УДК 519.852

М.О. ГАВРИЛОВА
Пермский государственный технический университет

О ЗАДАЧАХ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С КУСОЧНО-ПОСТОЯННЫМИ КОЭФФИЦИЕНТАМИ

Рассматривается задача линейного программирования с кусочно-постоянными коэффициентами системы ограничений и целевой функции. Приводятся алгоритмы поиска решения выпуклой задачи с кусочно-постоянными коэффициентами.

Рассмотрим задачу линейного программирования с переменными коэффициентами.

$$\min(\max)\ z(x) = \sum_{j=1}^{n} c_j(x)\, x_j \qquad (1)$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij}(x)\, x_j \geq b_i(x),\ \left(i = \overline{1,m}\right), \qquad (2)$$

$$\left(\sum_{j=1}^{n} a_{ij}(x)\, x_j \leq b_i(x)\right),\ x_j \geq 0,\ \left(j = \overline{1,n}\right), \qquad (3)$$

где $a_{ij}(x)$, $b_i(x)$ и $c_j(x)$ – некоторые кусочно-постоянные функции аргумента $x = (x_1, x_2, \ldots, x_n)$.

Функции $a_{ij}(x)$, $b_i(x)$ и $c_j(x)$ определены на одном и том же множестве $G = \left\{x \in G \subset R^n\right\}$, и существует такое конечное разбиение $G = \bigcup G_k$, $\left(k = \overline{1,l}\right)$, что в каждом подмножестве G_k функции постоянны, причем G_k и G_{k+1} могут пересекаться только по своим границам.

Область допустимых решений M, т.е. множество $x = (x_1, x_2, \ldots, x_n)$, удовлетворяющих системе (2)–(3), во многом в так поставленной задаче зависит от семейства подобластей G и функций коэффициентов $a_{ij}(x)$, $b_i(x)$ и $c_j(x)$. Здесь, безусловно, возможна потеря областью M выпуклости, связности и других свойств, которые обычно используются при решении подобных задач.

Приведем пример нарушения связности области допустимых решений. Рассмотрим область, заданную следующей системой уравнений:

$$\begin{cases} 3x+3y\le 48, \\ x-6y\le -33, \\ x\le 9,\ y\ge 7, \end{cases} \qquad \begin{cases} -2x+3y\le 3, \\ x-6y\le -33, \\ x\ge 9,\ y\ge 7, \end{cases} \qquad \begin{cases} -2x+3y\le 3, \\ x+y\le 16, \\ x\ge 9,\ y\le 7. \end{cases} \tag{4}$$

Область допустимых решений, заданная указанной системой, обозначена на рисунке пунктиром.

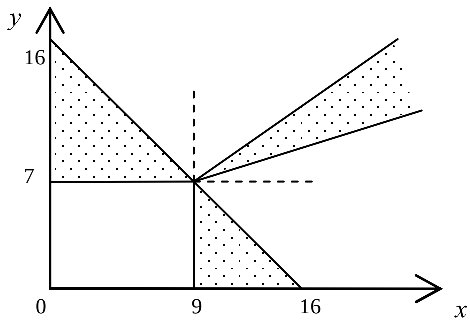

Рис.

Отсюда следует, что говорить о решении поставленной задачи (1)–(3) имеет смысл, если есть дополнительные предположения. Такими предположениями, например, могут являться выпуклость целевой функции и функций ограничений. Целью данной статьи является изучение именно указанных случаев.

Для удобства изложения далее будем рассматривать задачу минимизации функции (1) при соответствующих ограничениях (2), (3).

Прежде всего потребуем дополнительных условий на области G и G_k. Будем считать, что G – параллелепипед: $G=\Pi= =[0,a_1]\times[0,a_2]\times\ldots\times[0,a_n]$. Таким образом, условие (3) можно не рассматривать. Каждое G_k также является параллелепипедом, грани которого параллельны координатным гиперплоскостям. Более того, мы будем считать выполненным следующее требование: существует конечное разбиение параллелепипеда Π, например $\Pi=\bigcup\limits_{k_a}\Pi^a_{k_a}$ $\left(k_a=\overline{1,s_a}\right)$, где

$\left\{\Pi_{k_a}^a\right\}$ – набор параллелепипедов, на каждом из которых функция $a_{ij}\left(x\right)$ является постоянной. Аналогично можно представить $\Pi = \bigcup\limits_{k_b} \Pi_{k_b}^b$ или $\Pi = \bigcup\limits_{k_c} \Pi_{k_c}^c$ $\left(k_b = \overline{1, s_b},\ k_c = \overline{1, s_c}\right)$, где $\left\{\Pi_{k_b}^b\right\}$ и $\left\{\Pi_{k_c}^c\right\}$ – конечные наборы параллелепипедов с постоянными $b_i\left(x\right)$ и $c_j\left(x\right)$ соответственно. Из этого следует, что существует конечное разбиение множества $G = \Pi$ на параллелепипеды $G_k = \Pi_k^{abc}$, такое, что $\Pi = \bigcup\limits_k \Pi_k^{abc}$, $\left\{\Pi_k^{abc}\right\} = \left\{\Pi_{k_a}^a\right\} \cap \left\{\Pi_{k_b}^b\right\} \cap \left\{\Pi_{k_c}^c\right\}$ и функции $a_{ij}\left(x\right)$, $b_i\left(x\right)$ и $c_j\left(x\right)$ постоянны для каждого Π_k^{abc}. При этом соседние параллелепипеды Π_k^{abc} и Π_{k+1}^{abc} пересекаются только по границе.

Наконец, потребуем, чтобы в задаче (1)–(3) целевая функция $z(x)$ и функции ограничений $f_i(x) = \sum\limits_{j=1}^{n} a_{ij}\left(x\right) x_j - b_i\left(x\right)$ были непрерывны и выпуклы. Условия для выпуклости целевой функции и функции ограничений мы не выписываем. Такие условия можно получить, используя условие выпуклости для дифференцируемых функций и аппроксимацию полиэдральных функций дифференцируемыми функциями.

Можно выделить несколько методов решения задачи (1)–(3) при таких предположениях.

1. Самым очевидным из них [1] является простой перебор конечного числа параллелепипедов Π_k^{abc}, т.к. в каждом из Π_k^{abc} мы будем иметь дело с обычной задачей линейного программирования с линейной целевой функцией и выпуклой областью допустимых решений. Для каждого Π_k^{abc} находится оптимальный план x^k, а решением задачи (1)–(3) является x^*, для которого целевая функция минимальна: $z^*(x^*) = \min\limits_k \left\{ z\left(x^k\right)\right\}$.

2. Другой подход к решению задачи (1)–(3) основан на итерационном методе решения экстремальных задач [2]. Для этого используем следующий алгоритм.

На множестве $\Pi = \bigcup\limits_k \Pi_k^{abc}$ зададим некоторую сетку точек с достаточно малым шагом $\alpha > 0$. Отметим, что для каждого параллелепипеда Π_k^{abc} соответствующая часть области допустимых решений системы (2) представляет собой выпуклый многогранник $M_k \subseteq \Pi_k^{abc}$, а целевая функция линейна.

На предварительном этапе решения задачи (1)–(3) находится начальный опорный план на произвольно выбранном Π_k^{abc}. Для этого решается стандартная задача линейного программирования:

$$z^k(x) = \sum_{j=1}^{n} c_j^k x_j \to \min, \tag{5}$$

$$\sum_{j=1}^{n} a_{ij}^k x_j \ge b_i^k,\ i = \overline{1, m}, \tag{6}$$

$$x_j \ge 0,\ j = \overline{1, n}, \tag{7}$$

$$x = (x_1, x_2, \ldots, x_n) \in \Pi_k^{abc}$$

с помощью любого известного метода, например симплекс-методом. В результате решения данной задачи находится точка $x^k \in M_k \subseteq \Pi_k^{abc}$.

Первый шаг алгоритма заключается в переходе из x^k в ближайшую узловую точку сетки, принадлежащую области допустимых решений M, т.е. в такую точку $x_\alpha^k \in M$, для которой выполнено соотношение $\left|x^k - x_\alpha^k\right| = \min\limits_{x_\alpha \in M}\left|x^k - x_\alpha\right|$ по всем окрестным узловым точкам $x_\alpha \in M$.

На втором шаге алгоритма определяются смежные с x_α^k узловые точки сетки $x_{\alpha_t}^k$ $\left(t = \overline{1, p}\right)$, принадлежащие области допустимых решений. Каждая из найденных узловых точек $x_{\alpha_t}^k$ может принадлежать одному или одновременно нескольким параллелепипедам из П. Для каждого «затронутого» точками $x_{\alpha_t}^k$ параллелепипеда решается соответствующая задача линейного программирования (5)–(7), затем среди множества полученных решений выбирается оптимальное x^{k+1}, такое, для которого целевая функция принимает минимальное значение.

Далее опять производится переход из x^{k+1} в ближайшую узловую точку x_α^{k+1} и осуществляется повтор второго шага алгоритма. Алгоритм завершает свою работу, когда найденное решение x^{k+1} окажется равным оптимальному решению x^k, полученному на предыдущем этапе.

3. Далее рассмотрим способ решения, который основан на использовании метода линейной аппроксимации и выпуклого симплекс-метода [3].

Для этого возьмем разбиение множества Π на параллелепипеды Π_k^{ab}, такое, что $\Pi = \bigcup_k \Pi_k^{ab}$, $\left\{\Pi_k^{ab}\right\} = \left\{\Pi_{k_a}^{a}\right\} \bigcap \left\{\Pi_{k_b}^{b}\right\}$. Функции коэффициентов $a_{ij}\left(x\right)$ и $b_i\left(x\right)$ постоянны для каждого Π_k^{ab}, следовательно, для каждого Π_k^{ab} соответствующая часть области допустимых решений задачи является выпуклым многогранником $M_k \subseteq \Pi_k^{ab}$. В силу наложенных ранее условий целевая функция $z(x)$ выпукла и непрерывна на каждом Π_k^{ab}. Таким образом, на каждом Π_k^{ab} мы имеем дело с задачей нелинейного программирования с линейными ограничениями. Методы решения таких задач известны под названиями метода линейной аппроксимации и выпуклого симплекс-метода и подробно рассмотрены, например в работе [3].

Алгоритм решения задачи (1)–(3) заключается в следующем. Как и ранее, решение задачи начинается с нахождения начального опорного плана на произвольно выбранном множестве Π_k^{ab}. Для этого с помощью известных методов необходимо решить задачу нелинейного программирования с линейными ограничениями:

$$z\left(x\right) = \sum_{j=1}^{n} c_j\left(x\right) x_j \to \min, \tag{8}$$

$$\sum_{j=1}^{n} a_{ij}^k x_j \ge b_i^k, \; i = \overline{1, m}, \tag{9}$$

$$x_j \ge 0, \; j = \overline{1, n}, \tag{10}$$

$$x = (x_1, x_2, \ldots, x_n) \in \Pi_k^{ab}.$$

В результате решения указанной задачи находится точка $x^k \in M_k \subseteq \Pi_k^{ab}$.

Далее необходимо осуществить переход к следующему параллелепипеду из Π. Для этого можно предложить итерационный метод, известный как метод проекции субградиента [2]:

$$x_\alpha^{k+1} = P_M\left(x_k - \alpha_k c_k\right), \; \alpha_k > 0, \; c_k \in \partial z\left(x_k\right), \; k = 1, 2, \ldots, \tag{11}$$

где α_k – длина шага, c_k и $\partial z\left(x_k\right)$ – соответственно субградиент и субдифференциал функции $z\left(x\right)$ в точке x_k, P_M – проекция точки

$(x_k - \alpha_k c_k)$ на множество M. Субградиент c_k выбирается из $\partial z(x_k)$ произвольным образом. Для выбора величины α_k в градиентных методах существует насколько способов, наиболее распространенные из них рассмотрены в работе [2].

Найденная с помощью (11) точка x_α^{k+1} определяет некоторый параллелепипед $\Pi_{k+1}^{ab} : x_\alpha^{k+1} \in \Pi_{k+1}^{ab}$, для которого снова решается задача (8)–(10) и находится последующее решение x^{k+1}. Если при некотором k окажется, что $x^{k+1} = x^k$, то процесс прекращается, а x^k является решением задачи (1)–(3).

4. Рассмотрим далее подход к решению задачи (1)–(3), основанный на переходе к двойственной задаче.

Для этого рассмотрим еще один вариант разбиения множества Π на параллелепипеды Π_k^{ac}, таким образом, что $\Pi = \bigcup_k \Pi_k^{ac}$, $\{\Pi_k^{ac}\} = \{\Pi_{k_a}^{a}\} \cap \{\Pi_{k_c}^{c}\}$. В силу разбиения для каждого Π_k^{ac} постоянными являются функции $a_{ij}(x)$ и $c_j(x)$, и для каждого Π_k^{ac} мы можем записать следующую задачу с кусочно-постоянными коэффициентами:

$$z^k(x) = \sum_{j=1}^{n} c_j^k x_j \to \min, \tag{12}$$

$$\sum_{j=1}^{n} a_{ij}^k x_j \ge b_i(x),\ i = \overline{1,m}, \tag{13}$$

$$x_j \ge 0,\ j = \overline{1,n}, \tag{14}$$

$$x = (x_1, x_2, \ldots, x_n) \in \Pi_k^{ac}.$$

Далее для задачи (12)–(14) запишем двойственную задачу:

$$g(y) = \sum_{i=1}^{m} b_i(x) y_i \to \max, \tag{15}$$

$$\sum_{i=1}^{m} a_{ij}^k y_i \le c_j^k, \qquad j = \overline{1,n}, \tag{16}$$

$$y_i \ge 0,\ i = \overline{1,m}. \tag{17}$$

Целевая функция двойственной задачи $g(y)$ является непрерывной и выпуклой [4]. Таким образом, задача (15)–(17) представляет собой задачу нелинейного программирования с линейными ограничениями, аналогичную задаче (8)–(10), и также может быть решена с помощью метода линейной аппроксимации или выпуклого симплекс-метода.

Найденному решению y^k двойственной задачи (15)–(17) соответствует решение прямой задачи (12)–(14) $x^k \in \Pi_k^{ac}$. Для перехода к следующему параллелепипеду Π_{k+1}^{ac} может быть использован метод проекции субградиента, описанный соотношениями (11).

Далее, аналогично алгоритму предыдущего пункта, для Π_{k+1}^{ac} решается задача (12)–(14) путем перехода к двойственной (15)–(17) и находится последующее решение x^{k+1}. Алгоритм завершает работу, когда $x^{k+1} = x^k$, т.к. в этом случае x^k является решением задачи (1)–(3).

Сходимость каждого из перечисленных методов решения задачи (1)–(3) устанавливается известными способами [2], [3].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гаврилова М.О., Первадчук В.П., Севодин М.А. К задаче о наилучшем использовании ресурсов с переменной матрицей технологических коэффициентов // Сборник научных трудов. Шестая международная научно-прикладная конференция. – Варна, 2006.
2. Васильев Ф.П. Численные методы решения экстремальных задач. – М.: Наука, 1988. – 552 с.
3. Зангвилл У. Нелинейное программирование. Единый подход. – М.: Советское радио, 1973. – 312 с.
4. Обен Ж.-П. Нелинейный анализ и его экономические приложения. – М.: Мир, 1988. – 264 с.