РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
АВТОМЕТРИЯ
№ 3 1999

УДК 517.518.8

А. А. Мицель, А. Н. Хващевский
(Томск)

НОВЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ

Рассматривается алгоритм решения задачи квадратичного программирования. Отличием от известных алгоритмов является отсутствие искусственных переменных. В некоторых частных случаях алгоритм сводится к решению системы линейных алгебраических уравнений.

Введение. Одним из классов выпуклого программирования является квадратичное программирование (КП). Задачи квадратичного программирования (ЗКП) встречаются в различных областях науки, техники, экономики [1–4]. Так, одна из постановок обратной задачи лазерного зондирования газового состава атмосферы на основе дескриптивных сглаживающих сплайнов сводится к задаче минимизации квадратичной функции при линейных ограничениях [1]. Другим примером является задача минимизации потерь мощности в электрических цепях [2]. Большой интерес представляет задача выбора оптимизационного портфеля ценных бумаг [3, 4], которая также сводится к поиску экстремума квадратичной функции при линейных ограничениях.

Модель задачи квадратичного программирования. Задача квадратичного программирования имеет следующий вид:

$$f(x) = x'Qx - c'x \to \min_x \tag{1}$$

$$\text{при } Ax \le b, \quad x \ge 0. \tag{2}$$

Здесь x – n-мерный вектор с неизвестными компонентами, подлежащими определению; Q – матрица с компонентами q_{ij} (положительно-определенная); c – n-мерный вектор, определяющий линейный член целевой функции; A – матрица ограничений размерностью $m \times n$; b – m-мерный вектор ограничений.

Известно [2], что необходимые и достаточные условия существования решения задачи (1) выражаются условиями Куна – Такера, которые имеют вид:

$$2Qx - c - \mu + A'\lambda = 0, \tag{3}$$

$$Ax - b + s = 0, \tag{4}$$

$$x, s \geq 0, \quad \mu, \lambda \geq 0, \tag{5}$$

$$\mu' x + s' \lambda = 0. \tag{6}$$

Здесь s – вектор дополнительных переменных, используемый для преобразования ограничения неравенства (2) в ограничения равенства; μ, λ – векторы неопределенных множителей Лагранжа размерностью n и m соответственно.

Одним из методов решения задачи квадратичного программирования может быть метод решения (3)–(6) как задачи линейного программирования с использованием искусственного базиса и модифицированного симплекс-метода [3]. При этом условие (6) неявно учитывается в процедуре симплекс-метода при помощи правила ограниченного ввода в базис переменных s_j и μ_i.

Этот метод применим только в случае положительной определенности матрицы Q. Если матрица Q положительно полуопределена, то метод может расходиться.

Другим методом решения задачи (3)–(6) является так называемый метод решения задачи о дополнительности, разработанный Лемке [3]. В процедуре решения этим методом также предусматривается введение искусственной переменной. Недостаток метода – его громоздкость.

В данной работе предлагается новый алгоритм решения задачи квадратичного программирования, на наш взгляд, более простой в численной реализации.

Описание алгоритма. Разрешим систему (3), (4) относительно переменных x_i и λ_j. Для этого введем составные векторы y, g и матрицу M:

$$y = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad g = \begin{pmatrix} c + \mu \\ b - s \end{pmatrix} = \vartheta + \tilde{I} p, \quad M = \begin{pmatrix} 2Q & A' \\ A & 0 \end{pmatrix}, \tag{7}$$

где $\vartheta = \begin{pmatrix} c \\ b \end{pmatrix}$; $p = \begin{pmatrix} \mu \\ s \end{pmatrix}$; $\tilde{I}$ – единичная матрица, у которой нижние m строк умножены на -1.

В результате получим систему линейных алгебраических уравнений размерностью $(n + m)$:

$$My = g. \tag{8}$$

Если определитель матрицы $\det M$ отличен от нуля, то существует единственное решение системы (8). Это решение будет зависеть от параметров μ и s. Так как параметры μ и s не определены, то решить систему (8) можно лишь при дополнительных условиях (5) и (6), которые полностью доопределяют исходную задачу.

Суть ограничений (5) и (6) состоит в том, что если все компоненты $x_i \geq 0$, $i = \overline{1, n}$; $s_j \geq 0$; $j = \overline{1, m}$, то соответствующие им компоненты μ_i и s_j равны нулю. Следовательно, можно положить $p = 0$ и решать систему

$$My = \vartheta \tag{9}$$

любым классическим методом (например, методом симплексного преобразования системы либо методом Гаусса). Сформулируем теперь алгоритм решения задачи (8).

Шаг 1. Решаем систему (9) и находим y.

Если все $y_i \geq 0, i = \overline{1, n+m}$, то полагаем

$$x_j = y_j, \qquad j = \overline{1, n}, \tag{10}$$

и заканчиваем вычисления.

Если полученное решение (9) содержит отрицательные компоненты $y_i < 0$, то на шаг 2.

Шаг 2. Решаем систему (8) методом симплексного преобразования. Подобная процедура используется при решении задачи линейного программирования для поиска одной из угловых точек [5]. Поскольку решение системы (8) единственно, то найденное решение и будет искомым решением задачи. Отличие метода симплексного преобразования при решении данной задачи состоит в необходимости учета ограничений (5), (6), которые и обеспечивают единственность решения (разумеется, если оно вообще существует). Модифицированный алгоритм симплексного преобразования выглядит следующим образом. Предполагается, что все компоненты $\vartheta_i > 0$. (Если $\vartheta_i < 0$, соответствующие строки системы (8) домножаем на (-1).) Переписываем систему (8) в виде

$$My - Ip = \vartheta \tag{11}$$

или

$$Wz = \vartheta, \tag{12}$$

где

$$W = (M, -\widetilde{I}); \qquad z = (y, p)'. \tag{13}$$

Этап 1. Среди столбцов y коэффициентов W_{ij} матрицы W выбирается столбец, в котором имеется хотя бы один положительный элемент. (Если таких столбцов нет, то решения не существует.) Если в таком столбце несколько положительных элементов, то выбирается тот элемент в столбце, который дает наименьшее частное от деления элементов вектора ϑ на соответствующие элементы этого столбца. Выделенный таким образом разрешающий элемент примем за W_{kj}^0.

Этап 2. Рассчитываются элементы матрицы $W_{il}^{(1)}$ и столбца $\vartheta_i^{(1)}$ по следующим формулам:

$$\begin{gathered}
W_{kl}^{(1)} = \frac{W_{kl}^0}{W_{kj}^0}, \qquad l = \overline{1, L}; \quad L = 2(n+m), \\
W_{il}^{(1)} = W_{il}^0 - \frac{W_{ij}^0}{W_{kj}^0} W_{kl}^0, \qquad i = \overline{1, n+m}; \quad i \neq k; \quad l = \overline{1, L}, \\
\vartheta_k^{(1)} = \frac{\vartheta_k^0}{W_{kj}^0}, \\
\vartheta_i^{(1)} = \vartheta_i^0 - \frac{W_{ij}^0}{W_{kj}^0} \vartheta_k^0, \qquad i = \overline{1, n+m}; \quad i \neq k.
\end{gathered} \tag{14}$$

Исходная

x_1	x_2	x_3	x_4	λ_1	λ_2	λ_3
943,6	70,4	36,8	-14,4	1	1	0
70,4	118,2	2,76	-1,08	1	0	1
36,8	2,76	111,8	-0,56	1	0	1
-14,4	-1,08	-0,56	20,8	1	1	1
1	1	1	1	0	0	0
1	0	0	1	0	0	0
0	1	1	1	0	0	0

Переход на этап 1. Столбец с номером j в матрице $W^{(1)}$ состоит из нулей и единиц ($W_{kj}^{(1)} = 1$). При поиске разрешающего элемента в матрице $W^{(1)}$ не рассматриваем столбец с номером $j + N$, $N = n + m$, который соответствует переменной p_j (условие дополнительной нежесткости (6)). Если при поиске разрешающего элемента в $W^{(1)}$ мы окажемся в столбце с номером $N + r$, то это значит, что на следующем этапе при работе с матрицей $W^{(2)}$ исключается из рассмотрения столбец с номером r, соответствующий переменной y_r (условие (6)).

После выполнения $N = n + m$ преобразований матрицы W и вектора ϑ получим матрицу $W^{(N)}$, в которой N столбцов состоят из нулей и единиц, причем единичные элементы расположены в разных строках и разных столбцах. Номера столбцов, состоящих из нулей и единиц, соответствуют тем номерам

Конечная

x_1	x_2	x_3	x_4	λ_1	λ_2	λ_3
1	0	0	0	0,001	0,002	0
0	1	0	0	0,008	-0,001	0,009
0	0	1	0	0,009	0	0,009
0	0	0	1	0,049	0,049	0,049
0	0	0	0	-0,067	-0,050	-0,066
0	0	0	0	-0,050	-0,051	-0,048
0	0	0	0	-0,066	-0,048	-0,067

Примечание. В матрице результатов при выводе на экран округление чисел производи-

Таблица 1

матрица W

μ_1	μ_2	μ_3	μ_4	s_1	s_2	s_3	ϑ
-1	0	0	0	0	0	0	11,57
0	-1	0	0	0	0	0	4,44
0	0	-1	0	0	0	0	3,79
0	0	0	-1	0	0	0	3,18
0	0	0	0	1	0	0	0,5
0	0	0	0	0	1	0	0,25
0	0	0	0	0	0	1	0,4

переменных, которые совпадают с номерами строк, содержащих единицы на пересечении столбцов, состоящих из нулей и единиц. Значения этих переменных равны значениям компонент вектора $\vartheta^{(N)}$, расположенных в указанных строках.

После того как решение z системы (13) найдено, вычисляем компоненты искомого решения:

$$x_j = z_j, \qquad j = \overline{1, n}. \tag{15}$$

Результаты моделирования. В качестве примера рассмотрим решение следующей задачи квадратичного программирования:

$$f(x) = x'Qx - 11{,}57x_1 - 4{,}44x_2 - 3{,}79x_3 - 3{,}18x_4 \to \min,$$

Таблица 2

матрица W

μ_1	μ_2	μ_3	μ_4	s_1	s_2	s_3	ϑ
-0,001	0,001	0	-0,001	0	0	0	0,01120807
0,001	-0,009	0	0	0	0	0	0,03167195
0	0	-0,009	0	0	0	0	0,03022456
-0,001	0	0	-0,049	0	0	0	0,16310286
0,001	0,008	0,009	0,049	1	0	0	0,26377144
0,002	0,001	0	0,049	0	1	0	0,07568906
0	0,009	0,009	0,049	0	0	1	0,17497951

лось до трех знаков после запятой.

$$x_1 + x_2 + x_3 + x_4 \le 0{,}5,$$

$$x_1 + x_4 \le 0{,}25,$$

$$x_2 + x_3 + x_4 \le 0{,}4, \quad x_1, x_2, x_3, x_4 \ge 0,$$

$$Q = \begin{pmatrix} 471{,}8 & 35{,}2 & 18{,}4 & -7{,}2 \\ 35{,}2 & 59{,}1 & 1{,}38 & -0{,}54 \\ 18{,}4 & 1{,}38 & 55{,}9 & -0{,}28 \\ -7{,}2 & -0{,}54 & -0{,}28 & 10{,}4 \end{pmatrix}.$$

Результаты моделирования приведены в таблицах: в табл. 1 – исходная матрица, в табл. 2 – конечная матрица (после всех преобразований). Шаг 2 алгоритма при решении данной задачи выполняется 7 раз. Искомое решение данной задачи имеет вид: $x^* = (0{,}01120807;\ 0{,}03167195;\ 0{,}03024567;\ 0{,}16310286)$; $f(x^*) = -0{,}451799$.

Заключение. Для анализа результатов работы описанного алгоритма проведено сравнение с методом решения задачи о дополнительности [3]. Сравнение алгоритмов показало, что описанный алгоритм может быть как более, так и менее громоздким по объему вычислений в сравнении с задачей о дополнительности. Если брать максимально возможное число вычислений каждого алгоритма, то преимущество будет отдано вышеописанному алгоритму, так как при одинаковом числе итераций число вычислений в задаче о дополнительности больше ввиду того, что матрица W больше на один столбец (столбец искусственной переменной). Следует также отметить, что объем вычислений в каждом из алгоритмов зависит от искусства программиста. К преимуществу описанного алгоритма можно также отнести следующее:

– для большинства практических задач решение задачи (1) сводится к не представляющему трудностей решению системы (9);

– при решении задачи (1) в общем виде (см. (12)) нет необходимости вводить искусственные переменные и вспомогательную целевую функцию.

СПИСОК ЛИТЕРАТУРЫ

1. **Воскобойников Ю. Е., Катаев М. Ю., Мицель А. А.** Решение обратной задачи зондирования газового состава атмосферы на основе дескриптивных сглаживающих сплайнов // Оптика атмосферы. 1991. **4**, № 2. С. 177.
2. **Базара М., Шетта К.** Нелинейное программирование. М.: Мир, 1982.
3. **Реклейтис Г., Рейвиндран А., Рэгсдел К.** Оптимизация в технике. М.: Мир, 1986. Т. 2.
4. **Первозванский А. А., Первозванская Г. И.** Финансовый рынок: расчет и риск. М.: Финансы и статистика, 1994.
5. **Заварыкин В. М., Житомирский В. Г., Лапчик М. П.** Численные методы. М.: Просвещение, 1990.

Поступила в редакцию 22 июня 1998 г.