

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ МУЛЬТИПЛИКАТИВНОГО АЛГОРИТМА СИМПЛЕКС-МЕТОДА ПРИ РЕШЕНИИ БОЛЬШИХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ НА ЭВМ

П. А. АХМЕТОВ, У. Х. МАЛКОВ

(Москва)

В работе исследуются разные вычислительные схемы мультипликативного алгоритма симплекс-метода. Авторам удалось разработать вычислительную схему повторения (периодического возобновления мультипликативного представления обратной матрицы), позволяющую хранить ненулевые элементы мультипликативного представления обратной матрицы в массиве ячеек, число которых не более чем в 2 раза превышает количество ненулевых элементов в прямой базисной матрице.

При решении больших задач линейного программирования с малой заполненностью ненулевыми коэффициентами матрицы условий самым эффективным алгоритмом является мультипликативный алгоритм (МА) симплекс-метода с повторениями, так как объем промежуточной ненулевой информации и количество требуемых арифметических действий минимальны относительно других алгоритмов симплекс-метода.

Обозначим исходную матрицу условий задачи линейного программирования через A , а столбцы через P_j ; m — число строк, а n — число столбцов

$$A = [P_1, \dots, P_n], P_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T.$$

Пусть базис на v итерации составляют векторы $P_{n_1}, P_{n_2}, \dots, P_{n_m}$:

$$B = [P_{n_1}, \dots, P_{n_m}].$$

Основной объем в промежуточной ненулевой информации на каждой итерации в случае МА занимает мультипликативное представление обратной базисной матрицы

$$B_v^{-1} = E_{r_v}^v E_{r_{v-1}}^{v-1} \dots E_{r_1}^1, \quad (1)$$

где

$$E_{r_s}^s = \begin{pmatrix} 1 & 0 & \dots & y_{1s} & \dots & 0 \\ 0 & 1 & \dots & y_{2s} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & y_{r_s s} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & y_{ms} & \dots & 1 \end{pmatrix};$$

$$y_{is} = \begin{cases} -x_{ik_s}/x_{r_s k_s}, & i = r_s; \\ 1/x_{r_s k_s}, & i = r_s; \end{cases}$$

$$X_{k_s} = B_s^{-1} P_{k_s} = \begin{pmatrix} x_{1k_s} \\ x_{2k_s} \\ \vdots \\ x_{mk_s} \end{pmatrix}.$$

Здесь v — номер итерации; r_s — номер главной строки на s -й итерации; k_s — номер главного столбца на s -й итерации; P_{k_s} — k_s -столбец в матрице условий A .

Как известно, повторение в мультипликативном алгоритме симплекс-метода заключается в нахождении такого мультипликативного представления (1) матрицы B_v^{-1} , которое содержит ровно m элементарных матриц, т. е. сводится к повторному вводу в базис векторов $P_{n_1}, \dots, P_{n_m}$ при этом главные столбцы и строки выбираются так, чтобы переменные, входившие в последний перед повторением базис, были снова введены в него.

При реализации мультипликативного алгоритма на ЭВМ естественно хранить информацию о матрице B_v^{-1} в виде массива ненулевых компонент y_{is} неединичных векторов элементарных матриц $E_{r_s}^{k_s}$, причем ненулевой элемент y_{is} хранится вместе с индексом строки i в одной ячейке оперативной памяти ЭВМ.

Обозначим количество ненулевых элементов y_{is} ; $s = 1, \dots, v$; $i = 1, \dots, m$, в неединичных столбцах элементарных матриц $E_{r_s}^{k_s}$, образующих B_v^{-1} после повторения, через t . Далее будем называть число t количеством ненулевых элементов матрицы B_v^{-1} . Число t существенно зависит от порядка ввода в базис векторов P_{n_i} , $i = 1, \dots, m$, во время повторения.

Как правило, в программах порядок ввода в базис векторов во время повторения фиксирован: например, векторы вводятся последовательно в порядке убывания индексов n_i . Назовем такой способ обычным повторением. В качестве главной строки при обычном повторении выбираем свободную строку, содержащую максимальный по абсолютной величине элемент разложения вводимого в базис столбца.

Мы предлагаем три схемы повторения, существенно сокращающие количество ненулевых элементов в обратной базисной матрице (и уменьшающие время счета задач). Назовем эти схемы повторениями 1, 2 и 3.

Рассмотрим схему повторения 1. Определим понятие меры связности столбцов. Обозначим количество ненулевых элементов в i -й строке базисной матрицы $B = [P_{n_1}, P_{n_2}, \dots, P_{n_m}]$ через s_i . Вычислим сумму S_{n_i} величин s_i , соответствующих строкам, в которых столбец P_{n_i} имеет ненулевые компоненты

$$S_{n_i} = \sum' s_i,$$

где $\sum'$ обозначает, что сумма берется только для тех строк, которые соответствуют ненулевым элементам вектора P_{n_i} матрицы B . Величина S_{n_i} называется мерой связности вектора P_{n_i} относительно матрицы $B = [P_{n_1}, \dots, P_{n_m}]$. В повторении 1 в качестве вводимого столбца выбирается столбец, имеющий минимальную меру связности, а в качестве главной строки выбирается строка среди свободных, содержащая ненулевой элемент разложения вводимого в базис столбца и имеющая минимальное количество элементов среди свободных строк. Мера связности пересчитывается после каждой итерации относительно матрицы еще не введенных в базис векторов P_{n_i} .

Количество ненулевых элементов в строках и мера связности определяются сначала относительно начальной базисной матрицы $B = [P_{n_1}, \dots,$

$P_{nm}]$, а затем ее подматриц, получаемых вычеркиванием векторов, введенных в базис.

Опишем схему повторения 2. Пусть введены в базис k векторов и вычислены разложения остальных $m - k$ векторов. Вычислим меру связности S_n^i для остальных $m - k$ векторов и количество ненулевых элементов s_i в строках относительно не прямой подматрицы, содержащей $m - k$ векторов P_n , а преобразованной. Главный столбец и главная строка выбирается так же, как в повторении 1.

В повторении 3 сначала выбирается главная строка, в качестве которой используется свободная строка, содержащая минимальное количество ненулевых элементов в разложениях, еще не введенных в базис векторов. А в качестве вектора, вводимого в базис, выбираем столбец, в разложении которого имеется ненулевой элемент в главной строке и который имеет в разложении минимальное количество ненулевых элементов.

Сложность повторения растет в такой последовательности: обычное повторение, повторение 1, повторение 2. Повторение 2 и повторение 3 имеют одинаковую сложность: разница между ними состоит в том, что в повторении 2 выбирается главный столбец, а затем главная строка, а в повторении 3 сначала определяем главную строку, а затем главный столбец.

Повторение 1 отличается от обычного повторения только подсчетом меры связности базисных векторов, поэтому эти повторения занимают примерно одинаковое время. Но количество ненулевых элементов в обратной матрице после повторения 1 на 15—30% меньше, чем после обычного повторения. Повторения 2 и 3 работают в среднем в 1,5—2 раза медленнее, чем обычное повторение, но количество ненулевых элементов в обратной матрице сокращается в среднем в 1,5—3,5 раза по сравнению с обычным повторением.

Во время работы повторений 2 и 3 всю информацию, в том числе и разложения еще не введенных в базис векторов, можно хранить компактно, без ненулевых элементов, так же как в обычном повторении. При вводе базисного вектора P_{n_i} в базис для вычисления мер связности $m - k$ векторов, еще не введенных в базис, нужно получать их разложение в новом базисе на каждой итерации. Для разложения этих векторов можно использовать в памяти поле, отведенное для элементов обратной матрицы B^{-1} , так что дополнительной памяти повторения 2 и 3 не требуют.

Опишем основную вычислительную схему повторения 3. Пусть при повторении введено в базис s векторов. Пусть рабочее поле повторения, состоящее из T ячеек (причем достаточно взять $T \geq 2l$, где l — количество ненулевых элементов в исходной базисной матрице) имеет следующую структуру:

$$\begin{array}{c} B_s^{-1} \quad \{B_s^{-1}P_j\} \quad \{B_s^{-1}P_j\} \\ | \text{---} | \text{---} | \text{---} | \text{---} | \text{---} | \\ 0 \quad t \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad T \end{array}.$$

Здесь частичное мультипликативное представление обратной базисной матрицы после s итераций

$$B_s^{-1} = E_{r_s}^s E_{r_{s-1}}^{s-1} \dots E_{r_1}^1$$

занимает первые t ячеек, при этом записываются только ненулевые элементы неединичных столбцов элементарных матриц $E_{r_i}^i$, $i = 1, \dots, s$. Разложения $X_j^{(s)} = B_s^{-1}P_j$ еще не введенных в базис векторов занимают ячейки начиная с t_3 . Если эти разложения не помещаются в поле t_3, t_4 , то разложения неуместившихся векторов располагаются в поле t_1, t_2 .

При пересчете разложений $X_j^{(s)}$ и записи их в рабочем поле t_3, t_4 перед записью разложения очередного вектора $X_j^{(s)}$ проверяется, сколько ячеек остается до конца рабочего поля T . Если оставшееся поле содержит менее m ячеек, в этом случае запись очередного разложения начинается с ячейки t_1 . Поэтому имеет место неравенство $T - m \leq t_4 \leq T$. Ячейки от t до t_1 (m ячеек, $t_1 = t + m$) отведены для записи информации об очередной элементарной матрице $E_{r_{s+1}}^{s+1}$.

Шаг повторения 3 заключается в следующем:

1. Определим, как указано выше, сначала номер главной строки и затем номер главного столбца.

2. Добавим следующую элементарную матрицу $E_{r_{s+1}}^{s+1}$ к B_s^{-1} , записав ее ненулевые элементы в ячейки, начиная с $t + 1$. После этого t увеличится на количество ненулевых элементов в $E_{r_{s+1}}^{s+1}$.

3. Преобразуем разложения еще не введенных в базис векторов

$$X_j^{(s+1)} = E_{r_{s+1}}^{s+1} X_j^{(s)}$$

в той последовательности, в которой они расположены в ячейках t_3 до t_4 и затем от t_1 до t_2 , причем первое преобразованное разложение (только его ненулевые коэффициенты) запишем в ячейки, начиная с $t_2 + 1$ (т. е. с начала свободного поля), таким образом $t_3' = t_2 + 1$. Следующее преобразованное разложение запишем вслед за первым, продолжая этот процесс до тех пор, пока от конца последнего разложения до T остается более чем $m - 1$ ячеек. Когда в конец рабочего поля уже не поместится очередное разложение, запишем его, начиная с ячейки $t_1 + m$, т. е. взяв $t_1' = t_1 + m$. Разложение введенного в базис вектора не перезаписывается, т. е. количество разложений сокращается на единицу на каждой итерации.

Таблица

Ненулевые элементы	Число ограничений $m = 94$			Число ограничений $m = 64$			
	v_1	v_2	v_3	v_1	v_2	v_3	v_4
l_v	259	302	289	141	189	213	256
t_0	909	1140	808	626	839	*	*
t_1	757	907	557	347	503	901	1169
t_2	504	610	482	270	367	578	714
t_3	259	380	296	191	251	376	414

Приведем результаты некоторых задач, решенных при помощи программы мультипликативного алгоритма с различными повторениями на АЛЬФА-языке на машине БЭСМ-3М. В таблице указаны две задачи линейного программирования с небольшой заполненностью (менее 15%) исходной матрицы условий с числом ограничений $m = 94$ и $m = 64$.

В первой задаче повторения производились на $v_1 = m$, $v_2 = m + 1/4m$, $v_3 = 1/2m$ итерациях, а во второй — на $v_1 = m$, $v_2 = m + 1/3m$, $v_3 = m + 1/3m$, $v_4 = 2m$ итерациях. В таблице через l_v обозначено количество ненулевых элементов в базисной матрице $B_v = [P_{n_1}, \dots, P_n]$ на v -й итерации (но не в обратной к ней!), через t_0 — количество ненулевых элементов в обратной базисной матрице B_v^{-1} после обычного повторения на v -й ите-

рации, t_1 — после повторения 1, t_2 и t_3 — после повторений 2 и 3 соответственно. Символ * означает, что $t > 1300$, т. е. количество ненулевых элементов превышает поле, отводимое для элементов обратной матрицы. В этом случае задача с применением обычного повторения не проходила, т. е. не решалась до конца. Исследования проводились в основном для задач со слабо заполненными матрицами условий (заполненность 4—15%).

Как видно из таблицы, повторение 1 всегда лучше обычного повторения и сокращает количество элементов в обратной базисной матрице почти на одну треть по сравнению с обычным повторением. Из повторений 2 и 3, одинаковых по сложности, повторение 3 всегда лучше, так как оно дает больший эффект.

Повторение 3 сокращает количество ненулевых элементов в обратной базисной матрице примерно в 2—3,5 раза по сравнению с обычным повторением. В случае вычеркиваемости базисной матрицы $B = [P_{n_1}, \dots, P_{n_m}]$ в повторении 3 роста ненулевых элементов в обратной базисной матрице не происходит и $l_v \equiv t_{(3)}^{(v)}$, что видно из таблицы на первом примере при $v_1 = m$.

Результаты показали, что в повторении 3 рост количества ненулевых элементов в матрице B_v^{-1} по отношению к ненулевым элементам базисной матрицы B был от 1,1 до 1,7 раза. Можно предположить, что максимальный рост $t_{(3)}^{(v)}$ в повторении 3 не должен превышать более чем в 2 раза l_v . В обычном повторении $t_0^{(v)}$ в некоторых случаях превышало в 5,5 раза l_v .

Любым из повторений можно воспользоваться для решений системы линейных алгебраических уравнений

$$Ax = b, \quad A = [A_1, \dots, A_m],$$

где A — неособая матрица, определив при помощи повторения мультипликативное представление матрицы A^{-1} и затем вычислив $x = A^{-1}b$. Самое эффективное по точности, времени и размерам решаемых задач при ограниченности памяти повторение 3.

ЛИТЕРАТУРА

1. Г. Зойтендейк. Методы возможных направлений. М., Изд. иностр. лит., 1963.
2. R. P. Tewarson. On the product form of inverses of sparse matrices. SIAM Rev. 1966, v. 8, № 3.

Поступила в редакцию
8 X 1969