ_j 0$ для $i \in I$. Пусть $y = (y_i)$ — частичное решение. Для $j \in J$ положим $i_j(y) = i_j(I^1(y))$, если $I^1(y) \neq \varnothing$, и $i_j(y) = 0$ в противном случае. Для $i \in I$ через $J_i(y)$ обозначим множество $\{j \in J \mid i \succ i_j(y)\}$. Рассмотрим множества

$$R(y) = \Big\{ i \in I^0(y) \cup I^*(y) \mid \sum_{j \in J_i(y)} p_{ij} < g_i \Big\},$$

$$S(y) = \Big\{ i \in I^*(y) \mid \sum_{j \in J_i(y)} (p_{ij} - p_{i_j(y)j}) \leqslant f_i$$
$$\text{и для всякого } j \in J_i(y) \text{ если } i \succeq_j k \succ_j i_j(y), \text{ то } k \in R(y) \Big\}.$$

Содержательный смысл множеств $R(y)$ и $S(y)$ поясняет

Лемма 1. *Пусть y — частичное решение. Для любого допустимого решения $(X, \widetilde{Z})$, $X = ((x_i), (x_{ij})$, $\widetilde{Z} = ((\widetilde{z}_i), (\widetilde{z}_{ij}))$, задачи $(\mathfrak{L}(y), \mathfrak{F})$ выполняется равенство $R(y) \cap I^1(\widetilde{z}) = \varnothing$, где $\widetilde{z} = (\widetilde{z}_i)$. Существует оптимальное некооперативное решение $(X, \overline{Z})$, $X = ((x_i), (x_{ij}))$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, задачи $(\mathfrak{L}(y), \mathfrak{F})$ такое, что $S(y) \cap I^1(x) = \varnothing$, где $x = (x_i)$.*

Доказательство. Покажем справедливость первого соотношения. Предположим, что для некоторого $i \in R(y)$ имеем $\widetilde{z}_i = 1$. Поскольку $J_i(x \vee \widetilde{z}) \subset J_i(y)$, то

$$\sum_{j \in J_i(x \vee \widetilde{z})} p_{ij} \leqslant \sum_{j \in J_i(y)} p_{ij} < g_i.$$

Это противоречит тому, что $\widetilde{Z}$ — оптимальное решение задачи $\mathfrak{F}$. Покажем, что выполнено второе утверждение. Предположим, что рассматриваемое оптимальное некооперативное решение $(X, \widetilde{Z})$ таково, что $x_{i_0} = 1$ для некоторого $i_0 \in S(y)$. Рассмотрим решение $X' = ((x'_i), (x'_{ij}))$ задачи $\mathfrak{L}(y)$, в котором вектор $x' = (x'_i)$ отличается от вектора $x = (x_i)$ только тем, что $x'_{i_0} = 0$. Задачу $\mathfrak{F}$ при фиксированных решениях X и X' обозначим через $\mathfrak{F}(X)$ и $\mathfrak{F}(X')$ соответственно. Покажем, что множества оптимальных решений задач $\mathfrak{F}(X)$ и $\mathfrak{F}(X')$ совпадают. Для этого ограничения (6) задачи $\mathfrak{F}(X)$ перепишем в виде

$$z_i + \sum_{k \in I | i \succ_j k} z_{kj} \leqslant 1, \quad i \in I,\ j \in J \tag{15}$$

$$\sum_{k | i_j(x) \succeq_j k} z_{kj} = 0, \quad j \in J. \tag{16}$$

Поскольку в силу доказанного выше для любого оптимального решения $\widetilde{Z} = ((\widetilde{z}_i), (\widetilde{z}_{ij}))$ задачи $\mathfrak{F}(X)$ выполняется равенство $\widetilde{z}_i = 0$ для $i \in R(y)$, к её ограничениям можно также добавить равенства

$$z_i = 0, \quad i \in R(y).$$

Задача $\mathfrak{F}(X')$ отличается от задачи $\mathfrak{F}(X)$ только ограничениями (16), которые заменяются на

$$\sum_{k | i_j(x') \succeq k} z_{kj} = 0, \quad j \in J. \tag{17}$$

Покажем, что множества оптимальных решений задач $\mathfrak{F}(X)$ и $\mathfrak{F}(X')$ совпадают. Действительно, если Z — допустимое решение задачи $\mathfrak{F}(X)$, то поскольку $i_j(x) \succeq_j i_j(x')$ для любого $j \in J$, будут выполняться и равенства (17). Обратно, пусть Z — допустимое решение задачи $\mathfrak{F}(X')$, и пусть для некоторого $j \in J$ имеем $i_j(x) \succ_j i_j(x')$. Так как $i_0 = i_j(x) \succ_j i_j(x') \succeq_j i_j(y)$ и $i_0 \in S(y)$, для всякого k такого, что $i_0 \succ_j k \succ_j i_j(x')$, имеем $k \in R(y)$, следовательно, $z_k = 0$. Поэтому равенства (16) выполняются.

Тем самым множества оптимальных решений задач $\mathfrak{F}(X)$ и $\mathfrak{F}(X')$ совпадают, и если $\widetilde{Z}$ — оптимальное решение, то $(X,\widetilde{Z})$ и $(X',\widetilde{Z})$ — допустимые решения задачи $(\mathfrak{L}(y),\mathfrak{F})$. Для этих решений с учётом монотонности доходов справедливы соотношения

$$L(X,\widetilde{Z})-L(X',\widetilde{Z})=-f_{i_0}+\sum_{j\in J_{i_0}(x)}(p_{i_0j}-p_{i_j(x')j})$$
$$\leqslant -f_{i_0}+\sum_{j\in J_{i_0}(y)}(p_{i_0j}-p_{i_j(y)j})\leqslant 0.$$

Следовательно, для допустимого некооперативного решения $(X,\overline{Z})$ найдётся допустимое некооперативное решение $(X',\overline{Z})$ такое, что

$$S(y)\cap I^1(x')=\varnothing,\quad L(X,\overline{Z})\leqslant L(X',\overline{Z}).$$

Лемма 1 доказана.

Далее, рассматривая оптимальные некооперативные решения, будем считать, что они обладают установленным леммой 1 свойством. В основе предлагаемого способа вычисления величины $H(y)$ лежит построение системы подмножеств $\{I_j(y)\}$, $I_j(y)\subset I$, $j\in J$, с использованием которой удаётся сформулировать достаточные условия захвата потребителей Последователем.

При заданном частичном решении $y=(y_i)$ для фиксированного элемента $j_0\in J$ сформулируем условия, позволяющие для всякого $i_0\in I$ выяснить, $i_0\in I_{j_0}(y)$ или $i_0\notin I_{j_0}(y)$. Если $y_{i_0}=0$, то $i_0\notin I_{j_0}(y)$. Пусть $y_{i_0}\neq 0$. Рассмотрим множество $N(i_0)=\{i\in I\mid i\succ_{j_0} i_0\}$. Если $N(i_0)=\varnothing$, то $i_0\in I_{j_0}(y)$. Пусть $N(i_0)\neq\varnothing$. Если $N(i_0)\cap I^1(y)\neq\varnothing$, то $i_0\notin I_{j_0}(y)$. Пусть $N(i_0)\neq\varnothing$ и $N(i_0)\cap I^1(y)=\varnothing$. Рассмотрим частичное решение $y'=(y'_i)$ такое, что $y'_i=y_i$ для $i\neq i_0$ и $y'_{i_0}=1$. Рассмотрим множества $R(y')$, $S(y')$ и $J(y,i_0)=\{j\in J\mid \text{если } i\succ i_j(y'), \text{ то } i\in N(i_0)\cup R(y')\}$. Заметим, что $J(i_0)\neq\varnothing$, поскольку $j_0\in J(i_0)$. Для всякого $k\in N(i_0)$ рассмотрим множество

$$J(y,k,i_0)=\left\{j\in J(y,i_0)\mid \text{если } i\succ_j k, \text{ то } i\in N(i_0)\cup I^0(y')\cup S(y')\right\}.$$

Считаем, что $i_0\in I_{j_0}(y)$, если для каждого $k\in N(i_0)$ выполняется неравенство $g_k>\sum\limits_{j\in J(y,k,i_0)}p_{kj}$, и $i_0\notin I_j(y)$, если найдётся $k\in N(i_0)$, для которого указанное неравенство нарушается. Так определённую систему подмножеств $\{I_j(y)\}$, $j\in J$, будем называть системой подмножеств,

определённой строгими неравенствами. Содержательный смысл множества $I_{j_0}(y)$, $j_0 \in J$, поясняет следующая лемма, устанавливающая, что если Лидер планирует получить прибыль от потребителя j_0, используя предприятие $i_0 \notin I_{j_0}(y)$, то потребитель j_0 будет захвачен Последователем.

Лемма 2. *Пусть y — частичное решение, $\{I_j(y)\}$, $j \in J$, — система подмножеств, определённая строгими неравенствами, а $(X, \overline{Z})$, $X = ((x_i), (x_{ij}))$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, — допустимое некооперативное решение задачи $(\mathfrak{L}(y), \mathfrak{F})$. Для всякого $j_0 \in J$ такого, что $p_{i_0j_0}x_{i_0j_0} > 0$ для некоторого $i_0 \notin I_{j_0}(y)$, выполняется равенство $\sum\limits_{i \in I} \overline{z}_{ij_0} = 1$.*

Доказательство. Рассмотрим $(0,1)$-векторы $x = (x_i)$ и $\overline{z} = (\overline{z}_i)$. Предположим, что для некоторых $j_0 \in J$, $i_0 \notin I_{j_0}(y)$ имеем $p_{i_0j_0}x_{i_0j_0} > 0$, но требуемое равенство не выполняется. Рассмотрим частичное решение $y' = (y'_i)$ такое, что $y'_i = y_i$ для $i \neq i_0$, $y'_{i_0} = 1$ и множество $N(i_0) = \{i \in I \mid i \succ_{j_0} i_0\}$. Поскольку $i_0 \notin I_{j_0}(y)$, то $N(i_0) \neq \varnothing$. Заметим, что $x_i = 0$, $\overline{z}_i = 0$ для $i \in N(i_0)$. Рассмотрим также множества $R(y')$ и $S(y')$. В силу леммы 1 имеем $\overline{z}_i = 0$ для $i \in R(y')$ и можем считать, что $x_i = 0$ для $i \in S(y')$. Построим множество

$$J(y, i_0) = \{j \in J \mid \text{если } i \succ_j i_j(y'), \text{ то } i \in N(i_0) \cup R(y')\}$$

и заметим, что $i_j(x) \succ_j i_j(\overline{z})$ для $j \in J(y, i_0)$. Так как $i_0 \notin I_{j_0}(y)$, найдётся $k \in N(i_0)$, для которого множество

$$J(y, k, i_0) = \{j \in J(i_0) \mid \text{если } i \succ_j k, \text{ то } i \in N(i_0) \cup I^0(y') \cup S(y')\}$$

таково, что $g_k \leqslant \sum\limits_{j \in J(y,k,i_0)} p_{kj}$. Заметим, что $k \succ_j i_j(x) \succ_j i_j(\overline{z})$ для $j \in J(y, k, i_0)$. Рассмотрим множества

$$J_L(k) = \{j \in J \mid k \succ_j i_j(x) \succ_j i_j(\overline{z})\},$$

$$J_F(k) = \{j \in J \mid k \succ_j i_j(\overline{z}) \succ_j i_j(x)\}$$

и решение $Z = ((z_i), (z_{ij}))$ задачи $\mathfrak{F}$, которое отличается от решения $\overline{Z}$ тем, что $z_k = 1$, $z_{kj} = 1$ для $j \in J_L(k)$, $z_{kj} = 1$, $z_{i_j(\overline{z})j} = 0$ для $j \in J_F(k)$. С учётом монотонности доходов и того, что $J(y, k, i_0) \subset J_L(k)$, имеем

$$F(Z) - F(\overline{Z}) = -g_k + \sum_{j \in J_L(k)} p_{kj} + \sum_{j \in J_F(k)} (p_{kj} - p_{i_j(\overline{z})j})$$

$$\geqslant -g_k + \sum_{j \in J(y,k,i_0)} p_{kj} \geqslant 0.$$

Следовательно, Z — оптимальное решение задачи $\mathfrak{F}$, а (X, Z) — допустимое решение задачи $(\mathfrak{L}(y), \mathfrak{F})$. Для допустимых решений (X, Z) и $(X, \overline{Z})$ имеем

$$L(X, \overline{Z}) - L(X, Z) \geqslant \sum_{j \in J(y,k,i_0)} p_{i_j(x)j} \geqslant p_{i_0 j_0} > 0.$$

Это противоречит тому, что $(X, \overline{Z})$ — допустимое некооперативное решение. Лемма 2 доказана.

Лемма 3. *Пусть y — частичное решение, $\{I_j(y)\}$, $j \in J$, — система подмножеств, определённая строгими неравенствами, а $(X, \overline{Z})$, $X = ((x_i), (x_{ij}))$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, — допустимое некооперативное решение задачи $(\mathfrak{L}(y), \mathfrak{F})$. Для всякого $j \in J$ справедливо равенство*

$$\Big(\sum_{i \in I} p_{ij} x_{ij}\Big)\Big(1 - \sum_{i \in I} \overline{z}_{ij}\Big) = \Big(\sum_{i \in I_j(y)} p_{ij} x_{ij}\Big)\Big(1 - \sum_{i \in I} \overline{z}_{ij}\Big).$$

Доказательство. Если $p_{ij} x_{ij} = 0$ для любого $i \in I$, то равенство выполняется. Пусть $p_{i_0 j} x_{i_0 j} > 0$ для некоторого $i_0 \in I$. Если $i_0 \in I_j(y)$, то равенство также верно. Если же $i_0 \notin I_j(y)$, то равенство справедливо, поскольку $\sum\limits_{i \in I} \overline{z}_{ij} = 1$ в силу леммы 1. Лемма 3 доказана.

При фиксированном частичном решении $y = (y_i)$ рассмотрим следующую задачу, которую будем называть *оценочной*:

$$\max_{(x_i),\,(x_{ij})} \Big(-\sum_{i \in I} f_i x_i + \sum_{j \in J} \sum_{i \in I_j(y)} p_{ij} x_{ij}\Big),$$

$$x_i + \sum_{k \in I | i \succ_j k} x_{kj} \leqslant 1, \quad i \in I,\ j \in J,$$

$$x_i \geqslant x_{ij}, \quad i \in I,\ j \in J,$$

$$x_i = y_i, \quad i \in I^0(y) \cup I^1(y),$$

$$x_i, x_{ij} \in \{0, 1\}, \quad i \in I,\ j \in J.$$

Оценочную задачу с системой подмножеств $I_j(y)$, $j \in J$, определённой строгими неравенствами, обозначим через $\mathfrak{B}(y)$, а значение её целевой функции на решении $X = ((x_i), (x_{ij}))$ — через $B(X)$.

Теорема 1. *Пусть y — частичное решение, $\{I_j(y)\}$, $j \in J$, — система подмножеств, определённая строгими неравенствами, $(X, \overline{Z})$, $X = ((x_i), (x_{ij}))$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, — оптимальное некооперативное решение задачи $(\mathfrak{L}(y), \mathfrak{F})$, X^0 — оптимальное решение задачи $\mathfrak{B}(y)$. Справедливо неравенство $L(X, \overline{Z}) \leqslant B(X^0)$.*

Доказательство. В силу леммы 3 для оптимального значения целевой функции задачи $(\mathfrak{L}(y), \mathfrak{F})$ выполняется неравенство

$$L(X, \overline{Z}) = -\sum_{i \in I} f_i x_i + \sum_{j \in J} \Big(\sum_{i \in I} p_{ij} x_{ij}\Big)\Big(1 - \sum_{i \in I} \overline{z}_{ij}\Big) \leqslant -\sum_{i \in I} f_i x_i + \sum_{j \in J} \sum_{i \in I_j(y)} p_{ij} x_{ij}.$$

Поскольку X — допустимое решение задачи $\mathfrak{B}(y)$, отсюда получаем

$$L(X, \overline{Z}) \leqslant B(X) \leqslant B(X^0).$$

Теорема 1 доказана.

Пусть X^0 — оптимальное решение задачи $\mathfrak{B}(y)$, а $(X^0, \overline{Z})$ — соответствующее допустимое некооперативное решение задачи $(\mathfrak{L}(y), \mathfrak{F})$. В качестве следствия из теоремы 1 сформулируем достаточное условие того, что оценка $B(X^0)$ является точной, а решение $(X^0, \overline{Z})$ — оптимальным некооперативным решением задачи $(\mathfrak{L}(y), \mathfrak{F})$.

Следствие 1. *Допустимое некооперативное решение $(X^0, \overline{Z})$, $X^0 = \big((x_i^0), (x_{ij}^0)\big)$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, оптимально для задачи $(\mathfrak{L}(y), \mathfrak{F})$, если для всякого $j \in J$ выполняется равенство*

$$\Big(\sum_{i \in I_j(y)} p_{ij} x_{ij}^0\Big) \sum_{i \in I} \overline{z}_{ij} = 0.$$

Отсюда вытекает, в частности, что если $\overline{Z}$ — нулевое решение, то $(X^0, \overline{Z})$ — оптимальное некооперативное решение задачи $(\mathfrak{L}(y), \mathfrak{F})$. Используя приведённый признак оптимальности решения $(X, \overline{Z})$, укажем правило выбора элемента $i(y) \in I^*(y)$, который «наилучшим» образом увеличивает множество единичных компонент частичного решения y. Пусть $\overline{z} = (\overline{z}_i)$ и

$$J_0 = \Big\{j \in J \mid \Big(\sum_{i \in I_j(y)} p_{ij} x_{ij}^0\Big) \sum_{i \in I} \overline{z}_{ij} \neq 0\Big\} \neq \varnothing.$$

Для $i \in I^*(y)$ положим $J_0(i) = \{j \in J_0 \cap J_i(\overline{z}) \mid i \in I_j(y)\}$, а через $i(y)$ обозначим такой элемент $i \in I^*(y)$, для которого величина $\sum\limits_{j \in J_0(i)} p_{ij}$ наибольшая. Определим функцию вычисления верхней границы $H(y)$ следующим образом. Если $y \in B^m$, то рассмотрим допустимое некооперативное решение $(X, \overline{Z})$ задачи $(\mathfrak{L}, \mathfrak{F})$, соответствующее $(0,1)$-вектору y. Положим $H(y) = L(X, \overline{Z})$ и $x(y) = y$. Если $y \notin B^m$, то берём $(0,1)$-вектор $x^0 = \big(x_i^0\big)$ из оптимального решения X^0 оценочной задачи $\mathfrak{B}(y)$ и полагаем $H(y) = B(X^0)$ и $x(y) = x^0$.

Воспользуемся определёнными в данном разделе функциями $i(y)$, $x(y)$, а также функцией вычисления верхней границы $H(y)$ для построения алгоритма поиска $(1-\varepsilon)$-приближённого решения задачи последовательного конкурентного размещения.

4. Алгоритм ветвей и границ. Результаты вычислительных экспериментов

Как показано выше, исследуемая задача $(\mathfrak{L}, \mathfrak{F})$ представляется в виде задачи максимизации некоторой псевдобулевой функции. Алгоритм ветвей и границ для задачи максимизации рассматриваемых псевдобулевых функций $f(x)$ включает начальный шаг и конечное число основных шагов и представляется следующим образом.

Начальный шаг. Имеем упорядоченное частичное решение $y = (y_i)$, $y_i = *$ для $1 \leqslant i \leqslant m$ и $q = 0$. Вычисляем верхнюю границу $H(y)$ и решение $x(y)$. Далее к вектору $x(y)$ применяем процедуру локального поиска, в результате чего получаем локально оптимальное решение, которое принимается за начальное рекордное решение x^0. Если $(1-\varepsilon)H(y) \leqslant f(x^0)$, то STOP, иначе GOTO движение вниз.

Основной шаг. Имеем упорядоченное частичное решение $y = (y_i)$ и вектор порядка $(i_1, \dots, i_q)$, $1 \leqslant q \leqslant m$. Если $q = m$, то вычисляем $f(x(y))$. Если $f(x(y)) > f(x^0)$, то $x^0 := x(y)$, GOTO движение вверх. Если $q < m$, то вычисляем $H(y)$ и $x(y)$. Если $(1-\varepsilon)H(y) \leqslant f(x^0)$, то GOTO движение вверх, иначе GOTO движение вниз.

Движение вверх. Полагаем $r := 0$ и вычисляем наибольший номер r, $1 \leqslant r \leqslant q$, для которого $y_{i_r} = 1$. Если $r = 0$, то STOP, иначе $y_{i_k} := *$ для $r < k \leqslant q$, $y_{i_r} := 0$, $q := r$ и GOTO основной шаг.

Движение вниз. Строим $I^1(x(y)) \backslash I^1(y) = \{i_{q+1}, \dots, i_{q+r}\}$, $r \geqslant 0$. Если $r = 0$, то вычисляем номер $i(y)$ и полагаем $i_{q+1} = i(y)$, $y_{i_{q+1}} = 1$, $q := q+1$, иначе полагаем $y_{i_{q+k}} := 1$ для $1 \leqslant k \leqslant r$, $q := q+r$. Строим

множество $S(y)=\{i_{q+1},\dots,i_{q+r}\},\ r\geqslant 0$. Если $r>0$, то $y_{i_{q+k}}:=0$ для $1\leqslant k\leqslant r,\ q:=q+r$. GOTO основной шаг.

Исследование качеств предлагаемого метода производилось на случайно сгенерированных тестовых примерах задачи конкурентного размещения предприятий на сети в виде дерева, взятых в классе примеров $TreeNE$ библиотеки тестовых примеров «Дискретные задачи размещения». Для данных примеров множества I и J совпадают с множеством V вершин некоторого случайного дерева. Процедура построения дерева итеративна и начинается с тривиального дерева, состоящего из единственной вершины. На каждой следующей итерации к уже построенному дереву добавляется новая висячая вершина, которая присоединяется к случайной вершине текущего дерева «длинным» ребром с вероятностью 0.1, либо «коротким» ребром с вероятностью 0.9. Длины «коротких» рёбер — случайные числа из интервала $[1,15]$, а «длинных» рёбер — из интервала $[100,150]$. Стоимость f_i и g_i открытия предприятия в вершине $i\in V$ для Лидера и Последователя — случайно выбранное целое число из интервала $[30,39]$ и $[20,29]$ соответственно. Предпочтения потребителей задаются расстояниями в дереве. Для каждого потребителя $j\in V$ из целочисленного интервала $[5,9]$ случайно выбирается значение его бюджета b_j. Доход p_{ij} от обслуживания потребителя j предприятием i полагается равным b_j в случае, если расстояние от потребителя j до предприятия i не превосходит 100. Для значений m, равных 20, 30, 40, 50 и 60, сгенерированы по 20 наборов входных данных задачи конкурентного размещения. Параллельная (многопоточная) реализация алгоритма осуществлена на языке программирования C#. Тестирование проводилось на компьютере под управлением операционной системы Windows Server 2008 R2 с двумя шестиядерными процессорами Intel Xeon X5675 с тактовой частотой ядер 3.07 ГГц и объёмом ОЗУ 96 Гб. Для решения задач ЦЛП использовалась библиотека классов Microsoft Solver Foundation 3.1, базирующаяся на решателе Gurobi 4.5. В качестве начального приближения выбиралось решение, полученное с помощью алгоритма из [4]. Время работы алгоритма ветвей и границ ограничено одним часом для каждого из примеров.

В табл. 1 и 2 приведены ключевые показатели работоспособности алгоритма при поиске $(1-\varepsilon)$-приближённого решения для различного числа рабочих потоков:

m — число точек на плоскости в примерах данной группы;

P — число рабочих потоков;

ε — гарантированная относительная погрешность решения;

Таблица 1

Результаты численных экспериментов

m	P	ε	S	p_{avg}	t_{min}	t_{avg}	t_{max}	N_{min}	N_{avg}	N_{max}
20	1	0	20	100	0	2	6	1	230	1041
20	1	0.3	20	100	0	1	6	1	177	959
20	1	0.5	20	100	0	1	6	1	125	727
20	12	0	20	100	0	4	11	12	247	1208
20	12	0.3	20	100	1	5	14	12	202	1143
20	12	0.5	20	100	3	6	14	16	150	903
30	1	0	20	100	9	261	807	668	6462	15548
30	1	0.3	20	100	1	202	751	54	4864	14404
30	1	0.5	20	100	0	131	700	1	3247	12450
30	12	0	20	100	15	81	199	836	5854	16120
30	12	0.3	20	100	17	60	154	126	4438	14873
30	12	0.5	20	100	14	55	164	71	2943	13032
40	1	0	3	69	301	3291	3601	3954	55018	165895
40	1	0.3	8	82	2	2738	3601	98	43082	164913
40	1	0.5	14	86	0	1595	3601	1	26781	164742
40	12	0	13	98	125	2306	3602	3490	112770	270576
40	12	0.3	13	98	87	1748	3605	260	78577	266396
40	12	0.5	16	99	70	1210	3603	218	47733	268737
50	1	0	0	27	3600	3600	3607	16279	84001	273633
50	1	0.3	6	50	335	2761	3601	2098	61266	269407
50	1	0.5	11	66	1	1933	3601	1	39936	199281
50	12	0	2	79	1405	3429	3608	47392	320299	1114543
50	12	0.3	8	86	91	2402	3603	2181	216383	1104509
50	12	0.5	12	93	54	1655	3602	124	175120	1016613
60	1	0	0	5	3600	3600	3602	9433	69233	298759
60	1	0.3	1	14	415	3441	3602	1435	47094	118441
60	1	0.5	3	26	3	3065	3601	1	24551	52917
60	12	0	0	34	3600	3601	3603	119133	295625	527817
60	12	0.3	1	46	2145	3455	3603	95273	191101	461580
60	12	0.5	4	77	134	3052	3604	134	116379	355659

S — число решённых примеров данной группы из 20 возможных;

p_{avg} — средняя по всем примерам данной группы доля отброшенных решений в процентах;

t_{min}, t_{avg} и t_{max} — минимальное, среднее и максимальное время в секундах, затраченное алгоритмом на решение примера из данной группы;

N_{min}, N_{avg} и N_{max} — минимальное, среднее и максимальное числа

вершин дерева ветвления, просмотренных алгоритмом при решении примеров данной группы;

$GAP_{\min}$, GAP_{avg} и $GAP_{\max}$ — минимальное, среднее и максимальное по всем примерам данной группы значения величины H/L^{rec}, где H — значение верхней границы на всем множестве допустимых решений, а L^{rec} — рекордное значение целевой функции на момент остановки алгоритма;

VL_{avg} — среднее по всем примерам данной группы значение целевой функции Лидера на лучшем найденном некооперативном решении;

VF_{avg} — среднее по всем примерам данной группы значение целевой функции Последователя на лучшем найденном некооперативном решении;

$|x|_{\text{avg}}$ — среднее по всем примерам данной группы число открытых предприятий Лидера в лучшем найденном некооперативном решении;

$|z|_{\text{avg}}$ — среднее по всем примерам данной группы число открытых предприятий Последователя в лучшем найденном некооперативном решении.

Как видно из табл. 1, тестовые примеры размерности 20 оказываются простыми: $(1-\varepsilon)$-приближённое некооперативное решение для всех рассматриваемых значений ε находится быстрее чем за 10 секунд. При этом алгоритм, работающий в одном потоке, справляется с задачами быстрее, и при росте ε время его работы сокращается, как и следовало ожидать. В случае двенадцати потоков время работы возрастает: при балансировке нагрузки рабочих потоков производится дополнительная работа, которая оказывается излишней. Уже на размерности 30 работа в параллельных потоках даёт преимущество, однопоточная реализация начинает проигрывать в скорости. Ускорение, однако, оказывается не двенадцатикратным: время работы сокращается приблизительно в три раза, что опять же является следствием неоптимальной балансировки нагрузки. Время, требуемое для доказательства оптимальности решения в примерах данного класса, исчисляется в зависимости от примера считанными секундами, либо несколькими минутами, но не превышает 15 минут для $P = 1$ и 4 минут — для $P = 12$. Перелом наблюдается при $m = 40$. Однопоточная реализация алгоритма доказывает оптимальность найденного решения только для трёх из двадцати предложенных примеров. Однако увеличение числа рабочих потоков, а также увеличение ε приводят к существенному росту продуктивности. Для того чтобы в этом убедиться, обратим внимание на значение p_{avg}, показывающее, какая доля от общего числа решений в среднем по примерам данного

класса оказывается просмотренной алгоритмом за отведённое ему время. Примеры малой размерности $m = 20$ и $m = 30$ решены полностью, и для них $p_{\text{avg}} = 100\%$. Для $m = 40$, $\varepsilon = 0$ и $P = 1$ в среднем 69% решений оказываются просмотренными, тогда как при $P = 12$ в среднем просмотрены уже 98% решений, а число примеров, в которых оптимальность найденного решения доказана, возрастает по сравнению с однопоточной реализацией с 3 до 13. Примеры большой размерности с $m = 50$ допускают поиск $\frac{1}{2}$-приближённых решений, соответствующих $\varepsilon = 0{,}5$: более половины таких примеров решены и при $P = 1$, и при $P = 12$. Следует отметить, что в последнем случае величина p_{avg} составляет 93%. Сравнительно небольшое увеличение временной квоты, выделенной алгоритму, позволит ему полностью завершить просмотр множества допустимых решений. Как видно из таблицы, среди сгенерированных примеров размерности $m = 60$ встречаются простые. При $P = 1$ и $\varepsilon = 0.3$ для решения одного из примеров потребовалось около семи минут. Однако лавинообразный рост трудоёмкости не позволяет справиться с большинством примеров данного набора.

Т а б л и ц а 2

Численные характеристики примеров

m	$GAP_{\min}$	GAP_{avg}	$GAP_{\max}$	VL_{avg}	VF_{avg}	$\lvert x\rvert_{\text{avg}}$	$\lvert z\rvert_{\text{avg}}$
20	1,0	2,7	8,3	9,3	23,8	1,6	1,6
30	1,6	5,3	17,7	12,5	57,3	2,3	2,2
40	1,7	4,7	13,6	17,6	84,0	3,1	3,1
50	1,8	4,6	14,0	25,1	128,2	3,5	3,3
60	1,6	3,7	6,4	27,7	145,7	4,7	3,6

В табл. 2 приведены некоторые численные характеристики, касающиеся сгенерированных примеров и найденных некооперативных решений. Значения величин $GAP_{\min}$ и $GAP_{\max}$ для различных m показывают, что в каждой группе тестовых примеров имеются такие, в которых верхняя граница довольно точно оценивает оптимальное значение целевой функции: для всех представленных m имеем $GAP_{\min} < 2$. Однако для всех рассматриваемых m встречаются и такие примеры, для которых оценка верхней границы оказывается существенно завышенной. Такие примеры требуют для решения больше времени, поскольку отбрасывание неперспективных подмножеств множества допустимых решений происходит на более поздних этапах, когда мощности данных подмножеств сравнительно невелики. В среднем оценка верхней границы превосходит величину F^{rec} в 2,7–5,3 раза. Как можно заметить из данных о величинах VL_{avg} и VF_{avg}, а также $|x|_{\text{avg}}$ и $|z|_{\text{avg}}$, в среднем Лидер получает

меньшую прибыль нежели Последователь при том, что открывает столько же предприятий.

ЛИТЕРАТУРА

1. **Береснев В. Л.** Алгоритмы локального поиска для задачи конкурентного размещения предприятий // Автоматика и телемеханика. — 2012. — № 3. — С. 12–27.
   Beresnev V. L. Local search algorithms for the problem of competitive location of enterprises // Automation and Remote Control. — 2012. — Vol. 73, N 3. — P. 425–439.
2. **Береснев В. Л.** Верхние оценки для целевых функций дискретных задач конкурентного размещения предприятий // Дискрет. анализ и исслед. операций. — 2008. — Т. 15, № 4. — С. 3–24.
   Beresnev V. L. Upper bounds for objective functions of discrete competitive facility location problems // J. Appl. Industr. Math. — 2009. — Vol. 3, N 4. — P. 419–432.
3. **Береснев В. Л.** Дискретные задачи размещения и полиномы от булевых переменных. — Новосибирск: Изд-во ин-та математики, 2005. — 408 с.
4. **Береснев В. Л., Гончаров Е. Н., Мельников А. А.** Локальный поиск по обобщённой окрестности для задачи оптимизации псевдобулевых функций // Дискрет. анализ и исслед. операций. — 2011. — Т. 18, № 4. — С. 3–16.
   Beresnev V. L., Goncharov E. N., Mel'nikov A. A. Local search with a generalized neighborhood in the optimization problem for pseudo-Boolean functions // J. Appl. Industr. Math. — 2012. — Vol. 6, N 1. — P. 22–30.
5. **Береснев В. Л., Мельников А. А.** Приближённые алгоритмы для задачи конкурентного размещения предприятий // Дискрет. анализ и исслед. операций. — 2010. — Т. 17, № 6. — С. 3–19.
   Beresnev V. L., Melnikov A. A. Approximate algorithms for the competitive facility location problems // J. Appl. Industr. Math. — 2011. — Vol. 5, N 2. — P. 180–190.
6. **Васильев И. Л., Климентова К. Б.** Метод ветвей и отсечений для задачи размещения с предпочтениями клиентов // Дискрет. анализ и исслед. операций. — 2009. — Т. 16, № 2. — С. 21–41.
   Vasil'ev I. L., Klimentova K. B. The branch and cut method for the facility location problem with clients' preferences // J. Appl. Indust. Math. — 2010. — Vol. 4, N 3. — P. 441–454.
7. **Васильев И. Л., Климентова К. Б., Кочетов Ю. А.** Новые нижние оценки для задачи размещения с предпочтениями клиентов // Журн. вычисл. математики и мат. физики. — 2009. — Т. 49, № 6. — С. 1055–1066.
   Vasilyev I. L., Klimentova K. B., Kochetov Yu. A. New lower bounds for the facility location problem with clients' preferences // Comput. Math. Math. Phys. — 2009. — Vol. 49, N 6. — P. 1010–1020.

8. **Кононов А. В., Кочетов Ю. А., Плясунов А. В.** Конкурентные модели размещения производства // Журн. вычисл. математики и мат. физики. — 2009. — Т. 49, № 6. — С. 1037–1054.
Kononov A. V., Kochetov Yu. A., Plyasunov A. V. Competitive facility location models // Comput. Math. Math. Phys. — 2009. — Vol. 49, N 6. — P. 994–1009.
9. **Плясунов А. В., Панин А. А.** Задача ценообразования. Ч. 1. Точные и приближённые алгоритмы решения // Дискрет. анализ и исслед. операций. — 2012. — Т. 19, № 5. — С. 83–100.
10. **Плясунов А. В., Панин А. А.** Задача ценообразования. Ч. 2. Вычислительная сложность // Дискрет. анализ и исслед. операций. — 2012. — Т. 19, № 6. — С. 56–71.
11. **Beresnev V.** Branch-and-bound algorithm for competitive facility location problems // Comput. Oper. Res. — 2013. — Vol. 40. — P. 2062–2070.
12. **Canovas L., Garcia S., Labbe M., Marin A.** A strengthened formulation for the simple plant location problem with order // Oper. Res. Lett. — 2007. — Vol. 35. — P. 141–150.
13. **Davydov I., Kochetov Yu., Plyasunov A.** On the complexity of the $(r|p)$-centroid problem in the plane // TOP (accepted). DOI: 10.1007/s11750-013-0275-y.
14. **Dempe S.** Foundations of bilevel programming. — Dordrecht: Kluwer Acad. Publ., 2002. — 332 p.
15. **Hammer P. L., Rudeanu S.** Pseudo-Boolean programming // Oper. Res. — 1969. — Vol. 17. — P. 233–261.
16. **Krarup J., Pruzan P. M** The simple plant location problem: survey and synthesis // Eur. J. Oper. Res. — 1983. — N 12. — P. 36–81.
17. **Mitten L. G.** Branch-and-bound method: general formulation and properties // Oper. Res. — 1970. — Vol. 18. — P. 24–34.
18. **Stackelberg H.** The theory of the market economy. — Oxford: Oxford Univ. Press, 1952. — 289 p.

Береснев Владимир Леонидович,
e-mail: beresnev@math.nsc.ru
Мельников Андрей Андреевич,
e-mail: melnikov@math.nsc.ru

Статья поступила
4 июля 2013 г.

Переработанный вариант —
15 октября 2013 г.