EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. 30. KÖIDE
FÜÜSIKA * MATEMAATIKA. 1981, NR. 2

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ 30
ФИЗИКА * МАТЕМАТИКА. 1981, № 2

https://doi.org/10.3176/phys.math.1981.2.04

УДК 519.854.3

*Х. НЯРИПЯ*

# ОДИН МЕТОД РЕШЕНИЯ ЗАДАЧИ ГРУППОВОЙ МИНИМИЗАЦИИ С ДОПОЛНИТЕЛЬНЫМ ЛИНЕЙНЫМ ОГРАНИЧЕНИЕМ

*(Представил А. Хумал)*

## 1. Постановка задачи и некоторые понятия

Рассматривается задача

$$\min \sum_{j=1}^{n} c_j x_j \tag{1}$$

при условиях

$$\sum_{j=1}^{n} \beta_j x_j = \beta_0, \tag{2}$$

$$\sum_{j=1}^{n} a_j x_j \leqslant a_0, \tag{3}$$

$$x_j \geqslant 0 \text{ и целое,} \quad j = 1, \ldots, n, \tag{4}$$

где $c = (c_1, \ldots, c_n) \geqslant 0$, $\beta_j \in G$ при $j = 0, 1, \ldots, n$ и $G$ является конечной абелевой группой порядка $d$, т. е. $G = \{\alpha_0 = 0, \alpha_1, \ldots, \alpha_{d-1}\}$, а в условии (2) под сложением, умножением на скаляр и сравнением подразумеваются групповые операции. В дальнейшем будем пользоваться еще следующими обозначениями:

$$a = (a_1, \ldots, a_n), \quad \beta = (\beta_1, \ldots, \beta_n), \quad x = (x_1, \ldots, x_n)^T, \quad \beta x = \sum_{j=1}^{n} \beta_j x_j,$$

а через $d_j$ будем обозначать порядок циклической подгруппы группы $G$, генерируемой элементом $\beta_j$.

Связь между задачей групповой минимизации (ГМ) (1), (2), (4) и задачами целочисленного линейного программирования (ЦЛП) подробно изложена, например, в [1] (с. 387—400). Дополнительное линейное ограничение (3) в задаче (1)—(4) усложняет ее решение, но зато значительно расширяет область применения задач ГМ. К виду (1)—(4) приводятся задача о рюкзаке и задача нахождения оптимального значения линейной формы на целочисленных точках любого симплекса. Далее, задачей (1)—(4) можно пользоваться в качестве оценочной к задачам ЦЛП. Один способ нахождения сравнительно сильных оценок изложен в [2]. Кроме того, к задаче (1)—(4) приводит математическая формулировка некоторых важных проблем планирования народного хозяйства. Такой проблемой является, например, комплектование оборудования многоступенчатых конвейерных систем с дискретной продукцией и с дополнительными условиями, что все узлы могут работать только с заданной скоростью и вся промежуточная продукция должна употребляться самой системой.

Ниже будет описан комбинаторный алгоритм решения задачи (1)—(4), состоящий из двух этапов. На первом этапе для всех $\alpha_j \in G$ решается задача ГМ

$$\min\{cx \mid \beta x = \alpha_j,\ x \geqslant 0 \text{ и целое}\},$$

обозначаемая в дальнейшем как задача $G(\alpha_j)$. Дополнительно находится вектор $u = (u_1, \ldots, u_n)$ верхних границ переменных $x_j$, которому удовлетворяют компоненты по крайней мере одного оптимального решения задачи (1)—(4). На втором этапе применяется метод типа ветвей и границ для нахождения оптимального решения задачи (1)—(4).

Приведем некоторые понятия, среди которых понятие доминирования сыграет в дальнейшем существенную роль.

Определение 1. *Псевдорешением задачи* (1)—(4) *называется любое допустимое решение задачи* $G(\beta_0)$.

Определение 2. *Неотрицательный $n$-вектор $y$ доминирован таким же вектором $x$ (или $x$ доминирует $y$) относительно векторов $c, a$ и $\beta$, если $\beta x = \beta y$, $cx \leqslant cy$, $ax \leqslant ay$ и по меньшей мере одно из двух последних неравенств выполняется строго.*

Так как в дальнейшем $c, a$ и $\beta$ являются постоянными, то будем просто говорить, что $x$ доминирует $y$, обозначая это через $x \prec y$.

Определение 3. *Вектор $y$ считается доминированным, если существует некоторый другой вектор $x$ такой, что $x \prec y$.*

Определение 4. *Расширением вектора $x$ называется вектор $z = x + y$, где $y$ — произвольный неотрицательный целочисленный вектор.*

Легко убедиться, что любое расширение доминированного вектора также доминировано.

Пусть $J^0 = \{j \mid c_j = 0\}$. В дальнейшем предполагается, что $a_j \geqslant 0$ при всех $j \in J^0$. Это предположение естественно, так как в случае существования $k \in J^0$ такого, что $a_k < 0$, задача (1)—(4) решается очень просто. Действительно, искомым решением $x^*$ является вектор $x^* = x(\beta_0) + td_k e_k$, где $x(\beta_0)$ — оптимальное решение задачи $G(\beta_0)$, $t = \max\left\{0, \left]\dfrac{a_0 - ax(\beta_0)}{a_k d_k}\right[\right\}$ и $e_k$ — $k$-й единичный вектор. *

Учитывая сделанное предположение, можно утверждать, что среди оптимальных решений задачи (1)—(4) всегда существуют недоминированные решения, нахождением которых и будем ограничиваться.

---

* Через $]s[$ обозначается наименьшее целое, не меньше $s$, а через $[s]$ — целая часть $s$.

## 2. Описание первого этапа алгоритма

На этом этапе алгоритма при всех $\alpha_j \in G$ вычисляются оптимальные решения $x(\alpha_j)$ задач $G(\alpha_j)$. Для этого можно пользоваться любым алгоритмом решения задач ГМ, гарантирующим неприводимость получаемых решений. Таким является, например, второй вариант алгоритма, предложенный в [3].

Для работы алгоритма необходимы также верхние границы $u_j$ компонент оптимального решения $x^*$ задачи (1)—(4). Введем следующие обозначения:

$$J^+=\{j\,|\,a_j\geqslant 0\},\quad J^-=\{j\,|\,a_j<0\},\quad A^-=\min_i ax(\alpha_i),$$

$$A^+=\max_i ax(\alpha_i),\quad M^1_j=\left]\frac{a_0-A^+}{a}\right[,\quad M^2_j=\max_i x_j(\alpha_i).$$

Теорема. *Если у задачи* (1)—(4) *существуют оптимальные решения, то среди них имеется по крайней мере одно решение* $x^*$, *компоненты которого удовлетворяют следующим условиям:*

$$x^*_j\leqslant\begin{cases}u^+_j=d_j-1, \text{ если } j\in J^+,\\ u^-_j=M^2_j+\max\{0,M^1_j\}, \text{ если } j\in J^-.\end{cases}\tag{5}$$

Доказательство. Пусть $x^0$ — произвольное оптимальное решение задачи (1)—(4) и $N=\{j\in J^-|\,M_j{}^1>0,\ x_j{}^0>u_j{}^-\}$. Тогда искомым оптимальным решением, удовлетворяющим условиям теоремы, является вектор $x^*=\bar{x}+x(\beta_0-\beta\bar{x})$, где $\bar{x}=\sum_{j\in N}M_j{}^1e_j$. Действительно, $x^*\geqslant 0$ и $\beta x^*=\beta_0$, а по определению величин $u_j{}^+$, $u_j{}^-$ и $M_j{}^1$ выполняются условия $ax^*\leqslant a_0$, $x_j{}^*\leqslant u_j{}^+$ при $j\in J^+$ и $x_j{}^*\leqslant u_j{}^-$ при $j\in J^-$. Наконец, $cx^*=c\bar{x}+cx(\beta_0-\beta\bar{x})\leqslant c\bar{x}+c(x^0-\bar{x})=cx^0$. При $N=\varnothing$ и $P=\{j\in J^+|\,x^0_j>u^+_j\}\neq\varnothing$ искомым решением является $x^*=x^0-\sum_{j\in P}t_jd_je_j$, где $t_j=\left[\dfrac{x^0_j}{d_j}\right]$. Теорема доказана.

Границы $u_j{}^+$ и $u_j{}^-$ вычисляются легко, но они слишком грубые и носят чисто теоретический характер.

Пользуясь понятием доминирования, можно вычислить следующие границы:

$$\bar{u}_j=\min\{t-1\,|\,te_j\succ x(t\beta_j),\ t\geqslant 1 \text{ и целое}\}.\tag{6}$$

Отметим, что при $j\in J^+$ всегда $\bar{u}_j\leqslant u_j{}^+$, но при $j\in J^-$ может оказаться $\bar{u}_j=\infty$. Чтобы сократить объем вычислений при определении границ $\bar{u}_j$, воспользуемся следующим очевидным утверждением: *если для некоторого* $j\in J^-$ *не существует целочисленного* $t_j$ *такого, что*

$$1\leqslant t_j\leqslant\min\left\{d_j\left[\frac{A^-}{a_j}\right]\right\}\ \text{ и }\ t_je_j\succ x(t_j\beta_j),$$

*то* $\bar{u}_j=\infty$.

Таким образом, для вычисления каждой $\bar{u}_j$ требуется не более $d_j$ проверок выполнения условий доминирования. Отметим еще, что границы $u_j^-$ не могут влиять на процесс упорядоченного перебора на втором этапе алгоритма, а все границы, выводимые из доминирования векторов, могут существенно ускорить работу алгоритма.

### 3. Описание второго этапа алгоритма

Процесс нахождения оптимального решения задачи (1)—(4) можно описать деревом просмотра, каждой вершине $T^k$ которого соответствует следующая задача определения оптимального расширения некоторой целочисленной неотрицательной точки $x^k$:

$$\min\{c^k+cy \mid \beta y=\alpha^k,\quad ay\leqslant a^k,\quad 0\leqslant y\leqslant u^k,\quad y \text{ — целое} \mid\}, \tag{7}$$

где $c^k = cx^k$, $\alpha^k = \beta_0 - \beta x^k$ и $a^k = a_0 - ax^k$. Каждая задача (7) однозначно определяется параметрами задачи (1)—(4), вектором $x^k$ и вектором верхних границ $u^k$. Обозначим через $X^k$ множество допустимых (относительно задачи (7)) расширений вектора $x^k$, т. е.

$$X^k=\{x \mid x=x^k+y,\ \beta y=\alpha^k,\ ay\leqslant a^k,\ 0\leqslant y\leqslant u^k,\ y \text{ — целое}\}.$$

Легко убедиться, что каждое $x\in X^k$ является допустимым решением задачи (1)—(4), а при $x^0 = 0$ и $u^0 = (\bar{u}_1, \ldots, \bar{u}_n)$ (определяется по формуле (6)) в $X^0$ содержится оптимальное решение задачи (1)—(4).

Опишем применяемый в алгоритме принцип ветвления. Обозначим через $y^k = x(\alpha^k)$ оптимальное решение задачи $G(\alpha^k)$, найденное на первом этапе алгоритма. Если $ay^k \leqslant a^k$ и $y^k \leqslant u^k$, то $y^k$ является оптимальным решением задачи (7). В противном случае задача (7) разбивается на $n$ подзадач такого же типа. При этом используется следующее обстоятельство: так как неприводимый вектор $y^k$ не удовлетворяет ограничениям задачи (7), то при каждом допустимом решении $y$ этой задачи должно выполняться по меньшей мере одно из условий $y_j \geqslant \geqslant y_j^k + 1$, где $j = 1, \ldots, n$.

Пусть на основе некоторого правила $R_B$ множество натуральных чисел $\{1, \ldots, n\}$ преобразуется в упорядоченное множество $< j(1), \ldots \ldots, j(n) >$. Векторы $x^{k,j(i)}$ и $u^{k,j(i)}$, определяющие $i$-ю подзадачу, вычисляются по следующим формулам:

$$x^{k,j(i)}=x^k+(y^k_{j(i)}+1)\,e_{j(i)}, \tag{8}$$

$$u^{k,j(i)}_{j(s)}=\begin{cases}\min\{y^k_{j(s)},\ u^k_{j(s)}\}, & \text{если } s<i,\\ u^k_{j(s)}-y^k_{j(s)}-1, & \text{если } s=i,\\ u^k_{j(s)}, & \text{если } s>i.\end{cases} \tag{9}$$

Аналогично множеству $X^k$ можно определить множества $X^{k,j}$, которые, в силу формул (8) и (9), удовлетворяют следующим соотношениям:

$$X^{k,j}\cap X^{k,i}=\varnothing,\ \text{если } i\neq j,\quad \text{и}\quad \bigcup_{j=1}^{n} X^{k,j}=X^k\setminus\{x^k+y^k\}.$$

Тем самым отпадает необходимость в повторном генерировании эквивалентных задач и гарантируется конечность алгоритма.

Алгоритм обладает двумя списками: списком перспективных вершин $T$ и списком псевдорешений $S$ задачи (1)—(4). Список $T$ характерен для каждого метода ветвей и границ, а список $S$ свойствен лишь описываемому алгоритму и используется в некоторых тестах исключения доминированных векторов. Если отказаться от этих тестов, то можно отказаться и от списка $S$. Запись $T^k$ списка $T$ должна содержать информацию, необходимую для восстановления соответствующей ей подзадачи (7), а запись $S^i$ списка $S$ должна определять некоторое псевдорешение $z^i$ задачи (1)—(4). Поэтому обозначим

$$T^k = (x^k, u^k) \quad \text{и} \quad S^i = (z^i).$$

Опишем теперь принципиальную схему работы алгоритма. Исходное состояние списков $T$ и $S$ следующее: $T = \{T^0 = (x^0, u^0)\}$ и $S = \{s^0 = (z^0)\}$, где $x^0 = 0$, $u^0 = \bar{u}$ и $z^0 = x(\beta_0)$ определены на первом этапе алгоритма. Если $az^0 \leqslant a_0$, то $z^0$ есть оптимальное решение задачи (1)—(4). В противном случае рекорду $r$ — стоимости наилучшего уже найденного допустимого решения задачи (1)—(4) — присваивается достаточно большое положительное значение и проводится пошаговый анализ перспективных вершин дерева просмотра.

На каждом шаге по некоторому правилу $R_N$ выбирается одна запись $T^k$ из списка $T$ и, тем самым, одна вершина дерева просмотра вместе с соответствующей задачей (7). По правилу $R_B$ и формулам (8) и (9) рассматриваемая задача разбивается на $n$ новых подзадач. Каждая из них подвергается анализу с помощью ряда тестов. Если хотя бы один из тестов завершается отрицательно, то оцениваемая задача не является перспективной, список $T$ не дополняется и соответствующая вершина дерева просмотра считается терминальной. После оценивания всех новых подзадач запись $T^k$ исключается из списка $T$ и начинается новый шаг. Работа алгоритма завершается, если на некотором шаге $T = \varnothing$.

Приступим к описанию применяемых в алгоритме тестов, которые можно разбить на четыре группы: 1) тесты по допустимости $Fi$, 2) тесты по оптимальности $OPTi$, 3) тест прямого перебора $DS$, 4) тесты по исключению доминированных векторов $DOMi$.

Тесты могут завершаться положительно (задача является перспективной) или отрицательно. Они также дополняют список $S$, определяют новые значения рекорда $r$ и могут вносить поправки в параметры оцениваемой задачи. В дальнейшем предполагается, что оценивается задача, соответствующая вершине $T^t = T^{k,i}$, получаемой при ветвлении от родительской вершины $T^k$ по переменной $x_i$.

*F1.* Если $u_i^t < 0$, т. е. $y_i^k \geqslant u_i^k$, то оцениваемая задача не имеет допустимых решений и тест завершается отрицательно.

*F2.* Этот тест проверяет, удовлетворяется ли линейное ограничение $ay \leqslant a^t$ при векторе верхних границ $u^t$. Пусть $\Delta_j = a_j u_j^t$ и $\Delta_s = \min\limits_{j \in J^-} \Delta_j$. Если $\Delta = a^t - \sum\limits_{j \in J^-} \Delta_j < 0$, то тест завершается отрицательно. В противном случае вычисляется еще $l = \left]\dfrac{\Delta + \Delta_s}{a_s}\right[$ и при $l > 0$ вектору $x^t$ присваивается новое значение $x^t + le_s$, а $u_s^t$ одновременно уменьшается на $l$ единиц.

*OPT1.* Если $r^t = cx^t + cx(\alpha^t) \geqslant r$, то тест завершается отрицательно, так как стоимости всех допустимых решений оцениваемой задачи не меньше найденного рекорда. Если $r^t < r$ и $ax^t + ax(\alpha^t) \leqslant a^t$, то $z^t = x^t + x(\alpha^t)$ является допустимым решением задачи (1)—(4) и определяет новое значение рекорда $r = r^t$. Вектор $z^t$, как кандидат на оптимальное решение задачи (1)—(4), сохраняется в списке $S$, а тест завершается отрицательно.

*OPT2.* Этот тест определяет стоимость $\bar{r}^t$ оптимального непрерывного расширения вектора $x^t$, т. е.

$$\bar{r}^t = cx^t + \min\{cy \mid ay \leqslant a^t,\ 0 \leqslant y \leqslant u^t\}.$$

Если $\bar{r}^t \geqslant r$, то тест завершается отрицательно.

*DS.* Этот тест прямого перебора всевозможных расширений вектора $x^t$ предназначен для сокращения числа записей в списке $T$. Прежде всего вычисляются величины

$$u_s = u_s^t = \max_j u_j^t \quad \text{и} \quad N_t = (u_s + 1)^{-1} \prod_{j=1}^{n} (u_j^t + 1).$$

Если $N_t > K$, где $K$ — заданная константа, то оцениваемая задача не подлежит прямому перебору и тест завершается положительно. В противном случае образуется $N_t$ всевозможных различных целочисленных векторов $y$ таких, что $0 \leqslant y \leqslant u^t$ и $y_s = 0$. Для каждого $y$ делается попытка найти его наилучшее расширение в виде $z = y + le_s$, при котором

$$0 \leqslant l \leqslant u_s, \quad az \leqslant a^t, \quad \beta z = \alpha^t \quad \text{и} \quad cz < r - cx^t.$$

Если это удается, то найден новый кандидат $x^t + z$ на оптимальное решение задачи (1)—(4). Он включается в список $S$, а рекорд получает новое значение $r = cx^t + cz$.

*DOM1.* В этом тесте, как и во всех остальных тестах по доминированию, вектор $x^t$ сравнивается с некоторым другим вектором (или векторами). Сравнение можно проводить с любым неотрицательным целочисленным вектором $y$, при котором $\beta y = \beta x^t$. В данном тесте $y = x(\beta_0 - \alpha^t)$. Если $x^t \succ x(\beta_0 - \alpha^t)$, то тест завершается отрицательно, так как все расширения вектора $x^t$ являются доминированными векторами.

*DOM2.* Здесь $x^t$ сравнивается с векторами $z^s - x(\alpha^t)$, где $z^s$ — некоторое псевдорешение из списка $S$. Если найдется хотя бы одно $z^s \geqslant x(\alpha^t)$ такое, что $x^t \succ z^s - x(\alpha^t)$, то тест завершается отрицательно. Кроме того, если $z_l^s < x_l(\alpha^t)$, $z_j^s \geqslant x_j(\alpha^t)$ при $j \neq l$ и остальные условия доминирования $x^t \succ z^s - x(\alpha^t)$ выполняются, то значением $u_l = x_l(\alpha^t) - z_l^s - 1$ можно пользоваться в качестве $l$-го компонента вектора $u^t$ (если, конечно, $u_l < u_l^t$).

*DOM3.* Тест работает аналогично тесту *DOM2*, но сравнение $x^t$ проводится с векторами $z^s + x(-\alpha^t)$, которые имеют всегда неотрицательные компоненты.

*DOM4.* Тест сравнивает вектор $x^t$ с вектором $x^k + x(\alpha^k) - x(\alpha^t)$, работает аналогично тесту *DOM2* и может определять один новый компонент вектора $u^t$.

Сделаем еще некоторые замечания относительно рассмотренных выше тестов:

1. Описаны только те тесты, которые учитывают специфику решаемой задачи и не требуют решения каких-либо вспомогательных задач (значение $\bar{r}^t$ выписывается по простой формуле).

2. Если отказаться от сравнительно трудоемкого теста *DOM2*, можно в списке $S$ сохранить только пары чисел $(az^i,\ cz^i)$, соответствующие недоминированным псевдорешениям задачи (1)—(4).

Остается описать правило выбора подзадачи $R_N$ и правило ветвления $R_B$, которые определяют стратегию алгоритма и могут существенно влиять на число генерируемых подзадач, на длину списка перспективных вершин $T$ и, в конечном счете, на эффективность алгоритма. Так как проблемам, связанным с выбором стратегии метода ветвей и границ, посвящена обширная литература, в настоящей работе ограничимся некоторыми замечаниями, касающимися прежде всего специфики алгоритма и решаемой задачи.

1. Обычно все правила выбора подзадачи (вершины) разбиваются на правила одностороннего ветвления и правила ветвления вширь. Первые из них рациональны с точки зрения сохранения информации и возможности выбора очередной подзадачи. Идеи одностороннего ветвления реализуются и в настоящем алгоритме. При этом удается отказаться от одновременного генерирования всех подзадач родительской задачи, а также сократить число элементов в каждой записи списка $T$.

2. В случае ветвления вширь обычно пользуются оценками функции на подзадачах. В качестве этих оценок в данном алгоритме могут выступать значения $r^t$ или $\max\{r^t, \bar{r}^t\}$.

3. Оценивать перспективность подзадачи можно также на основе значения $a^t = a_0 - ax^t$, которое показывает, насколько удовлетворяется линейное ограничение (3) задачи (1)—(4). При выборе подзадач с наибольшими значениями $a^t$ можно ожидать быстрого нахождения допустимых решений задачи (1)—(4) и сравнительно малого числа записей в списке $T$.

4. Длину списка $T$ можно регулировать и путем учета в правиле $R_N$ значений компонент вектора $u^t$.

5. Правилом $R_B$ определяется последовательность генерируемых подзадач одной родительской задачи, а тем самым и структура векторов верхних границ (см. (9)), которая также оказывает существенное влияние на работу алгоритма.

Алгоритм реализован на ФОРТРАНе. На основе вычислительного эксперимента (сравнительно малого объема) можно сделать следующие выводы:
1. Из числа тестов по доминированию оказались эффективными тесты *DOM1* и *DOM2*, которые исключили дополнительно (после всех остальных тестов) в среднем около 15% подзадач. Остальные тесты по доминированию существенного влияния на работу алгоритма не оказали.
2. Наиболее удачными оказались те варианты алгоритма, в которых подзадачи были генерированы в порядке убывания компонент вектора верхних границ родительской задачи.

## ЛИТЕРАТУРА

1. Ху Т., Целочисленное программирование и потоки в сетях, М., «Мир», 1974.
2. Нярипя Х., Изв. АН ЭССР, Физ. Матем., 28, № 2, 127—132 (1980).
3. Литвак Б. Г., Найвельт А. В., В сб.: Исследования по дискретной оптимизации, М., «Наука», 1976, с. 141—155.

*Тартуский государственный университет*

Поступила в редакцию
17/X 1980

H. NÄRIPÄ

## ÜKS LINEAARSE LISAKITSENDUSEGA RÜHMAMINIMEERIMISÜLESANDE LAHENDAMISE MEETODEID

Vaadeldava ülesande lahendamiseks on harude ja tõkete algoritmidele iseloomuliku skeemi raames rakendatud rühmateoreetilist lähenemisviisi. Esitatav algoritm on sarnane Wolsey üldistatud dünaamilise planeerimise meetodiga, kuid lahendusprotsessi efektiivsemaks muutmiseks on kasutatud mitmeid spetsiaalseid alamülesannete analüüsi teste. Nende uute testide eesmärk on kõrvaldada vaatluse alt domineeritud tugivektoritele vastavad alamülesanded.

H. NÄRIPÄ

## A METHOD FOR GROUP MINIMIZATION PROBLEMS WITH AN ADDITIONAL LINEAR CONSTRAINT

This paper combines the group theoretical approach with a branch-and-bound search to solve group minimization problems with an additional linear constraint. The proposed algorithm is similar to Wolsey's generalised dynamic programming method, but some special tests for subproblem analyses are used to increase the search efficiency. The general purpose of these new tests is to exclude subproblems corresponding to dominated integer support vectors.