УДК 5I2.25.5I9.8

И.В.РОМАНОВСКИЙ , С.С.СУРИН

# ТРИАНГУЛЯРИЗАЦИЯ БАЗИСНОЙ МАТРИЦЫ И МУЛЬТИПЛИКАТИВНЫЙ АЛГОРИФМ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В [2] предложена процедура решения систем линейных алгебраических уравнений высокого порядка с редко заполненной ненулевыми элементами матрицей. Эта процедура описана в рамках метода последовательного улучшения плана [I] .

В настоящей статье мы покажем,как можно её сочетать с мультипликативным методом решения задачи линейного программирования [3] . Поскольку мы еще не располагаем записью этого алгорифма на языке АЛГОЛ-60,мы ограничимся здесь его описанием .

Мультипликативный метод решения задачи л.п. отличается от метода последовательного улучшения плана лишь способом решения систем для определения значений двойственных переменных,соответствующих данному допустимому решению,и коэффициентов разложения вводимого в базис вектора. Этот способ состоит в использовании обратной матрицы в виде произведения элементарных матричных сомножителей.

Рассмотрим $r+k$ -ую итерацию решения задачи мультипликативным методом. Здесь $r$ -номер итерации,вслед за которой началось последнее повторение (итерации, необходимые для выполнения повторения здесь не учитываются), $k$ - целое положительное число.

На этой итерации мы имеем разложение её обратной базисной матрицы на $m+k$ элементарных матричных сомножителей ( $m$ - число условий в задаче), $m$ из которых образуют мультипликативное представление обратной базисной матрицы $r$ - й итерации $B^{(r)}$. Для того, чтобы решить на рассматриваемой итерации систему линейных алгебраических уравнений, надо правую часть её последовательно умножить на все $m+k$ элементарных сомножителей.

Имея в своем распоряжении эффективный метод непосредственного решения систем (например, описанный в [2] ), мы можем заменить умножение на первые $m$ элементарных сомножителей решением указанным методом вспомогательной системы с матрицей $B^{(r)}$, сохраняя умножение на оставшиеся $k$ элементарных сомножителей. Последовательность этих операций зависит от того, какая система решается.

Таким образом, для решения систем в рамках мультипликативного метода на каждой итерации мы должны иметь прямую базисную матрицу некоторой предыдущей итерации $B^{(r)}$ ( $r$ - номер этой итерации) и последовательность элементарных матриц, переводящую её в матрицу текущей итерации.

Если матрица $B^{(r)}$ уже разбита на подматрицы, как это указано в [2] , то решение вспомогательной системы легко находится.

Итак, $r+k$ -ая итерация предлагаемого алгорифма состоит из следующих операций. Здесь $r$, как и выше, номер последней итерации, на которой потребовалось повторение.

I) Умножение справа вектора базисных цен $c_0$ на $k$ элементарных матричных сомножителей. Результат умножения обозначим через $c_0^{(k)}$.

2) Решение вспомогательной системы, доставляющей окончательные значения двойственным переменным:

$$yB^{(r)} = c_0^{(k)} .$$

Здесь $y$ - вектор двойственных переменных.

3) Проверка условий оптимальности и выбор способа, вводимого в базис.

4) Решение вспомогательной системы, дающее промежуточные значения коэффициентов разложения :

$$B^{(r)} x^{(r)} = a .$$

Здесь $x^{(r)}$ - вектор промежуточных значений коэффициентов разложения, а $a$ - вводимый в базис на $r+k$ итерации вектор.

5) Умножение вектора $x^{(r)}$ слева на k элементарных сомножителей. В результате коэффициенты разложения получают окончательное значение.

6) Исправление плана.

7) Построение k+I -го элементарного матричного сомножителя.

Теперь рассмотрим ситуацию, в которой по правилам мультипликативного метода полагается выполнить повторение.

В этой ситуации мы располагаем полным списком базисных векторов, по которым можем построить базисную матрицу, триангуляризовать и разбить её на четыре подматрицы по правилам, описанным в [2]. На следующих итерациях эта матрица заменит матрицу $B^{(r)}$. Выполнив эти действия мы окажемся в условиях текущей итерации. Мы также можем для построенной выше базисной матрицы вычислить и обратить матрицу $B_4$ (см. [2]), которая будет использоваться, пока не наступит очередная ситуация "повторение". Это сократит вычисления при выполнении 2-ой и 4-ой операций текущей итерации. Однако останется необходимость на каждой итерации вычислять векторы

$$c_0^{(1)} - c_0^{(2)} B_4^{-1} B_3$$

и

$$a^{(1)} - B_2 B_4^{-1} a^{(2)}.$$

Используя мультипликативный способ представления базисной матрицы, можно избавиться и от этих вычислений. Для этого надо в качестве матрицы $B^{(r)}$ взять не ту базисную матрицу, которая была построена при "повторении", а матрицу специального вида, которая своими частями $B_2$ и $B_4$ совпадает с последней, а в качестве частей $B_1$ и $B_3$ для нее взяты соответственно треугольная и нулевая матрицы. Эта специальная матрица в силу своего построения сама является треугольной.

Базисная матрица представляется в виде произведения специальной треугольной матрицы и нескольких элементарных матричных сомножителей. Число их равно порядку подматрицы $B_1$ базисной матрицы.

В этом последнем варианте процедура "повторения" может быть кратко описана следующим образом.

I. Формирование базисной матрицы по списку базисных векторов последней выполненной итерации.

2. Полная триангуляризация этой матрицы и разбиение её на части: $B_1$, $B_2$, $B_3$ и $B_4$.

3. Замена в базисной матрице столбцов, составляющих подматрицы $B_1$ и $B_3$ столбцами специального вида (например, единичными ортами). В результате получается треугольная матрица.

4. Построение элементарных матриц, переводящих матрицу, построенную при выполнении 3-ей операции, в матрицу, построенную при выполнении I-ой и 2-ой операций.

Построенный нами алгорифм по быстроте выполнения текущих итераций превосходит алгорифм описанный в [2], т.к. не нуждается в триангуляризации и пересчете базисной матрицы на каждой итерации.

Он также более экономично, чем мультипликативный алгорифм, использует память ЭВМ. Это преимущество обнаруживается после первого же повторения, когда на $r+k$ -ой итерации мультипликативный алгорифм должен хранить $m+k$ элементарных матриц, а для выполнения одной итерации по нашему алгорифму $p+k$ таких матриц, где $p$ - порядок подматрицы $B_1$. Матрица же $B^{(r)}$ (специального вида) может частично восстанавливаться алгорифмически, а частично набираться из массива исходной симплекс-таблицы.

## Литература


I. Канторович Л.В. Экономический расчет наилучшего использования ресурсов. Изд. АН СССР 1959.
2. Сурин С.С. Решение задачи линейного программирования с большим числом нулевых элементов в исходной симплекс-таблице. Настоящий сборник, стр. 29-48.
3. Зойтендейк Г. Методы возможных направлений, ИЛ, 1963.



Поступила в редакцию
I5.XI.1967 г.