К программной реализации решения задачи геометрического программирования

М.А. Тынкевич

Кузбасский государственный технический университет, Кемерово

Рассматривается проблематика компьютерной реализации задач геометрического программирования и возможные пути ее полного решения.

Как известно, задача геометрического программирования сводится к поиску наименьшего значения позинома

$$g_0(x_1, x_2, ..., x_n) = \sum_{k=1}^{m_0} c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} ... x_n^{\alpha_{nk}} \quad (c_k > 0 \text{ при всех } k) \qquad (1)$$

при $x_1>0, .., x_n>0$ и так называемых вынужденных ограничениях $g_k(x_1, .., x_n) \le 1$, $k=1 \div p$, где все функции $g_k(x_1, .., x_n)$ – позиномы [1].

Если обозначить $m = \sum_{k=0}^{p} m_k$, где m_k – количество слагаемых в соответствующих позиномах, и ввести множества индексов:

$J_0 = \{1, 2, ..., m_0\}$, $J_1 = \{m_0 + 1, m_0 + 2, ..., m_0 + m_1\}$, …,

$$J_p = \left\{ \sum_{i=0}^{p-1} m_i + 1, \sum_{i=0}^{p-1} m_i + 2, ..., \sum_{i=0}^{p-1} m_i + m_p \right\},$$

то все позиномы запишутся в виде

$$g_k(x_1, x_2, ..., x_n) = \sum_{i \in J_k} c_k x_1^{\alpha_{1i}} x_2^{\alpha_{2i}} ... x_n^{\alpha_{ni}}$$

и информация о решаемой задаче представима вектором коэффициентов $c_k>0$ ($k=1,2,\ldots,m$) и *матрицей экспонент* (показателей степени), составленной из n строк и m столбцов.

Двойственная задача состоит в максимизации функции

$$V(\delta) = \left[\prod_{k=1}^{m} \left(\frac{c_k}{\delta_k} \right)^{\delta_k} \right] \prod_{s=1}^{p} \lambda_s(\delta)^{\lambda_s(\delta)} \quad , \qquad (2)$$

где

$$\lambda_s(\delta) = \sum_{j \in J(s)} \delta_j,\ s=1 \div p\ ;\ \delta_k \geq 0\ (k=1 \div m); \tag{3}$$

$$\delta_1 + \delta_2 + \ldots \delta_{m_0} = 1\ ,\ \sum_{i=1}^{m} \alpha_{ij} \delta_i = 0\ , j=1,2,..,n\ . \tag{4}$$

(функция $V(\delta)$ непрерывна при всех δ, поскольку при δ=0 соблюдается $\delta^{\pm\delta}$=1).

Теоремы двойственности геометрического программирования [1] утверждают, что при условии сильной совместности ограничений прямой задачи и существовании точки X^*, в которой достигается минимум $g_0(X)$ (хотя бы локальный), существует точка δ^*, в которой достигается максимум $V(\delta)$ (глобальный или локальный), причем $g_0(X^*) = V(\delta^*)$.

Для оптимального решения двойственной задачи δ^* любая точка минимума X^* прямой задачи удовлетворяет системе

$$c_i \prod_{s=1}^{n} x_s^{\alpha_s^{is}} = \begin{cases} \delta_i^* \times V(\delta^*),\ i \in J(0) \\ \delta_i^* / \lambda_k(\delta^*),\ i \in J(k), \end{cases} \tag{5}$$

где k принимает все положительные значения такие, что $\lambda_k(\delta^*)>0$.

Алгоритмическая реализация решения задачи сталкивается с принципиальными трудностями, связанными с соблюдением условия $m=n+1$ для минимизируемого позинома.

Так, при решении задачи [1. С.16] минимизации $g_0(X)=x_2$ при ограничениях $g_1(x_1, x_2) = \dfrac{x_1^4}{x_2^4} + \dfrac{\sqrt{x_2}}{x_1} \leq 1$; $x_1, x_2>0$, обнаруживаем m=3; соответственно, $J(0)=\{\ 1\ \}$, $J(1)=\{\ 2\ 3\ \}$, вектор C=[1 1 1] и матрицу экспонент

$$A = \begin{pmatrix} 0 & 4 & -1 \\ 1 & -4 & 0{,}5 \end{pmatrix},$$

откуда получаем

$$V(\delta) = \left(\frac{1}{\delta_1}\right)^{\delta_1} \left(\frac{1}{\delta_2}\right)^{\delta_2} \left(\frac{1}{\delta_3}\right)^{\delta_3} (\delta_2 + \delta_3)^{\delta_2 + \delta_3}$$

и множество планов, определяемое положительными (в пределе неотрицательными) решениями системы $n+1=3$ уравнений с m=3 неизвестными

$$\begin{aligned} \delta_1 &= 1, \\ 4\,\delta_2 - \delta_3 &= 0, \\ \delta_1 - 4\,\delta_2 + 0{,}5\,\delta_3 &= 0. \end{aligned}$$

Обнаружив без труда таковые δ = [1 0,5 2], убеждаемся в существовании решения обеих задач и равенства $V(\delta) = g_0(X) = 3{,}4939$. Обратившись к (5), получаем нелинейную, но сводимую логарифмированием к линейной, систему m=3 зависимых уравнений с n=2 неизвестными (заметьте, что в роли коэффициентов этой системы выступает транспонированная матрица экспонент):

$$x_2 = \delta_1 \times V(\delta) = 3{,}4939\ ;\ x_1^4 \Big/ x_2^4 = \left(\frac{\delta_2}{\delta_2+\delta_3}\right) = 0{,}2\ ;\ \sqrt{x_2} \Big/ x_1 = \left(\frac{\delta_3}{\delta_2+\delta_3}\right) = 0{,}8,$$

откуда $z_1 = \lg(x_1) = 0{,}3683$, $z_2 = \lg(x_2) = 0{,}5433$ и, соответственно, $x_1 = 2{,}34$, $x_2 = 3{,}49$.

Однако решая, на первый взгляд, более простую задачу минимизации

$$g_0(X) = \frac{40}{x_1 x_2} + 40x_2 + 20x_1 + 10x_1x_2$$

при отсутствии вынужденных ограничений (так называемый коэффициент трудности $s=m-(n+1)=4-3>0$), приходим к максимизации

$$V(\delta) = \left(\frac{40}{\delta_1}\right)^{\delta_1} \left(\frac{40}{\delta_2}\right)^{\delta_2} \left(\frac{20}{\delta_3}\right)^{\delta_3} \left(\frac{10}{\delta_4}\right)^{\delta_4}$$

на множестве неотрицательных решений системы (m=4, n=2)

$$\begin{aligned} \delta_1 + \delta_2 + \delta_3 + \delta_4 &= 1, \\ -\delta_1 + \delta_3 + \delta_4 &= 0, \\ -\delta_1 + \delta_2 + \delta_4 &= 0. \end{aligned}$$

Если при нулевой трудности $s=m-(n+1)=0$ программирование вычислительного процесса действительно не вызывает особых трудностей, то при $s>0$, построив аналогичную систему относительно δ_k, мы вынуждены выбирать дальнейший путь поиска оптимального решения.

Один из вариантов связан с отказом от «компьютерного продолжения» и использованием человеческого знания и интуиции на пути понижения размерности (громоздкая подготовительная работа уже выполнена) – подобрать некоторый набор n базисных

переменных и выразить остальные через них. Однако при s >>1 этот вариант чреват возникновением не менее сложной задачи.

Другой вариант предполагает использование процедуры Монте-Карло (еще 20 лет назад нереальной при сколь-нибудь значительной размерности, не говоря уже о «детских годах» геометрического программирования).

Есть и третий путь, базирующийся на идеях методов квадратического программирования (например, метода Вулфа – Фрэнка). Здесь в рамках симплексной процедуры с полным начальным искусственным базисом отыскивается начальный опорный план, находятся компоненты градиента $V(\delta)$

$$\frac{\partial V}{\partial \delta_k} V(\delta) \times \left(\begin{array}{l} \left(\dfrac{c_k}{\delta_k} \right)^{\delta_k} \ln \dfrac{c_k}{\delta_k} \left(-\dfrac{c_k}{\delta_k^2} \right) + \\ + \begin{cases} \lambda_s(\delta)^{\lambda_s(\delta)} \ln \lambda_s(\delta), & k \in J(s) \\ 0, & k \notin J(s) \end{cases} \end{array} \right), k = \overline{1,m},$$

принимаемые за компоненты «линейной формы», и выполняется переход к очередному опорному плану в направлении ее возрастания. Увы, поскольку вогнутость $V(\delta)$ в общем случае не гарантирована, не гарантировано и получение глобального оптимума.

Литература

1. *Даффин Р., Питерсон Э., Зенер К.* Геометрическое программирование. М.: Мир, 1972. 311 с.
2. *Тынкевич М.А., Вересова К.А.* К решению задачи геометрического программирования // Вестник КузГТУ. 2009. №3. С. 141–146.