ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Март—апрель 2009. Том 16, № 2. С. 21–41

УДК 519.854.2

МЕТОД ВЕТВЕЙ И ОТСЕЧЕНИЙ ДЛЯ ЗАДАЧИ РАЗМЕЩЕНИЯ С ПРЕДПОЧТЕНИЯМИ КЛИЕНТОВ*)

И. Л. Васильев, К. Б. Климентова

Аннотация. Проводится численное исследование методов решения задачи размещения производства, в которой клиенты выбирают поставщиков исходя из собственных предпочтений. Рассматриваются различные формулировки данной задачи в виде задач целочисленного линейного программирования. Для предложенного ранее семейства правильных неравенств, возникающего из связи с задачей о паре матриц, реализован метод отсечений. Проведён вычислительный эксперимент по его тестированию. Поиск точного решения задач осуществляется двумя вариантами метода ветвей и отсечений с использованием указанного выше метода отсечений. Для поиска верхних оценок оптимального решения в точных методах предложено использовать метод имитации отжига. Вычислительный эксперимент подтверждает эффективность реализованного подхода по сравнению с известными.

Ключевые слова: задача размещения с предпочтениями клиентов, метод отсечений, локальный поиск, метод ветвей и отсечений.

Введение

Задача размещения с предпочтениями клиентов представляет собой специальный случай задач размещения с двумя уровнями принятия решений [2, 4]. На верхнем уровне поставщиком выбирается подмножество открываемых предприятий. Затем на нижнем уровне происходит прикрепление клиентов к этим предприятиям согласно известным предпочтениям клиентов. Ставится задача на верхнем уровне выбрать открываемые предприятия так, чтобы обслужить всех клиентов с минимальными суммарными затратами. Задачу размещения с предпочтениями клиентов также называют *задачей размещения с упорядочением* [6].

Впервые такая задача размещения рассматривалась в [8]. Позже аналогичные модели независимо были предложены в [4, 5]. Отметим, что

*)Исследование первого автора выполнено при финансовой поддержке Фонда содействия отечественной науке и Фонда «Научный потенциал».

в случае, когда предпочтения клиентов на нижнем уровне согласуются с матрицей транспортных затрат на верхнем уровне, имеем классическую задачу размещения [2]. Следовательно, данная двухуровневая задача размещения с предпочтениями клиентов является NP-трудной. В [4, 5] установлена её тесная связь с псевдобулевыми функциями.

В [1] исследуется задача с фиксированным числом открываемых предприятий. Для поиска приближённого решения такой задачи в работе предложен генетический алгоритм, тестирование которого проводилось авторами на примерах с большим разрывом целочисленности.

Для поиска точного решения в задаче размещения с предпочтениями клиентов обычно используются сведения к задачам целочисленного линейного программирования (ЦЛП). В [6] рассматриваются известные и предлагается ряд новых правильных неравенств. С их помощью удаётся улучшить нижние оценки оптимального значения и повысить эффективность метода ветвей и границ. В [1, 9] также рассмотрены различные формулировки задачи в терминах ЦЛП, предложена формулировка, основанная на взаимосвязи задачи размещения с предпочтениями клиентов и задачи о паре матриц. С помощью такой формулировки удаётся получить лучшую нижнюю оценку за счёт увеличения числа переменных.

В работе [3] предложена новая формулировка, основанная на анализе упомянутой выше формулировки из [1, 9]. Она обеспечивает нижнюю оценку не хуже, чем в [9], но улучшение достигается не за счёт увеличения числа переменных, а с помощью нового семейства правильных неравенств.

Целью данной работы является проверка эффективности семейства неравенств, предложенного в [3]. Для поиска соответствующей нижней оценки был реализован метод отсечений. Проведён вычислительный эксперимент на серии тестовых примеров. Затем реализованный метод отсечений использовался при поиске точного решения двумя вариантами метода ветвей и границ. Кроме того, для поиска верхних оценок в указанных точных методах использовался метод имитации отжига. Проведённый вычислительный эксперимент подтвердил эффективность использования предложенного семейства неравенств и реализованной эвристики.

Статья организована следующим образом. В разд. 1 изложена постановка задачи. В разд. 2 представлен обзор известных формулировок ЦЛП, описана общая схема метода отсечений для семейства правильных неравенств из [3] и изложены особенности его реализации. В разд. 3

описан алгоритм имитации отжига. Результаты вычислительного эксперимента по тестированию метода отсечений, метода имитации отжига и поиску оптимального решения задач представлены в разд. 4.

1. Постановка задачи

Данный раздел посвящён постановке задачи размещения с предпочтениями клиентов. Будет рассмотрена комбинаторная постановка этой задачи, а затем эквивалентная одноуровневая формулировка в виде задачи целочисленного линейного программирования (ЦЛП).

Пусть заданы следующие множества и величины:

$I = \{1, \ldots, m\}$ — множество предприятий;

$J = \{1, \ldots, n\}$ — множество клиентов;

$f = \{f_i\},\ f_i \geqslant 0,\ i \in I,$ — затраты на открытие предприятия i;

$C = \{c_{ij}\},\ c_{ij} \geqslant 0,\ i \in I,\ j \in J,$ — матрица производственно-транспортных затрат на обслуживание клиентов;

$G = \{g_{ij}\},\ g_{ij} \geqslant 0,\ i \in I,\ j \in J,$ — матрица предпочтений клиентов, если $g_{i_1 j} < g_{i_2 j}$, то j-й клиент из открытых предприятий i_1, i_2 выберет предприятие i_1.

Ставится задача открыть некоторое подмножество предприятий $S \subseteq I$, минимизируя затраты на обслуживание клиентов и открытие предприятий, учитывая при этом предпочтения клиентов.

Математическая модель задачи размещения с предпочтениями клиентов [1, 4] в виде задачи комбинаторной оптимизации может быть записана следующим образом:

$$\min_{S \subseteq I} \Big\{ \sum_{j \in J} c_{s^j j} + \sum_{i \in S} f_i \Big\}, \tag{1}$$

$$s^j \in I(j, S) \overset{\triangle}{=} \operatorname*{Argmin}_{k \in S} g_{kj}, \quad j \in J. \tag{2}$$

Целевая функция (1) минимизирует затраты поставщиков на открытие предприятий и обслуживание клиентов, в то время как элементы s^j выбираются из множеств наиболее предпочтительных для клиентов $j \in J$ предприятий.

Сформулированная таким образом задача имеет естественную двухуровневую структуру. Действительно, на верхнем уровне (1) поставщиком осуществляется выбор некоторого подмножества S открываемых предприятий, а затем на нижнем уровне (2) клиенты выбирают из открытых предприятий наиболее предпочтительные.

Отметим, что понятие решения в задаче (1)–(2) требует уточнения, поскольку на нижнем уровне клиенты могут осуществлять выбор неединственным образом. В двухуровневом программировании принято рассматривать понятия оптимистического и пессимистического решений [7]. Известно [3], что для рассматриваемой двухуровневой задачи размещения задача поиска оптимистического и пессимистического решений сводится к случаю, когда для любого выбора $S \subseteq I$ существует единственное решение задачи нижнего уровня, т. е. когда

$$I(j,S) \stackrel{\triangle}{=} \operatorname*{Argmin}_{k\in S} g_{kj} = \{s^j\}, \quad j \in J.$$

В этом случае понятие оптимистического и пессимистического решений совпадают. Необходимым и достаточным условием существования единственного решения на нижнем уровне при любом выборе S на верхнем уровне является условие, что все элементы в столбцах матрицы G различны, т. е. $\forall j \in J\ g_{ij} \neq g_{kj}\ \forall i \neq k,\ i,k \in I$. Другими словами, каждый клиент для любой пары предприятий может сказать, какое из них для него предпочтительнее.

Известно [1, 4, 8], что для рассмотренной двухуровневой задачи комбинаторной оптимизации (1)–(2) с условием единственности оптимального выбора клиентов можно предложить формулировку в виде задачи ЦЛП. С этой целью введём бинарные переменные:

$$y_i = \begin{cases} 1, & \text{если открывается } i\text{-е предприятие;} \\ 0 & \text{в противном случае;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{если } j\text{-й клиент обслуживается из } i\text{-го предприятия;} \\ 0 & \text{в противном случае.} \end{cases}$$

Определим множества: $S_{ij} = \{k \in I \mid g_{kj} < g_{ij}\}$ — это множество предприятий, которые для клиента j лучше, чем предприятие i; $T_{ij} = \{k \in I \mid g_{kj} > g_{ij}\}$ — это множество предприятий, которые для клиента j хуже, чем предприятие i, $i \in I, j \in J$.

Тогда задачу (1)–(2) можно переписать в виде следующей целочисленной линейной задачи [4, 8, 9]

$$\min \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i, \tag{3}$$

$$y_i + \sum_{k\in T_{ij}} x_{kj} \leqslant 1, \quad i \in I,\ j \in J, \tag{4}$$

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J, \tag{5}$$

$$0 \leqslant x_{ij} \leqslant y_i, \quad i \in I,\ j \in J, \tag{6}$$

$$x_{ij}, y_i \in \{0, 1\}, \quad i \in I,\ j \in J. \tag{7}$$

Целевая функция (3) задаёт суммарные затраты поставщиков на обслуживание клиентов и открытие предприятий. Ограничения (5) обеспечивают обслуживание каждого клиента в точности одним предприятием. Неравенства (6) позволяют обслуживать клиентов только из открытых предприятий. Ключевым моментом здесь является появление ограничений (4), благодаря которым удаётся избавиться от двухуровневой структуры задачи. Эти ограничения гарантируют обслуживание клиентов из наиболее предпочтительных для них предприятий. Действительно, если предприятие i открыто, то клиент j не будет обслуживаться из менее предпочтительных предприятий, т. е. предприятий множества T_{ij}.

2. Правильные неравенства и метод отсечений

Как известно [12], во многих случаях для задачи ЦЛП можно предложить несколько эквивалентных формулировок. Качество формулировки принято оценивать разрывом целочисленности: gap = (Opt − LP)/Opt, где Opt — оптимальное значение, LP — значение линейной релаксации. Чем меньше величина gap, тем сильнее формулировка. Идеальным случаем является формулировка, в точности описывающая выпуклую оболочку допустимых целочисленных точек задачи. Однако построение такой формулировки с точки зрения сложности оказывается, как правило, эквивалентно решению исходной задачи. Во многих случаях получение такой формулировки на практике невыполнимо.

Существуют различные способы построения новых формулировок задачи ЦЛП. Одним из таких способов является конструирование правильных неравенств.

Определение 1. Пусть U — множество точек в $\mathbb{R}^n$. Неравенство $a^T u \leqslant b$ называют *правильным* для U, если $a^T u \leqslant b$ для всех $u \in U$.

Обозначим через P многогранник задачи (3)–(7), т. е. выпуклую оболочку целочисленных точек, удовлетворяющих ограничениям (4)–(7); через LB_1 — оптимальное значение задачи линейного программирования (ЛП) (3)–(6).

Для многогранника P известен ряд правильных неравенств. Они порождают различные формулировки, отличающиеся числом ограничений

и, как следствие, разрывом целочисленности. В частности, в [6] были предложены следующие семейства правильных неравенств.

1. Усиление неравенств (4). Пусть $j_1, j_2 \in J,\ i \in I$. Неравенства

$$C2(i, j_1, j_2): \quad \sum_{k \in T_{ij_1}} x_{kj_1} + \sum_{k \in T_{ij_2} \cap S_{ij_1}} x_{kj_2} + y_i \leqslant 1 \tag{8}$$

являются правильными для многогранника P.

2. Обобщение неравенств (8). Пусть $j_1, \ldots, j_s \in J$ и $i \in I$. Неравенства

$$Cs(i, j_1, \ldots, j_s): \sum_{k \in T_{ij_1}} x_{kj_1} + \sum_{t=2}^{s} \sum_{k \in T_{ij_t} \cap \left(\bigcap\limits_{q=1}^{t-1} S_{ij_q} \right)} x_{kj_t} + y_i \leqslant 1, \tag{9}$$

являются правильными для P. Данные неравенства порождают экспоненциальное число дополнительных ограничений. Некоторые из них могут доминировать другие. Потому целесообразно использовать только часть этих неравенств. Авторами [6] предлагается выбирать такие элементы $j_1, \ldots j_s \in J,\ i \in I$, для которых множества $T_{ij_t},\ t = 1, \ldots, s$, попарно не пересекаются.

3. Неравенства, доминирующие (6). Пусть $j_1, j_2 \in J$ и $i \in I$. Тогда

$$x_{ij_1} \leqslant x_{ij_2}, \text{ если } \ S_{ij_2} \subseteq S_{ij_1}. \tag{10}$$

При $S_{ij_1} = S_{ij_2}$ получаем $x_{ij_1} = x_{ij_2}$.

Обозначим через LB_2 нижнюю оценку, представленную в [6], т. е. оптимальное значение в задаче линейного программирования (3), (5), (6), (10) с предложенным в [6] подмножеством неравенств (9). Очевидно, что $LB_1 \leqslant LB_2$.

В [3] предложено ещё одно семейство неравенств. Остановимся на нём более подробно. Построение этого семейства основано на анализе целочисленной формулировки, предложенной в [1, 9]. Эта целочисленная формулировка возникает из взаимосвязи задачи размещения с предпочтениями клиентов и задачи о паре матриц.

Для начала рассмотрим постановку задачи о паре матриц. Пусть заданы матрицы $A = (a_{ij}),\ i \in I,\ j \in J_1$, и $B = (b_{ij}),\ i \in I,\ j \in J_2$, с одинаковым числом строк. Задача о паре матриц [2] состоит в нахождении непустого множества $S^* \subseteq I$ такого, что

$$R(S^*) = \min_{S \subseteq I} R(S) \stackrel{\triangle}{=} \min_{S \subseteq I} \Big\{ \sum_{j \in J_1} \max_{i \in S} a_{ij} + \sum_{j \in J_2} \min_{i \in S} b_{ij} \Big\}. \tag{11}$$

В [4, 5] предложено сведение задачи (1)–(2) к задаче о паре матриц. На основе этого сведения в [1, 9] получена новая формулировка исходной двухуровневой задачи размещения в виде задачи ЦЛП.

Итак, для того чтобы свести задачу (1)–(2) к задаче о паре матриц, необходимо представить матрицу (c_{ij}) в виде суммы двух матриц $c_{ij} = a_{ij} + b_{ij}$. С этой целью для каждого $j \in J$ найдём по матрице (g_{ij}) перестановку $\tau(j) = (i_1, \ldots, i_m)$ элементов множества I такую, что

$$g_{i_1 j} \leqslant g_{i_2 j} \leqslant \ldots \leqslant g_{i_m j}.$$

Положим:

$$\left.\begin{aligned}
&a_{i_1 j} = 0, \quad b_{i_1 j} = c_{i_1 j},\\
&a_{i_k j} = \sum_{l=1}^{k-1} \min\{0, c_{i_{l+1} j} - c_{i_l j}\}, \quad k = 2, \ldots, m,\\
&b_{i_k j} = c_{i_1 j} + \sum_{l=1}^{k-1} \max\{0, c_{i_{l+1} j} - c_{i_l j}\}, \quad k = 2, \ldots, m.
\end{aligned}\right\} \tag{12}$$

Введём обозначения. Пусть $\Delta_l^j = \min\{0; c_{i_{l+1} j} - c_{i_l j}\},\ l = 1, \ldots, m-1$, и $L_j = \{l \in \{1, \ldots, m-1\} \mid \Delta_l^j < 0\}$. Заметим, что при заданном $j \in J$ по номеру $l \in L_j$ однозначно восстанавливается номер $i_l \in I$.

В [1, 9] предложена формулировка задачи о паре матриц (11) в виде задачи ЦЛП, где матрицы A и B выбраны по правилу (12). Для её представления введём дополнительные переменные $v_l^j \in \{0, 1\},\ l \in L_j,\ j \in J$. Получим следующую формулировку [1]:

$$\min\Big\{\sum_{j\in J}\sum_{l\in L_j} -\Delta_l^j v_l^j + \sum_{i\in I}\sum_{j\in J} b_{ij}x_{ij} + \sum_{i\in I} f_i y_i\Big\} + \sum_{j\in J}\sum_{l\in L_j}\Delta_l^j, \tag{13}$$

$$y_i + \sum_{k\in T_{ij}} x_{kj} \leqslant 1,\ i \in I,\ j \in J, \tag{14}$$

$$\sum_{i\in I} x_{ij} = 1,\ j \in J, \tag{15}$$

$$0 \leqslant x_{ij} \leqslant y_i,\ i \in I,\ j \in J, \tag{16}$$

$$v_l^{j_1} \geqslant \sum_{i\notin T_{i_l j_1}} x_{ij_2},\ l \in L_j,\ j_1, j_2 \in J, \tag{17}$$

$$v_l^j, y_i, x_{ij} \in \{0, 1\},\ l \in L_j,\ j \in J,\ i \in I. \tag{18}$$

Обозначим через LB' оптимальное значение в задаче линейного программирования (13)–(17). Можно показать [1], что $LB' \geqslant LB_1$.

Фактически построенная формулировка задачи размещения с предпочтениями клиентов (13)–(18) представляет собой другой путь получения нижних оценок — конструирование расширенных формулировок. Действительно, полученная формулировка является задачей в более широком пространстве переменных $(x, y, v) \in \mathbb{B}^{m \cdot n} \times \mathbb{B}^m \times \mathbb{B}^{|L_1|+\ldots+|L_n|}$. Исходная формулировка определялась в пространстве переменных $(x, y) \in \mathbb{B}^{m \cdot n} \times \mathbb{B}^m$. Очевидным недостатком расширенных формулировок является увеличение числа переменных, в то время как попытки усиления формулировки в исходном пространстве, как правило, приводят к большому числу дополнительных ограничений, например, неравенствам (9). Одним из путей преодоления чрезмерного увеличения размерности расширенных формулировок может быть конструирование на их основе новых правильных неравенств и соответствующих алгоритмов отделения [12, 13]. Так, на основе анализа формулировки (13)–(18) в [3] было предложено следующее семейство неравенств.

Теорема. *Неравенства*

$$\sum_{i \in T_{i_l j_1}} x_{ij_1} + \sum_{i \notin T_{i_l j_1}} x_{ij_2} \leqslant 1, \quad l \in L_j,\ i \in I,\ j_1, j_2 \in J,\ \ j_1 \neq j_2, \tag{19}$$

являются правильными для P.

Обозначим через LB_3 нижнюю оценку, получаемую при решении линейной релаксации улучшенной формулировки из [6] с добавленными в неё неравенствами (19), т. е. оптимальное значение задачи ЛП (3), (5), (6), (10), (19) с предложенным в [6] подмножеством неравенств (9).

Непосредственным следствием из теоремы является неравенство $LB' \leqslant LB_3$, т. е. новая нижняя оценка оказывается не хуже известной оценки LB', при этом для её получения необходимо решать задачу линейного программирования в пространстве меньшей размерности.

На практике неравенств в семействе (19) может оказаться чрезвычайно много, причём некоторые из них могут быть неактивны и бесполезны. Это приводит к неоправданному увеличению количества ограничений в формулировке. Для преодоления чрезмерного разрастания формулировки используется метод отсечений. Прежде чем переходить к описанию особенностей реализации метода отсечений для семейства неравенств (19), рассмотрим несколько базовых определений.

Определение 2. Пусть заданы некоторое множество X и точка $\bar{x} \notin X$. Правильное для X неравенство $a^T x \leqslant b$ называется *отсекающей плоскостью* (*отсечением*), если $a^T \bar{x} > b$.

Определение 3. Пусть заданы некоторое множество X и семейство $\mathcal{C}$ правильных для X неравенств. *Задача отделения* для некоторой точки $\bar{x}$ заключается в том, чтобы

либо доказать, что $\bar{x}$ удовлетворяет всем неравенствам семейства $\mathcal{C}$;

либо определить отсекающие плоскости $(a^T x \leqslant b) \in \mathcal{C}$.

Определение 4. Алгоритм, с помощью которого решается задача отделения, называется *алгоритмом отделения.*

Пусть задано некоторое семейство $\mathcal{C}$ правильных для X неравенств, где X — это множество допустимых точек некоторой задачи ЦЛП. Идея метода отсечений заключается в том, чтобы не добавлять сразу все правильные неравенства семейства $\mathcal{C}$ в формулировку (поскольку большинство из них будет неактивно и бесполезно), а включать их последовательно по мере необходимости. Сначала рассматривается ЛП-релаксация на исходном множестве X и находится её решение. Затем в формулировку добавляются только те неравенства из семейства $\mathcal{C}$, которые нарушаются полученным решением ЛП-релаксации. И далее вновь находится решение ЛП-релаксации уже на новом допустимом множестве. Этот процесс продолжается, пока решение релаксированной задачи не будет удовлетворять всем неравенствам рассматриваемого семейства.

При проведении предварительного вычислительного эксперимента по тестированию метода отсечений для неравенств (19) оказалось, что за счёт использования метода отсечений удаётся существенно сократить количество добавляемых в формулировку неравенств, однако их число остаётся чрезвычайно большим. Поэтому были применены два приёма, позволившие дополнительно сократить число ограничений в формулировке. Таким образом были реализованы следующие три варианта метода отсечения.

1. На каждой итерации метода добавлялись все нарушенные текущим дробным решением неравенства (классическая схема метода отсечений).

2. Среди всех нарушенных неравенств выявлялись наиболее нарушенные, а именно: пусть $a^T x \leqslant 1$ — некоторое нарушенное неравенство, тогда расстояние от дробного решения $\bar{x}$ до этого неравенства вычисляется по формуле $r(\bar{x}, a) = (a^T \bar{x} - 1)/\|a\|$. Во втором варианте метода отсечений добавлялось M неравенств, расположенных дальше всего от рассматриваемого дробного решения, где M — это параметр, определяющий максимальное количество правильных неравенств, которое может быть добавлено на каждой итерации метода отсечений.

3. Выявлялись нарушенные неравенства, соответствующие гиперплоскости которых оказываются почти параллельными. Следовательно, мож-

но добавить лишь одно из таких неравенств. Рассмотрим два неравенства $a^T x \leqslant 1$ и $b^T x \leqslant 1$. Косинус угла между данными неравенствами вычисляется по формуле $\cos(a, b) = \langle a, b\rangle/\|a\|\|b\|$. Таким образом, в третьем варианте метода отсечений на каждой итерации добавлялось не более M правильных неравенств, косинус угла между которыми не превышает некоторого порогового значения η.

3. Верхняя оценка оптимального значения

Метод отсечений, представленный в предыдущем разделе, может быть использован для ускорения работы точных методов решения, например, в методе ветвей и отсечений при поиске нижних оценок в вершинах дерева поиска. Другим способом ускорения поиска оптимального решения может быть задание в начале работы метода ветвей и границ хороших верхних оценок оптимального решения. Для поиска верхних оценок принято использовать эвристики. При решении каждого класса задач возникает вопрос выбора подходящей эвристики. Так, для тестовых примеров с неполными матрицами C методы локального поиска оказываются неэффективными в силу того, что в окрестности локального оптимума может оказаться мало допустимых точек. Для таких задач более удачным представляется использование генетических алгоритмов. Например, в работе [1] предложен генетический алгоритм для решения задачи о p-медиане с предпочтениями клиентов. Его тестирование проводилось на тестовых примерах с сильно разреженными матрицами. Тем не менее методы локального поиска могут оказаться весьма успешными на задачах с полными матрицами, т. е. когда величины c_{ij} определены для любой пары предприятие-клиент [11]. Поскольку тестовые примеры, предложенные в работе [6], имеют полные матрицы, для поиска верхних оценок оптимального решения в этих примерах предлагается использовать известный метод имитации отжига [10, 14].

Идея метода имитации отжига заимствована из статистической механики. Метод основывается на имитации физического процесса, который происходит при кристаллизации вещества из жидкого состояния в твёрдое, в том числе при отжиге металлов. Действительно, материал с понижением температуры изменяет свою структуру, стремясь к состоянию минимума энергии. Если температура понижается достаточно медленно, гарантируя достижение состояния термического равновесия на каждой температурной стадии, то система достигнет кристаллической структуры [10, 14]. Если же термическое равновесие достигаться не будет, то в результате охлаждённое вещество примет аморфную структуру.

Общая схема метода имитации отжига представлена на рис. 1. Метод начинает свою работу с некоторой допустимой точки z^0 и начальной температуры t_0. Целевая функция F — это функция энергии, система стремится к состоянию минимума энергии. На внешней итерации метода (шаг 1) осуществляется понижение температуры (шаг 1.2), в то время как внутренняя итерация (шаг 1.1) гарантирует достижение термического равновесия на каждой температурной стадии. Функция $\mathrm{Rand}(0,1)$ возвращает некоторое случайное число из промежутка $(0,1)$.

Шаг 0. Выбрать начальные значения (z^0, t_0), $k := 0$.
Положить:
лучшее известное решение $z^* := z^0$;
лучшее значение целевой функции $F^* := F(z^0)$;
текущее значение температуры $t = t_0$.
Шаг 1. Пока система не «кристаллизована»,
выполнять следующие действия.
Шаг 1.1. Пока система не достигла термического
равновесия, выполнять следующие действия:
a. Выбрать точку z' из окрестности
$N(z^k)$ точки z^k.
b. Если $\Delta(z^k, z') \overset{\triangle}{=} F(z') - F(z^k) < 0$ и
$F(z') < F^*$,
то $z^k := z'$, $z^* := z'$, $F^* := F(z')$;
$k := k + 1$. Перейти на шаг 1.1.
c. Если $\Delta(z^k, z') < 0$, то $z^k := z'$;
$k := k + 1$. Перейти на шаг 1.1.
d. Если $\Delta(z^k, z') \geqslant 0$ и
$e^{\left(\frac{-\Delta(z^k, z')}{t}\right)} > \mathrm{Rand}(0,1)$,
то $z^k := z'$; $k := k + 1$. Перейти на шаг 1.1.
Шаг 1.2. Уменьшить температуру $t := q \cdot t$.
STOP. Оптимальное значение F^*, решение z^*.

Рис. 1. Метод имитации отжига

Для того чтобы определить полностью описанный метод имитации отжига для исследуемой задачи, необходимо уточнить некоторые его детали.

1. В качестве допустимых точек z метода выступают допустимые точки (x, y) задачи (3)–(7), удовлетворяющие ограничениям (4)–(7).

2. В качестве функции энергии $F(z)$ рассматривается целевая функция (3) задачи (3)–(7).

3. Под окрестностью $N(z)$ допустимой точки $z = (x, y)$ будем понимать flip-окрестность, которая представляет собой множество допустимых точек (x', y'), векторы y' которых отличаются от y только в одной координате, т. е. $d(y, y') = 1$, где d — это расстояние Хэмминга между векторами, а вектор x' выбирается так, чтобы пара (x', y') была допустимой в задаче.

4. Система считается «кристаллизованной», если $t = t_{\min}$ или лучшее значение целевой функции не улучшается в течение заданного числа внутренних итераций M_{fr}.

Одним из существенных моментов при реализации метода имитации отжига и повышении его эффективности является выбор параметров этого метода, которые, как правило, нужно подбирать индивидуально для каждой серии тестовых примеров. Параметры, использовавшиеся для серии тестовых примеров, на которых был проведён вычислительный эксперимент, представлены в разд. 4.2.

4. Вычислительный эксперимент

Вычислительный эксперимент с использованием семейства правильных неравенств (19) и описанного метода имитации отжига проходил в три этапа. На первом этапе было проведено сравнение известных нижних оценок оптимального решения LB_1, LB_2, LB_3 (см. разд. 2). Для поиска оценки LB_3 с использованием семейства неравенств (19) были реализованы три варианта метода отсечений, описанных в разд. 2. Результаты первого этапа вычислительного эксперимента представлены в разд. 4.1. На втором этапе проводилось тестирование метода имитации отжига. Результаты представлены в разд 4.2. На третьем этапе реализованные метод отсечений и метод имитации отжига использовались для поиска оптимального решения методом ветвей и отсечений. Результаты этого этапа представлены в разд. 4.3.

Для решения задач ЛП и ЦЛП использовался пакет Xpress-MP, версия Xpress-Optimizer 19.00.00 [15]. Тестирование проводилось на ПК Intel Core 2 1.8 Ггц, 1 ГБ ОЗУ.

В качестве тестовых использовались примеры, представленные в работе [6], которые объединяются в три блока. Первый блок — это задачи размерности $m = 50,\ n = 50$, во втором блоке — задачи средней размерности $m = 50,\ n = 75$, и, наконец, в третьем — $m = 75,\ n = 100$.

4.1. Вычисление нижних оценок. На первом этапе вычислялись оценки LB_1, LB_2, а также LB_3 с использованием трёх вариантов метода отсечений, описанных выше (см. разд. 2). Были проведены эксперимен-

ты с различными значениями параметров M и η. Наиболее удачными оказались значения $M = 2n$, $\eta = 0{,}95$.

Результаты вычислений для первой группы примеров с 50 предприятиями и 50 клиентами представлены в табл. 1, где использовались следующие обозначения. В столбце Name указано имя задачи, в соответствие с [6]. В столбце gap $LB_1(\%)$ указан разрыв целочисленности (в процентах) для исходной формулировки (3)–(7). В столбце cgap(%) представлены результаты процентного улучшения разрыва целочисленности gap LB_1 для оценок LB_2 и LB_3, т. е. величины cgap $= \frac{LB_i - LB_1}{\mathrm{Opt} - LB_1} \cdot 100\%$, $i = 2, 3$, где Opt — оптимальное значение задачи. В столбце Time указано время вычисления соответствующих оценок (в секундах), причём для оценки LB_3 указаны три величины, соответствующие времени вычисления оценки LB_3 с использованием первого, второго и третьего вариантов метода отсечений ($LB_3(v1)$, $LB_3(v2)$, $LB_3(v3)$).

Таблица 1

Нижние оценки для тестовых примеров размерности 50×50

		cgap(%)		Time			
Name	gap $LB_1(\%)$	LB_2	LB_3	LB_2	$LB_3(v1)$	$LB_3(v2)$	$LB_3(v3)$
132-1	10,3	15,5	80,6	0,1	25,9	7,2	4,8
132-2	14,4	18,1	68,8	0,1	25,3	6,6	5,2
132-3	12,0	15,8	70,0	0,1	35,8	6,4	4,2
132-4	6,8	10,3	76,5	0,1	14,8	3,6	2,9
133-1	9,5	10,5	96,8	0,0	17,2	4,1	2,7
133-2	6,2	16,1	91,9	0,1	10,7	3,8	3,2
133-3	12,6	7,1	76,2	0,1	53,0	10,0	7,5
133-4	7,6	17,1	78,9	0,1	16,8	4,3	2,7
134-1	12,1	35,5	83,5	0,1	40,6	9,4	6,0
134-2	7,1	23,9	100,0	0,1	21,5	4,4	3,0
134-3	12,7	3,9	66,1	0,1	50,5	7,9	6,2
134-4	13,2	10,6	57,6	0,1	36,9	6,5	4,6

Прежде всего отметим, что несмотря на то, что время поиска нижней оценки LB_3 оказывается существенно больше, чем время поиска оценки LB_2, это не является главным критерием данного этапа вычислительного эксперимента. Более важным здесь представляется улучшение значения разрыва целочисленности для каждой из рассматриваемых формулировок.

Для задач небольшой размерности благодаря использованию нового семейства неравенств удалось существенно уменьшить разрыв целочис-

ленности, а одну из задач (134-2) даже удалось решить с помощью метода отсечений (cgap = 100%).

Результаты вычислений для задач средней и большой размерности представлены в табл. 2 и 3 соответственно. Улучшение нижней оценки для этих групп также оказалось весьма существенно. Для задач второго класса разрыв целочисленности уменьшился в среднем на 48 процентов, для третьего класса — на 34 процента.

Таблица 2

Нижние оценки для тестовых примеров размерности 50×75

Name	gap LB_1(%)	cgap(%)		Time			
		LB_2	LB_3	LB_2	$LB_3(v1)$	$LB_3(v2)$	$LB_3(v3)$
a75-50-1	27,7	11,6	40,4	0,4	234,7	34,0	27,4
a75-50-2	27,2	12,9	39,7	0,4	178,8	36,1	22,3
a75-50-3	27,1	12,5	40,6	0,3	151,7	36,6	25,7
a75-50-4	24,3	10,3	40,3	0,4	175,4	34,4	20,7
b75-50-1	28,1	12,8	51,2	0,3	159,1	35,0	21,7
b75-50-2	31,1	12,5	43,1	0,3	172,1	29,4	20,4
b75-50-3	28,2	13,5	48,2	0,3	133,1	27,5	16,5
b75-50-4	27,3	16,8	56,0	0,3	150,6	28,0	19,5
c75-50-1	31,9	16,0	53,6	0,3	100,0	23,6	15,4
c75-50-2	29,0	13,1	55,2	0,2	83,2	22,5	14,9
c75-50-3	28,5	21,8	60,7	0,3	57,4	18,6	11,2
c75-50-4	30,5	12,8	47,5	0,3	99,9	21,1	14,6

Кроме того, благодаря использованию третьего варианта метода отсечений удалось уменьшить время поиска оценки LB_3 по сравнению с первым вариантом для всех задач в среднем в 9 раз.

Таким образом, результаты вычислительного эксперимента подтверждают эффективность реализованного метода отсечений для нового семейства правильных неравенств и могут быть использованы для поиска оптимального решения задачи.

4.2. Вычисление верхних оценок. На втором этапе вычислительного эксперимента было проведено тестирование описанного в разд. 3 метода имитации отжига. Как уже упоминалось, выбор параметров данного метода оказывается зачастую ключевым моментом в повышении эффективности его работы. Проведённые предварительные эксперименты показали, что наиболее удачными из протестированных одновременно для всех трёх групп задач являются следующие значения параметров: начальная температура $t_0 = 23000$, конечная температура $t_{\min} =$

4000, число внешних итераций $M_{\text{out}} = 2m$, число внутренних итераций $M_{\text{in}} = 5n$, число итераций без уменьшения значения целевой функции $M_{\text{fr}} = 5nm$. С использованием данных параметров коэффициент понижения температуры на шаге 1.2 (см. разд. 3) вычисляется по формуле $q = \left(\frac{t_{\min}}{t_0}\right)^{\frac{1}{M_{\text{out}}-1}}$.

Таблица 3

Нижние оценки для тестовых примеров размерности 75×100

Name	gap LB_1(%)	cgap(%)		Time			
		LB_2	LB_3	LB_2	$LB_3(v1)$	$LB_3(v2)$	$LB_3(v3)$
a100-75-1	21,3	10,8	43,7	1,7	2658,1	269,8	159,1
a100-75-2	27,2	7,4	30,9	1,8	3155,8	293,3	178,0
a100-75-3	26,0	6,9	31,9	1,8	2433,2	237,2	143,0
a100-75-4	24,6	9,3	35,0	1,4	2591,6	245,8	144,3
b100-75-1	31,8	10,4	32,1	1,6	1108,7	148,0	89,1
b100-75-2	33,1	7,9	29,6	1,5	1760,8	187,1	101,4
b100-75-3	34,5	10,1	31,0	1,7	1644,7	187,9	117,9
b100-75-4	28,6	5,2	32,9	1,3	1392,7	190,9	96,6
c100-75-1	32,2	13,0	36,3	1,7	1084,7	165,7	84,3
c100-75-2	31,5	9,8	36,2	1,8	982,7	150,6	75,0
c100-75-3	32,3	10,5	35,6	1,5	784,2	128,9	79,1
c100-75-4	32,4	9,3	34,6	1,0	765,2	137,7	72,6

В табл. 4 представлены результаты тестирования описанного метода имитации отжига. С целью сбора статистических данных о качестве работы эвристики метод запускался по 100 раз для каждой задачи. В столбце Name здесь, как и в разд. 4.1, указано имя задачи в соответствии с [6]; в столбце N_{opt} указано количество раз из 100, когда методу удалось найти оптимальное решение; в столбце Err указана средняя для $N = 100$ запусков погрешность найденных верхних оценок, которая вычислялась по формуле $\text{Err} = \frac{1}{N}\sum\limits_{i=1}^{N}\frac{UB_i - \text{Opt}}{\text{Opt}} \cdot 100\%$, где UB_i — верхняя оценка, полученная при i-м запуске метода; в столбцах Time_A и Time указано соответственно среднее время поиска верхней оценки для данной задачи и общее время работы при 100 запусках метода.

Для задач небольшой размерности указанный выбор параметров оказался не очень удачным. Оптимальное значение удавалось найти лишь в 60 случаях из 100 в среднем для задач данной группы. Это можно объяснить тем, что для данных задач оптимальные значения Opt в среднем меньше значений для двух других групп, поэтому начальная и конечная

температуры $t_0 = 23000$ и $t_{\min} = 4000$ оказываются слишком велики. Тем не менее благодаря достаточному количеству внутренних и внешних итераций даже при таком температурном расписании для группы тестовых примеров небольшой размерности во всех проведённых экспериментах после 10 запусков удавалось получить оптимальное решение. Заметим также, что погрешность получаемых верхних оценок для данной серии задач оказывается достаточно мала и не превышает 2%.

Для тестовых примеров средней и большой размерностей данный выбор параметров был более удачным. Для этих групп в среднем в 80 случаях из 100 методу имитации отжига удавалось получить оптимальное решение, кроме того, погрешность получаемой верхней оценки для данных групп задач оказывается ещё меньше, чем для первой группы задач, и не превышает 0,5%.

4.3. Точные методы решения. Как уже упоминалось, для поиска точного решения задач использовался пакет Xpress-MP. В нём реализован известный метод ветвей и границ. Данный пакет представляет собой библиотеку вызываемых функций, причём пользователь может подключать свои функции, например, эвристики, метод отсечений, что и было сделано для алгоритма имитации отжига из разд. 3 и метода отсечений, представленного в разд. 2.

С целью проверки эффективности предложенного семейства правильных неравенств и реализованной эвристики для поиска оптимального решения было протестировано три варианта точного метода. В первом случае для формулировки, представленной в работе [6] (т. е. формулировки, соответствующей оценке LB_2), запускался метод ветвей и границ, реализованный в пакете Xpress-MP. Во втором случае для той же самой формулировки в начале работы пакета задавалась верхняя оценка, полученная реализованным методом имитации отжига. Наконец, в третьем случае в пакете помимо эвристики использовался также реализованный метод отсечений для нового семейства правильных неравенств.

В табл. 5 представлены результаты для первой группы задач с 50 предприятиями и 50 клиентами. Помимо указанных выше, в таблице использовались следующие обозначения: T_1 — время решения задачи в формулировке из [6] (так как время работы методов не было ограничено, оптимальное решение найдено во всех задачах); T_2 — время решения задачи в формулировке из [6] с использованием верхней оценки, полученной разработанным методом имитации отжига; T_3 — время решения задачи методом ветвей и отсечений с использованием метода имитации отжига и предложенного семейства правильных неравенств. Жирным

шрифтом в таблице выделены задачи, для которых время T_3 оказалось меньше времени T_1.

Т а б л и ц а 4

Метод имитации отжига

Name	N_{opt}	Err(%)	Time$_A$	Time
132-1	93	0,34	0,05	5,46
132-2	14	1,96	0,05	4,93
132-3	95	0,27	0,05	4,95
132-4	75	0,15	0,05	4,80
133-1	40	1,40	0,05	4,78
133-2	79	0,79	0,05	4,89
133-3	70	0,96	0,05	4,90
133-4	93	0,01	0,05	4,83
134-1	57	0,43	0,05	5,03
134-2	56	1,10	0,05	4,79
134-3	66	0,54	0,05	4,79
134-4	8	1,63	0,05	4,80
a75-50-1	73	0,06	0,12	11,60
a75-50-2	86	0,26	0,12	11,55
a75-50-3	84	0,01	0,11	11,50
a75-50-4	93	0,09	0,12	11,71
b75-50-1	86	0,38	0,11	11,16
b75-50-2	90	0,01	0,11	11,20
b75-50-3	69	0,17	0,11	11,10
b75-50-4	100	0,00	0,11	11,02
c75-50-1	14	0,33	0,11	10,82
c75-50-2	72	0,08	0,11	10,94
c75-50-3	99	0,01	0,11	10,88
c75-50-4	99	0,01	0,11	10,94
a100-75-1	95	0,21	0,36	36,42
a100-75-2	78	0,36	0,42	42,02
a100-75-3	68	0,17	0,44	44,33
a100-75-4	52	0,26	0,43	42,56
b100-75-1	79	0,20	0,41	41,17
b100-75-2	94	0,07	0,41	41,10
b100-75-3	91	0,07	0,42	41,62
b100-75-4	72	0,46	0,41	40,83
c100-75-1	52	0,28	0,41	41,30
c100-75-2	87	0,14	0,41	40,92
c100-75-3	100	0,00	0,41	40,63
c100-75-4	91	0,03	0,42	41,60

Для задач небольшой размерности существенного улучшения по срав-

нению с формулировкой из [6] добиться не удалось. Лишь в некоторых задачах с помощью разработанного метода ветвей и отсечений удалось улучшить время T_1. Что касается сравнения с временем T_2, здесь улучшение получено лишь в задаче 134-3. Но следует отметить, что время решения тестовых примеров этой группы чрезвычайно мало и составляет в среднем 2 секунды.

Таблица 5

Время решения задач размерности 50×50

Name	Opt	T_1	T_2	T_3
132-1	1122749,51	1,46	1,25	2,14
132-2	1157721,88	**3,17**	2,15	**2,87**
132-3	1146301,43	1,69	1,71	1,87
132-4	1036779,43	**1,26**	0,94	**1,08**
133-1	1103271,99	0,82	0,68	1,57
133-2	1035442,98	0,77	0,64	0,87
133-3	1171331,30	2,14	2,01	2,49
133-4	1083636,49	0,99	0,84	1,55
134-1	1179639,43	**2,62**	1,60	**1,89**
134-2	1121632,95	1,05	0,82	1,32
134-3	1171408,55	**4,73**	**3,03**	**2,71**
134-4	1210465,88	2,97	2,55	3,19

Для задач средней размерности в табл. 6 помимо упомянутых выше величин представлено также процентное уменьшение времени работы разработанного нами метода ветвей и отсечений по сравнению с временем поиска T_1 и T_2 в формулировке из [6]. Отметим, что жирным шрифтом в этой и следующей таблицах выделено процентное улучшение в задачах, время T_3 для которых оказалось меньше времени T_2.

Таким образом, в данной группе улучшение по сравнению с временем T_2 было получено для пяти тестовых примеров. Можно также отметить, что время решения задач, для которых не удалось улучшить время T_2, несколько меньше времени решения других задач (например, задачи b75-50-1, b75-50-4, c75-50-3)

Что касается сравнения с результатами, представленными в работе [6] (т. е. без использования верхней оценки), для данной группы тестовых примеров время решения задач уменьшено в среднем на 30 процентов (6-й столбец).

Результаты решения тестовых примеров большой размерности представлены в табл. 7.

Для задач этой группы не удалось улучшить время поиска в формулировке из [6] с использованием верхней оценки лишь для первых четырёх примеров. Вновь, как и для примеров средней размерности, можно заметить, что время решения этих четырёх задач существенно меньше времени решения остальных задач, для которых получено улучшение времени T_2 в среднем на 8 процентов.

Т а б л и ц а 6

Время решения задач размерности 50×75

Name	Opt	T_1	T_2	T_3	$\frac{T_1-T_3}{T_1} \cdot 100\%$	$\frac{T_2-T_3}{T_2} \cdot 100\%$
a75-50-1	1661269,26	149,10	**84,68**	**76,25**	48,86	**9,95**
a75-50-2	1632906,96	116,67	**70,83**	**68,19**	41,56	**3,72**
a75-50-3	1632212,56	93,82	84,68	88,77	5,38	-4,84
a75-50-4	1585027,69	83,05	52,48	54,77	34,05	-4,37
b75-50-1	1252803,55	62,40	31,72	44,99	27,90	-41,82
b75-50-2	1337446,44	137,95	101,88	109,25	20,81	-7,23
b75-50-3	1249750,19	103,03	**51,18**	**43,07**	58,19	**15,84**
b75-50-4	1217508,08	47,40	28,39	39,15	17,41	-37,87
c75-50-1	1310192,74	74,88	64,91	73,18	2,27	-12,75
c75-50-2	1244255,03	46,17	**31,46**	**29,85**	35,35	**5,12**
c75-50-3	1201706,42	29,25	16,47	21,39	26,86	-29,86
c75-50-4	1334782,49	109,53	**73,32**	**67,41**	38,46	**8,06**

Т а б л и ц а 7

Время решения задач размерности 75×100

Name	Opt	T_1	T_2	T_3	$\frac{T_1-T_3}{T_1} \cdot 100\%$	$\frac{T_2-T_3}{T_2} \cdot 100\%$
a100-75-1	2286397,49	479,33	171,11	172,22	64,07	-0,65
a100-75-2	2463186,71	781,68	682,03	700,40	10,40	-2,69
a100-75-3	2415835,79	741,49	674,03	677,23	8,67	-0,47
a100-75-4	2380149,73	597,92	490,47	495,04	17,21	-0,93
b100-75-1	1950231,44	3577,56	**2349,45**	**2110,47**	41,01	**10,17**
b100-75-2	2023097,40	4754,34	**2655,15**	**2464,11**	48,17	**7,19**
b100-75-3	2062594,97	4731,20	**3359,11**	**3095,46**	34,57	**7,85**
b100-75-4	1865322,75	2379,22	**1061,46**	**1049,13**	55,90	**1,16**
c100-75-1	1843620,48	2454,78	**1653,67**	**1416,92**	42,28	**14,32**
c100-75-2	1808867,01	2537,67	**1713,86**	**1655,86**	34,75	**3,38**
c100-75-3	1820587,32	1902,55	**952,61**	**908,71**	52,24	**4,61**
c100-75-4	1839007,24	2194,25	**1664,57**	**1423,39**	35,13	**14,49**

Кроме того, по сравнению с результатами из [6] для данной группы задач получено улучшение в среднем на 37 процентов.

Таким образом, на основании проведённых вычислений, не принимая во внимание задачи маленькой размерности, время решения которых, вообще говоря, незначительно, можно сделать вывод об эффективности разработанного метода ветвей и отсечений. При этом, если в задачах средней размерности улучшение удалось получить главным образом за счёт реализованной эвристики, то для больших задач можно говорить также и об эффективности предложенного семейства правильных неравенств, которое позволяет заметно сократить время решения задач.

ЛИТЕРАТУРА

1. **Алексеева Е. В., Кочетов Ю. А.** Генетический локальный поиск для задачи о p-медиане с предпочтениями клиентов // Дискрет. анализ и исслед. операций. — 2007. — Т. 14, № 1. — С. 3–31.
2. **Береснев В. Л.** Дискретные задачи размещения и полиномы от булевых переменных. — Новосибирск: Изд-во Института математики, 2005. — 408 с.
3. **Васильев И. Л., Климентова К. Б., Кочетов Ю. А.** Новые нижние оценки для задачи размещения с предпочтениями клиентов // Журн. вычисл. математики и мат. физики. (принято в печать).
4. **Горбачевская Л. Е.** Полиномиально разрешимые и NP-трудные двухуровневые задачи стандартизации // Дис. ... канд. физ.-мат. наук. — Новосибирск, 1998. — 131 с.
5. **Горбачевская Л. Е., Дементьев В. Т., Шамардин Ю. В.** Двухуровневая задача стандартизации с условием единственности оптимального потребительского выбора // Дискрет. анализ и исслед. операций. — 1999. — Т. 6, № 2. — С. 3–11.
6. **Cánovas L., García S., Labbé M., Marín A.** A strengthened formulation for the simple plant location problem with order // Operations Research Letters. — 2007. — Vol. 35, N 2. — P. 141–150.
7. **Dempe S.** Foundations of bilevel programming. — Dordrecht: Kluwer Acad. Publ., 2002. — 320 p.
8. **Hanjoul P., Peeters D.** A facility location problem with clients' preference orderings // Regional Sci. Urban Econom. — 1987. — Vol. 17. — P. 451–473.
9. **Hansen P., Kochetov Y., Mladenovic N.** Lower bounds for the uncapacitated facility location problem with user preferences. Les Charies du GERAD G–2004–24, 2004. — P. 1–10.
10. **Kirkpatrick S., Gelatt C. D., Vecchi M.P.** Optimization by simulated annealing // Science. — 1983. — Vol. 220, N 4598. — P. 671–680.
11. **Kochetov Yu., Ivanenko D.** Computationally difficult instances for the uncapacitated facility location problem // Metaheuristics: progress as real solvers. — New York: Springer, 2005. — P. 351–367.

12. **Nemhauser G. N., Wolsey L. A.** Integer and combinatiorial optimization. — New York: A Wiley-Interscience Publication, 1999. — 766 p.
13. **Pochet Y., Wolsey L. A.** Production planning by mixed integer programming. — New York: Springer-Verl., 2006. — 477 p.
14. **Van Laarhoven P. J. M., Aarts E. H. L.** Simulated annealing: theory and application. Dordrecht: D. Reidel Publ. Comp., 1988. — 198 p.
15. http://www.dashoptimisation.com

Васильев Игорь Леонидович,
e-mail: vil@icc.ru

Климентова Ксения Борисовна,
e-mail: Xenia.Klimentova@icc.ru

Статья поступила
7 ноября 2008 г.
Переработанный вариант —
4 февраля 2009 г.