

МЕТОД РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

В. С. ТАНАЕВ

(Минск)

Появление многочисленных работ по формулировке отдельных практически важных задач в терминах дискретного, в частности, целочисленного линейного программирования потребовало разработки эффективных алгоритмов их решений. В настоящее время мы располагаем алгоритмами, обладающими значительной универсальностью, такими, например, как алгоритм Р. Гомори решения задач целочисленного линейного программирования и различные его модификации, метод динамического программирования Р. Беллмана, локальные методы оптимизации и др.

Определенный интерес представляет метод «ветвей и границ», предложенный в различных вариантах для решения задач целочисленного линейного и нелинейного программирования [1—4], некоторых комбинаторных задач [5, 6], задач теории расписаний [7, 8]. В основу этого метода положен общий принцип поиска минимального (максимального) значения функции, заключающийся в построении такого покрытия области определения G функции конечным числом множеств $G_1, G_2, \dots, G_s$, при котором значение оптимизируемой функции в некоторой точке области определения оказывается не больше (не меньше) заранее вычисленных нижних (верхних) ограничителей значений функции на множествах $G_1, G_2, \dots, G_s$.

Объем вычислений, необходимых для построения такого покрытия, в значительной степени зависит от оптимизируемой функции, метода расчета ее ограничителей, принципа построения множеств G_i и, как правило, превышает возможности современной вычислительной техники. Тем не менее этим методом оказалось возможным решить ряд дискретных задач сравнительно большого размера.

Мы рассмотрим одно из возможных направлений применения указанного принципа при решении задач целочисленного линейного программирования. Характерной особенностью предлагаемого варианта метода «ветвей и границ» является использование в качестве множеств, покрывающих многогранник допустимых решений задачи, целочисленных многогранников относительно произвольной конфигурации, что во многих случаях значительно упрощает процесс поиска оптимального решения (п. 1°). Этот прием, в частности, позволяет получить целочисленное решение транспортной задачи с дополнительными ограничениями достаточно общего вида в результате решения ряда обычных транспортных задач (п. 2°). К сожалению, заранее оценить число (и, следовательно, размер) этих задач не представляется возможным, а результаты решения нескольких конкретных комбинаторных задач (п. 3°) носят скорее иллюстратив-

ный характер и не позволяют сделать сколь-нибудь обоснованных выводов относительно эффективности предлагаемого метода.

1°. Обозначая через A матрицу размером $m \times n$, X и b — вектор-столбцы с n и m компонентами соответственно, c — вектор-строку с n компонентами, задачу целочисленного линейного программирования можно записать в виде

$$cX \rightarrow \max, \quad (1)$$

$$AX \leq b, \quad (2)$$

$$X \geq 0 \quad (3)$$

и компоненты вектора X принимают целочисленные значения.

Будем предполагать, что эта задача имеет конечное решение $X_{\text{опт}}$ и точка $X_{\text{опт}}$ принадлежит некоторому целочисленному многограннику $L = \{X | A'X \leq B', X \geq 0\}$, причем матрица $A' (m' \times n)$ обладает свойством унимодулярности [9]. Для построения L достаточно расширить соответствующим образом допустимую область (2) — (3) до выпуклой оболочки целочисленных точек так, чтобы эта оболочка охватывала допустимую область или ее часть, содержащую точку $X_{\text{опт}}$. В частности, поскольку при решении большинства практических задач известны границы изменения каждой переменной, в качестве L можно использовать соответствующий гиперпараллелепипед.

Обозначим через G — множество целочисленных точек $X \in L$, удовлетворяющих условию (2) (рис. 1, а).

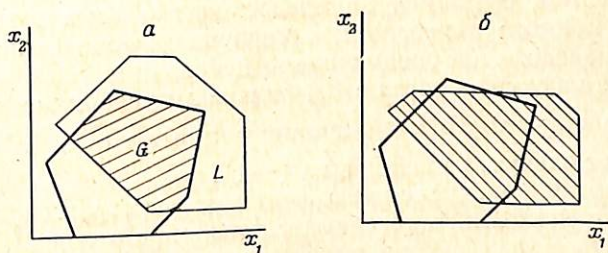


Рис. 1

Задача, очевидно, заключается в нахождении такого покрытия множества G конечным числом множеств $G_1, G_2, \dots, G_s$, чтобы значение функции cX в некоторой точке $X^* \in G$ оказалось не менее верхних ограничений на множествах $G_1, G_2, \dots, G_s$. В качестве значений этой функции на множествах, получаемые из L последовательным наложением дополнительных ограничений вида $x_j \leq r_j$ или $x_j \geq r_j$, где r_j — целое неотрицательное число (рис. 1, б).

Для определения точки X^* и построения требуемого покрытия множества G воспользуемся следующим алгоритмом.

Пусть $cX_0^* = \max\{cX | X \in L\}$. Если $AX_0^* \leq b$, то X_0^* — решение задачи (1) — (3). В противном случае нарушаются некоторые (или все) условия (2). Выберем одно из них $A_{t_1}X_0^* > b_{t_1}$. Обозначим $G_{h_1}^1 = \{X | \text{sign}(a_{t_1, h_1})x_{h_1} \leq \text{sign}(a_{t_1, h_1})r_{h_1}\}$, где $r_{h_1} = \max[x_{h_1} - \text{sign}(a_{t_1, h_1}), 0]$. Очевидно, $G \subseteq \bigcup_{k_1=1}^{s_1} G_{h_1}^{k_1}$, где $G_{h_2}^1 = L \cap G_{h_1}^1$, $a_{t_1, h_1}^{s_1} \neq 0$, $1 \leq s_1 \leq s_1$, $k_1 = k_1^1, k_1^2, \dots, k_1^{s_1}$. Чтобы найти верхние ограничители значений функции cX на множествах $G_{h_1}^{k_1}$, достаточно решить s_1 задач $cX_{h_1}^{k_1} = \max\{cX | X \in G_{h_1}^{k_1}\}$, $k_1 = k_1^1, k_1^2, \dots, k_1^{s_1}$. Найдем $\max\{cX_{h_1}^{k_1}\} = cX_{h_1}^{k_1^1}$.

Обозначим $X_i^* = X_{i+1}$. На этом заканчивается первый шаг итерационного процесса.

Пусть X_i^* — точка, полученная на i -м шаге. Если $AX_i^* \leq b$, то X_i^* — искомое решение задачи (1) — (3). В противном случае, повторяя описанные рассуждения, приходим к покрытию множества G множествами $G_{k_j}^j$, $1 \leq j \leq i+1$, $k_j = k_j^1, \dots, k_j^{s_j}$ и $k_j \neq l_j$ при $j \leq i$. Верхние ограничители значений функции cX_j на этих множествах равны соответственно $cX_{k_j}^j = \max\{cX | X \in C_{k_j}^j\}$. Пусть наибольший из этих ограничителей

$$cX_{l_{i+1}}^{p_{i+1}}. \text{ Обозначим } X_{i+1}^* = X_{l_{i+1}}^{p_{i+1}}.$$

Множества $G_{k_j}^j$ получаются из множества L путем наложения дополнительных ограничений вида $x_l \leq r_l$ или $x_l \geq r_l$. Исключая возможность последовательного наложения противоречивых ограничений указанного вида (что не представляет особых затруднений), мы через конечное число шагов найдем значение $X = X_v^*$, удовлетворяющее условию (2). Поскольку значение функции cX в точке X_v^* не меньше верхних ограничителей значений этой функции на множествах, покрывающих множество G , X_v^* — искомое решение задачи (1) — (3). Целочисленность полученного решения гарантирует использование наиболее употребительных методов линейного программирования при вычислении верхних ограничителей значений линейной функции на целочисленных многогранниках. Объем вычислений, очевидно, не превосходит объема вычислений, необходимых для полного перебора целочисленных точек множества L , и в ряде случаев оказывается сравнительно небольшим.

Отметим, что в случае, когда в (2) включены условия вида $A_{t_i} X = b_{t_i}$, и на i -м шаге описанного итерационного процесса $A_{t_i} X_i^* < b_{t_i}$, последнее неравенство следует умножить на (-1) .

2°. Среди задач целочисленного линейного программирования существуют задачи, представляющие значительный практический интерес, для которых область L фактически определяется некоторыми из ограничений (2). Показательной в этом отношении является транспортная задача

$$\sum_{i,j} c_{ij} x_{ij} \rightarrow \min, \quad (4)$$

$$\sum_j x_{ij} = a_i, \quad (5)$$

$$\sum_i x_{ij} = b_j, \quad (6)$$

$$x_{ij} \geq 0, 1 \leq i \leq n, 1 \leq j \leq m, \quad (7)$$

с дополнительными ограничениями

$$\sum_{i,j} d_{ij}^k x_{ij} \leq e_k, \quad 1 \leq k \leq p, \quad (8)$$

и требованием целочисленности решения. Здесь d_{ij}^k равно 0 или 1, a_i , b_j , e_k — целые числа.

Чтобы найти целочисленное решение этой задачи, можно непосредственно использовать описанный алгоритм, поскольку искомая точка $X_{\text{опт}}$ лежит на поверхности многогранника (5) — (7). Оптимальное решение при этом получаем в результате решения ряда обычных транспортных задач. Действительно [10, 11], решение транспортной задачи с матрицей условий A и дополнительными ограничениями $x_{ij} \leq r_i < b_{ij}$, $j = 1, 2, \dots, s$, равносильно решению транспортной задачи, матрица условий которой содержит, вообще говоря, s дополнительных столбцов j'_l , элементы которых равны элементам соответствующих столбцов j_l за исключением $c_{ij'_l} = \infty$, $b_{j'_l} = r_i$ и $b_{j_l} = b_{j_l} - r_i$.

Алгоритм решения общей задачи целочисленного линейного программирования в данном случае принимает следующий вид.

Решая задачу (4) — (7), находим значения $x_{ij} = \bar{x}_{ij}^0$. Если $\bar{x}_{ij}^0$ удовлетворяют (8), то $\bar{x}_{ij}^0$ — искомое решение. Выбирая в противном случае одно из условий (8) $\sum_{i,j} d_{ij}^t \bar{x}_{ij}^0 > e_{t_1}$, решаем s_1 транспортных задач (4) — (7)

с дополнительным ограничением $x_{i_{k_1} j_{k_1}} \leq \max(\bar{x}_{i_{k_1} j_{k_1}}^0 - 1, 0)$ где $k_1 =$

$= k_1^1, \dots, k_1^{s_1}, d_{i_{k_1} j_{k_1}}^{t_1} = 1$, остальные $d_{ij}^{t_1} = 0$. Если значения $x_{ij} = \bar{x}_{ij}^1$, соответствующие наименьшему из найденных s_1 значений оптимизируемой функции, удовлетворяют условиям (8), то $\bar{x}_{ij}^1$ — искомое решение. В случае нарушения хотя бы одного из условий (8) описанные операции повторяются, причем на каждом шаге рассматриваются лишь те значения оптимизируемой функции, соответствующие матрицы условий которых не подвергались изменению до данного шага включения. Дополнительные ограничения, наложенные на задачу (4) — (7) на предыдущих шагах итерационного процесса, сохраняются на последующих.

В результате через конечное число шагов найдем значения $x_{ij} = \bar{x}_{ij}^v$, удовлетворяющие условиям (8) и, следовательно, являющиеся решением задачи (4) — (8), либо убедимся в противоречивости системы (5) — (8).

3°. В качестве примера рассмотрим решение следующей комбинаторной задачи. Требуется построить связный псевдосимметрический граф с заданной полустепенью r_i исхода (захода) каждой вершины $1 \leq i \leq n$, эйлеров контур которого имеет наименьшую длину.

Обозначая через l_{ij} длину дуги (i, j) , $1 \leq i, j \leq n$, эту задачу нетрудно сформулировать в терминах целочисленного линейного программирования [12]

$$\sum_{i,j} l_{ij} x_{ij} \rightarrow \min, \quad (9)$$

$$\sum_j x_{ij} = r_i, \quad (10)$$

$$\sum_j x_{ji} = r_i, \quad (11)$$

$$x_{ij} \geq 0, \quad 1 \leq i, j \leq n. \quad (12)$$

Для каждого подмножества M множества $\{1, 2, \dots, n\}$, содержащего $1 \leq k \leq [n/2]$ элементов:

$$\sum_{i,j \in M} x_{ij} \leq \sum_{i \in M} r_i - 1. \quad (13)$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	276	142	314	150	178	284	70	163	312	283	467	160	515	237	174	416	141	545	312	410
2	174	625	700	374	890	346	218	471	516	520	180	292	372	613	21	662	145	564	130	210
3	807	140	80	437	200	372	464	413	662	307	633	373	870	682	156	713	36	326	210	317
4	412	998	516	34	823	173	62	712	172	340	259	918	686	145	484	160	707	876	46	562
5	267	819	57	152	912	997	993	722	759	264	814	297	252	319	399	662	392	601	648	154
6	815	390	139	520	828	73	150	264	860	517	630	496	416	88	689	900	161	943	156	154
7	135	625	174	93	890	341	698	284	463	485	961	763	94	137	546	516	880	343	127	215
8	364	170	362	291	455	46	339	154	172	752	467	333	578	302	607	174	703	440	178	743
9	163	910	773	477	264	563	401	172	473	214	770	548	957	131	538	438	94	32	778	763
10	700	524	708	312	805	778	283	336	221	927	538	39	571	597	368	171	311	318	664	543
11	680	150	971	620	79	691	792	609	814	637	982	870	171	256	942	190	810	522	144	139
12	264	488	503	668	852	976	700	213	80	71	519	147	853	281	967	698	281	850	413	320
13	141	645	823	417	256	148	97	417	256	270	163	192	656	162	154	441	877	149	993	247
14	413	145	702	172	636	982	562	32	188	162	496	297	154	117	930	198	913	802	607	316
15	961	16	181	274	414	903	139	481	730	381	466	905	662	244	68	225	151	545	650	701
16	312	774	188	186	453	326	176	764	801	313	172	434	224	967	342	416	935	447	150	37
17	178	665	163	227	107	271	154	282	197	399	114	721	307	818	593	613	50	319	371	413
18	398	145	451	482	278	648	215	465	196	520	806	81	518	333	238	165	320	16	750	197
19	116	162	237	181	270	500	120	453	268	665	719	947	755	668	524	174	156	918	437	215
20	413	340	140	686	571	513	828	152	662	538	239	264	215	181	516	35	229	171	215	128
r_i	1	5	3	4	2	4	6	3	3	2	1	5	3	4	8	1	3	7	2	2

Оптимальное решение получено при наложении на задачу (9) — (12) дополнительных ограничений $x_{20, 20} = 0$, $x_{15, 15} \leq 2$, $x_{14, 6} \leq 2$, $x_{12, 18} \leq 2$. Искомый граф изображен на рис. 3. Длина эйлерова контура равна 4451.

На каждом шаге алгоритма из ограничений (13), которым не удовлетворяет полученное на предыдущем шаге решение, выбиралось ограничение с наименьшим числом отличных от нуля коэффициентов. Не исключена возможность иного принципа выбора соответствующего ограничения. Зависимость объема вычислений от принципа выбора ограничения на каждом шаге алгоритма требует дополнительного исследования.

Автор благодарен Л. А. Славиной и Д. Н. Кравчуку за участие в проведении машинного эксперимента.

ЛИТЕРАТУРА

1. A. H. Land, A. G. Doig. An automatic method of solving discrete programming programming; Integer programming. Rev. frans. rech. operat., 1964, v. 8, N 31.
2. G. L. Thompson. The stopped simplex method; I. Basic theory for mixed integer programming; Integer programming. Rev. frans. rech. operat., 1964, v. 8, N 31.
3. M. E. Balas. Un algorithme additif pour la résolution des programmes linéaires en variables bivalentes. C. R. Acad. Sci. Paris, 1964, v. 258, N 15.
4. M. E. Balas. Extension de l'algorithme additif à la programmation en nombres entiers et à la programmation non linéaire. C. R. Acad. Sci. Paris, 1964, v. 258, N 21.
5. Дж. Литл, К. Мурти, Д. Сунни, К. Кэрел. Алгоритм для решения задачи о коммивояжере. Экономика и матем. методы, 1965, т. I, вып. 1.
6. В. С. Танаев. Некоторые оптимизируемые функции одностадийного производства. Докл. АН БССР, 1965, т. 9, № 1.
7. Z. A. Lomnicki. A «branch-and-bound» algorithm for exact solution of three-machine scheduling problem. Operat. Res. Quart., 1965, v. 16, N 1.
8. E. Ignall, L. Schrage. Application of the branch and bound technique to some flow-shop scheduling problems. Operat. Res., 1965, v. 13, N 3.
9. А. Дж. Гофман, Дж. Б. Краскал. Целочисленные граничные точки выпуклых многогранников. Сб. Линейные неравенства и смежные вопросы. М., Изд-во иностр. лит., 1959.
10. Л. В. Канторович, М. К. Гавурич. Применение математических методов в вопросах анализа грузопотоков. Сб. Проблемы повышения эффективности работы транспорта, М., Изд-во АН СССР, 1949.
11. А. Л. Лурье. О математических методах решения задач на оптимум при планировании социалистического хозяйства, М., «Наука», 1964.
12. В. С. Танаев. Об одной задаче теории расписаний. Изв. АН БССР. Сер. физ.-техн. н., 1964, № 4.
13. А. Е. Залесский. Сведение некоторых комбинаторных задач к целочисленному линейному программированию. Изв. АН БССР. Сер. физ.-техн. н., 1965, № 3.

Поступила в редакцию
6 VI 1966

ОБ ОДНОЙ ЗАДАЧЕ ОПТИМАЛЬНОГО ПЛАНИРОВАНИЯ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

А. И. ВЕРЕСКОВ

(Москва)

1. Двухэтапная постановка стохастической задачи линейного программирования подробно рассматривается в книге [1]. Приведем здесь формулировку двухэтапной задачи, ограничиваясь случаем дискретного распределения случайных параметров.

Найти вектор $x \geq 0$, удовлетворяющий условиям

$$Ax = d \quad (1)$$

и требованию, чтобы при любом векторе b^t (из заданного множества значений случайного вектора b) задача

$$Bx + Cy^t = b^t, y^t \geq 0, (f, y^t) \rightarrow \min \quad (2)$$

имела решение. Таким образом, векторы y^t вводятся для того, чтобы компенсировать невязку между Bx и b^t , а величина (f, y^t) представляет собой штраф за перестройку y^t . Критерием для выбора вектора x является требование минимальности ожидаемых затрат

$$(c, x) + \sum_{t=1}^N p_t [\min_{y^t} (f, y^t)] \rightarrow \min.$$

Матрицы A, B, C и векторы c, d, f фиксированы; известны также значения случайного вектора b^t и соответствующие им вероятности $p_t, t = 1, \dots, N$.

В работе [2] описан частный случай двухэтапной задачи, характеризующийся тем, что матрица $C = (E, -E)$. Условия (2) принимают вид

$$Bx + u^t - v^t = b^t, u^t \geq 0, v^t \geq 0. \quad (3)$$

Ясно, что такие условия не накладывают ограничений на выбор вектора x : всякий $x \geq 0$, удовлетворяющий (1), оказывается допустимым. В связи с этим задача с условиями (1) и (3) была названа «полной» задачей (*complete problem*); как отмечается в [2], существуют и другие двухэтапные задачи, которые естественно относить к классу «полных».

В настоящей работе рассматривается двухэтапная задача, у которой условия (2) имеют вид

$$B^t(x + u^t - v^t) = b^t, x + u^t - v^t \geq 0, u^t \geq 0, v^t \geq 0. \quad (4)$$

Нетрудно видеть, что задача с условиями (1) и (4) является полной в том случае, если каждое из множеств $\{z | B^t z = b^t, z \geq 0\}, t = 1, \dots, N$ не пусто. В случае несовместности хотя бы одной из групп условий $B^t z = b^t, z \geq 0$ вся задача планами не обладает.

Ниже излагается возможный экономический смысл задачи и описывается метод ее решения, использующий специфику ограничений (4).