СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

11 - 9557

Й.Г.Митев

МОДИФИКАЦИЯ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ ЦЕЛОЧИСЛЕННЫХ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

1976

11 - 9557

Й.Г.Митев

МОДИФИКАЦИЯ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ ЦЕЛОЧИСЛЕННЫХ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

§1. Алгоритм для решения целочисленной задачи с выпуклой целевой функцией и линейными ограничениями

Рассмотрим задачу целочисленного математического программирования в следующей, довольно общей форме:

$$F(x) \to \min \qquad /1/$$

$$g_i(x) = 0, \quad i = 1, 2, \ldots, m \qquad /2/$$

$$x \geqq 0 \text{ - целые}, \qquad /3/$$

где x – n-мерный вектор; $F(x)$ и $g_i(x)$, $i = 1, \ldots, m$ - выпуклые функции /в частности, линейные/.

Планом /допустимым решением/ называется любой вектор x, удовлетворяющий ограничениям /2/, /3/, а оптимальное решение - допустимое решение, которое минимизирует целевую функцию $F(x)$.

Перед изложением алгоритма ветвей и границ для этой задачи, цель которой - нахождение оптимального решения, докажем следующую теорему.

Теорема. Пусть $x^\circ = (x_1^\circ, \ldots, x_n^\circ)$ является планом задачи /1/, /2/, и для $x_i = x_i' > x_i^\circ$ она не имеет допустимого решения. В таком случае для каждого значения $x_i > x_i'$ задача не будет иметь допустимого решения.

Доказательство. Допустим, что $x^2 = (x_1^2, \ldots, x_n^2)$ является планом задачи /1/, /2/ и $x_i^2 > x_i'$. Тогда существуют числа

$$\alpha \geqq 0, \ \beta \geqq 0, \quad \alpha + \beta = 1 \quad \text{и} \quad \alpha x_i^\circ + \beta x_i^2 = x_i',$$

но в силу выпуклости /2/ $\alpha x^{\circ}+\beta x^{2}$ является планом задачи /1/, /2/, в которой $x_i = x_i'$. Это противоречие доказывает теорему. Доказательство идет аналогично и в случае, если $x_i' < x_i^{\circ}$.

В дальнейшем изложении предполагаем, что можно решать задачу /1/, /2/ для конечного числа шагов /без ограничения на целочисленность/. Ее решение будем называть непрерывным.

Алгоритм метода ветвей и границ /так называемый случай одностороннего ветвления/ состоит в следующем[1].

1. Решаем задачу /1/, /2/, т.е. получаем непрерывное решение задачи ($x_1^{\circ}, x_2^{\circ}, \dots, x_n^{\circ}$). Если в нем все x_i°, $i=1,2,..,n$ являются целыми числами, то задача решена, если нет, - переходим к пункту 2.

2. Выбираем одну из переменных и задаем ей целое значение. Переменная x_i получает значение $[x_i^{\circ}]$ или $[x_i^{\circ}]+1$. Переходим к 3.

3. Решаем непрерывную задачу уже при фиксированной очередной переменной /например, x_i /. Если задача имеет решение, переходим к 5. Если нет, то множество допустимых целых значений переменной x_i сокращается в силу доказанной теоремы. Переходим к 4.

4. Фиксируем заново x_i , если это возможно. Если нет, т.е. возможности фиксации x_i уже исчерпались, возвращаемся к переменной, фиксированной до x_i , и фиксируем ее заново. При этом полностью восстанавливается множество целых допустимых значений для x_i. Переходим к 3.

Если при возвращении допустимые целые значения для первой из фиксированных переменных исчерпались, то переходим к 7.

5. Если все переменные уже фиксированы, т.е. нашли допустимое целочисленное решение, переходим к 6, если нет - фиксируем следующую переменную. При этом используем информацию из последнего решения задачи /1/, /2/. Переходим к 3.

6. Нашли допустимое целочисленное решение. Значение целевой функции F^* для него используется в дальнейшем как ограничение задачи, т.е. к /2/ добавляется ограничение $F(x) \leqq F^*$, или, иными словами, продолжаем искать только такие допустимые целочисленные решения, которые

по значению целевой функции не хуже найденного. Переходим к 4.

7. Конец решения задачи. Оптимальным решением является последнее из найденных допустимых нелочисленных решений. Если такового не существует, то доказано, что задача не имеет решения.

Из описания алгоритма видно, что процесс решения задачи похож на движение по дереву, конечные узлы которого представляют все возможные целочисленные комбинации для вектора x. При этом движении алгоритм можно рассматривать как бы состоящим из двух этапов: на первом достигается оптимум, на втором идет доказательство, что это именно оптимум. В процессе первого этапа могут быть найдены и некоторые другие целочисленные решения, для которых значение целевой функции монотонно убывает. Это ускоряет процесс завершения первого этапа, так как область искомых решений постоянно сужается.

Хотя мы и дали описание алгоритма ветвей и границ для выпуклой задачи целочисленного программирования, его реализация при столь общем классе задач довольно малоэффективна. Дело в том, что на каждом шагу решается непрерывная задача, причем, так как происходит фиксация или рефиксация некоторой из переменных, каждый раз приходится искать начальный план задачи, что чрезвычайно замедляет процесс решения.

В предлагаемом ниже алгоритме при его реализации используется модификация метода ветвей и границ для решения частично-целочисленной задачи с выпуклой целевой функцией и линейными ограничениями. Для решения непрерывной задачи применяется метод линейной аппроксимации целевой функции, описанный в[2].

Итак, решается задача:

$$F(x;y) \to \min \qquad /4/$$

$$Ax + By = d, \quad x \geqq 0, \quad y \geqq 0 \qquad /5/$$

$$x \text{ - целые.} \qquad /6/$$

Множество /5/ предполагается ограниченным, так как в противном случае может оказаться, что мы не в состоянии решить какой-нибудь из промежуточных непрерывных задач.

Метод линейной аппроксимации для решения задачи /4/, /5/ состоит в следующем: находится начальный опорный план множества /5/. Пусть это $(x^1; y^1)$. Вычисляется вектор

$$\Phi_1 = \left(\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \ldots, \frac{\partial F}{\partial y_n}\right)_{(x^1; y^1)}$$

и при ограничении /5/ минимизируется линейная функция

$$(\Phi_1, (x; y)).$$

Пусть минимум достигается в точке $(x^2; y^2)$. После этого функция F минимизируется на отрезке $(x^1; y^1), (x^2, y^2)$ и пусть точкой минимума является $(x; y)_1$. Процесс продолжается таким же образом после вычитания антиградиента функции F в $(x; y)_1$. Решение задачи заканчивается, если

$$(\Phi_k, (x; y)_k - (x; y)_{k-1}) \leqq 0. \qquad /7/$$

Доказательство сходимости метода дано в /2/.

Из описания метода решения непрерывной задачи видно, что если оптимум не находится в угловой точке области /2/, то в общем случае имеем:

$$\lim_{k \to \infty} (x; y)_k = (x, y)_{\text{опт.}},$$

но сходимость может быть очень медленной. Использование условия /7/ в качестве критерия об окончании решения непрерывной задачи не является очень удобным в рамках метода ветвей и границ. В процессе ветвления по дереву в некоторых случаях нет необходимости решать задачу до конца. В связи с этим в программе предусмотрены следующие возможности:

а/ целевая функция ограничена $F(x;y) \leqq \gamma$. Если

$$F(x; y)_k - F(x; y)_{k+1} \leqq \frac{1}{p}(F(x; y)_{k+1} - \gamma),$$

то считаем, что для оптимума ограничение $F \leqq \gamma$ нарушается;

б/ переменная, которую фиксировали, не является последней целочисленной переменной. Решение непрерывной задачи заканчивается, если

$$F(x; y)_k - F(x; y)_{k+1} \leqq \frac{1}{5}(\gamma - F(x; y)_{k+1});$$

в/ все переменные фиксированы и $F(x; y)_k < \gamma$. Решение задачи заканчивается /в этом случае получаем новый допустимый целочисленный план задачи/, если

$$F(x; y)_k - F(x; y)_{k+1} \leqq 10^{-3}.$$

При выполнении численных экспериментов оказалось, что быстродействие алгоритма сильно зависит от выбора константы p. Чем больше p, тем медленнее устанавливается бесперспективность данной ветви. С другой стороны, если p имеет небольшое значение, то это может привести к ошибке /объявить бесперспективной ветвь, которая содержит лучшее целочисленное решение, чем найденное/. Доказать, что существует такое значение p, при котором утверждение а/ является доказанным, нам в общем случае не удалось. В программе, при помощи которой делались эксперименты, $p = 5$.

В процессе экспериментирования была сделана попытка использовать в качестве метода решения непрерывных задач движение по антиградиенту целевой функции. Полученные результаты являются неудовлетворительными в отношении времени решения задачи. Во-первых, даже если непрерывное решение находится внутри допустимой области, после фиксации решение следующей непрерывной задачи очень часто выходит на границу допустимой области. Во-вторых, после каждого шага по направлению антиградиента мы должны проверять, не покинули ли допустимую область, и каждая такая проверка по объему вычислений почти совпадает с одной симплексной итерацией. Очень трудным является и определение величины шага.

В *табл. 1* приведены характеристики решения нескольких примеров при одном и том же органичении, но с раз-

Таблица 1

№	1				2			
А1	11				15			
А2	83				93			
А3	244				385			
А4	75				75			
А5	5				15			
В	лин.	вып.	вып.	вып.	лин.	вып.	вып.	вып.
С1	620	418	954	841	535	317	945	614
С2	815	418	1114	1254	914	342	1072	-
С3	1341	530	1322	1417	3512	463	1178	1025
Д1	754	1364	5719	4275	1240	1319	5518	3873
Д2	793	1364	6090	4908	1363	1456	6740	-
Д3	1075	1507	6231	5364	4248	1615	7052	7233
Е1	2	1	3	2	4	2	3	1
Е2	106	100	109	110	115	110	116	-
Е3	16	26	92	87	74	36	176	238

А1 – число строк,
А2 – число столбцов,
А3 – $\neq 0$ элементы матрицы,
А4 – булевые переменные,
А5 – целые переменные,
В – тип целевой функции,
С1 – число исследованных узлов до 1-го решения,
С2 – число исследованных узлов до оптимума,
С3 – число исследованных узлов до конца вычисления,
Д1 – число симпл. итераций до 1-го решения,
Д2 – число симпл. итераций до оптимума,
Д3 – число симпл. итераций до конца вычисления,
Е1 – число полученных решений,
Е2 – отклонение (в %) 1-го решения от оптимума,
Е3 – время в секундах.

ными целевыми функциями. Таблица содержит следующие данные: размерность матрицы ограничений задачи /число строк и число столбцов/ и отличные от нуля элементы в ней; количество переменных и их тип; исследованные узлы /или количество решенных промежуточных задач при движении по дереву ветвления/ до нахождения первого допустимого целочисленного решения, до нахождения оптимума и до доказательства оптимальности; соответствующее число симплексных итераций; число полученных допустимых целочисленных решений /включая оптимум/ и отклонение /в процентах/ первого полученного решения от оптимального.

Каждый пример решался один раз - с линейной целевой функцией /с использованием программы для решения частично-целочисленных линейных задач/ и три раза - с выпуклой целевой функцией.

Из проведенных экспериментов видно, что для исследования одного узла дерева ветвления при выпуклой целевой функции необходимо сделать в среднем в два раза больше симплексных итераций, чем при линейной целевой функции. В то же время видна и другая, очень важная закономерность, а именно: после нахождения допустимого целочисленного решения /или оптимального/ сравнительно быстро заканчивается второй этап решения задачи /доказательство оптимальности/. Это можно объяснить тем, что при строго выпуклой целевой функции и при наличии допустимых целочисленных точек в окрестности непрерывного решения среди них должен находиться и оптимум. После нахождения допустимого решения и включения в ограничение задачи условия типа $F \leqq \gamma$, даже при небольшом отклонении некоторой переменной значение целевой функции сильно возрастает, и, таким образом, быстро устанавливается бесперспективность данной ветви. При линейных задачах, как известно, дело обстоит не так. Даже при наличии целочисленных точек вблизи непрерывного решения целочисленный оптимум может быть расположен довольно далеко от него. Грубо говоря, можно сказать, что задачи с выпуклой целевой функцией являются значительно более устойчивыми, чем линейные, при поиске целочисленного решения.

§2. Алгоритм для одного класса нелинейных задач с булевыми переменными

Рассмотрим другой класс задач математического программирования - с нелинейной целевой функцией специального вида, линейными ограничениями и булевыми переменными /0 или 1/. Задачи этого типа легко сводятся к линейным путем введения новых переменных и ограничений. Полученная линейная целочисленная задача с булевыми переменными имеет специальную структуру, что позволяет применять к ней сравнительно эффективно модификацию метода ветвей и границ. Пусть дана задача *A*:

$$F(x) = \sum_{j=1}^{p} f_j(x) \to \min \qquad /8/$$

$$f_j(x) = c_j x_{j_1} x_{j_2} \dots x_{j t_j}$$

$$Ax = b, \quad x \in \{0,1\}. \qquad /9/$$

Функцию вида /8/ часто называют позиномом /это название используется в задачах геометрического программирования/.

Практические проблемы, которые могут быть описаны математической моделью типа задачи *A*, возникают, например, при планировании комплекса взаимосвязанных экспериментов, где один из них может быть проведен лишь после окончания некоторых предыдущих. Оптимизационные задачи такого типа встречаются при планировании работ по селекции растений, где эксперименты взаимосвязаны, дорогостоящи и занимают довольно длительные периоды времени.

Исходя из задачи *A*, сформулируем новую задачу следующим образом /задача *B*/:

$$\bar{F}(y) = \sum_{j=1}^{p} c_j y_j \to \min \qquad /10/$$

$$Ax = b \qquad /11/$$

$$\frac{1}{t_j}(x_{j_1} + x_{j_2} + \dots + x_{j t_j}) = y_j + z_j \qquad /12/$$

$$0 \leq z_j \leq 1 - \frac{1}{t_j}, \quad j = 1, 2, \dots, p$$

$$x, y \in \{0, 1\}.$$

Размерность задачи *B* увеличивается по сравнению с задачей *A* на p булевых переменных, p непрерывных переменных и 2p линейных ограничений, где p - число групп в целевой функции. На самом деле число ограничений увеличивается только на p , так как используемая модификация двойственного симплекс-метода позволяет не включать в симплексную таблицу ограничения сверху на переменные.

Покажем, что задачи *A* и *B* эквивалентны.

То, что каждый допустимый план задачи *A* является допустимым и для задачи *B*, очевидно, так как ограничения /12/ допускают любой набор нулей и единиц по отношению к переменным x. По этой причине любой план задачи *B* является планом задачи *A* /только переменные x /. Если имеем $x_{j_1} = x_{j_2} = \dots = x_{j t_j} = 1$, то $f_j(x) = c_j$, но из j -того ограничения /12/ следует, что $y_j = 1$, следовательно, c_j участвует в целевой функции $\bar{F}(y)$. С другой стороны, если хотя бы одно $x_{jk} = 0$, то $f_j(x) = 0$ и $\frac{1}{t_j}(x_{j_1} + x_{j_2} + \dots + x_{j t_j}) < 1$, следовательно, $y_j = 0$.

Таким образом, эквивалентность задач *A* и *B* доказана.

При численных экспериментах для решения задачи *B* была использована программа решения частично-целочисленной линейной задачи с некоторыми изменениями:

1. Все целочисленные переменные должны быть булевыми.

2. В процессе ветвления первыми фиксируются переменные y .

3. После фиксации всех целочисленных переменных /по достижении допустимого целочисленного плана/ ветвление начинается с последней фиксированной перемен-

Таблица 2

№	1		2		3		4	
A1	11		15		11		15	
A2	80		80		100		100	
A3	244		385		410		532	
A4	80		80		100		100	
"/	M	H	M	H	M	H	M	H
C1	325	283	552	315	604	632	551	287
C2	714	283	1043	368	–	703	815	–
C3	970	347	3034	492	2350	1055	2040	3436
Д1	492	317	927	446	933	982	905	398
Д2	1025	317	1614	504	–	1016	1203	–
Д3	1173	351	4192	815	3419	1220	2942	4416
E1	3	1	4	2	2	2	3	1
E2	130	100	114	107	–	118	112	–
E3	17	5	49	14	70	21	64	113

"/ M – как обычная линейная целочисленная задача,
H – с программой, в которой предусмотрены особенности задачи.

ной y. Это можно сделать, так как после фиксации всех переменных y целевая функция больше не изменяет своего значения /т.е. не зависит от фиксации переменных x /. Таким образом, мы можем потерять некоторые из допустимых целочисленных решений с таким же значением целевой функции, но с другой стороны, время решения задачи значительно сокращается.

Экспериментальные результаты даны в *табл. 2.* Структура таблицы аналогична структуре *табл. 1.* В первом столбце даны результаты решения задачи как обычной с булевыми переменными. Во втором столбце - с использованием особенности задачи *B* /пункты 3 и 4/.

Литература

1. *М.Д.Иванчев, Й.Г.Митев, Н.И.Янев. Сб. "III пролетна конференция по математика и механика - Бургас, 1974", София, 1975.*
2. *У.И.Зангвилль. Нелинейное программирование - единый подход. Москва, "Советское радио", 1973.*

Рукопись поступила в издательский отдел
24 февраля 1976 года.