ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Ноябрь—декабрь 2010. Том 17, № 6. С. 3–19

УДК 519.725

ПРИБЛИЖЁННЫЕ АЛГОРИТМЫ ДЛЯ ЗАДАЧИ КОНКУРЕНТНОГО РАЗМЕЩЕНИЯ ПРЕДПРИЯТИЙ *)

В. Л. Береснев, А. А. Мельников

Аннотация. Рассматривается задача конкурентного размещения предприятий, в которой две соперничающие стороны (Лидер и Последователь) открывают последовательно свои предприятия, а каждый потребитель выбирает одно из открытых предприятий, исходя из своих предпочтений. Задача состоит в том, чтобы выбрать размещение предприятий Лидера так, чтобы получить максимальную прибыль, учитывая последующее размещение предприятий Последователем, который также стремится получить максимальную прибыль. Задача формулируется как задача двухуровневого целочисленного программирования. Предлагается способ вычисления верхней границы для величины максимальной прибыли Лидера. Соответствующий алгоритм состоит в построении классической задачи размещения предприятий на максимум и отыскании оптимального решения этой задачи. Одновременно с вычислением верхней границы строится начальное приближённое решение задачи конкурентного размещения предприятий. Предлагаются алгоритмы локального поиска для улучшения начального приближённого решения. Приводятся результаты вычислительного эксперимента с предложенными алгоритмами, позволяющие оценить точность получаемых приближённых решений и дать сравнительную оценку качества рассматриваемых алгоритмов построения приближённых решений исследуемой задачи.

Ключевые слова: задача двухуровневого программирования, оптимальное некооперативное решение, верхняя граница, приближённое решение, локальный поиск.

---

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 09–01–00059), Президиума РАН (программа фундаментальных исследований № 2, проект № 227), ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № 14.740.11.0219), АВЦП «Развитие научного потенциала высшей школы (2009–2010 гг.)» (проект 2.1.1/3235).

Введение

В статье исследуется задача, являющаяся обобщением хорошо известной задачи размещения предприятий (средств обслуживания) на максимум [1, 6]. В этой модели, в отличие от классической задачи размещения предприятий, рассматриваются две соперничающие стороны (Лидер и Последователь), которые открывают свои предприятия, преследуя цель максимизации своей прибыли. При этом каждый потребитель, исходя из собственных предпочтений, среди открытых предприятий выбирает наилучшее и приносит тем самым доход одной из сторон. По аналогии с игрой Штакельберга [13] процесс принятия решений в данной модели представляется состоящим из трёх этапов. На первом этапе Лидер открывает свои предприятия. На втором этапе Последователь, зная размещения предприятий Лидера, открывает свои предприятия. Наконец, на третьем этапе каждый потребитель, имея информацию обо всех открытых предприятиях, выбирает наилучшее.

Литературу, в которой указанный процесс конкурентного размещения предприятий формализован с использованием задач математического программирования, к настоящему времени можно считать обширной [3, 5, 7, 8, 10–12]. Большое внимание в этих работах уделяется обсуждению различных концепций оптимальных решений. Кроме того, в большинстве рассматриваемых моделей используются ограничения в виде равенств на число открытых предприятий. В связи с этим следует выделить работу [7], в которой, как и в классической задаче размещения предприятий, введены фиксированные затраты на их открытие.

Исследуемая задача конкурентного размещения предприятий формулируется как задача двухуровневого математического программирования [4]. Такая задача рассматривается в [2], где для «линейного» случая приводится эквивалентная формулировка в виде задачи двухуровневого псевдобулева программирования и предложен способ вычисления верхней границы для значений целевой функции задачи.

В настоящей статье уточняется понятие оптимального решения задачи конкурентного размещения предприятий и рассматриваются решения, названные оптимальными некооперативными решениями задачи. В развитие результатов работы [2] предлагается метод вычисления верхней границы для оптимального значения целевой функции. Соответствующий алгоритм состоит в построении классической задачи размещения предприятий специального вида и отыскании оптимального значения её целевой функции. Область применения этого метода шире, чем «линейные» задачи конкурентного размещения предприятий. При его использовании предполагается, что для каждого потребителя доход, получаемый предприятиями, есть величина, не возрастающая на множестве предприятий, упорядоченных по убыванию предпочтений данного потребителя. Кроме того, предлагаются алгоритмы построения приближённого решения задачи, представляющие собой процедуру локального поиска [9] относительно окрестности специального вида. Локальный поиск ведётся из начального приближённого решения, получаемого одновременно с вычислением верхней границы, а используемая в алгоритмах окрестность строится таким образом, чтобы включать в себя относительно небольшое число «перспективных» вариантов изменения текущего решения.

В разд. 1 приводится формулировка задачи конкурентного размещения предприятий в виде задачи двухуровневого целочисленного программирования и вводится понятие оптимального некооперативного решения. В разд. 2 предлагается метод построения верхней границы значения целевой функции задачи на оптимальном некооперативном решении и приводится алгоритм вычисления значения верхней границы. Разд. 3 посвящён описанию алгоритмов построения приближённых решений задачи, представляющих собой процедуры локального поиска по окрестности специального вида и различающиеся способами выбора элемента окрестности, улучшающего текущее решение. Приводятся результаты вычислительных экспериментов с этими алгоритмами, позволяющие оценить точность получаемых приближённых решений и провести сравнительный анализ качества предложенных алгоритмов.

1. Формулировка задачи

Обозначим через $I = \{1, \dots, m\}$ множество предприятий (возможных мест размещения предприятий), а через $J = \{1, \dots, n\}$ — множество потребителей. Считаем, что для всякого $i \in I$ заданы величины f_i и g_i, равные фиксированным затратам на открытие предприятия i Лидером и Последователем соответственно. Для $i \in I$ и $j \in J$ через p_{ij} обозначим величину дохода, получаемого предприятием i при обслуживании потребителя j.

Считаем, что для всякого $j \in J$ на множестве I задано отношение порядка $\succ_j$, показывающее предпочтения потребителя j при выборе им предприятия. Отношение $i \succ_j k$ для $i, k \in I$ означает, что из двух открытых предприятий i и k потребитель j выберет предприятие i. Считаем также, что отношение $i \succcurlyeq_j k$ для $i, k \in I$ означает, что либо $i \succ_j k$, либо $i = k$.

Для формальной записи задачи используем следующие переменные:

$$x_i = \begin{cases} 1, & \text{если Лидер открывает предприятие } i \in I, \\ 0 & \text{в противном случае;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{если предприятие } i \in I \text{ является наилучшим для потребителя } j \in J \text{ среди всех предприятий, открытых Лидером,} \\ 0 & \text{в противном случае;} \end{cases}$$

$$z_i = \begin{cases} 1, & \text{если Последователь открывает предприятие } i \in I, \\ 0 & \text{в противном случае;} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \in I \text{ является наилучшим для потребителя } j \in J \text{ среди всех предприятий, открытых Лидером и Последователем,} \\ 0 & \text{в противном случае.} \end{cases}$$

С использованием указанных переменных и введённых обозначений задача конкурентного размещения предприятий записывается в следующем виде:

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

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

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

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

$$((\widetilde{z}_i), (\widetilde{z}_{ij})) \text{ — оптимальное решение задачи} \tag{5}$$

$$\max_{(z_i),(z_{ij})} \Big\{ -\sum_{i\in I} g_i z_i + \sum_{i\in I}\sum_{j\in J} p_{ij} z_{ij} \Big\}; \tag{6}$$

$$x_i + z_i + \sum_{k | i \succ_j k} z_{kj} \leqslant 1, \quad i \in I,\ j \in J; \tag{7}$$

$$z_i \geqslant z_{ij}, \quad i \in I,\ j \in J; \tag{8}$$

$$z_i, z_{ij} \in \{0,1\}, \quad i \in I,\ j \in J. \tag{9}$$

Целевая функция (1) сформулированной задачи выражает величину прибыли, получаемой Лидером с учётом потери части потребителей, «захваченных» Последователем. Неравенство (2) реализует правило выбора потребителем наиболее предпочтительного предприятия среди всех предприятий, открытых Лидером. Это же неравенство гарантирует, что каждый потребитель может выбрать для своего обслуживания не более одного открытого предприятия. Ограничение (3) означает, что потребитель для своего обслуживания может выбрать только открытое предприятие. Аналогичный смысл имеют целевая функция и ограничения задачи (6)–(9). Целевая функция (6) выражает суммарную прибыль, получаемую Последователем, а неравенство (7) обеспечивает выполнение правила выбора потребителем наиболее предпочтительного для него предприятия среди всех предприятий, открытых как Лидером, так и Последователем. Помимо этого ограничение (7) показывает, что если предприятие открыто Лидером, то оно не может быть открыто Последователем.

Представленная задача является задачей двухуровневого математического программирования. Как и всякая такая задача, она включает задачу первого уровня (1)–(4), которую будем называть *задачей Лидера* и обозначать через L, и задачу второго уровня (6)–(9), которую будем называть *задачей Последователя* и обозначать через F. Для задачи (1)–(9) будем использовать обозначение (L, F).

Обозначим через $X = ((x_i), (x_{ij}))$ допустимое решение задачи L, а через $Z = ((z_i), (z_{ij}))$ и $\widetilde{Z} = ((\widetilde{z}_i), (\widetilde{z}_{ij}))$ — допустимое и оптимальное решения задачи F при фиксированном допустимом решении X соответственно. Пару $(X, \widetilde{Z})$ будем называть *допустимым решением задачи* (L, F).

Обозначим через $L(X, \widetilde{Z})$ значение целевой функции (1) на допустимом решении $(X, \widetilde{Z})$, а через $F(Z)$ — значение целевой функции (6) на допустимом решении Z. Если предположить, что при любом допустимом решении X для любых допустимых решений $(X, \widetilde{Z}_1)$ и $(X, \widetilde{Z}_2)$ выполняется равенство $L(X, \widetilde{Z}_1) = L(X, \widetilde{Z}_2)$, то оптимальным решением задачи (L, F) будет такое допустимое решение $(X^*, \widetilde{Z}^*)$, что неравенство $L(X^*, \widetilde{Z}^*) \geqslant L(X, \widetilde{Z})$ выполняется при любом допустимом $(X, \widetilde{Z})$.

Если указанное условие не выполняется, то для определения оптимального решения задачи (L, F) необходимы дополнительные сведения относительно выбора оптимального решения задачи F, используемого для вычисления целевой функции задачи (L, F). Другими словами, необходимо уточнить цель, которую преследует Последователь при выборе мест размещения своих предприятий. В зависимости от этого могут быть даны разные трактовки понятия оптимального решения задачи (L, F).

Далее будем считать, что Последователь при выборе своего наилучшего решения руководствуется правилом так называемого *некооперативного поведения*, когда из всех оптимальных решений задачи F выбирается такое, при котором значение целевой функции задачи (L, F) наименьшее.

При фиксированном допустимом решении X задачи L оптимальное решение $\overline{Z}$ задачи F будем называть *оптимальным некооперативным решением*, если для всякого оптимального решения $\widetilde{Z}$ задачи F выполняется неравенство $L(X,\overline{Z}) \leqslant L(X,\widetilde{Z})$. Допустимое решение $(X,\overline{Z})$ задачи (L,F) назовём *допустимым некооперативным решением*, если $\overline{Z}$ — оптимальное некооперативное решение задачи F. Допустимое некооперативное решение $(X^*,\overline{Z}^*)$ задачи (L,F) назовём *оптимальным некооперативным решением*, если для любого допустимого некооперативного решения $(X,\overline{Z})$ задачи (L,F) выполняется неравенство $L(X^*,\overline{Z}^*) \geqslant L(X,\overline{Z})$. При этом величину $L(X^*,\overline{Z}^*)$ будем называть *оптимальным* значением целевой функции задачи (L,F).

Допустимое некооперативное решение $(X,\overline{Z})$ задачи (L,F) будем называть также *приближённым* решением задачи (L,F). Понятно, что приближённое решение $(X,\overline{Z})$ задачи (L,F) можно построить по допустимому решению X задачи L. Соответствующее оптимальное некооперативное решение $\overline{Z}$ задачи F определяется с помощью алгоритма, состоящего из двух этапов.

На этапе 1 при фиксированном решении X решается задача F и вычисляется оптимальное значение $F(\widetilde{Z})$ её целевой функции.

На этапе 2 при фиксированном решении X решается следующая вспомогательная задача:

$$\min_{(z_i),(z_{ij})} \sum_{j\in J} \Big(\sum_{i\in I} p_{ij}x_{ij}\Big)\Big(1-\sum_{i\in I} z_{ij}\Big); \tag{10}$$

$$x_i + z_i \sum_{k|i\succ_j k} z_{kj} \leqslant 1, \quad i\in I,\ j\in J; \tag{11}$$

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

$$-\sum_{i\in I} g_i z_i + \sum_{i\in I}\sum_{j\in J} p_{ij}z_{ij} \geqslant F(\widetilde{Z}); \tag{13}$$

$$z_i, z_{ij} \in \{0,1\}, \quad i\in I,\ j\in J. \tag{14}$$

Оптимальное решение $\overline{Z} = ((\overline{z}_i),(\overline{z}_{ij}))$ этой задачи будет искомым оптимальным решением задачи F.

2. Верхняя граница для оптимального значения целевой функции задачи (L,F)

Для оценки величины близости приближённого и оптимального некооперативного решений задачи (L,F) построим алгоритм вычисления верхней границы для оптимального значения целевой функции задачи (L,F). При построении верхней границы будем дополнительно предполагать, что для всякого $j\in J$ величины p_{ij}, $i\in I$, обладают *свойством невозрастания* относительно порядка $\succ_j$, т. е. для любых $i,k\in I$ таких, что $i \succ_j k$, выполняется неравенство $p_{ij} \geqslant p_{kj}$. Указанным свойством обладают, например, величины p_{ij}, $i \in I$, $j \in J$, определённые следующим образом. Пусть для всякого $j \in J$ указан элемент $i_j \in I$, задающий множество $A_j = \{i \in I \mid i \succcurlyeq_j i_j\}$ «допустимых» предприятий для обслуживания потребителя j. И пусть для всякого $i \in A_j$ доход, получаемый предприятием i при обслуживании потребителя j, не зависит от предприятия и равен величине b_j. В рассматриваемой ситуации для всякого $j \in J$ величины p_{ij}, $i \in I$, определяются формулой

$$p_{ij} = \begin{cases} b_j, & \text{если } i \in A_j, \\ 0 & \text{иначе} \end{cases}$$

и удовлетворяют указанному свойству.

Для всякого $j \in J$ определим множества $I_j \subset I$, с использованием которых будет построена искомая верхняя граница. Для этого при фиксированном $j_0 \in J$ сформулируем условия, позволяющие для всякого $i \in I$ выяснить, будет ли $i \in I_{j_0}$ или $i \notin I_{j_0}$.

Для $i \in I$ рассмотрим множества

$$N(i) = \{k \in I \mid k \succ_{j_0} i\},$$
$$J(i) = \{j \in J \mid i \succcurlyeq_j k \text{ для всякого } k \notin N(i)\}.$$

Заметим, что $J(i) \neq \varnothing$, поскольку $j_0 \in J(i)$.

Если $N(i) = \varnothing$, то считаем, что $i \in I_{j_0}$. Пусть $N(i) \neq \varnothing$. Для всякого $k \in N(i)$ построим множество

$$J(k,i) = \{j \in J(i) \mid k \succ_j i\}.$$

Считаем, что $i \in I_{j_0}$, если для каждого $k \in N(i)$ выполняется неравенство

$$g_k > \sum_{j \in J(k,i)} p_{kj},$$

и $i \notin I_{j_0}$, если найдётся $k \in N(i)$, для которого указанное неравенство нарушается.

Содержательный смысл множества I_j поясняет следующая лемма, устанавливающая, что если Лидер планирует получить доход от потребителя $j \in J$ и при этом не открывает ни одного предприятия из множества I_j, то потребитель j будет «захвачен» Последователем.

При фиксированном множестве $I_0 \subset I$ для всякого $j \in J$ через $i_0(j)$ обозначим такой элемент $i_0 \in I_0$, что $i_0 \succcurlyeq_j i$ для всякого $i \in I_0$. для $(0,1)$-вектора $x = (x_i)$, $i \in I$. Обозначим $I_0(x) = \{i \in I \mid x_i = 1\}$ для

$(0,1)$-вектора $x=(x_i)$, $i\in I$. Если $x=(x_i)$ и $y=(y_i)$ — $(0,1)$-векторы, то через $x\cup y$ будем обозначать $(0,1)$-вектор $z=(z_i)$ такой, что $z_i=\max\{x_i,y_i\}$, $i\in I$.

Лемма 1. *При любом допустимом некооперативном решении* $(X,\overline{Z})$, $X=((x_i),(x_{ij}))$, $\overline{Z}=((\overline{z}_i),(\overline{z}_{ij}))$, *для* $j_0\in J$ *такого, что* $p_{i_0j_0}x_{i_0j_0}>0$ *для некоторого* $i_0\notin I_{j_0}$, *выполняется равенство*

$$\sum_{i\in I}\overline{z}_{ij_0}=1.$$

Доказательство. Для заданных $(0,1)$-векторов $x=(x_i)$ и $\overline{z}=(\overline{z}_i)$ рассмотрим множество $I_0=I_0(x\cup\overline{z})$ и элементы $i_0(j)$, $j\in J$. Предположим, что для некоторого $j_0\in J$ имеем $p_{i_0j_0}x_{i_0j_0}>0$ и $i_0\notin I_{j_0}$, но требуемое равенство не выполняется. Рассмотрим множества $N(i_0)$ и $J(i_0)$ и заметим, что $\overline{z}_i=0$ для всякого $i\in N(i_0)$ и $i_0=i_0(j)$ для всякого $j\in J(i_0)$. Поскольку $i_0\notin I_{j_0}$, найдётся $k\in N(i_0)$, для которого существует множество $J(k,i_0)\subset J(i_0)$ такое, что

$$g_k\leqslant\sum_{j\in J(k,i_0)}p_{kj}.$$

Для данного $k\in N(i_0)$ рассмотрим множества

$$S_1(k)=\{j\notin J(i_0)\mid k\succ_j i_0(j),\, x_{i_0(j)}=1\},$$
$$S_2(k)=\{j\notin J(i_0)\mid k\succ_j i_0(j),\, \overline{z}_{i_0(j)}=1\}$$

и построим допустимое решение $\overline{Z}'=((\overline{z}'_i),(\overline{z}'_{ij}))$ задачи F, которое отличается от исходного допустимого решения $\overline{Z}=((\overline{z}_i),(\overline{z}_{ij}))$ тем, что $\overline{z}'_k=1$, $\overline{z}'_{kj}=1$ для $j\in J(k,i_0)$, $\overline{z}'_{kj}=1$ для $j\in S_1(k)$, $\overline{z}'_{kj}=1,\overline{z}'_{i_0(j)j}=0$ для $j\in S_2(k)$.

Оценим разность значений целевой функции задачи F на решениях $\overline{Z}'$ и $\overline{Z}$:

$$F(\overline{Z}')-F(Z)=-g_k+\sum_{j\in J(k,i_0)}p_{kj}+\sum_{j\in S_1(k)}p_{kj}+\sum_{j\in S_2(k)}(p_{kj}-p_{i_0(j)j}).$$

В силу свойства невозрастания величин p_{ij}, $i\in I$, относительно порядка $\succ_j$ последнее слагаемое неотрицательно и, следовательно, рассматриваемая разность неотрицательна. Отсюда получаем, что $\overline{Z}'$ — оптимальное решение задачи F. Заметим также, что поскольку $j_0\in J(k,i_0)$, то $L(X, \overline{Z}') < L(X, \overline{Z})$. Это противоречит тому, что $(X, \overline{Z})$ — допустимое некооперативное решение. Лемма 1 доказана.

Лемма 2. *При любом допустимом некооперативном решении* $(X, \overline{Z})$, $X = ((x_i), (x_{ij}))$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, *для всякого* $j \in J$ *выполняется неравенство*

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

Доказательство. Если $p_{ij} x_{ij} = 0$ для любого $i \in I$, то неравенство выполняется. Пусть $p_{i_0 j} x_{i_0 j} > 0$ для некоторого $i_0 \in I$. Если $i_0 \in I_j$, то неравенство также выполняется, поскольку

$$\sum_{i \in I} p_{ij} x_{ij} = p_{i_0 j} x_{i_0} \leqslant \max_{i \in I_j} p_{ij} x_i.$$

Если же $i_0 \notin I_j$, то неравенство справедливо, поскольку в силу леммы 1 выполняется равенство $\sum\limits_{i \in I} \overline{z}_{ij} = 1$. Лемма 2 доказана.

Определим матрицу (h_{ij}), $i \in I$, $j \in J$, положив

$$h_{ij} = \begin{cases} 1, & \text{если } i \in I_j, \\ 0 & \text{иначе.} \end{cases}$$

Имеет место следующая

Теорема. *Величина*

$$\max_{(x_i)} \Big\{ -\sum_{i \in I} f_i x_i + \sum_{j \in J} \max_{i | x_i = 1} p_{ij} h_{ij} \Big\}$$

является верхней границей для оптимального значения целевой функции задачи (L, F).

Доказательство. Для всякого допустимого некооперативного решения $(X, \overline{Z})$, $X = ((x_i), (x_{ij}))$, $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$, в силу леммы 2 и справедливости для всякого $j \in J$ равенства

$$\max_{i \in I_j} p_{ij} x_i = \max_{i | x_i = 1} p_{ij} h_{ij}$$

выполняются соотношения

$$L(X, \overline{Z}) \leqslant -\sum_{i \in I} f_i x_i + \sum_{j \in J} \max_{i \in I_j} p_{ij} x_i = -\sum_{i \in I} f_i x_i + \sum_{j \in J} \max_{i | x_i = 1} p_{ij} h_{ij}.$$

Отсюда получаем, что величина

$$\max_{(x_i)}\Big\{-\sum_{i\in I} f_i x_i + \sum_{j\in J} \max_{i|x_i=1} p_{ij} h_{ij}\Big\}$$

будет искомой верхней границей. Теорема доказана.

Из доказанного следует, что вычисление верхней границы сводится к решению классической задачи размещения предприятий на максимум:

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

$$\sum_{i\in I} x_{ij} \leqslant 1, \quad j\in J;$$

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

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

Если $X^* = ((x_i^*), (x_{ij}^*))$ — оптимальное решение этой задачи, то в силу свойства невозрастания величин p_{ij}, $i\in I$, при фиксированном $j\in J$ это решение можно считать допустимым решением задачи L, которое порождает допустимое некооперативное решение $(X^*, \overline{Z})$ задачи (L, F).

В качестве следствия доказанной теоремы укажем на некоторые случаи, когда получаемое при вычислении верхней границы приближённое решение $(X^*, \overline{Z})$ является оптимальным некооперативным решением, а величина верхней границы равна оптимальному значению целевой функции задачи (L, F). Заметим прежде всего, что если X^* — нулевое решение, то верхняя граница равна нулю и, следовательно, $(X^*, \overline{Z})$ — оптимальное некооперативное решение. Заметим также, что если $\overline{Z}$ — нулевое решение, то $(X^*, \overline{Z})$ будет оптимальным некооперативным решением, поскольку значение целевой функции задачи (L, F) на этом решении равно величине верхней границы.

Таким образом, *алгоритм вычисления верхней границы* для оптимального значения целевой функции задачи (L, F) включает два этапа. На первом этапе строится матрица (h_{ij}), $i\in I$, $j\in J$, а на втором — определяется оптимальное решение $((x_i^*), (x_{ij}^*))$ указанной задачи размещения предприятий.

Процедура построения матрицы (h_{ij}) состоит из n шагов. На j-м шаге при фиксированном $j\in J$ выполняется m однотипных подшагов. В начале i-го подшага полагается $h_{ij} = 1$. После этого для множества $I_0 = I\setminus(N(i)$ определяются элементы $i_0(s)$, $s\in J$, и для всякого $k\in N(i)$

вычисляется величина

$$-g_k + \sum_{s | i \succ_s i_0(s),\, k \succ_s i} p_{ks}.$$

Если для некоторого $k \in N(i)$ эта величина неотрицательна, то полагается $h_{ij} = 0$.

Несложно видеть, что временна́я сложность одного подшага алгоритма оценивается величиной $O(mn)$. Поэтому временна́я сложность первого этапа алгоритма выражается величиной $O(m^2n^2)$.

На втором этапе алгоритма для решения задачи размещения предприятий может быть использован целый ряд алгоритмов [1], построенных на идеях локального поиска и неявного перебора, а также коммерческие средства решения задач целочисленного линейного программирования.

3. Алгоритмы построения приближённого решения задачи (L, F)

Выше отмечалось, что в качестве приближённого решения задачи (L, F) можно рассматривать любое допустимое решение X задачи L. Значение целевой функции задачи (L, F) на соответствующем допустимом некооперативном решении $(X, \overline{Z})$ определяется по решению X однозначно. При этом оптимальное некооперативное решение $\overline{Z}$ задачи F строится в результате решения задачи F и вспомогательной задачи (10)–(14). Кроме того, само допустимое решение $X = ((x_i), (x_{ij}))$ задачи L однозначно определяется $(0, 1)$-вектором (x_i). По любому такому вектору однозначно определяется значение $L(X, \overline{Z})$ целевой функции задачи (L, F) на соответствующем допустимом некооперативном решении $(X, \overline{Z})$. В силу этого задачу (L, F) можно рассматривать как задачу максимизации некоторой псевдобулевой функции $f(x)$, $x \in B^m$. Значение этой функции на произвольном $(0, 1)$-векторе x равно значению целевой функции задачи (L, F) на допустимом некооперативном решении $(X, \overline{Z})$, построенном по заданному $(0, 1)$-вектору x.

Поскольку одновременно с вычислением верхней границы для оптимального значения целевой функции задачи (L, F) определяется допустимое решение $X^* = ((x_i^*), (x_{ij}^*))$ задачи L, то $(0, 1)$-вектор (x_i^*) можно рассматривать как начальное приближённое решение задачи максимизации псевдобулевой функции $f(x)$.

Рассмотрим алгоритмы улучшения начального приближённого решения, построенные на основе стандартной процедуры локального поиска [1, 9]. Результатом работы таких алгоритмов является локально-оптимальное решение относительно заданной окрестности. В случае задачи оптимизации псевдобулевой функции $f(x)$, $x \in B^m$, в качестве окрестности обычно используются следующие множества:

$$N_1(x) = \{y \in B^m \mid d(x, y) = 1\},$$
$$N_2(x) = \{y \in B^m \mid d(x, y) = 2,\, d(0, x) = d(0, y)\},$$
$$N_3(x) = N_1(x) \cup N_2(x),$$

где $d(x, y)$ — расстояние Хэмминга, равное числу несовпадающих компонент $(0, 1)$-векторов x и y. Элементы окрестности $N_1(x)$ получаются из вектора x изменением значения одной компоненты, а окрестности $N_2(x)$ — двух компонент, сумма которых равна единице.

В нашем случае локальный поиск по «широкой» окрестности $N_3(x)$ может оказаться достаточно трудоёмкой процедурой хотя бы потому, что для любого элемента окрестности вычисление значения функции $f(x)$ предполагает решение двух задач целочисленного линейного программирования. Поэтому определим для $(0, 1)$-вектора x окрестность $N_0(x) \subset N_3(x)$, которая включает относительно небольшое число *существенных* вариантов изменения текущего решения x, и построим алгоритмы локального поиска, использующие окрестность $N_0(x)$. Пусть задано текущее решение $x = (x_i)$. В качестве основного показателя, с использованием которого будут строиться существенные вариации текущего решения x, примем величину *доходности* предприятия $k \in I$, открываемого Лидером.

Пусть $k \in I_0(x)$. Для множества $I_0 = I_0(x)$ определим элементы $i_0(j)$, $j \in J$, и рассмотрим величину

$$-g_k + \sum_{i \mid i_0(j) = k} p_{kj}.$$

Эту величину обозначим через $\Delta_k(x)$ и будем называть *доходностью* предприятия k относительно решения x.

Используя понятие доходности, выделим множество $N_0(x) \subset N_3(x)$ существенных вариаций текущего решения $x = (x_i)$. Окрестность $N_0(x)$ считаем состоящей из $(0, 1)$-векторов $x^k = \big(x_i^k\big)$, $k \in I$, которые строятся следующим образом:

если $x_k = 1$, то полагается $x_i^k = x_i$ для $i \neq k$ и $x_k^k = 0$;

если $x_k = 0$, то полагается $x_i^k = x_i$ для $i \neq k$ и $x_k^k = 1$.

Далее вычисляется величина $\Delta_k(x^k)$. Возможны два случая: $\Delta_k(x^k) \geqslant 0$ и $\Delta_k(x^k) < 0$. В первом случае, когда доходность предприятия k неотрицательная, для всякого $i \in I_0(x)$ вычисляется величина $\Delta_i(x^k)$. Если

$\Delta_i(x^k) \geqslant 0$ для всякого $i \in I_0(x)$, т. е. открытие «нового» предприятия k не приводит к тому, что некоторые «старые» предприятия, открытые Лидером, становятся убыточными, то вектор x^k считается построенным. Если же $\Delta_i(x^k) < 0$ для некоторых $i \in I_0(x)$, то определяется элемент $i_0 \in I_0(x)$ такой, что $\Delta_{i_0}(x^k) \leqslant \Delta_i(x^k)$ для всякого $i \in I_0(x)$, полагается $x^k_{i_0} = 0$, и вектор x^k считается построенным.

Если $\Delta_k(x^k) < 0$, т. е. доходность «нового» предприятия отрицательная, то среди «старых» предприятий отыскивается такое, удаление которого максимально увеличивает доходность «нового» предприятия k. Для этого при любом $l \in I_0(x)$ строится $(0,1)$-вектор $x^{kl} = \big(x^{kl}_i\big)$, где $x^{kl}_i = x^k_i$ при $i \neq l$ и $x^{kl}_l = 0$, и вычисляется величина $\Delta_k(x^{kl})$. Далее определяется номер $l_0 \in I_0(x)$ такой, что $\Delta_k(x^{kl_0}) \geqslant \Delta_k(x^{kl})$ для всякого $l \in I_0(x)$. Если $\Delta_k(x^{kl_0}) \geqslant 0$, то полагается $x^k_{l_0} = 0$. После этого вектор x^k считается построенным.

Алгоритм улучшения исходного приближённого решения $x^* = (x^*_i)$, полученного в результате вычисления верхней границы, представляет собой процедуру локального поиска по окрестности $N_0(x)$, начиная с вектора x^* до тех пор, пока не будет найден локальный максимум функции $f(x)$ относительно окрестности $N_0(x)$. Алгоритм состоит из предварительного шага и некоторого числа однотипных основных шагов.

На предварительном шаге имеется начальное приближённое решение $x = x^*$. Шаг состоит в вычислении значения L функции $f(x)$ на решении x.

На каждом основном шаге имеется решение x и значение L функции $f(x)$ на решении x. Шаг состоит в последовательном построении для всякого $k \in I$ векторов x^k и вычислении значения L^k функции $f(x)$ на решении x^k. Одновременно определяется решение x^{k_0}, $L^{k_0} > L$, *улучшающее* текущее решение x. Если такого решения найти не удаётся, то алгоритм заканчивает работу и x есть искомое приближённое решение. В противном случае полагается $L = L^{k_0}$, $x = x^{k_0}$ и начинается следующий шаг.

В представленной общей схеме алгоритма правило выбора улучшающего решения x^{k_0} в окрестности $N_0(x)$ требует уточнения. Рассматриваются четыре способа выбора элемента $k_0 \in I$. При первом способе просматриваются все элементы $k \in I$ и выбирается такой номер $k_0 \in I$, что $L^{k_0} > L$ и $L^{k_0} \geqslant L^k$ для всякого $k \in I$. Три других способа заключаются в просмотре элементов множества I в некотором заданном порядке и выборе в качестве номера k_0 первого элемента $k \in I$, для которого $L^k > L$. Во втором способе просмотр элементов множества I проводится в порядке возрастания номера k, а в третьем — элементы $k \in I$ упорядочиваются по убыванию числа вхождений элемента $k \in I$ в множества I_j, $j \in J$, определяемые при вычислении верхней границы. В четвёртом способе порядок просмотра элементов множества I задаётся случайным образом и при этом выбор некоторых номеров $k \in I$ запрещается. Список запретов формируется из элементов $k \in I$, выбранных на предыдущих шагах, и включает не более $m/3$ элементов.

В заключение приведём результаты вычислительного эксперимента с предложенными алгоритмами построения приближённых решений задачи (L, F), отличающихся способами выбора «наилучших» элементов окрестности. Алгоритмам присвоены номера 1, 2, 3 и 4 в соответствии с номерами используемых способов выбора улучшающего решения. Вычисления проводились для задачи (L, F) на сети [2], т. е. для задачи, где, во-первых, множество возможных мест размещения предприятий I и множество потребителей J совпадают с множеством вершин некоторого графа и отношение порядка для каждого потребителя определяется длинами кратчайших путей из соответствующей вершины во все другие вершины графа.

Исходные данные рассматриваемых примеров задачи (L, F) на сети формируются случайным образом по следующим правилам. В единичном квадрате случайным образом размещается n точек, образующих вершины графа. Расстояние r_{ij} между вершинами i и j вычисляется как расстояние между точками на плоскости. Наличие ребра между вершинами i и j определяется случайным образом посредством задания величины «вероятности» возникновения ребра. Эти величины задавались так, чтобы число рёбер в графе находилось в промежутке от $1,5n$ до $2n$.

Величины p_{ij}, $i \in I$, $j \in J$, вычисляются по формуле

$$p_{ij} = \begin{cases} b_j, & \text{если } d_{ij} \leqslant d_j, \\ 0 & \text{в противном случае,} \end{cases}$$

где d_{ij} — длина кратчайшего пути из вершины i в вершину j, d_j — параметр, b_j — случайная величина с равномерным распределением, принимающая целые значения из промежутка от b_0 до b^0. В рассматриваемых примерах $d_j = 0,7$, $b_0 = 10$, $b^0 = 20$.

Величины f_i, $i \in I$, равны 40, а каждая из g_i, $i \in I$, — случайная величина с равномерным распределением, принимающая целые значения от 25 до 35.

Вычисления проводились для серий из 40 задач при фиксированных значениях $n = 20, 30, 40, 50$. В приводимой ниже таблице для четырёх рассмотренных алгоритмов даются средние значения по серии задач следующих показателей:

UB — величина верхней границы;

VL — значение целевой функции задачи Лидера на приближённом решении;

$|x|$ — число открываемых Лидером предприятий, определяемое приближённым решением;

$|z|$ — число открываемых Последователем предприятий, определяемое приближённым решением;

N — число шагов алгоритма для получения приближённого решения;

ЦП — число решаемых задач линейно-целочисленного программирования;

UB/VL — оценка относительной точности полученного приближённого решения, равная отношению величин UB и VL.

Кроме того, в таблице для значений $n = 20, 30$ приводятся средние значения следующих величин:

VL^* — оптимальное значение целевой функции задачи Лидера, полученное с использованием процедур неявного перебора;

UB/VL^* — относительная точность вычисленной верхней границы, равная отношению величин UB и VL^*;

VL^*/VL — относительная точность полученного приближённого решения, равная отношению величин VL^* и VL.

Из таблицы видно, что средняя относительная точность приближённых решений примерно одинакова для всех рассматриваемых алгоритмов. Наиболее предпочтительным в этом плане является первый алгоритм, а наименее предпочтительным — второй. Однако преимущества первого алгоритма не являются абсолютными, поскольку для некоторых примеров лучшее решение даёт третий алгоритм.

Среднее значение верхней границы, как следует из таблицы, почти в два раза превосходит оптимальное значение целевой функции. Такая точность не является удовлетворительной для получения априорных оценок точности вычисляемых приближённых решений. Вместе с тем, начальное приближённое решение, получаемое одновременно с вычислением верхней границы, является хорошей основой для построения локального максимума целевой функции. Для этого, как следует из таблицы, в среднем достаточно 3–7 шагов. Получаемое первое локально-оптимальное решение во многих примерах либо является оптимальным решением, либо близко к нему по значению целевой функции. Поэтому точность приближённых решений, получаемых с использованием рассмотренных алгоритмов, можно считать вполне удовлетворительной. Следует отметить, что поиск «хороших» приближённых решений определённо не должен завершаться отысканием первого локально-оптимального решения, и, следовательно, у рассмотренных алгоритмов имеются реальные резервы для их результативной модификации.

Т а б л и ц а

n	Номер алг-ма	UB	VL	$\|x\|$	$\|z\|$	N	ЦП	$\frac{UB}{VL}$	VL^*	$\frac{UB}{VL^*}$	$\frac{VL^*}{VL}$
20	1	84,73	37,50	3,18	1,85	2,80	116,00	2,26	42,55	1,99	1,13
	2	84,73	36,20	3,03	1,78	3,80	85,05	2,34	42,55	1,99	1,18
	3	84,73	37,25	3,13	1,85	3,40	68,40	2,27	42,55	1,99	1,14
	4	84,73	36,78	3,20	1,85	3,53	132,00	2,30	42,55	1,99	1,16
30	1	132,45	58,68	4,80	2,60	3,57	218,50	2,26	67,90	1,95	1,16
	2	132,45	54,32	4,75	2,68	4,35	135,05	2,44	67,90	1,95	1,25
	3	132,45	55,87	4,75	2,60	4,35	109,85	2,37	67,90	1,95	1,22
	4	132,45	54,72	4,90	2,60	4,47	196,00	2,42	67,90	1,95	1,24
40	1	198,05	87,88	7,05	3,40	4,70	380,00	2,25			
	2	198,05	84,40	6,80	3,40	6,15	225,90	2,35			
	3	198,05	85,70	6,65	3,58	6,10	177,95	2,31			
	4	198,05	82,28	7,05	3,65	6,00	260,00	2,41			
50	1	240,20	104,13	7,90	4,08	4,82	486,50	2,31			
	2	240,20	101,53	7,80	3,98	7,15	315,65	2,37			
	3	240,20	101,03	7,43	4,04	7,05	249,10	2,38			
	4	240,20	100,43	7,78	4,08	6,95	324,00	2,39			

В ходе рассмотренного вычислительного эксперимента было также проведено сравнение используемой в предложенных алгоритмах окрестности $N_0(x)$ и более «широкой» окрестности $N_3(x)$. Замечено, что в более чем 85 % рассмотренных примеров найденное локально-оптимальное решение относительно окрестности $N_0(x)$ является также локально-оптимальным решением относительно окрестности $N_3(x)$. Это означает, что несмотря на то, что множество $N_3(x)$ содержит существенно больше элементов, чем множество $N_0(x)$, в большинстве случаев элемент окрестности $N_3(x)$, на котором значение целевой функции наибольшее, содержится также и в окрестности $N_0(x)$.

ЛИТЕРАТУРА

1. **Береснев В. Л.** Дискретные задачи размещения и полиномы от булевых переменных. — Новосибирск: Изд-во Ин-та математики, 2005. — 408 с.
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.
3. **Campos C., Moreno J. A.** Multiple voting location problems // Europ. J. Oper. Res. — 2008. — Vol. 191, N 2. — P. 436–452.
4. **Dempe S.** Foundations of bilevel programming. — Dordrecht: Kluwer Acad. Publ., 2002. — 332 p.
5. **Eiselt H. A., Laporte G.** Sequential location problems // Europ. J. Oper. Res. — 1996. — Vol. 96. — P. 217–231.
6. **Discrete location theory** / Eds. Mirchandani P. B., Francis R. L. — New York: John Wiley and sons, 1990. — 555 p.
7. **Dobson G., Karmarkar U.** Competitive location on network // Oper. Res. — 1987. — Vol. 35. — P. 565–574.
8. **Facility location: applications and theory** / Eds. Drezner Z., Hamacher H. W. — Berlin: Springer-Verl., 2002. — 457 p.
9. **Local search in combinatorial optimization** / Eds. Aarts E. H. L., Lensrta J. K. — Chichester: John Wiley and sons, 1997. — 512 p.
10. **Plastria F.** Static competitive facility location: an overview of optimization approaches // Europ. J. Oper. Res. — 2001. — Vol. 129. — P. 461–470.
11. **Plastria F., Vanhaverbeke L.** Discrete models for competitive location with foresight // Comput. Oper. Res. — 2008. — Vol. 35. — P. 683–700.
12. **Santos-Penãte D. R., Suárez-Vega R., Dorta-Gonzáles P.** The leader–follower location model // NETS. — 2007. — Vol. 7. — P. 45–61.
13. **Von 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: a.a.melnikov@hotmail.com

Статья поступила
10 мая 2010 г.