Управляемые системы, ИМ, ИК СО АН СССР 1974 г. выпуск 12.
УДК.512.25

АЛГОРИТМ НЕЯВНОГО ПЕРЕБОРА ДЛЯ ЗАДАЧИ ТИПА РАЗМЕЩЕНИЯ И СТАНДАРТИЗАЦИИ

В.Л.Береснев

В настоящей работе, которую можно считать продолжением исследований, начатых в /7/, /8/, рассматривается так называемая задача выбора оптимального набора /8/. В силу широты приложения этой задачи, ей посвящено большое количество работ с изложением различных методов решения. К настоящему времени можно выделить два основных подхода к точному решению задачи.

Первый подход /5/ - /8/ связан с рассмотрением некоторых классов задач выбора оптимального набора. Специфика этих классов позволяет построить эффективные алгоритмы со степенными оценками трудоемкости вычислений. К сожалению, многие важные практические задачи не подпадают ни под один из рассмотренных классов. Поэтому второй подход /2/ - /4/ связан с исследованием задачи в общем случае и предполагает использование для решения задачи метода неявного перебора /метода ветвей и границ//см. /1//. Алгоритмы такого типа имеют одну основу, но различаются способами ветвления и построения нижней границы.

В настоящей работе сделана попытка решить задачу в общем случае. Отсутствие в ранее опубликованных работах числовых примеров затрудняет сравнительную оценку приводимого алгоритма. Поэтому для демонстрации работоспособности алгоритма была проведена серия экспериментальных расчетов, результаты некоторых из этих расчетов приводятся.

§ 1. Постановка и некоторые свойства задачи выбора оптимального набора.

Задача выбора оптимального набора состоит в следующем. Пусть заданы конечные множества $\mathcal{U}=\{1,\dots m\}$ и $X=\{1,\dots,n\}$ и известны векторы $(\varphi_1,\dots,\varphi_n)$, $(g_1^0,\dots,g_m^0)$ с неотрицательными компонентами и матрица (g_{ij}), $i\in\mathcal{U}$, $j\in X$, с произвольными элементами. Сопоставим каждому непустому множеству $V\subset\mathcal{U}$ величину

$$S(V)=\sum_{i\in V} g_i^0+\sum_{j\in X}\varphi_j \min_{i\in V} g_{ij}.$$

Требуется отыскать такое множество V^*, при котором функция $S(V)$ принимает наимсньшее значение. Множество V^* будем называть оптимальным набором.

Содержательно сформулированная задача может рассматриваться как задача выбора оптимального параметрического ряда изделий /оптимальной системы технических устройств/. В этом случае $\mathcal{U}$ - множество типов изделий; g_i^o - затраты на организацию производства изделий i - го типа; X - множество видов работ, подлежащих выполнению; φ_j - количество работ j -го вида; g_{ij} - затраты на выполнение изделием i -го типа единичной работы j -го вида. Функция $S(V)$ в рассматриваемом случае означает суммарные затраты на выполнение всего комплекса работ изделиями из параметрического ряда V / системой устройств из V /.

Далее в силу равенства

$$\varphi_i \min_{i\in V} g_{ij} = \min_{i\in V} \varphi_i g_{ij}$$

не умоляя общности, будем считать φ_j = I для всякого $j\in X$.

Рассмотрим матрицу $G=(g_{ij})$, $i\in\mathcal{U}$, $j\in X$. Через $i_1^j,\dots,i_m^j$ будем обозначать перестановку множеств $\{1,\dots,m\}$, для которой

$$g_{i_1^j j}\le g_{i_2^j j}\le\dots\le g_{i_m^j j}\,.$$

Про такую перестановку будем говорить, что она задается j -м столбцом матрицы G . Отметим, что j -й столбец задает не единственную перестановку.

Л е м м а I. Пусть $i_1^j,\dots,i_m^j$ - некоторая перестановка, задаваемая j -м столбцом матрицы G . Пусть, далее $(x_1,\dots,x_m)$, $x_i\in\{0,1\}$, - не единичный вектор. Тогда

$$\min_{i/x_i=0} g_{ij} = g_{i_1^j j} + \sum_{\ell=1}^{m-1}(g_{i_{\ell+1}^j j}-g_{i_\ell^j j})\,x_{i_1^j}\dots x_{i_\ell^j}\,.$$

Д о к а з а т е л ь с т в о. Пусть $\min_{i/x_i=0} g_{ij}=g$ и K - наименьший номер, для которого $x_{i_K^j}=0$. Тогда $x_{i_\ell^j}=1$ при $\ell<K$ и $g_{i_K^j j}=g$. Следовательно,

$$g_{i_1^j j}+\sum_{\ell=1}^{m-1}(g_{i_{\ell+1}^j}-g_{i_\ell^j j})\,x_{i_1^j}\dots x_{i_\ell^j}=g_{i_1^j j}+\sum_{\ell=1}^{K-1}(g_{i_{\ell+1}^j}-g_{i_\ell^j j})=g_{i_K^j j}=g\,.$$

Лемма доказана.

Т е о р е м а I. Задача выбора оптимального набора эквивалентна задаче минимизации полинома $S(x_1,\dots,x_m)$, $x_i\in\{0,1\}$, равного

$$\sum_{i=1}^{m}(1-x_i)g_i^o+\sum_{j=1}^{n}\Big\{a_{j0}+\sum_{\ell=1}^{n-1}a_{j\ell}\,x_{i_1^j}\dots x_{i_\ell^j}\Big\}+F x_1\dots x_m\,,$$

где $i_1^j,\dots,i_m^j$, $j=1,\dots,n$, - некоторая перестановка, задаваемая j -м столбцом матрицы G ; $a_{j0}=g_{i_1^j j}$, $a_{j\ell}=g_{i_{\ell+1}^j j}-g_{i_\ell^j j}$, $j=1,\dots,n$, $\ell=1,\dots,m-1$; F - достаточно большое положительное число.

Доказательство. Пусть $(x_1, \ldots, x_m)$ – вектор с компонентами, принимающими значение 0,1. Рассмотрим задачу минимизации функции

$$S(x_1, \ldots, x_m) = \sum_{i=1}^{m} (1-x_i) g_i^0 + \sum_{j=1}^{n} \min_{i/x_i=0} g_{ij} + F x_1 \ldots x_m$$

где F – достаточно большое положительное число. Пусть $(x_1^*, \ldots, x_m^*)$ – оптимальный вектор $S(x_1, \ldots, x_m)$. Рассмотрим множество $V^* = \{i/x_i^* = 0\}$. Очевидно, $S(x_1^*, \ldots, x_m^*) = S(V^*)$ и V^* – оптимальный набор. Обратно, если V^* – оптимальный набор, то $(x_1^*, \ldots, x_m^*)$, $x_i^* = 0$ при $i \in V^*$, $x_i^* = 1$ при $i \notin V^*$ – оптимальный вектор $S(x_1, \ldots, x_m)$. Следовательно, задача выбора оптимального набора и задача минимизации функции $S(x_1, \ldots, x_m)$ эквивалентны. Для завершения доказательства остается воспользоваться леммой 1 и переписать функцию $S(x_1, \ldots, x_m)$ следующим образом:

$$\sum_{i=1}^{m} (1-x_i) g_i^0 + \sum_{j=1}^{n} \left\{ g_{i_1^j j} + \sum_{l=1}^{m-1} (g_{i_{l+1}^j j} - g_{i_l^j j}) x_{i_1^j} \ldots x_{i_l^j} \right\} + F x_1 \ldots x_m .$$

Теорема доказана.

Из построения $S(x_1, \ldots, x_m)$ видно, что полином

$$G(x_1, \ldots, x_m) = \sum_{j=1}^{n} \left\{ a_{j0} + \sum_{l=1}^{m-1} a_{jl} x_{i_1^j} \ldots x_{i_l^j} \right\}$$

полностью определяется матрицей G. Поэтому про полином $G(x_1, \ldots, x_m)$ будем говорить, что он соответствует матрице G. Пусть теперь имеется полином

$$H(x_1, \ldots, x_m) = \sum_{t=1}^{T} \left\{ b_{t0} + \sum_{l=1}^{m-1} b_{tl} x_{i_1^t} \ldots x_{i_l^t} \right\},$$

где $\{i_1^t, \ldots, i_m^t\}$, $t = 1, \ldots, T$, – некоторая система перестановок множества $\{1, \ldots, m\}$, а $b_{tl} \geq 0$ при любых t, l. Построим по полиному $H(x_1, \ldots, x_m)$ матрицу $H = (h_{it})$, которой соответствует полином $H(x_1, \ldots, x_m)$. Очевидно, матрица H должна иметь m строк и T столбцов. Нетрудно заметить, что элементы t-го столбца матрицы H определяются из следующих уравнений:

$$h_{i_1^t t} = b_{t0}, \quad h_{i_{l+1}^t t} - h_{i_l^t t} = b_{tl}, \quad l = 1, \ldots, m-1$$

Действительно, решив систему уравнений, получим для всякого

$l=1,2,\ldots,m$ $\quad h_{i_l^t t}=b_{t0}+\ldots+b_{t\,l-1}$

При этом, поскольку $b_{tl}\geq 0$, имеем

$$h_{i_1^t t}\leq h_{i_2^t t}\leq\ldots\leq h_{i_m^t t}\ ,$$

и, следовательно, $i_1^t,\ldots,i_m^t$ - перестановка, задаваемая t -м столбцом матрицы H .

Л е м м а 2. Для любого не единичного вектора $(x_1,\ldots,x_m)$ имеет место неравенство

$$g(x_1,\ldots,x_m)\geq\sum_{j\in X}\min_{i\in U}\{g_{ij}+w_{ij}\}\ ,$$

где w_{ij} , $i\in U$, $j\in X$, - неотрицательные числа, такие что

$$\sum_{j\in X}w_{ij}\leq g_i^0$$

для всякого $i\in U$.

Д о к а з а т е л ь с т в о. Пусть $V=\{i/x_i=0\}$ и $\{X_i\}$, $i\in V$,- совокупность областей применения для V /7/ .

Тогда

$$g(x_1,\ldots,x_m)=g(V)=\sum_{i\in V}\{g_i^0+\sum_{j\in X_i}g_{ij}\}\geq$$

$$\geq\sum_{i\in V}\{\sum_{j\in X}w_{ij}+\sum_{j\in X_i}g_{ij}\}\geq\sum_{i\in V}\sum_{j\in X_i}\{w_{ij}+g_{ij}\}\geq$$

$$\geq\sum_{i\in V}\sum_{j\in X_i}\min_{i\in U}\{w_{ij}+g_{ij}\}=\sum_{j\in X}\min_{i\in U}\{w_{ij}+g_{ij}\}\ .$$

Лемма доказана.

Л е м м а 3. Пусть $y_1,\ldots,y_p$ - фиксированные числа, равные 0 или 1. Тогда полином $G(y_1,\ldots,y_p,x_{p+1},\ldots,x_m)$ от переменных $x_{p+1},\ldots,x_m$ соответствует матрице $H=(h_{ij})$, $i=p+1,\ldots,m$; $j\in X$;

$$h_{ij}=\begin{cases}g_{ij}\ , & \text{если } g_{ij}\leq Q_{0j}\ ,\\ Q_{0j}\ , & \text{если } g_{ij}>Q_{0j}\ ,\end{cases}$$

где

$$Q_{0j}=\begin{cases}\min\limits_{i/y_i=0}g_{ij}\ , & \text{если } y_1,\ldots,y_p=0,\\ \infty & \text{если } y_1,\ldots,y_p=1,\end{cases}$$

Д о к а з а т е л ь с т в о. Рассмотрим полином

$$\mathfrak{S}_j(x_1,\dots,x_m)=a_{j0}+\sum_{l=1}^{m-1} a_{jl}\,x_{i_1 j}\dots x_{i_l j}.$$

Пусть l^* - наименьший номер, для которого $y_{i_{l^*}}=0$. Если номера l^* не существует, т.е. $y_1=\dots=y_p=1$, то положим $l^*=m$. Пусть $l_1,\dots,l_Q$ - все такие номера, что $i_{l_q}>p$, и пусть $l_q<l^*$ при $q=1,\dots,Q'$, $Q'\leqslant Q$. Тогда полином $\mathfrak{S}_j(y_1,\dots,y_p,x_{p+1},\dots,x_m)$ может быть записан следующим образом:

$$(a_{j0}+\dots+a_{jl_1-1})-\sum_{q=1}^{Q'-1}(a_{jl_q}+\dots+a_{jl_{q+1}-1})\,x_{i_{l_1}}\dots x_{i_{l_q}}+$$
$$+(a_{jl_{Q'}}+\dots+a_{jl^*-1})\,x_{i_{l_1}j}\dots x_{i_{l_{Q'}}j}.$$

Построим по этому полиному j-й столбец матрицы H. Элементы столбца определяются из следующей системы равенств:

$$h_{i_{l_1}j}=(a_{j0}+\dots+a_{jl_1-1});$$
$$h_{i_{l_{q+1}}j}-h_{i_{l_q}j}=(a_{jl_q}+\dots+a_{jl_{q+1}-1}),\quad q=1,\dots,Q'-1;$$
$$h_{i_{l_{Q'+1}}j}-h_{i_{l_{Q'}}j}=(a_{jl_{Q'}}+\dots+a_{jl^*-1});$$
$$h_{i_{l_{q+1}}j}-h_{i_{l_q}j}=0,\quad q=Q'+1,\dots,Q-1.$$

Отсюда получаем

$$h_{i_{l_q}j}=a_{j0}+\dots+a_{jl_q-1}=g_{i_{l_q}j},\quad q=1,\dots,Q';$$
$$h_{i_{l_q}j}=a_{j0}+\dots+a_{jl^*-1}=g_{i_{l^*}j},\quad q=Q'+1,\dots,Q.$$

Что и требовалось показать.

Из сказанного вытекает справедливость следующей леммы.

Л е м м а 4. Пусть первые p компонентов вектора $(x_1,\dots,x_m)$ равны соответственно $y_1,\dots,y_p$ и пусть среди остальных компонентов есть хотя бы один нуль. Тогда

$$S(x_1,\dots,x_m)\geqslant\sum_{i/y_i=0} g_i^0+\sum_{j\in X}\min_{i>p}\{w_{ij}+h_{ij}\},$$

где w_{ij}, $i=p+1,\dots,m$; $j\in X$ - неотрицательные числа, такие, что

$$\sum_{j\in X} w_{ij}\leqslant g_i^0$$

для всякого $i=p+1,\dots,m$.

§ 2. Алгоритм неявного перебора

Настоящий параграф посвящен описанию алгоритма типа неявного перебора, используемого для отыскания во множестве $S=\{(x_1,\dots,x_m)$, $x_i\in\{0,1\}\}$, оптимального вектора полинома $S(x_1,\dots,x_m)$.

В описанном ниже алгоритме, как и в любом алгоритме неявного перебора, поиск опимального решения осуществляется посредством разбиения множества решений на подмножества, отыскания среди элементов разбиения перспективного подмножества и исследования /зондирования/ этого подмножества. Будем говорить, что множество решений прозондировано, если в нем найдено наилучшее решение или с помощью нижней границы установлено, что множество не содержит решений, лучших уже известного.

Если удается прозондировать подмножество решений, то оно выбрасывается из рассмотрения. В противном случае множество подвергается делению /ветвлению/ и среди элементов полученного разбиения вновь отыскивается перспективное подмножество. Этот процесс продолжается до тех пор, пока число элементов разбиения не станет равным R_1, где R_1 - заданная величина. На этом одновременное ветвление /I/ элементов разбиения заканчивается и начинается одностроннее /I/ ветвление перспективного подмножества.

При одностороннем ветвлении в исследуемом множестве последовательно выделяются такие подмножества, которые удается прозондировать. Этот процесс продолжается до тех пор, пока не будет прозондировано все множество. После этого возвращаемся к рассмотрению элементов разбиения и отысканию перспективного множества.

Пусть t - подмножество множества S. Функцию B, ставящую в соответствие множеству t несобственные подмножества t_1, t_2, $t_1\cup t_2=t$, назовем одновременным ветвлением, а функцию b, сопоставляющую множеству t непустое подмножество t_1, - односторонним ветвлением. Величину $\beta(t)$ назовем нижней границей полинома $S(x)$ на множестве t, если $\beta(t)\le\min\limits_{x\in t} S(x)$ и $\beta(t)=\min\limits_{x\in t} S(x)$, когда множество t включает не более двух элементов.

Перейдем к описанию общей схемы алгоритма. Алгоритм состоит из конечного числа шагов, на каждом из которых рекуррентно пересчитываются:

t_1, ..., t_K - разбиение множества не выброшенных из рассмотрения решений;

x - наилучшее найденное решение;

R - метка, принимающая значение I, если на предыдущем шаге закончено исследование перспективного подмножества, и 0, если незакончено. Когда $R=0$, T - множество не рассмотренных решений перспективного множества;

τ - подмножество T, зондируемое на данном шаге.

На первом шаге $K=1$, $t_1=S$, $R=1$.

Если $R=1$, то шаг начинается с исключения из рассмотрения элементов разбиения t_i, для которых $\beta(t_i)\geqslant S(x)$. Далее, среди оставшихся элементов отыскиваем множество t, для которого $\beta(t)$ минимально. Если множества t не существует, т.е. все элементы разбиения выброшены из рассмотрения, то алгоритм заканчивает работу и x - оптимальный вектор. Пусть множество t найдено, положим тогда $\tau=t$, $T=t$ и рассмотрим величину $\beta(\tau)$. Когда $R=0$, сразу переходим к рассмотрению $\beta(\tau)$.

Пусть $\beta(\tau)\geqslant S(x)$, либо $\beta(\tau)<S(x)$, но удается отыскать вектор $y\in S$, для которого $\beta(\tau)=S(y)$. Положим тогда $T=T\setminus\tau$ и, если вектор y найден, $x=y$. Далее, если $T\neq\emptyset$, положим $R=1$, если $T=\emptyset$, выберем непустое подмножество τ множества T. После этого перейдем к следующему шагу.

Пусть $\beta(\tau)<S(x)$, но вектора y найти не удается. Если $R=0$ либо $K+2>R_1$, положим $R=0$, $\tau=b(\tau)$ и начнем новый шаг. Пусть $R=1$ и $K+2\leqslant R_1$ и пусть τ_1, τ_2 - результат применения к множеству τ функции B. Тогда рассмотрим разбиение $t_1,\dots,t_K$, τ_1,τ_2, и начнем новый шаг.

Заметим, что на каждом шаге описанного алгоритма происходит одно из трех: либо некоторое множество решений выбрасывается из рассмотрения, либо зондируемое множество делится на два подмножества с меньшим числом элементов в каждом /одновременное ветвление/, либо в зондируемом множестве выделяется подмножество с меньшим числом элементов, которое подвергается зондированию на следующем шаге /одностороннее ветвление/. В силу этого замечания, конечности множества S и свойства функции β следует, что за конечное число шагов будет найден оптимальный вектор полинома $S(x_1,\dots,x_m)$.

Заметим также, что если на каждом шаге $S(x)$ сравнивать не с $\beta(\tau)$, а с $R_2\beta(\tau)$, где $1\leqslant R_2$ - заданная величина, то в результате работы алгоритма будет найден вектор x^0, для которого

$$S(x^*)\leqslant S(x^0)\leqslant R_2S(x^*).$$

Здесь x^* - оптимальный вектор.

В самом деле, если $S(x^0)=S(x^*)$, то утверждение очевидно. Пусть $S(x^0)\neq S(x^*)$ и предположим, что $S(x^0)>R_2S(x^*)$. Пусть τ - множество, содержащее x^*, а x - наилучшее решение на некотором шаге. Тогда $S(x)\geqslant S(x^0)>$ $>R_2S(x^*)\geqslant R_2\beta(\tau)$. Отсюда следует, что в процессе работы алгоритма множество τ может быть выброшено из рассмотрения только в том случае, если в τ найден оптимальный вектор x^*. Но тогда $S(x^0)=S(x^*)$, а это противоречит предположению.

Таким образом, использование параметров R_2 позволяет отыскивать приближенное решение задачи минимизации полинома $S(x)$ с наперед заданной точностью.

Отметим теперь, что при описании алгоритма никак не использовалась специфика полинома $S(x_1,\dots,x_m)$ и множества S. Рассмотренная последовательность действий пригодна для отыскания оптимального решения произвольной функции $S(x)$, заданной на конечном множестве S. Специфика полинома $S(x_1,\dots,x_m)$ и множества S должна использоваться при записи информации о рассматриваемых в процессе работы алгоритма множествах и при построении нижней границы. При этом эффективность алгоритма зависит от того, насколько полно используется эта специфика, т.е. насколько просто записывается информация о множествах и насколько нижняя граница близка к действительному значению минимума.

Вектор $(y_1,\dots,y_p)$, $0<p\le m$, $y_i\in\{0,1\}$, назовем частичным решением, а вектор $(y_1,\dots,y_p,x_{p+1},\dots,x_m)$, $x_l\in\{0,1\}$, $l>p$, – продолжением частичного решения $(y_1,\dots,y_p)$. Если $p=0$, то будем говорить о пустом частичном решении. Будем говорить, что вектор $(y_1,\dots,y_m)$ лексикографически не меньше вектора $(x_1,\dots,x_m)$, если не существует номера K, $i\le K\le m$, такого что $x_i=y_i$, $i<K$, $x_K>y_K$. Частичное решение $(y_1,\dots,y_p)$ и номер q, $1\le q\le p$, задают три подмножества τ, t, T, $\tau\subset T\subset t$, множества S. Действительно t и τ суть множества продолжений частичных решений $(y_1,\dots,y_q)$, $(y_1,\dots,y_p)$, а T – множество векторов из t лексикографически не меньших вектора $(y_1,\dots,y_p,0,\dots,0)$. Если $p=q$, то очевидно, $t=\tau=T$.

Покажем, что в случае $S=\{(x_1,\dots,x_m),\ x_i\in\{0,1\}\}$ функции B и b можно определить таким образом, что рассматриваемые на каждом шаге алгоритма множества будут задаваться частичным решением и номером.

На первом шаге множество $t_1=S$ задается пустым частичным решением. Пусть на некотором шаге элементы разбиения $t_1,\dots,t_K$ задаются частичными решениями, а множества t, T, τ – частичным решением $(y_1,\dots,y_p)$ и номером q, $1\le q\le p$. Предположим, что множество τ выбрасывается из рассмотрения. Пусть q', $q<q'\le p$ – наибольший номер, для которого $y_{q'}=0$. Если номера q' не существует, то $\tau=T$ и исключение из рассмотрения множества τ сводится к зачеркиванию $(y_1,\dots,y_p)$ и q. Пусть номер q' существует, тогда рассмотрим частичное решение $(y_1,\dots,y_{q'-1},1)$ и номер q. Очевидно, они задают множества t, T', τ' такие, что $T'=T\setminus\tau$, $\tau'\subset T'$. Предположим теперь, что к множеству τ применяется функция b. В этом случае $p<m-1$, поэтому можем положить $b(\tau)$ равным множеству продолжений частичного решения $(y_1,\dots,y_p,0)$. Тогда частичное решение $(y_1,\dots,y_p,0)$ и номер q задают требуемые множества t, T, $b(\tau)$. Предположим, наконец, что к τ применяется функция B. Рассмотрим частичные решения $(y_1,\dots,y_p,0)$, $(y_1,\dots,y_p,1)$. Множества, задаваемые этими частичными решениями, будем считать результатом применения к τ функции B. Таким образом, во всех трех случаях множества, рассматриваемые на следующем шаге алгоритма, вновь задаются частичными решениями и номером.

Рассмотрим теперь вопрос о вычислении нижней границы $\beta(y_1,\dots,y_p)$ полинома $S(x_1,\dots,x_m)$ на множестве, задаваемом решением $(y_1,\dots,y_p)$. Пусть $(x_1^*,\dots,x_m^*)$ - оптимальный вектор $S(x_1,\dots,x_m)$ на этом множестве. Имеет место одно из двух: либо $x_{p+1}^*\cdot\ldots\cdot x_m^*=1$, либо $x_{p+1}^*\cdot\ldots\cdot x_m^*=0$.

В первом случае

$$S(x_1^*,\dots,x_m^*)=\beta_1(y_1,\dots,y_p)=\sum_{i/y_i=0} g_i^\circ+\sum_{j=1}^{n} Q_{oj}$$

Во втором случае, следуя лемме 4, имеем

$$S(x_1^*,\dots,x_m^*)\geqslant\beta_2(y_1,\dots,y_p)=\sum_{i/y_i=0} g_i^\circ+\sum_{j=1}^{n}\min_{i>p}\{h_{ij}+w_{ij}\}$$

Таким образом,

$$\beta(y_1,\dots,y_m)=\min\{\beta_1(y_1,\dots,y_m),\beta_2(y_1,\dots,y_m)\}.$$

Из сказанного следует, что нижняя граница зависит от выбора матрицы $W=(w_{ij})$, $i=p+1,\dots,m$; $j=1,\dots,n$, поэтому матрицу W нужно стремиться выбрать таким образом, чтобы получить величину $\beta_2(y_1,\dots,y_p)$ побольше.

Матрицу W для всякого столбца которой $(j=1,\dots,n)$ выполняются условия:

если $w_{ij}>0$, то $h_{ij}+w_{ij}=\min\{h_{kj}+w_{kj}\}$,

$$\min_{i/w_{ij}>0}\{g_i^\circ-\sum_{\ell=1}^{n} w_{i\ell}\}=0$$

назовем тупиковой для H матрицей. Тупиковая матрица обладает тем свойством, что ее локальное улучшение невозможно, т.е. нельзя увеличить $\beta_2(y_1,\dots,y_p)$, изменяя элементы лишь двух столбцов матрицы W.

Заметим теперь, что если $\beta(y_1,\dots,y_p)=\beta_1(y_1,\dots,y_p)$, то $(y_1,\dots,y_p,1,\dots,1)$ - оптимальный вектор на множестве продолжений частичного решения $(y_1,\dots,y_p)$. Отсюда следует, что если W - тупиковая матрица, то $\beta(y_1,\dots,y_p)=S(x_1^*,\dots,x_m^*)$ при $p=m-1$. Таким образом, построенная нижняя граница удовлетворяет требуемым свойствам.

Опишем теперь общую схему алгоритма построения тупиковой для H матрицы W. Алгоритм состоит из конечного числа шагов, на каждом из которых рекуррентно пересчитываются матрица $W=(w_{ij})$, $i=p+1,\dots,m$; $j=1,\dots,n$, и множество J не выброшенных из рассмотрения столбцов матрицы H.

На первом шаге $w_{ij}=0$, $J=\{1,2,\dots,n\}$. Шаг начинается с проверки условия $J=\varnothing$. Если $J=\varnothing$, то алгоритм заканчивает работу. Пусть $J\neq\varnothing$. Для $j\in J$ вычислим $Q_j=\min_{i>p}\{h_{ij}+w_{ij}\}$ и $J_j=\{i/h_{ij}+w_{ij}=Q_j\}$.

Далее отыщем номер $\ell\in J$, для которого число элементов в J_ℓ минимально, и вычислим

$$A = \min\left\{\min_{i\in\mathcal{J}_\ell}\left\{g_i^* - \sum_{j=1}^{n} w_{ij}\right\};\ \min_{i\notin\mathcal{J}_\ell}\left\{h_{i\ell} - Q_\ell\right\}\right\}.$$

Если $A = 0$, то положим $\mathcal{J} = \mathcal{J}\setminus\{\ell\}$ и начнем новый шаг. Если $A > 0$, то положим $w_{i\ell} = w_{i\ell} + A$ для $i \in \mathcal{J}_\ell$ и перейдем к следующему шагу. Таким образом, алгоритм отыскания оптимального вектора полинома $S(x_1,\dots,x_m)$ описан полностью.

Сделаем несколько замечаний относительно трудоемкости вычислений по этому алгоритму. Отметим прежде всего, что для алгоритмов неявного перебора характерны две особенности.

Во-первых, алгоритмы неявного перебора требуют большого количества памяти для хранения элементов разбиения. Заметим сразу, что в описанном выше алгоритме число одновременно рассматриваемых элементов разбиения регулируется и не превышает значения параметра R_1. Для вычисления нижней границы также не требуется большой памяти, поскольку информация о матрицах H и W может быть задана векторами (Q_{0j}), $j=1,\dots,n$, и (Q_j), $j=1,\dots,n$.

Вторая особенность алгоритмов неявного перебора состоит в том, что теоретически число арифметических и логических операций совпадает с числом операций при полном переборе. Хотя для решения конкретной задачи требуется число операций значительно меньшее, чем полный перебор, тем не менее оно может быть очень большим. Поэтому о практической применимости алгоритма можно судить лишь после того, как будет выяснено /хотя бы экспериментально/, как увеличивается время счета /число операций/ при росте размерности задачи.

Из приведенного способа вычисления нижней границы следует, что время решения задачи выбора оптимального набора рассмотренным алгоритмом слабо /линейно/ зависит от числа столбцов матрицы. Кроме того, время счета уменьшается с возрастанием величины R_2, поскольку в этом случае происходит искусственное "улучшение" нижней границы. К сожалению, о скорости возрастания счета при увеличении числа строк матрицы можно судить лишь по проведению серии экспериментальных расчетов. Г.М.Заикиной была составлена программа на языке АЛГОЛ--БЭСМ, реализующая описанный алгоритм. Ниже приводится таблица, составленная в результате решения некоторого числа как практических задач выбора оптимального набора, так и задач со случайно выбранными исходными данными. В таблице обозначено: m - число строк, n - - число столбцов матрицы t - среднее время в минутах решения задачи на ЭВМ ВЭСМ -6, K - среднее число шагов, проделанных алгоритмом.

m	n	R_1	R_2	t	K
20	50	200	I,0	0,5	20
30	50	200	I,0	I	35
50	50	200	I,0	6	200
60	50	200	I,0	I0	370
75	50	200	I,0	20	830
75	50	200	I,I	8	260
I00	50	200	I,0	30	I270
I00	50	200	I,I	I0	360

Поступила в ред.-изд.отдел
25.I.I974 г.

Литература

I. И.В.Романовский. Методы неявного перебора для решения задач целочисленного программирования с бивалентными переменными.-"Изв.вузов. Матем".,I970, № 4.

2. В.А.Емеличев, М.М.Ковалев. Решение некоторых задач вогнутого программирования методом построения последовательности планов.I.-"Изв. АН БССР", I970, № 6.

3. M.E.Efroumson, T.L. Ray. A branch-bound algorithm for plant location. Operat. Res., I966, I4, N3.

4. Sa G. Branch-bound and approximate solution to the capacitated plant location problem. Operat. Res., I969, I7, N 6.

5. Э.Х.Гимади. Выбор оптимальных шкал в одном классе задач типа размещения, унификации и стандартизации.- В кн.: Управляемые системы, Новосибирск, I969,вып.6.

6. Э.Х.Гимади, В.Т.Дементьев. О методах решения некоторых задач оптимизации параметрических рядов.- В кн.: Стандарты и качество,I97I, № I2.

7. В.Л.Береснев. Об одном классе задач оптимизации параметров однородной технической системы.- В кн.: Управляемые системы,Новосибирск, I97I, вып. 9.

8. В.Л.Береснев. Об одной задаче математической теории стандартизации. I.- В кн.: Управляемые системы, Новосибирск, I973,вып.II.