Автоматика и телемеханика, № 7, 2014

© 2014 г. **Г.Г. ЗАБУДСКИЙ, д-р физ.-мат. наук (zabudsky@ofim.oscsbras.ru), А.А. КОВАЛЬ (akoval@thumbtack.net)**
(Омский филиал Института математики им. С.Л. Соболева СО РАН)

ПОИСК РЕШЕНИЯ С ЗАДАННОЙ ТОЧНОСТЬЮ МАКСИМИННОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ НА ПЛОСКОСТИ[1]

Рассматривается задача оптимального размещения в ограниченной области на плоскости с фиксированными в ней объектами. Заданы минимально допустимые расстояния между размещаемыми и фиксированными и размещаемых объектов между собой. Критерий – максимум минимального взвешенного расстояния от размещаемых объектов до фиксированных. Предложен квазиполиномиальный комбинаторный алгоритм решения задачи с заданной точностью. Представлены результаты вычислительного эксперимента с использованием модели целочисленного программирования и пакета IBM ILOG CPLEX.

1. Введение

В оптимальном размещении средств обслуживания (объектов) можно выделить два типа задач. Размещать объекты "как можно ближе" к клиентам и "как можно дальше" от них. Исследованию и решению задач первого типа уделяется достаточно много внимания. Наиболее известные задачи данного класса – это задачи о p-центре и p-медиане, задача Вебера, простейшая задача размещения и ее обобщения [1–7]. Объектами в таких задачах могут быть предприятия, склады, магазины, аварийные службы и др.

Задачи второго типа также имеют много приложений на практике [8, 9], но их изучению в отечественной литературе уделяется недостаточно внимания. В этих задачах часто применяется максиминный критерий, т.е. объекты размещаются "как можно дальше" от имеющихся (фиксированных). Такой критерий может использоваться при размещении опасных объектов, которыми могут быть заводы, работающие с радиоактивными или химическими веществами, аэропорты, городские свалки и т.д. Они оказывают вредное воздействие на здоровье жителей близлежащих районов и должны располагаться по возможности дальше от них. Указанный критерий применяется также при размещении объектов, требующих экологической чистоты (санатории, дома отдыха и т.д.).

В данной статье рассматривается задача второго типа оптимального размещения средств обслуживания в прямоугольной области на плоскости с расположенными в ней клиентами (фиксированными объектами) [10, 11]. Заданы минимально допустимые расстояния между фиксированными и размещаемыми объектами, а также размещаемых между собой. Необходимо разместить объекты в указанной области таким образом, чтобы выполнялись заданные

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 13-01-00862).

ограничения и минимальное расстояние между фиксированными и размещаемыми было максимальным.

Для записи математической модели введем обозначения:

$V = \{v_1, \ldots, v_n\}$ – множество фиксированных объектов, которые соответствуют населенным пунктам, (a_i, b_i) – координаты объекта v_i;

S – прямоугольная область на плоскости, которая содержит множество фиксированных объектов, A – ее длина, B – ширина;

$O = \{o_1, \ldots, o_p\}$ – множество размещаемых объектов, (x_j, y_j) – координаты объекта o_j;

R_{ij} – минимальное допустимое расстояние между фиксированным объектом v_i и размещаемым o_j;

M – минимальное допустимое расстояние размещаемых объектов между собой;

$w_i > 0$ – вес фиксированного объекта v_i;

∂S – граница множества S;

ρ – метрика.

Необходимо разместить множество объектов O в области S таким образом, чтобы выполнялись заданные ограничения и взвешенное минимальное расстояние от размещаемых объектов до фиксированных было максимальным.

Введем дополнительную переменную L. Тогда модель имеет вид:

$$L \to \max, \tag{1}$$

$$\rho(o_j, v_i) \geqslant \max\left(R_{ij}, \frac{L}{w_i}\right), \quad o_i \in O, \quad v_j \in V, \tag{2}$$

$$\rho(o_i, o_j) \geqslant M, \quad o_i, o_j \in O, \tag{3}$$

$$(x_j, y_j) \in S, \quad o_j \in O. \tag{4}$$

Задача (1)–(4) является NP-трудной [12]. Допустимая область может быть несвязной и невыпуклой.

Для решения задачи (1)–(4) в метрике l_1 применяются комбинаторные алгоритмы и аппарат линейного программирования (ЛП). В [12] при решении задачи с условиями $R_{ij} = R_i$, $w_i = 1$ проводятся прямые, параллельные осям координат, через фиксированные объекты. Область S разбивается на прямоугольные ячейки. В каждой из ячеек определен взаимный порядок расположения размещаемого объекта относительно фиксированных. Для двух размещаемых объектов предложен алгоритм ветвей и границ в случае, когда в каждой ячейке располагается не более одного объекта. При условии допустимости размещения нескольких объектов внутри одной ячейки предлагается решать серию задач ЛП либо задач целочисленного линейного программирования (ЦЛП) [11]. Сведение непрерывной задачи к дискретной при условиях $R_{ij} = R_i$, w_i – произвольные положительные и метрика l_∞ описано в [13]. Для этого строится конечное множество допустимых значений целевой функции (1), и для каждого такого значения проверяется “можно или нельзя разместить объекты в указанной области”. Параллельные алгоритмы решения задачи в случае $R_{ij} = Lw_i$, $M = L$ для двух объектов в метриках l_1 и l_2 предложены в [14].

В данной работе рассматривается задача в метрике l_∞ в более общей постановке: p – произвольное целое положительное, $R_{ij} \neq R_i, w_i$ – произвольные положительные вещественные. Доказано, что если множество допустимых решений не пусто, то найдется оптимальное решение, в котором по крайней мере один объект находится в угловой точке этого множества. Это обобщение свойства, приведенного в [13]. С использованием этого свойства разработан комбинаторный алгоритм нахождения решения задачи с заданной точностью ε трудоемкости $O\left(n^{p-1} \log \frac{D(S)}{\varepsilon} \log n\right)$, где $D(S)$ – диаметр множества S. Проведен вычислительный эксперимент по сравнению эффективности предложенного алгоритма и решении задачи с помощью пакета IBM ILOG CPLEX. Для этого построена модель ЦЛП.

2. Алгоритм решения

Идея алгоритма состоит в переборе значений параметра L и в проверке существования допустимого решения для них. Для выбора очередного значения L решается вспомогательная задача – задача распознавания (ЗР) в допустимой области с применением декомпозиции [15, 16]. Указанная область строится как теоретико-множественная разность исходного множества S и объединения всех запрещенных зон – областей, образованных ограничениями (2). Радиус запрещенной зоны для фиксированных $v_i \in V$, $o_j \in O$ – это $\max\left(\frac{L}{w_i}, R_{ij}\right)$.

Задача (задача распознавания). *Можно или нельзя для фиксированного значения L разместить p объектов в области S вне запрещенных зон, определяемых параметром L, так, чтобы выполнялось ограничение на минимальное расстояние между ними?*

Если решение ЗР дает ответ “да”, то на следующей итерации происходит увеличение запрещенных зон пропорционально весам фиксированных объектов. Если “нет” – пропорциональное уменьшение. Таким образом, в алгоритме последовательно расширяются или сужаются запрещенные зоны до тех пор, пока не найдется допустимое значение L, удовлетворяющее заданной точности.

2.1. Два объекта

Значения параметра L выбираются из некоторого отрезка с помощью метода дихотомии. Для нахождения отрезка, содержащего оптимальное значение L, используется следующее свойство.

Утверждение. Если задача (1)–(4) *имеет решение, то для оптимального значения L^* целевой функции справедливы неравенства*

$$\min_{o_i \in O, v_j \in V} (w_i R_{ij}) \leqslant L^* \leqslant \max(A, B) \max_{o_i \in O} w_i.$$

Левое неравенство следует из того, что существует допустимое размещение при значении L таком, что $\max(R_{ij}, \frac{L}{w_i}) = R_{ij}$, т.е. $L \leqslant R_{ij} w_i$. Правое неравенство следует из того, что $\max(R_{ij}, \frac{L}{w_i}) \leqslant \rho(v_i, o_j) \leqslant \max(A, B)$.

Обозначим через $J_k = [g_k, r_k]$ отрезок, рассматриваемый на шаге k, $g_1 = \min_{i,j}(w_i R_{ij})$, $r_1 = (A+B)\max_i w_i$. Выбирается L' – середина отрезка J_k (метод дихотомии), и решается ЗР. Если ответ в ЗР "да", то $J_{k+1} = [L', r_k]$, иначе $J_{k+1} = [g_k, L']$. Для решения ЗР используется алгоритм нахождения контура объединения изотетичных прямоугольников с помощью плоского заметания (ПЗ) с использованием дерева отрезков [17]. Для каждого o_j изотетичные прямоугольники – это области

$$F_i^j(L') = \left\{ z \in R^2,\ \rho(z, v_i) \leqslant \max\left(R_{ij}, \frac{L'}{w_i}\right) \right\}, \quad v_i \in V.$$

Это – набор квадратов со сторонами, параллельными осям координат. Объединение $F^j(L')$ из n изотетичных квадратов $F_1^j(L'), \ldots, F_n^j(L')$ может состоять из нескольких непересекающихся компонент связности, а каждая компонента может содержать внутренние дыры. Контур (граница) $\partial F^j(L')$ состоит из набора непересекающихся циклов, образованных ориентированными вертикальными и горизонтальными ребрами. Считаем, что каждое ребро ориентировано так, что область из объединения квадратов расположена слева. Цикл ориентирован по часовой стрелке, если он ограничивает дыру, и против, если это внешняя граница компоненты связности. Отметим, что запрещенные зоны – это открытые множества $\mathrm{int}(F_i^j(L'))$. Обозначим допустимую область для размещения объекта o_j как $U_j = S \setminus \mathrm{int}(F^j)$.

Приведем описание алгоритма решения ЗР для $p = 2$ при фиксированном L', частный случай которого приведен в [13, 14].

Алгоритм 1.

С помощью алгоритма ПЗ строим контуры $\partial F^1(L')$, $\partial F^2(L')$.
Находим K_1,K_2 – множества угловых точек ∂U_1, ∂U_2.
u_1^*, u_2^*
$\rho^* = 0$
for $u_1 \in K_1$, $u_2 \in K_2$ **do**
  if $\rho^* < \rho(u_1, u_2)$ **then**
    $\rho^* = \rho(u_1, u_2)$
    $u_1^* = u_1$
    $u_2^* = u_2$
  end if
end for
if $\rho^* \geqslant M$ **then**
  return ответ "да", u_1^*, u_2^* – допустимое размещение
else
  return ответ "нет"
end if

Замечание 1. Множество K_i состоит из угловых точек множества S, не принадлежащих $F^i(L')$, угловых точек $F^i(L')$ в S и угловых точек $\partial S \cup \cup \partial F^i(L')$.

Замечание 2. Для нахождения u_1^* и u_2^* достаточно рассмотреть угловые точки с наибольшими и наименьшими значениями абсцисс и ординат.

Замечание 3. Если $R_{ij} = R_i$, то $F^1(L') = F^2(L')$. Для нахождения u_1^* и u_2^* достаточно найти наиболее удаленные точки ∂U_1 [13].

Количество итераций (решения ЗР) равно $\log \frac{r_1 - g_1}{\varepsilon}$. Трудоемкость построения контура изотетичных прямоугольников равна $O(n \log n + m \log(\frac{n^2}{m}))$, где m – число ребер в контуре [17]. Так как находится контур объединения квадратов, то $m = O(n)$ и общая трудоемкость алгоритма решения ЗР составляет $O(n \log n)$.

2.2. Общий случай

Значения параметра L выбираются как описано выше. При решении ЗР для $p > 2$ предлагается провести редукцию, т.е. свести проверку возможности размещения p объектов к проверке допустимости размещения $p-1$ объектов. Считаем, что значение параметра L, для которого решается ЗР, фиксировано, и поэтому далее будем опускать его в формулах.

Для произвольного $o_i \in O$ обозначим некоторое вертикальное ребро e множества ∂U_i через $(e^t, e^b; e_x)$, где e^t, e^b – значения верхней и нижней ординаты соответственно, e_x – абсцисса ребра. Аналогично через $(h^l, h^r; h_y)$ – горизонтальное ребро h. Пусть имеется некоторое допустимое размещение p объектов, т.е. определены (x_i, y_i) координаты $o_i \in O$. Такое допустимое решение задачи (1)–(4) будем обозначать через $o_i(x_i, y_i)$. Каждому объекту $o_i(x_i, y_i)$ соответствует окрестность $Q_i(x_i, y_i)$ диаметра M (квадрат со сторонами, параллельными осям координат) с центром в точке (x_i, y_i). Через $Q(x, y) = \{Q_1(x_1, y_1), \ldots, Q_p(x_p, y_p)\}$, $x = (x_1, \ldots, x_p)$, $y = (y_1, \ldots, y_p)$ обозначим множество всех таких окрестностей. Отметим, что для произвольных o_i, $o_j \in O$ справедливы условия

$$\operatorname{int}(Q_i(x_i, y_i)) \cap \operatorname{int}(Q_j(x_j, y_j)) = \varnothing. \tag{5}$$

Перенумеруем объекты $o_i(x_i, y_i)$ в порядке неубывания абсцисс x_i и определим отображение f_{left} "сдвиг влево" для них. Это отображение последовательно применяется к объектам в соответствии с их нумерацией. Так как каждому $o_i(x_i, y_i)$ соответствует окрестность, то указанное отображение можно рассматривать как "сдвиг влево" окрестностей $Q_i(x_i, y_i)$. Для объекта $o_j(x_j, y_j)$ отображение задается следующим образом:

$$f_{left}(x_j, y_j) = (\hat{x_j}, \hat{y_j}), \quad \hat{x_j} = x_j - k_j, \quad \hat{y_j} = y_j, \quad k_j = \min(k_j^0, k_j^1),$$
$$k_j^0 = \min_{i<j}(x_j - \hat{x_i} - M), \quad i : |y_j - \hat{y_i}| < M,$$
$$k_j^1 = \min_{v \in \partial U_j}(x_j - e_x), \quad e : e^b - M/2 < y_j < e^t + M/2, \quad e_x < x_j.$$

Значение k_j^0 соответствует перемещению $Q_j(x_j, y_j)$ до касания с ближайшей слева ранее перемещенной окрестностью $\hat{Q}_i(\hat{x_i}, \hat{y_i})$. Если такой окрестности нет, то полагаем $k_j^0 = \infty$. Значение k_j^1 соответствует перемещению $o_j(x_j, y_j)$ до ближайшего слева вертикального ребра $e \in \partial U_j$, такое всегда существует. Отметим, что размещение, полученное после применения отображения f_{left}, удовлетворяет условию (5).

Введем для размещаемых объектов отношение блокирования. Обозначим через q^i_{left} и q^i_{right} – левое и правое ребро окрестности Q_i соответственно.

Определение. Объект $o_i(x_i, y_i)$ блокирует $o_j(x_j, y_j)$, если

$$q^i_{right} \cap q^j_{left} \neq \varnothing.$$

Понятие блокирования можно применять к окрестностям. Справедлива теорема.

Теорема. Если ответ в ЗР "да" для заданного L, то существует допустимое решение задачи (1)–(4)*, в котором по крайней мере один объект o_i располагается в угловой точке допустимой области ∂U_i.*

Идея доказательства состоит в построении допустимого решения с использованием отображения f_{left}, в котором по крайнем мере один из объектов находится в угловой точке. Доказательство приведено в Приложении.

Из теоремы следует, что при решении ЗР для p объектов один объект можно считать расположенным в угловой точке, который рассматривается как фиксированный с соответствующей запрещенной зоной радиуса M. Таким образом, задача бóльшей размерности может быть сведена к задаче меньшей размерности. Идея алгоритма решения ЗР заключается в фиксировании очередного размещаемого объекта в угловой точке области его допустимых расположений и переходе к решению задачи с меньшим числом объектов. Пусть зафиксирован порядок на множестве P. Используя описанную редукцию, приведем алгоритм решения ЗР для $p \geqslant 3$.

Алгоритм 2.

function SOLVE ($facilities$ – число размещаемых объектов)
  if $facilities = 2$ **then**
    Применяем алгоритм 1.
    if ответ "да" **then**
      return ответ в исходной ЗР "да".
    else
      return ответ "нет".
    end if
  else
    $index = p - facilities + 1$
    Строим ∂F^{index} и множество K_{index} – угловых точек области U_{index}.
    for k_i in K_{index} **do**
      Фиксируем o_{index} в k_i
      if SOLVE ($facilities - 1$) = "да" **then**
        return ответ в исходной ЗР "да".
      end if
    end for
  end if
  return ответ "нет".
end function
return SOLVE (p);

Замечание 4. Отметим, что при фиксировании объектов в угловых точках добавляются запрещенные зоны радиуса M с центрами в этих точках.

Замечание 5. Трудоемкость алгоритма 2 не превосходит $O(n^{p-1}\log n)$, при этом используемая память $O(pn)$.

Действительно, в алгоритме используется $p-2$ вложенных цикла, в каждом из которых перебирается множество из $O(n)$ элементов, т.е. число операций $O(n^{p-2})$. Учитывая, что трудоемкость алгоритма 1 составляет $O(n\log n)$, получаем указанную выше трудоемкость. В [13] для одинаковых объектов ($R_{ij} = R_i$) используется память $O(n)$. Так как в алгоритме 2 необходимо хранить контуры для всех размещаемых объектов, то требуемая память составит $O(pn)$.

3. Модель целочисленного линейного программирования

Для построения модели ЦЛП используется свойство, что решения задачи в метриках l_∞ и l_1 связаны между собой [18]. Пусть $O^*, \bar{O}^*$ – оптимальные решения в метриках l_∞, l_1 соответственно, тогда справедливы соотношения:

$$T(\bar{O}^*) = O^*,$$
$$\bar{O}^* = T^{-1}(O^*),$$

где T – отображение вида $\left(\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right)$.

Запишем задачу (1)–(4) в метрике l_1. Обозначим $(\bar{a}_i, \bar{b}_i) = T^{-1}(a_i, b_i)$, $(\bar{x_j}, \bar{y_j}) = T^{-1}(x_j, y_j)$, $i \in \{1,\dots,n\}$, $j \in \{1,\dots,p\}$; W, H – достаточно большие положительные константы, например, $W = H = 2(A+B)$. Введем булевые переменные z_{ij}, t_{ij}, которые определяют взаимное расположение объектов, $z_{ij} = 1$, если $\bar{x_j} \leqslant \bar{a}_i$, и $z_{ij} = 0$ – иначе; $t_{ij} = 1$, если $\bar{y_j} \leqslant \bar{b}_i$, и $t_{ij} = 0$ – иначе. Ограничения (2) для $o_i \in O$ и $v_j \in V$ будут иметь вид:

$$(6)\quad \begin{cases} \bar{x_j} - \bar{a}_i + \bar{y_j} - \bar{b}_i + z_{ij}H + t_{ij}W \geqslant \max\left(R_{ij}, \dfrac{L}{w_i}\right), \\ -\bar{x_j} + \bar{a}_i + \bar{y_j} - \bar{b}_i + (1-z_{ij})H + t_{ij}W \geqslant \max\left(R_{ij}, \dfrac{L}{w_i}\right), \\ \bar{x_j} - \bar{a}_i - \bar{y_j} + \bar{b}_i + z_{ij}H + (1-t_{ij})W \geqslant \max\left(R_{ij}, \dfrac{L}{w_i}\right), \\ -\bar{x_j} + \bar{a}_i - \bar{y_j} + \bar{b}_i + (1-z_{ij})H + (1-t_{ij})W \geqslant \max\left(R_{ij}, \dfrac{L}{w_i}\right), \\ z_{ij}, t_{ij} \in \{0,1\}. \end{cases}$$

Аналогично введем булевые переменные s_{ij}, u_{ij}: $s_{ij} = 1$, если $\bar{x_j} \leqslant \bar{x_i}$, и $s_{ij} = 0$ – иначе, $u_{ij} = 1$, если $\bar{y_j} \leqslant \bar{y_i}$, и $u_{ij} = 0$ – иначе. Ограничения (3) для

$v_i, v_j \in V$ будут иметь вид:

$$(7)\qquad \begin{cases} \bar{x_j} - \bar{x_i} + \bar{y_j} - \bar{y_i} + s_{ij}H + u_{ij}W \geqslant M, \\ -\bar{x_j} + \bar{x_i} + \bar{y_j} - \bar{y_i} + (1 - s_{ij})H + u_{ij}W \geqslant M, \\ \bar{x_j} - \bar{x_i} - \bar{y_j} + \bar{y_i} + s_{ij}H + (1 - u_{ij})W \geqslant M, \\ -\bar{x_j} + \bar{x_i} - \bar{y_j} + \bar{y_i} + (1 - s_{ij})H + (1 - u_{ij})W \geqslant M, \\ s_{ij}, u_{ij} \in \{0, 1\}. \end{cases}$$

Ограничения $(x_j, y_j) \in S$ и $o_j \in O$ (после поворота) записываются так:

$$(8)\qquad \begin{cases} \bar{y_j} - \bar{x_j} \geqslant 0, \\ \bar{x_j} + \bar{y_j} \geqslant 0, \\ \bar{y_j} + \bar{x_j} \leqslant A, \\ \bar{y_j} - \bar{x_j} \leqslant B. \end{cases}$$

Если каждое из ограничений (6) представить в виде двух неравенств, то получим модель ЦЛП (1), (6)–(8).

4. Вычислительный эксперимент

Для проведения вычислительного эксперимента использовались модель ЦЛП (1), (6)–(8) и программный продукт IBM ILOG CPLEX версии 12.2. Комбинаторный алгоритм реализован на языке C++. Вычисления проводились на ПЭВМ со следующими параметрами: Intel Core i3 2310M, частота процессора 2100 МГц.

В табл. 1 отражена зависимость времени работы (в секундах) предложенного алгоритма и пакета ILOG CPLEX от количества фиксированных объектов. Параметры задачи: $p = 2$, S – $[1600, 1000]$, $R_{ij} = R_i \leqslant 25$, w_i – генерировались случайным образом, $\varepsilon = 0{,}001$.

Таблица 1

Число фиксированных объектов	Комбинаторный алгоритм	CPLEX
10	0,011	0,15
20	0,019	0,87
30	0,024	2,49
40	0,027	3,36
50	0,031	5,32

Таблица 2

Число размещаемых объектов	Комбинаторный алгоритм	CPLEX
3	0,028	3,93
4	0,032	5,56
5	0,037	9,01
6	0,055	51,13
7	0,060	93,75
8	0,078	98,7
9	0,170	>5 минут

В табл. 2 отражена зависимость времени работы (в секундах) алгоритма и пакета ILOG CPLEX от количества размещаемых объектов. Параметры задачи: $n = 30$, S – $[1600, 1000]$, $R_{ij} = R_i \leqslant 25$, w_i – генерировались случайным образом, $\varepsilon = 0{,}001$.

Анализируя результаты экспериментов, отметим, что комбинаторный алгоритм решает задачи на несколько порядков быстрее. С ростом размерности задачи его время работы увеличивается незначительно, а время решения задачи с помощью пакета существенно возрастает, несмотря на то что используются преимущества многоядерного процессора. Предложенный комбинаторный алгоритм в процессе работы использует только один поток. Можно сделать вывод, что применение алгоритма поиска решения с заданной точностью предпочтительнее использования продукта IBM ILOG CPLEX.

5. Заключение

Исследовалась максиминная задача размещения точечных объектов в прямоугольной области на плоскости с имеющимися в ней фиксированными объектами. Заданы ограничения на минимально допустимые расстояния между фиксированными и размещаемыми, а также размещаемых между собой. Рассмотрен вариант минимально допустимых расстояний, зависящих от типа размещаемых объектов. Предложен алгоритм нахождения приближенного решения с заданной точностью. Проведен вычислительный эксперимент по анализу эффективности предложенного алгоритма и решения задачи с помощью модели целочисленного линейного программирования и пакета IBM ILOG CPLEX. Эксперименты показали, что с помощью предложенного алгоритма находятся решения, близкие к оптимальным по значению целевой функции с существенно меньшими вычислительными затратами, чем при решении задачи пакетом IBM ILOG CPLEX. Разработанный алгоритм может применяться в системах автоматизации проектирования расположения объектов, например, представляющих какую-либо опасность.

ПРИЛОЖЕНИЕ

Доказательство теоремы. Очевидно, что утверждение справедливо для $p = 1$. Пусть $p \geqslant 2$.

Так как ответ в ЗР “да”, то обозначим через $o_i(x_i, y_i)$, $i \in \{1, \ldots, p\}$, некоторое допустимое размещение. Далее для упрощения обозначений, где это не вызывает трудностей в понимании, координаты (x_i, y_i) объектов o_i и соответствующих им областей Q_i будем опускать. На рис. 1 показан пример допустимого размещения для $p = 5$. Темные квадраты – это соответствующие объектам окрестности. Линиями изображены границы допустимых областей. Чтобы не перегружать рис. 1, в примере полагаем $\partial F^1 = \partial F^2$ и $\partial F^3 = \partial F^5$.

Построим новое допустимое размещение с помощью отображения f_{left}. На рис. 2 показан результат применения отображения f_{left}. Отображение f_{left} последовательно сдвигает объекты в порядке неубывания их абсцисс. На рис. 2 объекты o_1, o_2, o_3 и o_4 переместились до ближайших вертикальных ребер соответствующих допустимых областей. При сдвиге объект o_3 блокиру-

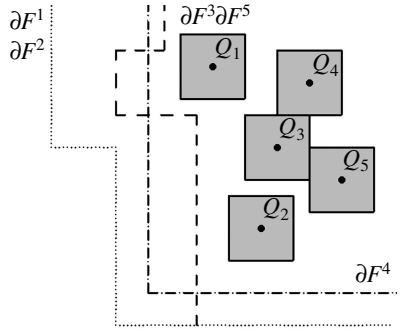

Рис. 1. Начальное размещение.

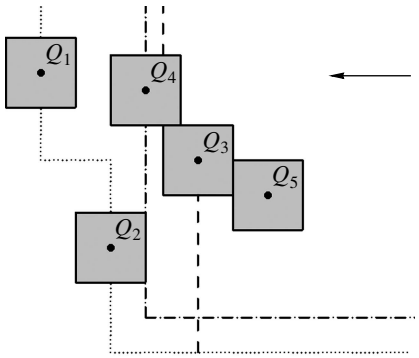

Рис. 2. Сдвиг влево.

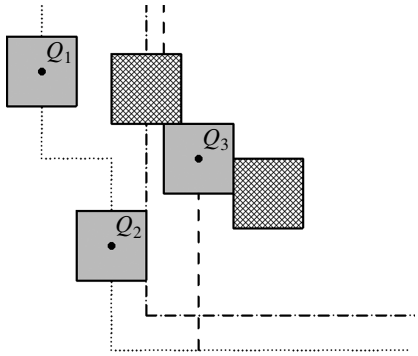

Рис. 3. Множество Q^l.

ет o_5. Отметим, что расстояния, на которые были перемещены объекты по оси абсцисс, следующие: $k_1 = k_1^1$, $k_2 = k_2^1$, $k_3 = k_3^1$, $k_4 = k_4^1$, $k_5 = k_5^0$. Если после применения отображения хотя бы один размещаемый объект попал в угловую точку, то теорема доказана. Пусть не попал.

Рассмотрим множество $Q^l = \{Q_1, \ldots, Q_p\}$ окрестностей, которые удовлетворяют следующим условиям: $Q_i \in Q^l$, если найдется ребро $(e^t, e^b; e_x) \in \partial U_i$ такое, что

$$x_i = e_x, \quad e^b \leqslant y_i \leqslant e^t,$$

и не существует Q_j такой, что

$$x_i > x_j, \quad |y_i - y_j| \leqslant M.$$

Очевидно, что $Q^l \neq \varnothing$, так как это множество содержит окрестность, центр которой имеет наименьшую абсциссу. Обозначим через Q_* окрестность из множества Q^l для которой

$$y_* = \min_{Q_i \in Q^l} y_i.$$

На рис. 3 изображен вариант, когда множество $Q^l = \{Q_1, Q_2, Q_3\}$. В качестве Q_* выбирается окрестность Q_2.

Покажем, что не существует окрестности, лежащей в левом нижнем квадранте, образованном прямыми, проходящими через нижнее и правое ребра окрестности Q_*. Предположим, что такая окрестность Q_0 существует. Так как $Q_0 \notin Q^l$ и Q_0 расположена ниже Q_*, то найдется Q_1 слева от Q_0, которая блокирует Q_0 или $|y_1 - y_0| \leqslant M$. Окрестность Q_1 не может быть элементом Q^l, поскольку это противоречит выбору Q_*. Следовательно, существует Q_2, которая блокирует Q_1 или $|y_2 - y_1| \leqslant M$. Очевидно, $Q_0 \neq Q_2$. Окрестность Q_2 также не может быть элементом Q^l, следовательно, существует окрестность, которая ее блокирует. Рассуждая таким образом дойдем до последней нерассмотренной в квадранте окрестности Q_s. Так как не существует окрестности, блокирующей Q_s или лежащей правее Q_s, то после сдвига влево центр Q_s должен лежать на вертикальном ребре своей допустимой области. Таким образом, $Q_s \in Q^l$ и $y_s < y_*$, что противоречит выбору Q_*. Следовательно, не существует окрестности, лежащей в квадранте, образованном нижним и правым ребрами окрестности Q_*.

Так как объект o_* находится на вертикальном ребре и нет окрестностей, расположенных в квадранте, то можно переместить o_* в точку (e_x, e^b). Полученное размещение останется допустимым.

Теорема доказана.

СПИСОК ЛИТЕРАТУРЫ

1. *Береснев В.Л.* Дискретные задачи размещения и полиномы от булевых переменных. Новосибирск: ИМ СО РАН им. С.Л. Соболева, 2005.
2. *Панюков А.В., Пельцвергер Б.Ф.* Оптимальное размещение дерева в конечном множестве // Журн. вычисл. матем. и мат. физики. 1988. Т. 28. С. 618–620.
3. *Panyukov A.V., Pelzwerger B.V.* Polynomial Algorithms to Finite Veber Problem for a Tree Network // J. Computational and Applied Mathematics. 1991. V. 35. P. 291–296.
4. *Агеев А.А., Гимади Э.Х., Курочкин А.А.* Полиномиальный алгоритм решения задачи размещения на цепи с одинаковыми производственными мощностями предприятий // Дискретный анализ и исследование операций. Новосибирск: ИМ СО РАН им. С.Л. Соболева, 2009. Т. 16. № 5. С. 3–18.
5. *Забудский Г.Г.* Построение модели и решение задач размещения на плоскости с запрещенными зонами // АиТ. 2006. № 12. С. 136–141.

   Zabudskii G.G. Model Building and Location Problem Solving in a Plane with Forbidden Gaps // Autom. Remote Control. 2006. V. 67. No. 12. P. 1986–1990.
6. *Забудский Г.Г., Амзин И.В.* Сужение области поиска решения задачи Вебера на плоскости с прямоугольными запрещенными зонами // АиТ. 2012. № 5. С. 71–83.

   Zabudskii G.G., Amzin I.V. Search Region Contraction of the Weber Problem Solution on the Plane with Rectangular Forbidden Zones // Autom. Remote Control. 2012. V. 73. No. 5. P. 821–830.
7. *Кочетов Ю.А., Пащенко М.Г., Плясунов А.В.* О сложности локального поиска в задаче о p-медиане // Дискретный анализ и исследование операций. Сер. 2. 2005. Т. 12. № 2. С. 44–71.
8. *Farahani R.Z., Hekmatfar M.* Facility Location: Concepts, Models, Algorithms and Case Studies. Heidelberg: Physica-Verlag Heidelberg, 2009.
9. *Nickel S., Puerto J.* Location Theory. Heidelberg: Springer-Verlag, 2005.

10. *Забудский Г.Г., Коваль А.А.* Размещение объектов на плоскости с максиминным критерием и минимально допустимыми расстояниями // XIV Всеросс. конф. “Математическое программирование и приложения”. Тез. докл. Екатеринбург: ИММ УрО РАН, 2011. С. 88.
11. *Забудский Г.Г., Мархоцкая Н.В.* Решение максиминной задачи размещения на плоскости с минимально допустимыми расстояниями // XIV Байкальская межд. шк.-семинар “Методы оптимизации и их приложения”. Тр. шк.-семинара. Иркутск: ИСЭ СО РАН, 2008. Т. 1. С. 380–387.
12. *Brimberg J., Mehrez A.* Multi-Facility Location Using a Maximin Criterion and Rectangular Distances // Location Science. 1994. V. 2. No. 1. P. 11–19.
13. *Katz M.J., Kedem K., Segal M.* Improved Algorithms for Placing Undesirable Facilities // Computers & Operations Research. 2002. No. 29. P. 1859–1872.
14. *Tamir A.* Locating Two Obnoxious Facilities Using the Weighted Maximin Criterion // Operations Research Lett. 2006. No. 34. P. 97–105.
15. *Левин Г.М., Танаев В.С.* Декомпозиционные методы оптимизации проектных решений. Минск: Наука и техника, 1978.
16. *Танаев В.С.* Декомпозиция и агрегирование в задачах математического программирования. Минск: Наука и техника, 1987.
17. *Препарата Ф., Шеймос М.* Вычислительная геометрия. Введение. М.: Мир, 1988.
18. *Dearing P.M., Francis R.L.* A Network Flow Solution to a Multifacility Minimax Location Problem Involving Rectilinear Distances // Transportational Science. 1974. V. 8. P. 126–141.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 16.11.2013