Геннадий Григорьевич ЗАБУДСКИЙ — старший научный сотрудник лаборатории дискретной оптимизации Омского филиала Института математики СО РАН, кандидат физико-математических наук, доцент

УДК 519.854

*РЕШЕНИЕ ЗАДАЧИ ВЕБЕРА НА ПЛОСКОСТИ С ЗАПРЕЩЕННЫМИ ЗОНАМИ**

АННОТАЦИЯ. Рассматривается задача Вебера на плоскости с прямоугольными запрещенными зонами. Расстояния измеряются в прямоугольной метрике. Построена модель целочисленного линейного программирования задачи и предложены алгоритмы ее решения. Проведен вычислительный эксперимент.

Weber Problem on a plane with rectangular forbidden areas is considered. Distances are measured by rectilinear metric. Model of integer linear programming is constructed and algorithms for solving the problem are proposed. Computational experiment is presented.

Введение

Задача Вебера относится к классу задач оптимального размещения. Такие задачи актуальны и часто возникают при проектировании, например, генеральных планов предприятий, либо их реконструкции. Выделена площадка под строительство предприятия, определено оборудование, связанное между собой различными коммуникациями, например, трубопроводами. На площадке возможны участки, запрещенные для установки оборудования (запрещенные зоны). Запрещенные зоны могут быть естественными (озера, овраги, горы и т. д.), а также искусственными (здания, оборудование и т. д.). Необходимо составить проект размещения оборудования на площадке с учетом запрещенных зон с минимальной суммарной стоимостью связывающей сети (суммарная стоимость трубопроводов).

Для решения сформулированной задачи предложена процедура построения модели целочисленного линейного программирования. Разработаны эвристический алгоритм и алгоритм ветвей и границ. Проведен вычислительный эксперимент.

1. Постановка задачи

Пусть $P_1, P_2, ..., P_m$ — множество фиксированных объектов на плоскости, $M=\{1, 2, ..., m\}$ — множество их номеров и (p_{1i}, p_{2i}), $i \in M$ — их координаты. Требуется разместить объекты $X_1, X_2, ..., X_n$ на той же плоскости, $N=\{1, 2, ..., n\}$ — множество номеров объектов и (x_j, y_j), $j \in N$ — координаты. Удельные стоимости связей между i-м фиксированным и j-м новым объектами обозначим через w_{ij}, $i \in M$, $j \in N$, а v_{ik} — удельные стоимости связей между i-м и k-м новыми объектами $i, k \in N$, $i<k$. Кроме того, задано z прямоугольных запрещенных зон $F_1, F_2, ..., F_z$ со сторонами параллельными осям координат (изотетичные прямоугольники), $Z=\{1, 2, ..., z\}$ — множество номеров зон. Пусть $F=\cup_{i \in Z} F_i$, $d(P_i, X_j)$, — расстояние между i-м фиксированным и j-м новым объектами, а $d(X_i, X_k)$ — расстояние между i-м и k-м новыми объектами. Требуется разместить объекты

* Работа выполнена при поддержке гранта РГНФ №04-02-00238а.

$X_1, X_2, ..., X_n$ вне запрещенных зон $F_1, F_2, ..., F_z$ так, чтобы суммарная стоимость связей между всеми объектами была минимальной, т.е.

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} d(P_i, X_j) + \sum_{i=1}^{n-1}\sum_{k=i+1}^{n} v_{ik} d(X_i, X_k) \rightarrow \min \quad (1.1)$$

$$X_i \notin Z, \quad i \in N \quad (1.2)$$

В работе рассматриваем случай, когда $d(\cdot,\cdot)$ прямоугольная метрика, т.е. L_1. Аналогичная задача с минимаксным критерием и $n=1$ рассматривалась в [1].

Задача (1.1) без условия (1.2) может быть сформулирована как две независимые одинаковые задачи линейного программирования (ЛП) с помощью введения переменных $s=(s_{ij})$, $l=(l_{ij})$, $i\in M$, $j\in N$, $t=(t_{jk})$, $u=(u_{ik})$, j, $k\in N$, $j<k$ [4].

Первая задача:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} s_{ij} + \sum_{j=1}^{n-1}\sum_{k=i+1}^{n} v_{jk} t_{jk} \rightarrow \min \quad (1.3)$$

$$-s_{ij} \le x_j - p_{1i} \le s_{ij},\ i\in M, j\in N \quad (1.4)$$

$$-t_{jk} \le x_j - x_k \le t_{jk},\ j,k\in N,\ j<k \quad (1.5)$$

Вторая задача:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} l_{ij} + \sum_{j=1}^{n-1}\sum_{k=i+1}^{n} v_{jk} u_{jk} \rightarrow \min \quad (1.6)$$

$$-l_{ij} \le y_j - p_{1i} \le l_{ij},\ j\in N, i\in M \quad (1.7)$$

$$-u_{jk} \le y_j - y_k \le u_{jk},\ j,k\in N,\ j<k \quad (1.8)$$

С учетом специфики указанных задач ЛП предложены более эффективные алгоритмы их решения [4, 5].

2. Построение целочисленной модели

Построим модель целочисленного линейного программирования задачи (1.1)-(1.3). С помощью целочисленных переменных учитывается, что допустимая область указанной задачи является невыпуклой и может быть несвязной. Дополнение к представим в виде объединения прямоугольных областей $R_1, R_2, ..., R_g$ (разрешенных зон), $G=\{1, 2, ..., g\}$ — множество их номеров. Зона R_k ограничена прямоугольником $[(a_k, c_k); (b_k, d_k)]$, где (a_k, c_k) — координаты левого нижнего угла, а (b_k, d_k) — правого верхнего. Аналогично будем обозначать F_k прямоугольником $\left[(a_k', c_k'); (b_k', d_k')\right]$. Для того чтобы построить модель, необходимо определить контур (границу) области F. Указанный контур находится с помощью метода плоского заметания и представляется в виде чередующихся вертикальных и горизонтальных ребер [3].

Алгоритм построения разрешенных зон

Ограничим область F прямоугольником $[(A, C); (B, D)]$, где $A=\min_{k\in Z} a_k'$ $C=\min_{k\in Z} c_k'$, $B=\max_{k\in Z} b_k'$, $D=\max_{k\in Z} d_k'$. Тогда выделяются четыре неограниченные разрешенные зоны: $x_i\le A$, $x_i\ge B$, $y_i\le C$, $y_i\ge D$, $i\in N$.

Пусть контур F построен. Будем определять разрешенные зоны внутри прямоугольника $[(A, C); (B, D)]$ «двигаясь» вдоль оси OY. Упорядочим в порядке возрастания координаты вертикальных ребер контура по оси OX: $A = x_1 < x_2 < \cdots < x_t = B$. Строим список точек пересечения прямой $x = x_2$ с горизонтальными ребрами. Упорядочиваем его по возрастанию координаты y: $(x_2, y_1), (x_2, y_2), \ldots, (x_2, y_s)$, где $y_1 = C$, $y_s = D$.

Проверяем, принадлежит или нет точка границе запрещенной зоны. Далее перебирая по порядку все точки из списка, начиная с (x_2, y_2), проверяем, принадлежит точка правой границе запрещенной зоны или правой границе разрешенной зоны. Признаком выхода из любой зоны служит точка, не являющаяся началом какого-либо горизонтального отрезка, либо точка, являющаяся одновременно концом одного горизонтального отрезка и началом другого, либо точка (x_2, y_s). Перебрав все точки из списка $(x_2, y_1), (x_2, y_2), \ldots, (x_2, y_s)$ и сформировав разрешенные зоны в полосе $[x_1, x_2]$, переходим к координате x_3 и повторяем процесс.

Алгоритм построения по шагам

Пусть k — счетчик числа разрешенных зон, i — номер очередной координаты по оси OX, j — счетчик очередной координаты по оси OY. Логическая переменная *flag* принимает значение *true*, если «движемся» по правой границе разрешенной зоны, и значение *false*, если «движемся» по правой границе запрещенной зоны.

Шаг 1. Упорядочим список вертикальных ребер по возрастанию координаты x, а список горизонтальных ребер по возрастанию координаты y. Полагаем $i{:=}2$, $j{:=}1$, $k{:=}4$ (т. к. у нас есть уже четыре разрешенных зоны). На Шаг 2.

Шаг 2. Находим список точек пересечения прямой $x = x_i$ с горизонтальными ребрами и прямыми $y = C$, $y = D$. Упорядочиваем его по возрастанию координаты y. На Шаг 3.

Шаг 3. Если точка (x_i, C) не является началом какого-либо горизонтального ребра, но принадлежит ему, то $flag := false$, $j := 2$. На Шаг 4.

Если точка (x_i, C) является началом какого-либо горизонтального ребра, либо не принадлежит никакому ребру, то $flag := true$, $k := k + 1$, $a_k := x_{i-1}$, $b_k := x_i$, $c_k := C$, $j := 2$. На Шаг 4.

Шаг 4.

4.1. Если ($y_j = D$ и $flag := false$), тогда если $x_i = B$, то **Стоп**. Иначе ($x_i \neq B$), $i := i + 1$, $j{:=}1$. На Шаг 2.

4.2. Если ($y_j = D$ и $flag := true$), то $d_k := y_j$. Включаем зону $[(a_k, c_k); (b_k, d_k)]$ в список разрешенных зон. Если $x_i = B$, то **Стоп**. Иначе ($x_i \neq B$) $i{:=}i{+}1$, $j := 1$. На Шаг 2.

4.3. Если $y_j \neq D$ и (x_i, y_j) является началом и не является концом горизонтального ребра, то $j := j + 1$. На Шаг 4.

4.4. Если $y_j \neq D$ и (x_i, y_j) является концом либо принадлежит горизонтальному ребру, то

•Если $flag = true$, то $d_k := y_j$, $flag := false$, $j := j + 1$. На Шаг 4.

•Если $flag = false$, то $k := k + 1$, $a_k := x_{i-1}$, $b_k := x_i$, $c_k := y_j$, $j{:=}j{+}1$, $flag{:=}true$. На Шаг 4.

Для уменьшения числа разрешенных зон некоторые из них можно объединить. Зоны R_k и R_j можно объединить если $b_k = a_j$, $d_k = d_j$, $c_k = c_j$ или $b_j = a_k$, $d_j = d_k$, $c_j = c_k$.

Запишем модель ЦЛП. Не ограничивая общности, можно считать, что все фиксированные объекты и запрещенные зоны находятся в первой четверти. Это можно сделать с помощью параллельного сдвига. Пусть $k_1 = \max_{i \in M, k \in G}\{p_{1i}, b_k\}$, $k_2 = \max_{i \in M, k \in G}\{p_{2i}, d_k\}$. Нетрудно показать, что оптимальное решение исходной задачи находится в прямоугольнике $[(0, 0); (k_1, k_2)]$. Условие принадлежности нового объекта i только одной из разрешенных зон записывается с помощью булевых переменных h_{ik}, $i \in N$, $k \in G$, $h_{ik} = 1$, если объект i находится в k-й разрешенной зоне, иначе $h_{ik}=0$. Тогда условия принадлежности каждого объекта $i\in N$ только одной из разрешенных зон $k\in G$ имеют вид:

$$\left.\begin{array}{l} x_i-a_k+a_k(1-h_{ik})\ge 0\\ y_i-c_k+c_k(1-h_{ik})\ge 0\\ -x_i+k_1-h_{ik}(k_1-b_k)\ge 0\\ -y_i+k_2-h_{ik}(k_2-d_k)\ge 0\\ \sum\limits_{k=1}^{g}h_{ik}=1\\ h_{ik}\in\{0,1\}\end{array}\right\}\quad i\in N,\ k\in G \tag{2.1}$$

Объединяя целевые функции (1.3) и (1.6) и записывая ограничения (1.4), (1.5), (1.7), (1.8), (2.1) получаем модель ЦЛП для задачи (1.1), (1.2).

3. Алгоритмы решения

3.1 Размещение одного объекта

В случае, когда *n=1,* нет необходимости использовать модель, построенную в предыдущем пункте, а можно предложить эффективный алгоритм. Известен алгоритм решения задачи (1.1) для *n=1*. Оптимальное решение находится в точках пересечения прямых, параллельных осям координат и проходящих через фиксированные объекты [2].

Так как *n=1*, то второй индекс у удельных стоимостей связей и номер размещаемого объекта опустим. Проведем через каждую точку P_i вертикальную c_i и горизонтальную d_i линии $i\in M$. Пронумеруем линии слева направо от 1 до *r*, а линии снизу вверх от 1 до *q*. Считаем, что слева от вертикальной линии 1 имеется линия 0, а справа от вертикальной линии *r* — линия *r + 1*, ниже горизонтальной линии 1 — линия 0 и выше линии *q* — линия *q + 1*. Обозначим через $[i, j]$ область, ограниченную *j*-й и (*j+1*)-й вертикальными линиями и *i*-й и (*i+1*)-й горизонтальными линиями, F' контур области F и $F^{\wedge}$— множество угловых точек F'. Справедливо

Утверждение. Существует оптимальное решение X^* такое, что $X^*\in(c_i\bigcap d_j)$ $\bigcup\ (c_i\bigcap F')\ \ (d_j\bigcap F')\ \bigcup\ F^{\wedge},\ i=1,2,\ldots,r;\ j=1,2,\ldots,q$.

Справедливость утверждения следует из свойств кусочно-линейной функции (1.1) для метрики L_I.

Опишем алгоритм решения задачи (1.1), (1.2) по шагам.

Шаг 1. Решаем задачу (1.1).

1.1. Определяем $D_i=\sum_{k:p_{2k}=d_i}w_k,\ i=1,2,\ldots,q;\ C_j=\sum_{k:p_{1k}=c_j}w_k,\ j=1,2,\ldots,r.$

1.2. Вычисляем величины:

$$M_0=-\sum_{j=1}^{r}C_j=\sum_{j=1}^{m}w_j \qquad N_0=-\sum_{j=1}^{q}D_j=\sum_{j=1}^{m}w_j$$

$$M_1=M_0+2C_1 \qquad N_1=N_0+2D_1$$

$$\ldots$$

$$M_r=M_{r-1}+2C_r=\sum_{i=1}^{m}w_i \qquad N_q=N_{q-1}+2D_q=\sum_{i=1}^{m}w_i$$

1.3. Определяем координаты оптимальной точки $X^* = (x^*, y^*)$. Рассмотрим четыре случая.

а) Если $M_{t-1} < 0, M_t > 0, N_{s-1} < 0, N_s > 0$, то $x^* = c_t, y^* = d_s$.

б) Если $M_{t-1} < 0, M_t = 0, N_{s-1} < 0, N_s > 0$, то x^* любая точка интервала $[c_t, c_{t+1}]$, а $[d_s, d_{s+1}]$.

в) Если $M_{t-1} < 0, M_t > 0, N_{s-1} < 0, N_s = 0$, то $x^* = c_t$, y^* любая точка интервала $[d_s, d_{s+1}]$.

г) Если $M_{t-1} < 0, M_t = 0, N_{s-1} < 0, N_s = 0$, то x^* любая точка интервала $[c_t, c_{t+1}]$, а y^* любая точка интервала $[d_s, d_{s+1}]$. На Шаг 2.

Шаг 2. Если $(x^*, y^*) \notin F$, то **Стоп**. Иначе на Шаг 3.

Шаг 3. Строим контур F'. На Шаг 4.

Шаг 4. Находим угловые точки $F^{\wedge}$ контура F' и точки пересечения F' с $c_j, j = 1, 2, ..., r$; $d_j, j = 1, 2, ..., q$. Вычисляем значение функции (1.1) в найденных точках и выбираем минимальное. **Стоп**.

3.2. Размещение произвольного числа объектов

Алгоритм ветвей и границ (АВГ). Схема алгоритма состоит в следующем. Сначала находим решение задачи (1.1). Затем проверяем, попадают или нет новые объекты в область F. Если нет, то задача решена. В противном случае начинаем ветвление. Ветвление осуществляется перебором вариантов расположения каждого нового объекта в каждой разрешенной зоне, т.е. вместо ограничений (2.1) полагаем $h_{ik} = 1, k \in G$ и в модель добавляются ограничения $x_i \ge a_k, x_i \le b_k, y_i \ge c_k, y_i \le d_k$. Нижние оценки целевой функции находятся с помощью решения задачи ЛП без учета запрещенных зон.

Эвристический алгоритм (ЭА).

Шаг 1. (Построение начального размещения). Фиксируем порядок объектов. Для очередного размещаемого объекта перебираем разрешенные зоны множества G и применяем алгоритм п. 3.1. Находим оптимальное расположение и фиксируем его. После просмотра всех объектов переходим на Шаг 2.

Шаг 2. (Корректировка размещения). В заданном порядке по одному «освобождаем» объекты, другие считаем фиксированными и алгоритмом п. 3.1. находим его оптимальное размещение.

Алгоритм останавливается, если рекордное значение после просмотра всех объектов на Шаге 2 не уменьшается.

4. Результаты вычислительного эксперимента

Эксперимент проводился на компьютере со следующими техническими характеристиками Intel Pentium 4 2.8GHz HT RAM DDR3200 512Mb dual. Решено более 300 задач. Данные генерировались случайным образом. Количество фиксированных объектов изменялось от 2 до 27, а количество новых объектов — от 1 до 22. Отклонение значения целевой функции, полученного с помощью эвристического алгоритма, от оптимального в среднем составило 1%. Частично результаты экспериментов представлены в табл. 4.1.

Выводы. В результате проведенных исследований можно сделать следующие выводы. Предложенный подход для решения задач размещения с невыпуклыми, несвязными областями является перспективным. Построение целочисленной модели задачи позволяет применять достаточно разработанный аппарат целочисленного линейного программирования (методы отсечения, ветвей и границ и другие) для получения точных решений. Кроме того, эвристический («жадный») алгоритм дает достаточно хорошие приближения к точному решению и может применяться для практических задач большой размерности.

Таблица 4.1

Результаты экспериментов

№	m	n	Оптим. решение	Решение **АВГ**	Время **АВГ**	Решение **ЭА**	Время **ЭА**
1	15	5	198030	198429	0:00:06	198429	0:00:00
2	8	6	104353	105625	0:00:03	105709	0:00:00
3	10	7	118861	118861	0:00:00	120094	0:00:00
4	13	7	201169	205381	0:00:32	205381	0:00:00
5	10	8	165587	166959	0:00:06	167036	0:00:00
6	25	8	60468	60473	0:04:25	61017	0:00:02
7	27	9	48850	48960	0:14:39	49081	0:00:05
8	7	12	183612	186306	0:00:09	186306	0:00:00
9	6	13	152407	153536	0:00:00	162422	0:00:00
10	8	13	225974	228058	0:00:13	228058	0:00:00
11	2	15	8008	8683	0:00:00	9493	0:00:00
12	6	15	272012	272019	0:00:09	272059	0:00:00
13	4	17	89117	89571	0:02:16	90072	0:00:00
14	8	17	267915	268023	0:07:12	272807	0:00:00
15	7	22	428010	429099	0:38:01	433963	0:00:02
16	3	22	79982	84310	0:03:03	85587	0:00:01

СПИСОК ЛИТЕРАТУРЫ

1. Забудский Г. Г. Алгоритм решения минимаксной задачи размещения на плоскости с запрещенными зонами // Автоматика и телемеханика. 2004. № 4. С. 93-100.

2. Исследование операций. Модели и их применение / Под ред. Дж. Моудера, С. Элмаграби. М.: Мир, 1981. 677 с.

3. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. М.: Мир, 1989. 478 с.

4. Трубин В. А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой // Кибернетика. 1978. № 6. С. 67-70.

5. Picard, J. C., Ratliff, D. H. A cut approach to the rectilinear distance facility location problem // Oper. Res. 1978. V. 26. № 3. P. 422-433.