EESTI NSV TEADUSTE AKADEEMIA TOIMETISED.
FÜÜSIKA * MATEMAATIKA
ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ФИЗИКА * МАТЕМАТИКА
PROCEEDINGS OF THE ACADEMY OF SCIENCES OF THE ESTONIAN SSR.
PHYSICS * MATHEMATICS

1988, 37, 1

https://doi.org/10.3176/phys.math.1988.1.12

УДК 519.854.62

Л. КИВИСТИК

ОБ УСКОРЕНИИ ПЕРВОГО АЛГОРИТМА ГОМОРИ

L. KIVISTIK. GOMORY ESIMESE ALGORITMI KIIRENDAMISEST

L. KIVISTIK. ON ACCELERATING GOMORY'S FIRST ALGORITHM

(Представил Г. Вайникко)

Рассмотрим первый алгоритм Гомори [1, 2] для решения задач целочисленного линейного программирования. Как известно, алгоритмы отсечения отличаются своей нерегулярностью ([2], с. 209). Поэтому представляют интерес приемы ускорения этих алгоритмов, которые в то же время уменьшают их нерегулярность. Ниже будет представлен вариант первого алгоритма Гомори, который по данным вычислительного эксперимента оказался наилучшим среди многих возможных вариантов этого алгоритма.

Рассмотрим задачу целочисленного программирования в следующем виде: максимизировать функцию

$$x_0 = a_{00} + \sum_{j\in J} a_{0j}(-x_j) \tag{1}$$

при условиях

$$x_i = a_{i0} + \sum_{j\in J} a_{ij}(-x_j) \geqslant 0 \quad (i \in I), \tag{2}$$

$$x_j = -(-x_j) \geqslant 0 \quad (j \in J), \tag{3}$$

$$x_j \text{ — целое } (j = 0, 1, \ldots, N), \tag{4}$$

где I — множество базисных, а J — множество небазисных индексов, причем $I \cup J = \{1, 2, \ldots, N\}$. Предположим, что вначале все коэффициенты a_{ij} — целочисленные. После каждой симплексной итерации вид задачи (1)—(4) сохраняется, изменяются только множества I, J и коэффициенты a_{ij}. Поэтому вид (1)—(4) можно считать текущим видом рассматриваемой задачи.

Предположим, что симплексная таблица задачи уже преобразована к оптимальному виду, т. е. она допустима и l-нормальна. Если какая-то дробная часть $\{a_{k0}\}$ больше нуля, то правильным отсечением является неравенство

$$x_{N+1} = -\{a_{k0}\} - \sum_{j\in J} \{a_{kj}\}(-x_j) \geqslant 0 \quad (x_{N+1} \text{ — целое}). \tag{5}$$

Обычно предполагают [2], что отсечение (5) строят по первой из таких строк, где $\{a_{k0}\}>0$. Будем называть этот вариант **базовым** вариантом первого алгоритма Гомори. Базовый вариант всегда конечен. Вектор коэффициентов каждого отсечения (5) генерирует аддитивную циклическую группу векторов коэффициентов правильных отсечений [1]. Порядок этой группы равен или общему знаменателю D элементов симплексной таблицы, или некоторому делителю этого. Операцией группы является сложение по модулю 1. Соответствующие отсечения имеют вид

$$x_{N+1}^{(u)}=-\{ua_{k0}\}-\sum_{j\in J}\{ua_{kj}\}(-x_j)\geqslant 0,\quad u\text{ — целое число.}\qquad (6)$$

Таким образом, более сильное отсечение можно искать среди отсечений (6). По данным вычислительного эксперимента, проведенного автором настоящей статьи, наилучшим вариантом первого алгоритма Гомори оказался следующий (будем его называть вариантом, использующим наиглубокие отсечения).

1. Решим соответствующую непрерывную задачу (1)—(3), преобразуя симплексную таблицу к l-нормальному виду (предположим, что это возможно).

2. Если решение нецелочисленно, построим отсечение (5) по первой сверху такой строки, где a_{k0} дробное, и по правилам лексикографического двойственного симплекс-метода определим для этого отсечения индекс l ведущего столбца.

3. Среди отсечений (6) найдем наиглубокое относительно оси x_l.

4. В случае возможности усилим еще это отсечение (получим новое отсечение вида (6)).

5. Для усиленного отсечения найдем ведущий столбец по правилам лексикографического двойственного симплекс-метода.

6. Вычислим приращение целевой функции как при базовом (5), так и при усиленном отсечении (6).

7. Если абсолютная величина приращения при усиленном отсечении больше абсолютной величины приращения при базовом отсечении, используем усиленное отсечение, в противном случае — базовое отсечение.

8. При помощи лексикографического двойственного симплекс-метода преобразуем симплексную таблицу снова к оптимальному виду. При этом добавленное отсечение отбросим сразу после одного симплексного шага и в дальнейшем не восстановим. Вернемся к пункту 2.

Конечность такого варианта первого алгоритма Гомори непосредственно следует из конечности базового варианта.

Остановимся подробнее на пунктах 3 и 4 этого варианта. Обозначим $\{a_{kl}\}=h_l/D$ и через (h_l, D) — наибольший общий делитель целых чисел h_l и D. Наиглубокое отсечение (6) можно найти описанным в [1] способом. Для этого нужно взять $u=D/(h_l, D)$, если h_0 не делится на (h_l, D); тогда $\{ua_{kl}\}=0$ и $\{ua_{k0}\}>0$. Если h_0 делится на (h_l, D), то нужно найти u, при котором $\{ua_{kl}\}=(h_l, D)/D$ (это минимальное возможное положительное значение дробной части $\{ua_{kl}\}$ при изменении u). Такое целое число u определяется равенством $(h_l, D)=uh_l-vD$ и может быть найдено, например, как числитель предпоследней подходящей дроби непрерывной дроби рационального числа D/h_l в предположении, что оно разложено в непрерывную дробь с нечетным числом элементов. Как для нахождения элементов подходящей дроби, так и для нахождения наибольшего общего делителя (h_l, D) можно использовать алгоритм Евклида (см., напр., [3], с. 70—80). Усиление полученного отсечения (пункт 4) может быть возможным, если $\{ua_{k0}\}=$ $=h_0/D<1/2$. В таком случае найдем максимальное натуральное число u_0, при котором $u_0 h_0/D < 1$, и переумножим генерирующую строку еще на u_0. Тогда отсечение

$$x_{N+1}^{(uu_0)} = -\{uu_0 a_{k0}\} - \sum_{j \in J} \{uu_0 a_{kj}\}(-x_j) \geqslant 0 \qquad (7)$$

не слабее отсечения (6), причем его глубина относительно оси x_l не изменяется.

В ходе вычислительного эксперимента сравнивали описанную модификацию с базовым алгоритмом и приемом Мартина [4], который, по данным литературы (см., напр., [2], с. 202), давал достаточно хорошие результаты. Эксперимент провели на персональной ЭВМ «Apple II+», которая в ходе вычислений сохраняет 9 значащих цифр. Вычисления проводили точно с обыкновенными дробями (см. [1, 4]), так как при вычислении с десятичными дробями возникало быстрое накопление ошибок округления, вследствие чего правильный результат не был гарантирован, если число итераций превышало 30—40. Переход к обыкновенным дробям увеличил время решения примерно на 40%, но зато позволял решить задачи, требующие тысячи итераций. Если числитель или знаменатель дробей становились больше, чем 9-значные целые числа, то такая ситуация истолковывалась как «переполнение» и решение задачи прекращали (хотя такое «переполнение» не всегда приводило к неправильному результату).

В таблице приводятся средние данные для 90 из 100 тестовых задач, которые использовали в [5]. Для сравнения прибавлены соответствующие данные из [5] в случае полностью целочисленного (третьего) алгоритма Гомори. Данные для остальных 10 задач (задачи № 5—8, 16, 17, 19, 24, 25, 27 из [5]) не было возможно использовать, так как при решении их возникало «переполнение» уже после небольшого количества итераций базового варианта или приема Мартина. Однако включены данные в случае двух задач, решение которых не доведено до конца базовым вариантом первого алгоритма. Причиной прекращения решения этих задач было слишком большое количество итераций (прекращено после 1000 и 800 итераций соответственно).

Из таблицы видно, что использование наиглубоких отсечений приводит к значительному улучшению по сравнению с базовым вариантом и приемом Мартина I алгоритма. Но этот вариант уступает ускоренному варианту [5] полностью целочисленного алгоритма Гомори.

	I алгоритм Гомори			Полностью целочисленный алгоритм	
	Вариант				
	базовый	прием Мартина	наиглубоких отсечений	базовый	ускоренный [5]
Среднее время решения, с	187* 100%	71 38%	33 18%	100* 53%	18 10%
Среднее число итераций	112* 100%	37 33%	18 16%	197* 176%	15 13%
Среднее число отсечений	65* 100%	18 28%	8,6 13%	197* 303%	4.3 7%
Число нерешенных до конца задач (из 100)	12	6	2	6	—

* Если решение не было доведено до конца из-за слишком большого количества итераций, то при вычислении средних использовали фактически затраченное время и достигнутые количества итераций и отсечений.

Наконец, отметим, что в ходе вычислительного эксперимента выяснились некоторые слабые стороны приема Мартина.

1. При использовании этого приема значения целевой функции изменяются немонотонно и часто в больших пределах, что иногда приводит к «переполнению» (6 случаев из 100).

2. В течение большой итерации (т. е. до преобразования симплексной таблицы снова к оптимальному виду) необходимо восстанавливать все предыдущие отсечения, соответствующие слабые переменные которых превращаются в базисную (после окончания большой итерации восстановленные строки можно отбрасывать). В ходе эксперимента попробовали также такой вариант приема Мартина, где этого не делали, но тогда неоднократно встречалось зацикливание.

В ходе вычислительного эксперимента попробовали еще некоторые другие возможные варианты первого алгоритма, в том числе вариант, использующий отсечения (из группы отсечений) с наибольшим свободным членом $\{ua_{k0}\}$. Хотя эти варианты обычно значительно ускоряли решение задач по сравнению с базовым вариантом, они в среднем хуже варианта, использующего наиглубокие отсечения.

ЛИТЕРАТУРА

1. *Gomory, R. E.* In: Recent Advances in Mathematical Programming. New York, McGraw-Hill Book Co., Inc., 1963, 269—302.
2. *Корбут А. А., Финкельштейн Ю. Ю.* Дискретное программирование. М., «Наука», 1969.
3. *Михелович Ш. Х.* Теория чисел. М., «Высшая школа», 1967.
4. *Martin, G. T.* In: Recent Advances in Mathematical Programming. New York, McGraw-Hill Book Co., Inc., 1963, 311—317.
5. *Кивистик Л.* Изв. АН ЭССР. Физ. Матем., 34, № 1, 11—19 (1985).

Тартуский государственный университет

Поступила в редакцию
28/IV 1987