объединенный
институт
ядерных
исследований
дубна

Р11 - 12405

Н.И.Янев

О РЕШЕНИИ ПРОСТОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ

1979

Р11 - 12405

Н.И.Янев

О РЕШЕНИИ ПРОСТОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ

Направлено в "Журнал вычислительной математики и математической физики"

Янев Н.И. P11 - 12405

О решении простой задачи размещения

Исследованы особенности распределения возможных решений простой задачи размещения в множестве вершин выпуклого многогранника соответствующей непрерывной задачи. Для решения задачи предложен алгоритм типа модифицированного симплексного алгоритма с компактной формой хранения обратной базисной матрицы. Показаны результаты выполнения одной программной реализации алгоритма на ЭВМ CDC-6500.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1979

Janev N.I. P11 - 12405

On a Simple Plant-Location Problem Solution

Distribution peculiarities of a simple plant-location problem feasible solutions into a set of convex polyhedra vertexes corresponding to a continuous problem are investigated. Suggested is a revised simplex type algorithm with a compact form of basis inverse. The results of CDC-6500 computer runs are shown.

The investigation has been performed at the Laboratory of Computing Techniques and Automations, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna 1979

В литературе хорошо известна следующая задача о размещении ("о размещении производства", "plant-location problem" и т.д.): из заданного множества возможных мест размещения "производства" найти подмножество, для которого минимизируется сумма расходов на производство и транспортировку продукции к заданным потребителям. Термин "простой" (употребляется также термин "неограниченные мощности") означает, что каждое производство может удовлетворить всех потребителей. В дальнейшем эту задачу будем называть задачей размещения.

Для решения задачи размещения предложено несколько алгоритмов направленного перебора, вычислительная эффективность которых подробно исследована в /1/. В /2/ предложен специальный метод для вычисления значения множителей Лагранжа применительно к данной задаче без решения задач линейного программирования. Интересное исследование проведено в /3/, где задача размещения с квадратной матрицей расходов получается как эквивалентное представление однопараметрической задачи унификации. Там доказано, что при выполнении некоторых весьма сильных ограничений задача линейного программирования, которая соответствует задаче размещения, имеет целочисленное опорное решение.

В данной работе исследуются свойства множества решений задачи размещения и доказывается, что в определенном смысле ее всегда можно решать как задачу линейного программирования. Практическая применимость полученных результатов подкреплена алгоритмом решения задачи и результатами его машинной реализации.

I. Некоторые свойства задачи размещения

Задача размещения рассматривается в следующей постановке:

$$L(X) \equiv \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} c_i x_i \longrightarrow \min , \qquad (I.I)$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = \overline{1,n} , \qquad (I.2)$$

$$0 \le x_{ij} \le x_i , \quad j = \overline{1,n} , \ i = \overline{1,m} , \qquad (I.3)$$

$$x_i \in \{0,1\} , \quad i = \overline{1,m} . \qquad (I.4)$$

Для компактности изложения введем следующие обозначения:

$M = \{1,2,,,m\}$, $N = \{1,2,\dots,n\}$, $MN = M \times N = \{(i,j)/i \in M,\ j \in N\}$,

$X' = (x_{11}, x_{12}, \dots x_{1n}, x_{21}, \dots, x_{mn}, x_1, \dots, x_m)$, $b' = (1,1,..1,0,..0)$.

Тогда условия (I.2), (I.3) можем записать матрично, как $AX \gtrless b$, где равенства соответствуют единичным компонентам вектора b , а матрица A имеет вид $A = \begin{bmatrix} B & C \\ E & D \end{bmatrix}$. Здесь E – единичная матрица, C – нулевая $n \times m$ матрица, а $B(n \times mn)$ и $D(mn \times m)$ являются матрицами специального вида:

$$B = \begin{bmatrix} 1 & & & 1 & & & \dots & 1 & & \\ & 1 & & & 1 & & & & 1 & \\ & & \ddots & & & \ddots & & & & \ddots \\ & & & 1 & & & 1 & & & & 1 \end{bmatrix} \quad D = \begin{bmatrix} -1 & & & \\ -1 & & & \\ \vdots & & & \\ -1 & & & \\ & -1 & & \\ & -1 & & \\ & \vdots & & \\ & -1 & & \\ & & \dots & -1 \\ & & & -1 \\ & & & \vdots \\ & & & -1 \end{bmatrix} .$$

В приводимых ниже доказательствах существенную роль играет следующая теорема, доказанная в эквивалентной формулировке в /4/.

Пусть F – матрица, элементы которой равняются 0 или ± 1.

Теорема I. Матрица F вполне унимодулярна , если:

1. Каждая строка содержит не более чем два ненулевых элемента.
2. Множество столбцов можно разбить на подмножества S_1 и S_2 так, что:

а) если строка содержит два ненулевых элемента с одинаковыми знаками, то столбцы, соответствующие этим элементам, принадлежат разным подмножествам;

б) если строка содержит два противоположных по знаку элемента, то соответствующие столбцы принадлежат одному и тому же подмножеству.

Напомним, что вполне унимодулярная матрица определяется как матрица, все миноры которой равняются 0 или ± 1 .

Обозначим через C множество всех $mn+n$ -мерных векторов, удовлетворяющих (I.2),(I.3), и пусть $C^0 = \{X \in C/X$ –бинарный вектор$\}$. Известно, что все точки C^0 являются крайними для многогранника C , но в общем случае ($m > 2$) C^0 есть собственное подмножество множества крайних точек C, и поэтому не всегда решение задачи размещения можно получить как решение задачи линейного программирования (I.I)–(I.3). Ниже доказываются некоторые свойства распределения точек C^0 , имеющие прямое отношение к возможности применения линейного программирования для решения задачи размещения.

Пусть $X^1 = (\dots x^1_{ij} \dots, x^1_i)$ и $X^2 = (\dots x^2_{ij} \dots, \dots x^2_i \dots)$ принадлежат C^0 и существуют $i \in M$, $j \in N$ такие, что $x^1_{ij} = x^2_{ij}$. Тогда множество точек, удовлетворяющих системе условий (I.2), (I.3), и уравнения:

$$x_{ij} = 0 \quad \text{для всех } i,j \text{ таких, что } x^1_{ij} = x^2_{ij} = 0, \qquad (I.5)$$

$$x_{ij} = 1 \quad \text{для всех } i,j \text{ таких, что } x^1_{ij} = x^2_{ij} = 1, \qquad (I.6)$$

$$x_i = 1 \quad \text{для всех } i \text{ таких, что } x^1_i = x^2_i = 1, \qquad (I.7)$$

по определению являются некоторой гранью Γ множества C , содержащей точки X^1 и X^2 . Запишем систему условий, определяющих Γ в виде

$$\Gamma = \{ X/A_1 X \gtrless b_1 ,\ X \geqslant 0 \} , \qquad (I.8)$$

где $A_1 = \begin{bmatrix} B_1 & C_1 \\ E_1 & D_1 \end{bmatrix}$ и b_1 получаются из A и b зачеркиванием столбцов, соответствующих уравнениям (I.5), и строк, являющихся тождеством или избыточным неравенством $x_{ij} \leqslant 1$ (следствие уравнений (I.6) или неравенства $x_i \geqslant 0$ (следствие из уравнений (I.5)). Пусть $MN_1 = M_1 \times N_1 \subset MN$, $M_1 \subset M, N_1 \subset N$ – соответствующие множества индексов столбцов матрицы A_1 .

Лемма I. Матрица A_1 - вполне унимодулярна.

Доказательство. Поскольку унимодулярность сохраняется при добавлении единичных столбцов (строк), то утверждение леммы достаточно доказать для матрицы $A_1 = \left[\begin{array}{c:c} B_1' & C_1' \\ \hdashline E_1' & D_1' \end{array}\right]$, которая получается из A_1 зачеркиванием всех единичных столбцов (строк). Обозначим через $MN_1' = M_1' \times N_1' \subset MN_1$ множество пар (i, j) соответствующих индексов столбцов матрицы $\left[\begin{array}{c} B_1' \\ \hdashline E_1' \end{array}\right]$. Пусть $T_i = \{i\} \times \{j / (i,j) \in MN_1'\}$, $i \in M_1'$. Отметим, что по способу получения каждая строка матрицы A_1' содержит ровно по два единичных элемента и для всех i, участвующих в условиях (I.7), выполняется $i \bar{\in} M_1'$.

Построим граф $K=(V,R)$, где вершины $V_i \in V$ соответствуют множествам T_i, а множество ребер R содержит только те пары (i,j), $i \in V$, $j \in V$, для которых существует $k \in N_1'$ такое, что $(i,k) \in T_i$, $(j,k) \in T_j$.

Покажем, что граф K - двудольный. Для этого достаточно показать, что он не содержит нечетные циклы (циклы с нечетным числом ребер). Допустим противное, т.е., что в графе K существует цикл $(i_1, i_2, \ldots, i_k, i_1)$ и $k = 2t + 1$, $t > 0$. Распишем ограничения системы $A_1 X \geq b_1'$, соответствующие этому циклу:

$$x_{i_1 1} + x_{i_2 1} = 1,$$
$$x_{i_2 2} + x_{i_3 2} = 1,$$
$$\ldots\ldots\ldots\ldots$$
$$x_{i_k k} + x_{i_1 k} = 1, \tag{I.9}$$
$$x_{i_p j} \leq x_{i_p}, \quad p = \overline{1,k}, \quad j = \overline{1,k}.$$

Здесь вторые индексы переменных для удобства обозначения взяты из множества $\{1,2,\ldots k\}$. Поскольку для любого i_p переменные с первым индексом i_p участвуют ровно в двух уравнениях, то любое (0,I)- решение системы (I.9) должно иметь по крайней мере $t+1$ (из $2t+1$) переменных $x_{i_p} = 1$. Так как X_1 и X_2 являются решением системы (I.9), то отсюда следует, что существует i_p, для которого $x^1_{i_p} = x^2_{i_p} = 1$, т.е. $i_p \bar{\in} M_1$, что эквивалентно $i_p \bar{\in} V$. Полученное противоречие доказывает, что граф K двудольный.

По определению двудольного графа существуют множества S_1 и S_2 такие, что: $S_1 \cup S_2 = V$; $S_1 \cap S_2 = \Phi$ и каждое ребро инцидентно одной вершине из S_1 и одной - из S_2. Очевидно, что S_1 и S_2 (если их интерпретировать как множества столбцов матрицы A_1') удовлетворяют условиям теоремы I, откуда следует унимодулярность матрицы A_1' и соответственно A_1.

<u>Теорема 2</u>. Любые $X^1 \in C^o$ и $X^2 \in C^o$, $X^1 \neq X^2$ принадлежат некоторой k-мерной грани Γ множества C, все крайние точки которой принадлежат C^o.

Доказательство.

Для $m=2$ множество C^o совпадает с множеством крайних точек C, поэтому будем считать, что $m > 2$. Покажем, что при $m > 2$ для любых X^1 и X^2 существуют $i \in M, j \in N$ такие, что $x^1_{ij} = x^2_{ij}$. Допустим противное, т.е., что $x^1_{ij} + x^2_{ij} = 1$ для всех i,j. Тогда из выпуклости C следует, что точка $\frac{1}{2}(X^1 + X^2)$ удовлетворяет уравнению (I.2), откуда получаем противоречие: $m=2$.

Грань Γ, задаваемая условием (I.8), содержит точки X^1 и X^2 и согласно лемме I имеет вполне унимодулярную матрицу условия A_1. Из этого следует, что все базисные решения системы (I.8) целочисленные, что и требовалось доказать.

Для $X \in C^o$ обозначим через $M(X)$ множество всех точек $Y \in C^o$ таких, что X и Y инцидентны некоторой одномерной грани множества C.

<u>Теорема 3</u>. Точка X^o является решением задачи размещения тогда и только тогда, когда $L(X^o) = \min\limits_{X \in M(X^o)} L(X)$.

Доказательство.

($\Longrightarrow$) Очевидно.

($\Longleftarrow$) Допустим, что условие теоремы выполнено и существует точка $Y \in C^o$, для которой $L(Y) < L(X^o)$. Тогда из теоремы 2 следует, что существует грань Γ, содержащая крайние точки X^o и Y, все крайние точки которой принадлежат C^o. Условие $L(X^o) = \min\limits_{X \in M(X^o)} L(X)$ является достаточным для оптимальности X^o в задаче линейного программирования $\min\limits_{X \in \Gamma} L(X)$. Отсюда следует, что $L(X^o) \leq L(Y)$, а это противоречит допущению $L(Y) < L(X^o)$.

Из теоремы 3 следует, что для решения задачи размещения может быть использована любая версия прямого симплексного алгоритма при условии, что все базисные решения, получаемые в ходе выполнения алгоритма, принадлежат множеству C^{o} . Соблюдение этого условия осуществляется простой модификацией правила выбора переменной, которая исключается из базиса (см. п.2). Более существенным является вопрос об эффективности алгоритма. Дело в том, что в задаче размещения обратная матрица базиса имеет большие размеры ($mn+ n \times mn+n$), что даже при небольших m и n может сильно увеличить время решения задачи, если в алгоритме не предусмотрены возможности для минимизации числа арифметических операций.

В следующем разделе описывается алгоритм решения задачи размещения, отличающийся большой экономностью требуемой памяти и числа операций, что позволяет эффективно решать (только в оперативной памяти) задачи, для которых число переменных x_{ij} порядка 10000.

2. Алгоритм решения задачи размещения

В /5/ доказано, что любая базисная матрица B для задач линейного программирования, в условиях которой имеются ограничения типа (I.3), представима в виде $B=\begin{bmatrix} C & \vdots & D \\ E & \vdots & I \end{bmatrix}$, где число строк единичной матрицы I равно числу ограничений (I.3) (для задачи размещения это число равняется mn). Тогда обратную матрицу B^{-1} можно представить как

$$B^{-1}= \begin{bmatrix} (C-DE)^{-1} & \vdots & (C-DE)^{-1}(-D) \\ \cdots\cdots & \cdots & \cdots\cdots \\ -E(C-DE)^{-1} & \vdots & -E(C-DE)^{-1}(-D)+I \end{bmatrix}. \qquad (2.I)$$

Из (2.I) видно, что для вычисления B^{-1} достаточно хранить только матрицу $(C-DE)^{-1}$, размерность которой $n \times n$. Здесь не будем приводить подробное описание алгоритма, поскольку он почти аналогичен модифицированному симплексному методу, который хорошо известен в литературе. Для того чтобы модифицированный симплексный алгоритм мог быть применен к решению задачи размещения, необходимо изменить правило выбора переменной, которая выключается из базиса и критерия оптимальности. С целью сохранения целочисленности опорных планов j -я внебазисная переменная, которая имеет отрицательный коэффициент Δ_j в целевой функции, может быть включена в базис, если существует r такое, что $\min\limits_{\alpha_{ij}>0} \frac{\alpha_{io}}{\alpha_{ij}}$ достигается для $i=r$ и $\alpha_{rj}=1$ (α_{ij} , α_{io} - коэффициенты разложения

j -го столбца матрицы A (b) по базисным векторам). Если r существует, из базиса исключается переменная, находящаяся на r -ом месте. Если такого r нет, то движение по j -ой координате в пространстве внебазисных переменных выводит из множества C^{o} , поэтому j -ая переменная выключается из множества кандидатов ($K=\{j/\Delta_j < 0\}$) на включение в текущий базис. Согласно теореме 3 критерий оптимальности данного опорного плана состоит в проверке условий:

I) $K=\Phi$ или 2) для каждого $j \in K$ выражение $\frac{\alpha_{io}}{\alpha_{ij}}$ достигает своего минимума на множестве $\alpha_{ij}>0$ для $\alpha_{rj}>1$.

Приведем формулы, задающие основные вычислительные операции для заданной итерации, обусловленные особенностями хранения обратной матрицы (2.I) и спецификой матрицы A . Для этого приведем задачу (I.I)-(I.4) в каноническом виде, добавляя дополнительные переменные $s_{ij} \geq 0$ в условиях (I.3), и пронумеруем переменные задачи размещения при помощи некоторого взаимнооднозначного соответствия f , где: $\{s_{11},..s_{mn},x_{11},...,x_{mn},x_1,...x_m\} \xrightarrow{f} \{-1,-2,...-mn, 1,2,..mn,mn+1,..mn+m\}$. Определим функции:

$$\left.\begin{aligned} \varphi(1,k) &= i \\ \varphi(2,k) &= j \end{aligned}\right\} , \quad \text{если } f^{-1}(k)=x_{ij} \text{ или } s_{ij}$$

($\varphi(1,k)$ и $\varphi(2,k)$ достаточны для определения столбца, соответствующего переменным x_{ij} или s_{ij}),

$$c(k) = \begin{cases} c_{ij} , & \text{если } f^{-1}(k)=x_{ij} , \\ 0 , & \text{если } f^{-1}(k) = s_{ij}, \\ c_i , & \text{если } f^{-1}(k) = x_i . \end{cases}$$

Номер столбца, который находится на k -ом месте в базисе, будем обозначать через $b(k)$. Пусть $\pi=-c_B B^{-1}$, $n+mn$ -мерный вектор оценок базисных столбцов, первые n -компонентов которого будем считать известными (обычно π включается как нулевая строка матрицы B^{-1} , и поэтому его надо задавать только для начального опорного плана). Далее для удобства записи будем отождествлять α_i и $\alpha[i]$, где α и i - любые символы.

а) Оценивание внебазисных столбцов ($\Delta_K = c_K + \pi a_K$).

Для вхождения в базис выбираются столбцы, для которых $\Delta_K<0$, где

$$\Delta_k = \begin{cases} c(k) + \pi[\varphi(2,k) + \pi[(\varphi(1,k)-1)n + \varphi(2,k)], & 1\le k\le mn,\\ \pi[(\varphi(1,k)-1)n + \varphi(2,k)], & k<0,\\ c(k) - \sum\limits_{j=1}^{n} \pi[(k-mn)n+j], & k>mn,\end{cases} \qquad (2.2)$$

где для $i > n$:

$$\pi[i] = \begin{cases} -c(b(i)) - \pi[\varphi(2,b(i))], & \text{если } f^{-1}(b(i)) = x_{ij},\\ 0, & \text{если } f^{-1}(b(i)) = s_{ij}. \end{cases}$$

б) <u>Разложение внебазисного столбца a_k по векторам базиса</u> ($\alpha = B^{-1}a_k$).

Пусть $\beta[i,j]$ - (i,j) -ый элемент матрицы $(C-DE)^{-1}$ и для $k>mn$ $S_k = \{ j/f^{-1}(\ell) = x_j,\ \ell = \overline{(k-mn)n+1,(k-mn)n+n}$. Положим $\alpha[i]=0$, $i=mn+1, mn+n$.

<u>Вычисление первых n компонентов вектора α</u>

Если $1\le k\le mn$, то

$$\begin{aligned} &\alpha[i] = \beta[i, \varphi(2,k)],\ i=\overline{1,n},\\ &\alpha[(\varphi(1,k)-1)n + \varphi(2,k)] = 1; \end{aligned} \qquad (2.3)$$

если $k<0$, то

$$\begin{aligned} &\alpha[i] = -\beta[i, \varphi(2,k)],\ i=1,n,\\ &\alpha[(\varphi(1,k)-1)n + \varphi(2,k)] = 1; \end{aligned} \qquad (2.4)$$

если $k>mn$, то

$$\begin{aligned} &\alpha[i] = \sum_{j\in S_k} \beta[i,j],\ i=\overline{1,n},\\ &\alpha[(k-mn)n+j] = -1,\ j=\overline{1,n}. \end{aligned} \qquad (2.5)$$

<u>Окончательное вычисление α.</u>

Обозначим через α^0 вектор, компоненты которого вычислены по формулам (2.3)-(2.5). Тогда окончательное разложение вектора a_k по векторам базиса задается вектором α^n, который вычисляется по формуле $\alpha^i = g(\alpha^{i-1})$, $i=\overline{1,n}$, или подробнее:

если $f^{-1}(b(i))=x_{ij}$ или s_{ij}, то

$$\alpha^i[(\varphi(1,b(i))-1)n + \varphi(2,b(i))] = \alpha^{i-1}[(\varphi(1,b(i))-1)n + \varphi(2,b(i))] - \alpha[i]; \qquad (2.6)$$

если $f^{-1}(b(i)) = x_i$, то

$$\alpha^i[in+j] = \alpha^{i-1}[in+j] + \alpha[i],\ j = \overline{1,n}. \qquad (2.7)$$

В конкретной программной реализации вычисления по формулам (2.6), (2.7) должны быть сделаны только для $\alpha[i]\ne 0$, $i=\overline{1,n}$, а α^i обозначает только текущее состояние одномерного массива $\alpha[1:mn+n]$.

в) <u>Преобразование матрицы $(C-DE)^{-1}$.</u>

Пусть k -ая переменная входит в базис на p -ое место. Тогда в зависимости от p имеем три возможности:

1) $1\le p\le n$.

В этом случае матрица $(C-DE)^{-1}$, соответствующая новому базису, преобразуется по формуле

$$\beta[i,j] = \beta[i,j] - \alpha[i]\beta[p,j],\quad i=\overline{1,n},\ i\ne p,\ j=\overline{1,n}; \qquad (2.8)$$

2) $p>n$, $b(p)+k=0$.

Пусть вектор $p[i]$, $i=\overline{1,n}$ определен по формуле:

$$p[i] = \begin{cases} 1 & b(i)\le mn \quad (\varphi(1,b(i))n + \varphi(2,b(i)) = p,\\ -1 & b(i)>mn \quad (b(i)-mn)n+1 \le p \le (b(i)-mn)n+n,\\ 0 & \text{в остальных случаях.} \end{cases} \qquad (2.9)$$

Вычисляем вектор $t[i]$, $i=\overline{1,n}$ по формуле

$$t[i] = \sum_{k=1}^{n} p[k]\beta[k,i],\quad i=\overline{1,n}. \qquad (2.10)$$

Формула (2.10) эквивалентна суммированию строк матрицы $(C-DE)^{-1}$, соответствующих ненулевым элементам вектора $p[i]$, $i=\overline{1,n}$. Если $p[i]=-1$, i -ая строка суммируется со знаком минус.

Окончательно матрица $(C-DE)^{-1}$ преобразуется по формуле

$$\beta[i,j] = \beta[i,j] - \alpha[i]t[j],\quad i=\overline{1,n};\ j=\overline{1,n}; \qquad (2.11)$$

3) $p>n$, $b(p)+k\ne 0$.

Пусть для некоторого i^*, $1\le i^*\le n$, $b(i^*)+p=0$. Заменим строку $\beta[i^*,j]$, $j=\overline{1,n}$ вектором $t[i]$, $i=\overline{1,n}$, вычисленным по (2.9)-(2.10). Тогда окончательное преобразование проводится по формуле (2.8) для $p=i^*$.

Рассмотрим подробнее операции преобразования в последнем случае. Условие $b(p)+k \neq 0$ означает, что если k-ая переменная войдет в базис на p-ое место, то новая матрица I не будет единичной. В силу теоремы, доказанной в /5/, существует такой базисный столбец i^*, что при замене мест i^* - го и p - го столбцов матрица I не изменяется. При замене мест столбцов i^*-го и p-го матрицы B в B^{-1} надо поменять местами i^*-ю и p-ю строки. Согласно (2.I) первые элементы p - й строки матрицы B^{-1} вычисляются по формулам (2.8)-(2.9). Корректность приведенных формул следует из (2.I) и структуры матрицы A (в данном случае их следует рассматривать как выражения основных вычислительных операций алгоритма решения задачи размещения).

3. Вычислительный эксперимент

Для экспериментальной проверки вычислительной эффективности алгоритма на языке ФОРТРАН IУ написана программа, которая выполнялась на ЭВМ CDC-6500. Здесь коротко перечислены те пункты программной реализации алгоритма, которые имеют прямое отношение к анализу полученных результатов.

а) Соответствие f (нумерация переменных) реализована по формуле

$$f(x)= \begin{cases} (i-1)n+j, & \text{если } x=x_{ij}, \\ -(i-1)n-j, & \text{если } x=s_{ij}, \\ mn+i, & \text{если } x=x_i. \end{cases}$$

Отметим, что при такой нумерации в базисе после n-го места могут находиться только переменные с номерами $\pm 1, \pm 2, \dots, \pm mn$ в указанном порядке.

б) Начальный базис $b(i), i=1,mn+n$ содержит переменные с номерами:

$-(n-1),-(n-2),\dots-2,-1,mn+1,1,2,3,\dots,n,-(n+1),-(n+2),\dots-mn.$

Этому базису соответствует вектор оценок $\pi=-c_B B^{-1}$, первыми n-компонентами которого являются

$$-c_{11},-c_{12},\dots,-c_{1n-1},-c_{1n}-c_1$$

и матрица $(C-DE)^{-1}$ со следующими ненулевыми элементами:

$$(C-DE)^{-1} = \begin{bmatrix} & & & & -1 & 1 \\ & & & -1 & & 1 \\ & & -1 & & & 1 \\ & \cdot & & & & \vdots \\ -1 & & & & & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

в) Правило выбора переменной для вхождения базиса следующее: кандидатом на вхождение в базис является первая переменная, для которой коэффициент $\Delta_k < 0$ (см. (2.2)). При этом порядок вычисления Δ_k задается последовательностью $\{mn+1,mn+2,\dots mn+m, -b(1),-b(2),\dots,-b(mn)\}$.

Переменная, для которой $\Delta_k < 0$, вводится в базис, если сохраняется целочисленность опорного плана.

В таблице показаны результаты решения 9 задач размещения. Для всех задач c_{ij} и c_i выбирались случайно с равномерным распределением в интервале 0,I00 за исключением задач с номерами 2 и 4, в которых 50% из чисел c_{ij} имели значение I0000. Обозначения в таблице I имеют следующий смысл:

$m(n)$	m - число производств, n - число потребителей (максимальные значения m и n определяются из неравенства $3mn+2n+m+n^2 \leq \frac{1}{2}V$, где V - объем оперативной памяти ЭВМ в байтах).
ОПТ -	оптимальное значение целевой функции.
ИОПТ -	число итераций до получения оптимального решения.
ИАКТ -	число итераций, в которых целевая функция уменьшалась.
ИСУМ -	число итераций до решения задачи.
t -	время решения задачи в минутах (для задач, в которых время указано в скобках, вычисления прерывались по истечении заданного времени).

В процессе решения задач по указанной программной реализации алгоритма отчетливо выявились две фазы: первая - достижение оптимальной точки или точки, близкой к оптимальной (как в задаче 4), и вторая - доказательство ее оптимальности. Количественным выражением этого факта служит число итераций до получения оптимума и до решения задачи. Удельный вес числа итераций второй фазы быстро растет вместе с ростом m(n) , что, наверное, можно объяснить сильной вырожденностью задачи и ограничениями на целочисленность.

В заключение отметим, что рассмотренный в данной работе метод решения задачи размещения допускает множество других программных реализаций, но отыскание лучшей из них является предметом отдельного исследования.

Таблица

№	m	n	опт	исум	иакт	исум	t
I	I00	80	963	853	98	>I0000	(50)
2	75	50	925	827	9I	>I0000	(40)
3	40	40	628	430	66	6I00	I2
4	40	40	735	902	I04	7000	I4
			73I	5663	I05		
5	30	30	45I	978	7I	I5I0	I
6	20	20	240	355	38	374	0.28
7	20	20	308	234	44	236	0.I5
8	20	20	296	253	43	278	0.I8
9	II	II	348	62	6	I97	0.03

Литература

I. Spielberg K. Algorithms for the simple plant-location problem with some side constrains Oper. Research, 17,1, 1969.

2. Лебедев С.С., Ковалевская М.И. Множители Лагранжа в простейшей задаче размещения. В сб.: Исследования по дискретной оптимизации. "Наука", М., I976.

3. Беркович М.М. О применимости линейного программирования к задачам унификации. В сб.: Исследования по дискретной оптимизации. "Наука", М., I976.

4. Heller I., Tompkins C.B. An extension of a theorem of dantzig. Linear inequalities and related systems, Prinston Univ. Press, 1958.

5. Schrage L. Implicit representation of generalized variable upper bounds in linear programming, Math. programming 14, 1978.

Рукопись поступила в издательский отдел
I9 апреля I979 года.