ЭКОНОМИКА И МАТЕМАТИЧЕСКИЕ МЕТОДЫ 1981 Том XVII, вып. 1

АЛГОРИТМ СОКРАЩЕННОГО ПЕРЕБОРА ДЛЯ РЕШЕНИЯ ЦЕЛОЧИСЛЕННЫХ ЗАДАЧ

И. Л. ЕРОХИН, С. Г. КАПЛАН

(Москва)

Рассматривается следующая задача. Оборудование предприятия — станки, каждый из которых имеет несколько технологических состояний. Переналадка станка из одного состояния в другое характеризуется определенной трудоемкостью. За один производственный цикл станок, находящийся в некотором состоянии, выпускает определенный набор изделий. Необходимо проверить, позволяет ли оборудование за ограниченное число производственных циклов удовлетворить некоторую заявку на изделия и, если она выполнима, выбрать план работы каждого станка с минимальной трудоемкостью переналадок оборудования.

Введем обозначения: Q — множество станков; I — номенклатура выпускаемых изделий; J_k — множество состояний станка k, $\bigcup\limits_{k\in Q} J_k=J \forall k\in Q \forall l\in Q\setminus\{k\}: J_k\cap J_l=\varnothing$; $n_k=|J_k|$ — количество состояний станка k; j_k — исходное состояние станка k, $j_k\in J_k$; b_i — количество изделий вида i, которое необходимо выпустить; a_{ij} — количество изделий вида i, выпускаемое станком, находящимся в состоянии j за один производственный цикл; d_k — максимально допустимое число производственных циклов станка k; c_{ij} — трудоемкость переналадки станка из состояния i в состояние j;

$$c_{ij}: \begin{cases} \geqslant 0, \text{ если } i\in J_k, \quad j\in J_k, \\ =0, \text{ если } i=j \text{ или } j=j_k, \\ =\infty, \text{ если } i\in J_k, \quad j\in J_l, \quad k\neq l; \end{cases}$$

$$u_j=\begin{cases} 1, \text{ если к началу производственного цикла} \\ \quad \text{станок находился в состоянии } j, \text{ т. е. } j=j_k, \\ 0 \text{ в противном случае.} \end{cases}$$

В модели используются следующие переменные: x_j — количество производственных циклов, совершаемое станком в состоянии j;

$$y_{ij}=\begin{cases} 1, \text{ если после состояния } i \text{ станок переналаживается} \\ \quad \text{в состояние } j, \\ 0 \text{ в противном случае;} \end{cases}$$

матрица $\|y_{ij}\|$ характеризует последовательность переналадок станков; z_j — порядковый номер состояния j в указанной последовательности.

Математическую модель можно записать как

$$\sum_{k\in Q} P_k(x)\to\min, \tag{1}$$

$$\sum_{j\in J} a_{ij}x_j \geqslant b_i, \quad i\in I, \tag{2}$$

$$\sum_{j\in J_k} x_j \leqslant d_k, \quad k\in Q, \tag{3}$$

$$x_j \geqslant 0, \text{ целое } j\in J, \tag{4}$$

где

$$P_k(x) = \min_y \sum_{i\in J_k} \sum_{j\in J_k} c_{ij}y_{ij}, \tag{5}$$

$$\sum_{j\in J_k} y_{ij} = \operatorname{sign}(x_i+u_i), \quad i\in J_k, \tag{6}$$

$$\sum_{i\in J_k} y_{ij} = \operatorname{sign}(x_j+u_j), \quad j\in J_k, \tag{7}$$

$$z_i - z_j + n_k y_{ij} \leqslant n_k - 1, \; i, \; j\in J_k \setminus \{j_k\}. \tag{8}$$

Неравенства (2) — условия удовлетворения заявки на изделия, формула (3) — ограничения на количество производственных циклов станков. Допущение, что станок после окончания работы переналаживается в исходное состояние, позволяет представить последовательность переналадок в виде замкнутого контура, который описывается системой уравнений (6)—(8). Действительно, если станок работает в состоянии j и (или) оно является исходным, переналадка из этого состояния и в него совершается один раз. В противном случае переналадок, связанных с данным состоянием, быть не должно. Однако ограничения такого рода не исключают нескольких, не связанных между собой контуров.

Чтобы последовательность переналадок станка была единственной и содержала исходное и все состояния, в которых станок работает, необходимо ввести ограничение (8). Идея этого приема [1] в том, что неравенство (8) справедливо для разомкнутого контура и не выполняется для замкнутого. Согласно (6), (7), все последовательности переналадок станка замкнуты. Разомкнем одну из них. Если контур — единственный, то условие (8) выполняется. Если же таких контуров несколько, оно справедливо для разомкнутого и не выполняется для остальных, т. е. в целом. Таким образом, исключаются не связанные последовательности переналадок станка. Разомкнуть последовательность можно исключением из J_k исходного состояния, что и сделано в (8). Минимизация трудоемкости переналадок оборудования достигается с помощью (1), (5). Видно, что целевой функционал (5)—(8) представляет собой параметрическую задачу о коммивояжере, городам которой соответствует подмножество компонент вектора $x+u$ с ненулевыми значениями. В настоящее время разработан ряд алгоритмов решения задачи о коммивояжере [1], поэтому на способе вычисления целевого функционала не останавливаемся. Отметим лишь одно из его свойств — монотонность: $x>y \Rightarrow P(x) \geqslant P(y)$.

Таким образом, задача (1)—(4) представляет собой систему целочисленных линейных неравенств с монотонно неубывающей целевой функцией. Далее предлагается алгоритм решения задачи, которая для удобства представлена в виде

$$\hat{x} = \operatorname*{Arg\,min}_{x \in S \subseteq G} P(x), \tag{9}$$

$$G = \left\{ x \in Z_+{}^n \,\middle|\, \forall k \in Q : \sum_{j \in J_k} x_j \leqslant d_k \right\}, \tag{10}$$

$$S = \left\{ x \in G \,\middle|\, \forall i \in I : \sum_{j \in J} a_{ij} x_j \geqslant b_i \right\}, \tag{11}$$

где G — множество решений задачи; S — множество допустимых решений задачи; $Z_+{}^n$ — целочисленная решетка неотрицательного ортанта пространства R^n; $\hat{x}$ — оптимальное решение.

В алгоритме использована схема неявного перебора, состоящая в следующем. Множество решений G разбивается некоторым образом на конечное число непересекающихся подмножеств $G_l : \bigcup_{l=1}^{m} G_l = G \quad \forall k \forall l : G_k \cap G_l = \varnothing$. полученные подмножества вносятся в качестве элементов в множество, которое здесь будет именоваться списком. Из списка выбирается одно подмножество и к нему применяется ряд критериев, позволяющих установить отсутствие в нем оптимальных $\hat{x} \notin G_l$ или даже допустимых $S \cap G_l = \varnothing$ решений. В этом случае G_l считается исследованным и из рассмотрения исключается. Из списка выбирается новое подмножество и подвергается аналогичному анализу с помощью указанных критериев. Если отсутствие допустимых и оптимальных решений в подмножестве выявить не удается, его разбивают и вновь полученные подмножества вносятся в список. Процесс ветвления (разбиения множества решений на подмножества) продолжается до тех пор, пока список не будет пуст. Если все решения некоторого подмножества являются допустимыми решениями задачи $G_l \subseteq S$ (это определяется с помощью одного из критериев), то существует способ нахождения наилучшего решения на подмножестве

$$\bar{x} = \operatorname*{Arg\,min}_{x \in G_l \subseteq S} P(x).$$

Данная схема дает возможность ограничиться просмотром малого числа решений, причем ее эффективность зависит от выбора критериев оценки подмножества, способа ветвления и организации списка.

1. Способ ветвления. Разбиение множества решений на подмножества осуществляется с помощью одного из неравенств (2). Полагая множество компонент векторов из R^n упорядоченным $J = [1, n]$, введем на R^n бинарные отношения лексикографического следования $\prec$

$$x \prec y \leftrightarrow \forall j \in [1, \ \alpha \leqslant n] : x_j = y_j \& x_\alpha < y_\alpha, \ x \text{ лексикографически меньше } y, \tag{12}$$

$$x \preccurlyeq y \leftrightarrow x \prec y \vee x = y, \ x \text{ лексикографически не более } y, \tag{13}$$

$$x \overset{L}{\prec} y \leftrightarrow \neg \exists z : x \in L \& y \in L \& z \in L \& x \prec z \& z \prec y \& x \prec y, \tag{14}$$

y следует за x на множестве L.

Предлагается следующий способ разбиения множества G, принадлежащего списку, на подмножества. Пусть $L = \{x^1, \ldots, x^{l-1}, x^l, \ldots, x^m\} \subseteq G$, $x^0 = \sup G$, тогда

$$G_l = \{x \in G \mid x^l \preccurlyeq x \prec x^{l-1} \dot{\&} x^l \leqslant x \& x^l \overset{L}{\prec} x^{l-1}\}, \ l \in [1, m], \tag{15}$$

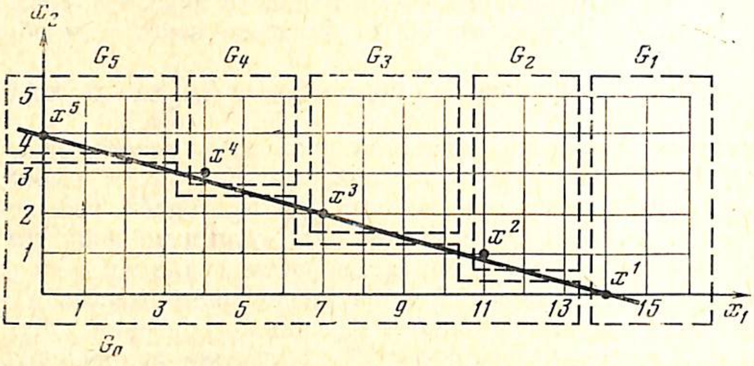
$$G_0 = G \setminus \bigcup_{l=1}^{m} G_l.$$

Если L содержит m элементов, то такое разбиение дает m множеств G_l и одно G_0.

Пусть ветвление осуществляется с помощью неравенства из системы (2) с индексом i

$$L = \left\{ x \in G \,\middle|\, \sum_{j \in J} a_{ij} x_j \geqslant b_i \,\&\, x \to \min \right\}, \tag{16}$$

где L представляет собой множество векторов Парето для неравенства $\sum_{j \in J} a_{ij} x_j \geqslant b_i$. На рисунке представлен двумерный случай разбиения для



Разбиение множества $G = \{x \in Z^2 \mid 0 \leqslant x_1 \leqslant 16 \;\; 0 \leqslant x_2 \leqslant 5\}$ на подмножества $G_0 - G_5$ с помощью неравенства $2x_1 + 7x_2 \geqslant 28$. Вектора $x^1 - x^5$ из множества L изображены точками. Каждое из подмножеств $G_0 - G_5$ ограничено пунктирной линией

неравенства $2x_1 + 7x_2 \geqslant 28$. Нетрудно показать, что полученные таким образом подмножества — непересекающиеся. Кроме того, разбиение (15), (16) обладает другими полезными свойствами: 1) все решения, содержащиеся в подмножествах G_l, $l \in [1, m]$, являются допустимыми для неравенства и ни одно решение из G_0 для него недопустимо; 2) вектор x^l — наименьший в множестве G_l: $x^l = \inf G_l$.

Раскрыв (15) с помощью (12) — (14), получаем

$$G_l = \{ x \in G \mid \forall j \in [1, \alpha) : x_j^l = x_j \,\&\, \forall j \in [\alpha, n] : x_j^l \leqslant x_j \,\&\, x_\alpha < x_\alpha^{l-1} \}, \tag{17}$$

где

$$\alpha \in J \,\&\, \forall j \in [1, \alpha) : x_j^l = x_j^{l-1} \,\&\, x_\alpha^l < x_\alpha^{l-1} \,\&\, x^l \overset{L}{\prec} x^{l-1}. \tag{18}$$

Итак, процедура ветвления осуществляется выбором очередного множества из списка и его разбиением на подмножества с помощью одного из неравенств (2), еще не принимавших участия в ветвлении.

2. Критерии оценки множеств. Важная особенность алгоритма заключается в том, что множество G_0, не содержащее допустимых решений задачи, в список не вносится. Множество G_l, извлеченное из списка, исследуется с участием следующих критериев.

1. Если наименьший элемент x^l множества G_l удовлетворяет всем неравенствам (2), то все решения из G_l являются допустимыми решениями задачи $G_l \subseteq S$. В силу монотонности целевой функции ее значение на наи-

меньшем элементе множества G_l удобно выбрать как оценку этого множества $F(G_l)=P(x^l)$.

Наилучшая оценка для всех полученных допустимых множеств используется в качестве рекорда $z=\min\limits_{G_l\subseteq CS} F(G_l)$.

2. Если оценка некоторого множества не меньше рекорда, то оптимального решения в данном множестве нет: $F(G_l)\geqslant z\longrightarrow \hat{x}\notin G_l$.

3. Если существует такое неравенство, которое не выполняется ни для одного решения из G_l, то в G_l не содержится допустимых решений.

3. Организация списка. Для получения множеств G_l, $l\in[1, m]$ достаточно иметь два алгоритма. Алгоритм A должен находить лексикографически наибольший вектор из L или сообщить, что $L=\varnothing$. Алгоритм B должен либо находить вектор, соседний с вектором x^{l-1} на L, $x^l=B(x^{l-1})$, либо сообщить, что такового не существует. Следует отметить, что критерий 3 оценки множества G_l реализуется применением алгоритма A для каждого из неравенств, не принимавших участия в ветвлении.

Таблица 1

Зависимость времени счета от количества переменных в задаче булева программирования *

n	N	t
10	10	0,3
15	10	2,0
20	10	13,7
25	5	88,4
28	5	226,3
30	5	690,3

* n — число переменных; N — число задач; t — среднее время решения одной задачи, сек.

Алгоритм B позволяет хранить не все подмножества, возникающие при разбиении множества с помощью неравенства, а только одно текущее подмножество G_l. Предполагается, что все предшествующие ему подмножества уже исследованы, а все последующие могут быть получены с помощью алгоритма B. Таким образом, остается поставить в соответствие каждому такому множеству G_l индекс неравенства, с использованием которого оно было получено. Список представляет собой последовательность индексов неравенств (2) (а их не более $|1|$), причем каждому индексу отвечает пара векторов x^l и x^{l-1}. Такая организация списка характеризуется малым временем доступа к данным и требует небольших затрат памяти, что очень важно в комбинаторных задачах. Один из возможных вариантов алгоритмов A и B описан в [2].

Предлагаемый способ ветвления позволяет осуществлять неявный перебор не на всем множестве решений, как, например, у Балаша [3], а только на множестве целочисленных векторов Парето линейных неравенств. Отсюда и название алгоритма — алгоритм сокращенного перебора.

Для исключения влияния времени счета задачи коммивояжера на результаты численного эксперимента алгоритм сокращенного перебора был реализован для задачи линейного программирования с булевыми переменными

$$\sum_{j\in J} c_j x_j \to \min, \tag{19}$$

$$\sum_{j\in J} a_{ij}x_j \geqslant b_i, \quad i\in I, \tag{20}$$

$$x_j\in\{0, 1\},$$

$$a_{ij}\geqslant 0,\ c_j\geqslant 0,\ b_i\geqslant 0,\ i\in I,\ j\in J, \tag{21}$$

и для задачи линейного целочисленного программирования с фиксированными доплатами и ограниченными переменными

Тестовый

i	j									
	1	2	3	4	5	6	7	8	9	10
1	11	12	20	20	20					
2		19	15	12		20	18		12	
3			19	14			9	1	1	5
4				8	16					
5					8	19				
6						5	3	15	11	12
7							19	16	18	8
8								4	17	
9	1		5		2				10	10
10							17	7		13
11										
12										
13		14			10					
14			9			3	14		9	1
15							7	12	20	
16		11								
17			2	10	9	4				
18							19	13	1	4
19									7	
20	7									
c	5	3	12	18	10	9	2	7	12	18
d	11	13	8	20	18	5	16	9	4	18
$\bar{x}$	4	5	2	3	4	2	4	6	7	1
$\hat{x}$	3	0	2	0	0	2	4	0	2	0

Таблица 2

пример № 1

j										b
11	12	13	14	15	16	17	18	19	20	
										70
										9
3	12	20	2							38
							18	8	18	44
									17	33
										29
16	17	7	13	5						160
	4				18		12	19	13	113
						8				21
2										20
10	10	9	5	3						50
	15	14	7	11	11			11	15	78
		7	14							22
			2	10						6
18	8		20	18	11					80
				12	18	8	19	13		90
						20	1	1	4	1
							16	17		9
			12		17			3	14	46
									11	8
12	19	12	19	17	5	5	3	12	18	
5	8	0	17	19	6	10	4	9	1	
1	3	6	7	2	4	1	3	1	3	
1	0	5	0	0	3	0	3	0	0	

Тестовый

i	j									
	1	2	3	4	5	6	7	8	9	10
1	9	3	15	12	18					
2		9	2	7	13	4	17	5	1	4
3			18	12						
4				2	9					
5		6	9		5	1				
6						9	1	4	20	2
7							20	20	20	2
8								3	13	
9			20		1				4	20
10									2	9
11										
12										
13					1					
14						16	18	10	8	
15								16	17	
16	4									
17		15		11						
18							9	1	4	
19										
20	5									
c	2	10	9	5	1	5	3	12	20	20
d	1	14	10	0	6	19	17	14	9	20
$\bar{x}$	4	3	2	5	7	4	7	1	5	3
x	0	0	1	0	1	0	0	0	1	0

Таблица 3

пример № 2

j										b
11	12	13	14	15	16	17	18	19	20	
										19
										6
16		1								5
						4		16	2	3
							5	4	18	16
8							17		4	4
8	17	2	10	8					17	33
		5						5	4	5
										16
4										2
18	9	2								4
	10	8	17			6	6		9	18
		1	4							1
			17	4						6
2	10	8		16	17					33
	16	18	9	2	6	7	13	5	3	30
			14			6	10	10		15
17							6	6		13
				7	11	14		8	17	18
					5	3	11	14	8	2
19	15	13	5	1	5	5	5	2	9	
15	3	17	19	18	4	5	10	0	12	
5	3	3	3	2	5	1	7	4	4	
0	2	0	0	0	0	0	0	3	0	

$$\sum_{j\in J}(c_jx_j+d_j\operatorname{sign}x_j)\to\min, \tag{22}$$

$$\sum_{j\in J}a_{ij}x_j\geqslant b_i,\quad i\in I, \tag{23}$$

$$0\leqslant\underline{x}_j\leqslant x_j\leqslant\overline{x}_j,\ j\in J, \tag{24}$$

$$m=|I|,\ n=|J|,\ a_{ij}\geqslant0,\ c_j\geqslant0,\ d_j\geqslant0,\ b_i>0,$$
$$i\in I,\ j\in J.$$

Параметры задач определялись с помощью генератора псевдослучайных чисел

$$a_{ij}\in R(1,20),\ \text{если}\ a_{ij}>0,$$

$$c_j\in R(1,20),\quad b_i\in R(1,p_i),\ \text{где}\ p_i=0{,}4\sum_{j=1}^{n}a_{ij}\overline{x}_j,$$

$$d_j\in R(0,20),\ \overline{x}_j\in R(0,7),$$

где $R(a, b)$ — множество равномерно распределенных на отрезке $[a, b]$ случайных величин с целыми значениями. Наконец, полагалось $m=n$, $\underline{x}_j=0$, $j\in J$, и коэффициент разреженности матрицы $\|a_{ij}\|$ равен 0,3.

Программы составлены на языке ПЛ-1, эксперименты проводились на ЭВМ ЕС-1022.

В табл. 1 приведены результаты счета по алгоритму, реализующему задачу (19)—(21). В табл. 2 и 3 показаны два тестовых примера для (22)—(24) с $m=n=20$. В центре этих таблиц расположена матрица $\|a_{ij}\|$, в правом столбце — вектор свободных членов — $\{b_i\}$. В нижней части помещены четыре вектора: цены — $\{c_j\}$, доплаты — $\{d_j\}$, верхние границы переменных — $\{\overline{x}_j\}$ и оптимальное решение — $\{\hat{x}_j\}$.

В первом примере решение найдено за 13 мин. 32,02 сек. Оптимальное значение целевой функции — 244. Во втором примере решение найдено за 1 мин. 29,36 сек. Оптимальное значение целевой функции — 94.

Следует отметить следующее положительное качество алгоритма, не отраженное в представленных экспериментах. При решении исходной задачи (1)—(8) среднее время поиска первого допустимого решения пренебрежимо мало по сравнению со средним временем поиска оптимального решения. Относительно быстро выясняется и отсутствие решения задачи.

Математическая модель (1)—(8) и алгоритм сокращенного перебора положены в основу комплекса программ месячного планирования производства железобетонных изделий в кассетном оборудовании домостроительного комбината, разработанного Управлением АСУС Главмособлстроя совместно с Центральным научно-исследовательским и проектно-экспериментальным институтом автоматизированных систем в строительстве (ЦНИПИАСС) Госстроя СССР.

ЛИТЕРАТУРА

1. *В. С. Танаев, В. В. Шкурба.* Введение в теорию расписаний. М., «Наука», 1975.
2. *А. В. Ланьшин, И. Л. Ерохин.* Алгоритм перебора множества Парето линейного неравенства. В сб. Тезисы докладов четвертой республиканской школы молодых ученых и специалистов по АСУ и автоматизации проектирования. Ташкент, 1978 (ЦК ЛКСМ Узбекистана. Узб. науч.-произв. объединение «Кибернетика» АН УзССР. Узб. респ. совет НТО. О-во «Знание» УзССР).
3. *Э. Балаш.* Аддитивный алгоритм для решения задач линейного программирования с переменными, принимающими значения 0 и 1. Кибернетический сборник. Новая серия. Вып. 6. М., «Мир», 1969.

Поступила в редакцию
8 XII 1978