2.7. Выбор ведущего элемента и алгоритм Блэнда, устраняющий зацикливание

В описанном выше симплекс-алгоритме имеется некоторая неопределенность: мы не сказали, как решить, какой именно столбец j (из тех столбцов, которые удовлетворяют условию $c_j - z_j < 0$[1]) ввести в базис, и мы также не сказали, как быть в том случае, если при вычислении θ_0, которое определяет строку l и переменную $x_{B(l)}$, выводимую из базиса, минимум достигается более чем при одном значении i.

Обратимся вначале к вопросе о выборе столбца. К сожалению, здесь нет хорошей теоретической базы и приходится полагаться на эмпирические наблюдения. Старейший и наиболее широко используемый критерий состоит просто в выборе наиболее отрицательного $\bar{c}_j < 0$. Как установлено выше, при возрастании на единицу переменной x_j, вводимой в базис, стоимость изменяется на $\bar{c}_j$, поэтому $\bar{c}_j$ можно рассматривать как производную стоимости по расстоянию в *пространстве внебазисных переменных*. Тогда выбор наиболее отрицательного $\bar{c}_j$ является разновидностью метода наискорейшего спуска, называемого в этом случае методом *внебазисного градиента* [KQ]. Однако отсюда вовсе не следует, что действительное уменьшение стоимости $\theta_0\bar{c}_j$ будет максимально возможным, поскольку мы не знаем θ_0, пока не вычислим отношений для выбора строки. Отсюда вытекает другой подход: выбрать столбец, который дает наибольшее уменьшение стоимости. Этот метод, называемый методом *наибольшего приращения*, требует дополнительных вычислений для каждой операции замещения, но дает возможность достигнуть оптимальности с использованием меньшего числа замещений, чем метод внебазисного градиента.

Увеличение внебазисной переменной x_j на единицу изменяет весь вектор x на

$$x_k = \begin{cases} +1, & k = j, \\ -x_{ij}, & k = B(i),\ i = 1, \ldots, m, \\ 0 & \text{в противном случае.} \end{cases}$$

Следовательно, можно вычислить производную по расстоянию в *пространстве всех переменных* $\bar{c}_j / \sqrt{1 + \sum_{i=1}^{m} x_{ij}^2}$. Подход к выбору столбца, соответствующий этой производной, называется методом *градиента по всем переменным*.

Кун и Куанд [KQ] сообщают о результатах обширных вычислительных экспериментов с этими и другими методами. Результаты для задач с числом строк до 25 указывают, что метод градиента по всем

[1] $z_j = c_B B^{-1} a_j$.

переменным приводит к оптимальному решению, используя меньше замещений, чем методы внебазисного градиента и наибольшего приращения, и является более быстрым. Голдфарб и Рейд [GR] описали быстрый способ вычисления производной по всем переменным и сообщают о хороших результатах при применении метода градиента по всем переменным. Однако читатель должен относиться к этим результатам с некоторой осторожностью. Во-первым, времена вычисления, о которых сообщают Кун и Куанд, указывают на ускорение в лучшем случае вдвое, а такое ускорение времени работы часто может быть результатом изменений в деталях программирования. Во-вторых, случайный класс задач ЛП, использованный в качестве тестов, может не отражать чьих-либо «типичных» задач ЛП. Наконец, важным преимуществом метода внебазисного градиента является простота, и он остается наиболее популярным методом, реально используемым в программировании.

Обратимся теперь к разрешению неопределенностей в процедуре выбора строки. Возможность вырожденности представляет здесь определенную опасность: если в качестве ведущего элемента в симплекс-алгоритме выбран элемент $x_{lj} < 0$ и компонента x_{l0} из бдр равна 0, то $\theta_0 = 0$ и приращение стоимости $\theta_0(c_j - z_j) = 0$. То есть стоимость z не убывает, даже если мы выбираем столбец j, такой, что $c_j - z_j < 0$. Возможно, далее, что, проделав последовательность таких замещений, мы вернемся в исходную точку. Это означает, что алгоритм будет неограниченно выполнять один и тот же цикл (в предположении, что выбор столбца и строки делается детерминированно), а такая ситуация была бы очень нежелательной. Это явление называется *зацикливанием.*

Пример 2.7 [Bel]. Рассмотрим таблицу

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
3	$-\frac{3}{4}$	$+20$	$-\frac{1}{2}$	$+6$	0	0	0
0	$\boxed{\frac{1}{4}}$	-8	-1	9	1	0	0
0	$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	0	1	0
1	0	0	1	0	0	0	1

3	0	-4	$-\frac{7}{2}$	33	3	0	0
0	1	-32	-4	36	4	0	0
0	0	$\boxed{4}$	$\frac{3}{2}$	-15	-2	1	0
1	0	0	1	0	0	0	1

3	0	0	-2	18	1	1	0
0	1	0	$\boxed{8}$	-84	-12	8	0
0	0	1	$\frac{3}{8}$	$-\frac{15}{4}$	$-\frac{1}{2}$	$\frac{1}{4}$	0
1	0	0	1	0	0	0	1

3	$\frac{1}{4}$	0	0	-3	-2	3	0
0	$\frac{1}{8}$	0	1	$-\frac{21}{2}$	$-\frac{3}{2}$	1	0
0	$-\frac{3}{64}$	1	0	$\boxed{\frac{3}{16}}$	$\frac{1}{16}$	$-\frac{1}{8}$	0
1	$-\frac{1}{8}$	0	0	$\frac{21}{2}$	$\frac{3}{2}$	-1	1

3	$-\frac{1}{2}$	16	0	0	-1	1	0
0	$-\frac{5}{2}$	56	1	0	$\boxed{2}$	-6	0
0	$-\frac{1}{4}$	$\frac{16}{3}$	0	1	$\frac{1}{3}$	$-\frac{2}{3}$	0
1	$\frac{5}{2}$	-56	0	0	-2	6	1

3	$-\frac{7}{4}$	44	$\frac{1}{2}$	0	0	-2	0
0	$-\frac{5}{4}$	28	$\frac{1}{2}$	0	1	-3	0
0	$-\frac{1}{6}$	-4	$-\frac{1}{6}$	1	0	$\boxed{\frac{1}{3}}$	0
1	0	0	1	0	0	0	1

3	$-\frac{3}{4}$	20	$-\frac{1}{2}$	6	0	0	0
0	$\frac{1}{4}$	-8	-1	9	1	0	0
0	$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	0	1	0
1	0	0	1	0	0	0	1

Будем, начиная с данного бдр, проводить операции замещения, используя следующие правила разрешения неопределенностей.

(а) Для введения в базис всегда выбирать внебазисную переменную с наиболее отрицательным $\bar{c}_j$.

(б) При появлении неоднозначности в выборе переменной, выводимой из базиса, всегда выбирать базисную переменную с наименьшим индексом. В результате получим следующую последовательность таблиц (ведущие элементы обведены прямоугольником).

После шести операций замещения *мы вернулись к исходному бдр*. При всех промежуточных замещениях новые базисные переменные вводились на нулевом уровне и стоимость не изменялась. Мы будем говорить, что симплекс-алгоритм (с этим специальным правилом замещения) *зацикливается*. □

Стоящую в данный момент перед нами проблему можно рассматривать как проблему такого разрешения неопределенностей в симплекс-алгоритме, которое устраняло бы зацикливание. Иногда сообщается, что зацикливание вообще не возникает в практике, даже несмотря на то что можно построить искусственные примеры. Однако это противоречит одному недавнему сообщению [KS]. Кроме того, формулировки некоторых комбинаторных задач в виде задач ЛП сильно вырождены, и не совсем ясно, можно ли положиться на счастливую случайность избежать зацикливания, не говоря уже об эстетических аспектах.

Простейший путь избежать зацикливания — разрешать неопределенности случайным образом. Тогда с вероятностью 1 мы выйдем из любого цикла. Однако это усложняет программирование проверки отношения и не является настолько же интеллектуально удовлетворительным, как детерминированное правило, гарантирующее конечность симплекс-алгоритма. Поэтому похоже, что такой подход не популярен.

В одном стандартном подходе к устранению зацикливания допускается произвольный выбор столбца, и неопределенности при вычислении θ_0 разрешаются таким образом, чтобы нулевая строка лексикографически возрастала, обеспечивая, таким образом, тот факт, что никакой базис никогда не повторяется. Этот метод обладает тем преимуществом, что допускает произвольное правило выбора столбца. Мы отложим описание этого метода до гл. 14. Ниже будет описан относительно недавний замечательный по своей простоте алгоритм Блэнда [B1], устраняющий зацикливание. Нам потребуется

Lemma 2.3. *Пусть $\bar{c}'$ — строка относительных стоимостей в любой таблице X_1 с единичным базисом, не обязательно соответствующим допустимому решению (т. е. некоторые x_{i0} нулевом столбце могут быть отрицательными). Пусть y — любое решение системы ограничений $Ay = b$, не обязательно соответствующее допустимому решению (т. е. некоторые y_j могут быть отрицательными). Пусть f — стоимость, соответствующая X_1, и g — стоимость, соответствующая y. Тогда $\bar{c}'y = g - f$.*

Доказательство. Непосредственными вычислениями получаем

$$\bar{c}'y = (c' - z')y = c'y - z'y = g - c_B'B^{-1}Ay = g - c_B'B^{-1}b = g - f,$$

поскольку $B^{-1}b$ — нулевой столбец таблицы X_1. $\square$

Theorem 2.9 (алгоритм Блэнда, устраняющий зацикливание). *Пусть столбец для введения в базис в симплекс-алгоритме выбирается по правилу*

$$j = \min\{j : c_j - z_j < 0\}$$

(выбрать прибыльный столбец с наименьшим номером), а строка выбирается по правилу

$$B(i) = \min\{B(i) : x_{ij} > 0 \text{ и } \frac{x_{i0}}{x_{ij}} \leqslant \frac{x_{k0}}{x_{kj}} \text{ для каждого } k, \text{ такого, что } x_{kj} > 0\}$$

(в случае неопределенности для вывода из базиса выбрать столбец с наименьшим номером). Тогда алгоритм заканчивает работу после конечного числа замещений.

Доказательство[2]. Покажем, что предположение о существовании цикла приводит к противоречию. Для того чтобы появился цикл, необходимо, чтобы после конечной последовательности замещений мы вернулись к некоторому бдр. Стоимость z должна оставаться постоянной в течение всего цикла, и значение $x_{i0} = 0$, соответствующее каждому замещению, должно равняться нулю, ибо в противном случае $\theta_0 > 0$, откуда следовало бы, что z убывает. Отсюда в свою очередь вытекает, что нулевой столбец x_{i0}, $i = 1, \ldots, m$ остается постоянным в течение всего цикла.

Отбрасывая строки и столбцы, не содержащие ведущих элементов во время цикла, получим новую задачу линейного программирования, которая также зацикливается и в которой все $x_{i0} = 0$ и z постоянно в течение цикла.

Пусть теперь q — *наибольший* индекс переменной, вводимой в базис во время цикла. Рассмотрим две таблицы: таблицу T_1, после которой x_q вводится в базис, и таблицу T_2, после которой x_q выводится из базиса (рис. 2.1). Обозначим элементы таблиц T_1 и T_2, соответственно через x_{ij} и $\hat{x}_{ij}$, а соответствующие им базисы — через $\mathcal{B}$ и $\hat{\mathcal{B}}$ и пусть столбец p вводится в T_2. Применим лемму 2.3, построив два решения. Для T_1 мы просто используем бдр x_0, отождествляя T_1 с X_1. Для T_2, определим решение y следующим образом:

$$y_j = \begin{cases} 1, & \text{если } j = p, \\ -\hat{x}_{ip}, & \text{если } A_j \in \hat{\mathcal{B}}, \\ 0 & \text{в противном случае.} \end{cases}$$

Заметим, что y хотя и не является ни базисным, ни допустимым, но все же является решением системы уравнений $Ay = b$ и, следовательно,

[2]Доказательство представляет собой упрощенный вариант доказательства Блэнда, приведенного в работе [Ku].

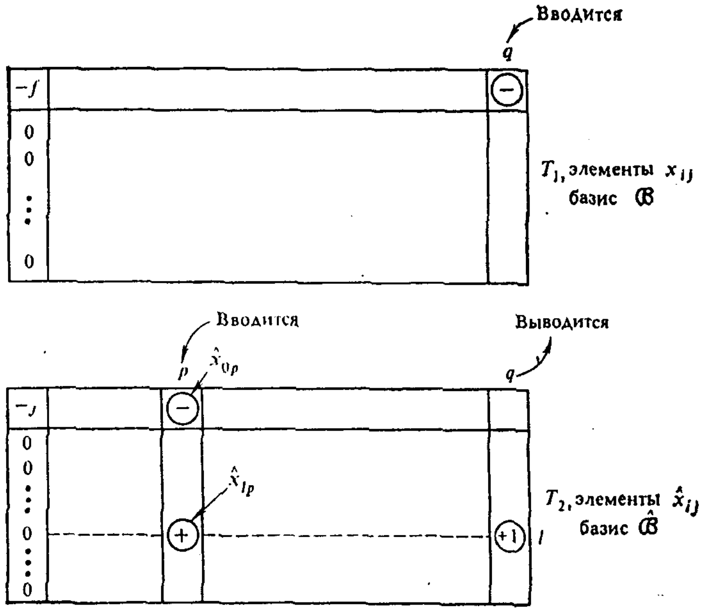

Рис. 2.1. Таблицы T_1 и T_2 из доказательства теоремы 2.9. Переменная x_q вводится в базис в T_1 и выводится из базиса в T_2.

удовлетворяет требованиям леммы 2.3. Кроме того, стоимость решения y равна $f + \hat{x}_{0p}$, поэтому заключение леммы 2.3 дает $\bar{c}'y = \hat{x}_{0p} < 0$. Неравенство вытекает из того, что столбец p вводится в T_2 и, следовательно, должен иметь отрицательную относительную стоимость $\hat{x}_{0p}$.

Согласно выбору ведущего столбца в T_1, получаем

$$\bar{c}_j \begin{cases} \geqslant 0, & j < q, \\ < 0, & j = q, \end{cases}$$

и, согласно выбору ведущей строки в T_2,

$$y_j = \begin{cases} -\hat{x}_{lp} < 0, & j = q, \\ \;\; 0, 1 \text{ или } \; -\hat{x}_{ip} \geqslant 0, & j < q. \end{cases}$$

Следовательно,

$$\bar{c}'y = \sum_{j<q} \bar{c}_j y_j + \bar{c}_q y_q \geqslant \bar{c}_q y_q > 0.$$

Пришли к противоречию. $\square$

Список литературы

[KQ] Kuhn H. W., Quandt R. E. An Experiemntal Study of the Simplex Method, pp. 107–124, in Proceedings of Simposia on Applied Mathematics, vol. XV, ed. N. Metropolis and others. American Mathematical Society, Providence, R.I., 1963.

[GR] Goldfarb D., Reid J. K. A Practicable Steepest-Edge Simplex Algorithm, Math. Prog., 12, No. 3 (June 1977), 361–371.

[Bel] Beale E. M. L. Cycling in the Dual Simplex Algorithm, Naval Research Logistics Quarterly, 2, No. 4 (1955), 269–275.

[KS] Kotiah T. C. T., Steinberg D. I. On the Possibility of Cycling with the Simplex Method, OR, 26, No. 2 (March–April 1978), 374–376.

[B1] Bland R. G. New Finite Pivoting Rules, Discussion Paper 7612, Center of Operations Research and Econometrics (CORE), Université Catholique de Louvain, Heverlee, Belgium, June 1976 (revised January 1977).

[Ku] Kuhn H. W. Class Notes, Princeton University, 1976.