Выпуск 22 (39) ОПТИМИЗАЦИЯ 1978 г.
Теория и машинная реализация массовых вычислений

УДК 681.3.06 : 51

АЛГОРИТМИЧЕСКОЕ ПОСТРОЕНИЕ ЗАДАЧ ОПТИМАЛЬНОЙ ОБРАБОТКИ СОСТАВНЫХ ФАЙЛОВ

Л.Т.Петрова

В [I] построено исчисление данных и разработана гомоморфная ему алгебра некоторых оптимизационных задач. Это позволяет для составных файлов, заданных формулами исчисления данных, по той же формуле вычислять (строить) некоторые задачи оптимальной обработки этих файлов. Там же рассмотрена задача о минимальном числе обращений к внешней памяти. Целью настоящей статьи является дальнейшая иллюстрация предложенной техники алгоритмического построения оптимизационных задач. Здесь рассматривается еще одна задача - о минимальном времени обмена информацией-и на различных примерах показывается, как именно строятся задачи для конкретных составных файлов, заданных формулами исчисления. Чтобы получить численное решение построенных геометрических программ, в статье изложен, наряду с краткой общей схемой решения задач, также и конкретный прием, использованный в примерах [2]. По поводу других методов решения задач геометрического программирования можно сослаться, например, на [3].

I. Задача о минимизации времени обмена информацией при обработке составных файлов

Пусть имеется база данных с различными уровнями внешней памяти. Каждый уровень памяти имеет две характеристики: x - время считывания одного кода, y - время поиска очередной порции информации.

Рассматривается исчисление данных $\mathcal{D}$, введенное в [I]. Составные файлы задаются формулами в этом исчислении. Для всякого терминального файла $a \in \mathcal{D}$ соответствующие характеристики уровня, на котором хранится данный файл, обозначаются через x_a, y_a. В остальном сохраняются обозначения и предположения [I]. Через l_a обозначается длина массива a, через t_a - размер отведенной ему резервной области; T - общий размер буфера в оперативной памяти. Всякий терминальный массив $a \in \mathcal{D}$ формально может быть разбит на подмассивы длины $\le t_a$ и представлен как объединение этих подмассивов. В операциях обмена эти подмассивы выступают как неделимые единицы информации. Величина g^a характеризует частоту обращений к внешней памяти при реализации массива a для терминального массива $g^a = l_a/t_a$.

Величину $\tau^a = l_a x_a + g^a y_a$ примем в качестве характеристики времени обмена при реализации терминального массива a. Поставим задачу о минимизации соответствующей величины τ для произвольного массива, заданного формулой в исчислении $\mathcal{D}$.

Определим следующее отображение f^* множества данных $\mathcal{D}$ в множество геометрических программ G: $f^*: \mathcal{D} \longrightarrow G$.

I) Для всякого терминального массива $a \in \mathcal{D}$ геометрическая программа $f^*(a)$ имеет вид

$$f^*(a) = \{\tau^a, C^a, B^a\},$$

где $\tau^a = l_a x_a + g_0^a y_a$, $g_0^a(t) = l_a/t_a$, $C^a = \{g_1^a\}$ - основные ограничения задачи, $B^a = \{g_2^a\}$ - простейшие ограничения, $g_1^a(t) = t_a/T \le 1$, $g_2^a(t) = t_a/l_a \le 1$. Геометрическая программа указанного вида представляет задачу $\min\{\tau^a(t) \mid g(t) \le 1 \ \forall g(t) \in C^a \cup B^a\}$.

2) Если массив $a \in \mathcal{D}$ задан выражением $a = b * c$, где звездочкой обозначена одна из операций X, $\cup$ или $\Uparrow$, а для массивов b и c уже построены образы

$$f^*(b) = \{\tau^b, C^b, B^b\},$$
$$f^*(c) = \{\tau^c, C^c, B^c\},$$

то программа $f^*(a)$ строится по приведенным ниже правилам из программ $f^*(b)$ и $f^*(c)$. При этом естественно, что ограничения на переменные и функционалы g_0 формируются по тем же самым правилам, что и в предыдущей задаче [I].

Геометрическая программа $f^*(a) = f^*(b) X f^*(c)$ при свободном вхождении операции X в формулу строится по следующим

правилам:

$$\begin{cases} \tau^a = \tau^b + g_0^b \tau^c, \\ g_0^a = g_0^b \cdot g_0^c, \\ C^a = \{g \mid g = g^b + g^c \; \forall (g^b, g^c) \in C^b \times C^c\} = F(C^b, C^c), \\ B^a = B^b \cup B^c. \end{cases} \tag{I.I}$$

В задаче (I.I) отражается следующий порядок считывания массивов. Считывается очередная порция массива b_1 и рядом с ней "протягиваются" поочередно все порции массива c. Затем рассматривается следующая порция b.

Геометрическая программа $f^*(a) = f^*(b) \times f^*(c)$ при связанном вхождении операции X в формулу строится по следующим правилам:

$$\begin{cases} \tau^a = \tau^b + l_b \tau^c, \\ g_0^a = g_0^b + l_b g_0^c, \\ C^a = F(C^b, C^c), \\ B^a = B^b \cup B^c. \end{cases} \tag{I.2}$$

Геометрическая программа $f^*(a) = f^*(b) \cup f^*(c)$ строится по правилам:

$$\begin{cases} \tau^a = \tau^b + \tau^c, \\ g_0^a = g_0^b + g_0^c, \\ C^a = C^b \cup C^c, \\ B^a = B^b \cup B^c. \end{cases} \tag{I.3}$$

Геометрическая программа $f^*(a) = f^*(b) \Uparrow f^*(c)$ строится по правилам:

$$\begin{cases} \tau^a = \tau^b + \tau^c, \\ g_0^a = g_0^b + g_0^c, \\ C^a = F(C^b, C^c), \\ B^a = B^b \cup B^c. \end{cases} \tag{I.4}$$

Таким образом, произвольной формуле $a \in \mathcal{D}$ отображение соотносит задачу $f^*(a)$ оптимальной реализации этой формулы.

Отображение f^* не сохраняет все свойства операций. Например, из соотношений (I.I) легко видеть, что операция (X) над геометрическими программами не обладает коммутативностью

при свободном вхождении этой операции в формулу. Это свидетельствует о неоднозначности отображения: для эквивалентных в $\mathcal{D}$ формул можно получить, вообще говоря, различные задачи. Однако это нежелательное само по себе обстоятельство можно использовать как полезный (хотя и несколько громоздкий) инструмент при оценке эффективности той или иной формулы. Улучшение решения соответствующей задачи может служить критерием при разработке тактики направленных эквивалентных преобразований формулы на уровне синтаксиса. Решение этих задач может быть полезным и при распределении данных по уровням, так как позволяет оценить и сравнить эффективность различных вариантов распределения.

2. Одна схема решения задач геометрического программирования

Рассмотрим одну из возможных схем решения задач геометрического программирования [2]. Пусть имеем прямую геометрическую программу (ГП)

$$\begin{cases} g_0(t) \longrightarrow min, \\ t_j > 0 \ (j=1,\dots,m), \\ t=(t_1,t_2,\dots,t_m), \\ g_k(t) \le 1 \quad (k=1,\dots,s). \end{cases} \qquad (2.1)$$

Здесь $g_k(t)=\sum_{i\in I(k)} c_i t_1^{a_{i1}} \dots t_m^{a_{im}}$, a_{ij} - произвольные вещественные числа, $c_i>0$, $I(k)=\{m_k, m_k+1,\dots,n_k\}$, $k=0,1,\dots,s$, $m_0=1, m_1=n_0+1,\dots,m_s=n_{s-1}+1, n_s=n$. A - матрица экспонент ГП (2.1),

$$A=\begin{bmatrix} a_{11} & a_{12} \dots a_{1m} \\ a_{21} & a_{22} \dots a_{2m} \\ \hline a_{n1} & a_{n2} \dots a_{nm} \end{bmatrix}$$

Прямой ГП (2.1) соотносится двойственная ГП относительно двойственных переменных δ_i, $i=1,\dots,n$:

$$v(\delta)=\left[\prod_{i=1}^{n}\left(c_i/\delta_i\right)^{\delta_i}\right]\prod_{k=1}^{s}\lambda_k(\delta)^{\lambda_k(\delta)} \longrightarrow max;$$

$$\left\{\begin{array}{l}\lambda_k(\delta)=\sum\limits_{i\in I(k)}\delta_i\,,\ k=1,\dots,s;\\ \text{ограничения на переменные:}\\ \delta_i\geq 0,\ i=1,\dots,n,\\ \sum\limits_{i\in I(0)}\delta_i=1,\\ \sum\limits_{i=1}^{n}a_{ij}\delta_i=0,\ j=1,\dots,m.\end{array}\right. \qquad (2.2)$$

Пусть $d=n-m-1$ есть степень трудности ГП (2.I) и $d>0$. Общее решение δ двойственных ограничений можно представить разложением по базису $\{b^{(j)}\}$ двойственного пространства

$$\delta=b^{(0)}+\sum_{j=1}^{d}r_j\,b^{(j)},$$

здесь $r_j\ (j=1,\dots,d)$ суть базисные переменные.

Выражая двойственную функцию v через базисные переменные, получим преобразованную ГП:

$$\left\{\begin{array}{l}v(r)=K_0\Big[\prod\limits_{j=1}^{d}K_j^{r_j}\Big]\Big[\prod\limits_{i=1}^{n}\delta_i(r)^{-\delta_i(r)}\Big]\Big[\prod\limits_{k=1}^{s}\lambda_k(r)^{\lambda_k(r)}\Big]\longrightarrow max;\\ K_j=\prod\limits_{i=1}^{n}c_i^{b_i(j)},\ j=0,1,\dots,d;\\ \delta_i(r)=b_i^{(0)}+\sum\limits_{j=1}^{d}r_j\,b_i^{(j)};\\ \lambda_k(r)=\lambda_k^{(0)}+\sum\limits_{j=1}^{d}r_j\,\lambda_k^{(j)},\ \lambda_k^{(j)}=\sum\limits_{i\in I(k)}b_i^{(j)},\ \left[\begin{array}{l}k=1,\dots,s,\\ j=0,1,\dots,d;\end{array}\right.\\ \text{на вектор } r \text{ наложены условия неотрицательности:} \qquad (2.3)\\ \quad b^{(0)}+\sum\limits_{j=1}^{d}r_j\,b^{(j)}\geq 0;\\ \text{вектор } b^{(0)} \text{ удовлетворяет условиям нормализации}\\ \quad \sum\limits_{i=1}^{n_0}b_i^{(0)}=1\\ \text{и условиям ортогональности}\\ \quad \sum\limits_{i=1}^{n}a_{ij}\,b_i^{(0)}=0,\ \ j=1,\dots,m;\\ \text{векторы невязки } b^{(j)} \text{ образуют базис пространства решений линейной однородной системы}\\ \quad \sum\limits_{i=1}^{n_0}y_i=0,\\ \quad \sum\limits_{i=1}^{n}a_{ij}\,y_j=0,\ \ j=1,\dots,m.\end{array}\right.$$

От задачи (2.3) можно перейти к эквивалентной ей задаче максимизации вогнутой функции $\ln v(\tau)$ при тех же ограничениях. Для решения этой последней задачи в выпуклом программировании имеются алгоритмы [2]. Вместо задачи выпуклого программирования можно решать некоторую систему нелинейных алгебраических уравнений. Эта система получается при анализе задачи о $\ln v(\tau)$ при приравнивании нулю логарифмической производной и состоит из так называемых максимизирующих уравнений:

$$K_j = \Big[\prod_{i=1}^{n} \delta_i(\tau)^{b_i^{(j)}}\Big] \prod_{k=1}^{s} \lambda_k(\tau)^{-\lambda_k^{(j)}}, \quad j=1,\dots,d. \qquad (2.4)$$

Если из соотношений (2.3) найти базис $\{b^{(j)}\}$ и вычислить базисные постоянные K_j, то соотношения (2.4) представляют систему d нелинейных уравнений относительно базисных переменных τ. Решение этой системы приводит к решению исходной задачи в силу соотношений, связывающих переменные прямой и двойственной задач:

$$u_k = c_k t_1^{a_{k1}} \dots t_m^{a_{km}} = \delta_k / \lambda_i, \quad k \in I(i),\ i=1,\dots,s. \qquad (2.5)$$

Соотношения (2.5) имеют особенно простой и удобный вид для наших задач – вследствие линейности ограничений в исходной ГП. Это же обстоятельство упрощает процесс построения базиса. Именно этот прием (решение уравнений (2.4)) использован в приведенных ниже примерах.

3. Примеры

Для иллюстрации метода построим и решим оптимизационные задачи для некоторых конкретных формул из $\mathcal{D}$.

ПРИМЕР I. Пусть имеем в $\mathcal{D}$ формулу

$$a = (a_1 \vee a_2) \times (a_3 \Uparrow a_4).$$

Запишем выражение для a в виде схемы

$$a = b \times c, \quad b = a_1 \vee a_2, \quad c = a_3 \Uparrow a_4. \qquad (3.I)$$

Построим для этой схемы задачу о минимизации числа обращений к внешней памяти [I]. По правилам (I.I)-(I.4) получаем функционалы: $g_0^a = g_0^b \cdot g_0^c$, $g_0^b = g_0^1 + g_0^2$, $g_0^c = g_0^3 + g_0^4$; $g_0^k = l_k / t_k$, $k=1,\dots,4$, т.е. $g_0^a = (l_1/t_1 + l_2/t_2)\cdot(l_3/t_3 + l_4/t_4)$.

По тем же правилам последовательно вычисляем ограничения:

$$C^k=\{g_1^k=t_k/T\},\ k=1,\dots,4;$$

$$C^c=F(C^3,C^4)=\{g_1^c\},\ g_1^c(t)=g_1^3+g_1^4=t_3/T+t_4/T;$$

$$C^b=C^1\cup C^2=\{g_1^b,g_2^b\},\ g_1^b(t)=g_1^1=t_1/T,\ g_2^b(t)=g_1^2=t_2/T;$$

$$C^a=F(C^b,C^c)=\{g_1^a,g_2^a\},$$

$$g_1^a(t)=g_1^b+g_1^c=\frac{1}{T}(t_1+t_3+t_4),$$

$$g_2^a(t)=g_2^b+g_1^c=\frac{1}{T}(t_2+t_3+t_4).$$

Учитывая соотношение $t_1=t_2$, запишем задачу в виде

$$\begin{cases} g_0(t)=\dfrac{c_1}{t_1t_3}+\dfrac{c_2}{t_1t_4}\longrightarrow min; \\ g_1(t)=\dfrac{1}{T}(t_1+t_3+t_4)\le 1. \end{cases} \qquad (3.2)$$

Здесь $c_1=(\ell_1+\ell_2)\ell_3$, $c_2=(\ell_1+\ell_2)\ell_4$, $c_3=c_4=c_5=1/T$. Матрица экспонент в этой задаче следующая:

$$A=\begin{bmatrix} -I & -I & 0 \\ -I & 0 & -I \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

Степень трудности $d=n-m-1=5-3-1=1$, $s=1$, $I(0)=\{1,2\}$, $I(1)=$ $=\{3,4,5\}$. Перейдем к решению полученной задачи (3.2).

В рассматриваемых задачах матрица A всегда содержит единичную подматрицу $E(m\times m)$. Остальные строки образуют матрицу $A_0(m\times k)$, $k=n-m=d+1$. Базисные векторы $B_0(k\times m)$ двойственного пространства получаются известной процедурой линейной алгебры. Пусть матрица A имеет вид

$$A=\begin{bmatrix} A_0(m\times k) \\ E(m\times m) \end{bmatrix}$$

Тогда матрица B_0 имеет вид

$$B_0 = \begin{bmatrix} E(k \times k) \\ -A_0^*(k \times m) \end{bmatrix}$$

где A_0^* есть матрица, транспонированная к A_0. По построению столбцы B_0 линейно-независимы и ортогональны к столбцам матрицы A. Остается удовлетворить условиям нормализации:

$$\sum_{i=1}^{n_0} b_i^{(0)} = 1, \quad \sum_{i=1}^{n_0} b_i^{(j)} = 0, \quad j = 1, \dots, d.$$

Нормализованные базисные векторы образуют матрицу $B(k \times n)$.

Возвратимся к примеру. Матрицы A, B_0 и B в нашем случае следующие:

$$\begin{array}{cc} & A \\ A_0 & \left[\begin{array}{ccc} -1 & -1 & 0 \\ -1 & 0 & -1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \\ E & \end{array} \qquad \begin{array}{cc} & B_0 \\ E & \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] \\ -A_0^* & \end{array} \qquad \begin{array}{c} B \\ \left[\begin{array}{cc} 1 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{array}\right] \end{array}$$

Первый столбец матрицы B_0 удовлетворяет условиям нормализации, это вектор нормализации $b^{(0)} = (1,0,1,1,0)$. Вектор невязки $b^{(1)}$ получается вычитанием из второго столбца матрицы B_0 вектора $\alpha b^{(0)}$, где α есть сумма первых n_0 компонент исправляемого вектора. В данном случае $\alpha = 1$, $b^{(1)} = (-1,1,0,-1,1)$. Базис построен. Разложение двойственных переменных по базису имеет вид:

$$\delta_1 = 1 - \tau, \quad \delta_2 = \tau, \quad \delta_3 = 1, \quad \delta_4 = 1 - \tau, \quad \delta_5 = \tau.$$

Вычислим базисную постоянную K_1 (см. (2.3)):

$$K_1 = \prod_{i=1}^{n} c_i^{b_i^{(1)}} = c_1^{-1} c_2\, c_3^0\, c_4^{-1} c_5 = \ell_4 / \ell_3 .$$

Составим максимизирующее уравнение (см. (2.4)):

$$K_1 = \left[\prod_{i=1}^{n} \delta_i(\tau)^{b_i^{(1)}}\right] \prod_{k=1}^{s} \lambda_k(\tau)^{-\lambda_k^{(1)}}$$

Здесь $s = 1$, $\lambda_1(\tau) = \delta_3 + \delta_4 + \delta_5 = 1 + (1-\tau) + \tau = 2$, $\lambda_1^{(1)} = 0$. Получаем уравнение $K_1 = (1-\tau)^{-1} \tau (1-\tau)^{-1} \tau$, т.е. $\ell_4 (1-\tau)^2 = \ell_3 \tau^2$. При $\ell_3 = \ell_4$ имеем решение $\tau = 1/2$. Из соотношений (2.5) в этом случае получаем

$$u_3 = t_1/T = \delta_3/\lambda_1(\tau) = 1/2,$$

$$u_4 = t_3/T = \delta_4/\lambda_1(\tau) = 1/4,$$

$$u_5 = t_4/T = \delta_5/\lambda_1(\tau) = 1/4 .$$

Таким образом, при $l_3 = l_4$ имеем следующее оптимальное решение задачи (3.2):

$$t_1^* = t_2^* = T/2 \ , \quad t_3^* = t_4^* = T/4 .$$

ПРИМЕР 2. Рассмотрим несколько усложненную формулу первого примера:

$$a = [(a_1 \vee a_2) \times (a_3 \uparrow a_4)] \times (a_5 \times a_6).$$

Принимая во внимание выкладки, выполненные в предыдущем примере, сразу выпишем для нашей формулы оптимизационную задачу [I]:

$$\begin{cases} g_0(t) = \dfrac{c_1}{t_1 t_3 t_5 t_6} + \dfrac{c_2}{t_1 t_4 t_5 t_6} \longrightarrow min; \\ g_1(t) = \dfrac{1}{T}(t_1 + t_3 + t_4 + t_5 + t_6) \le 1 . \end{cases} \qquad (3.3)$$

Здесь $c_1 = (l_1 + l_2) l_3 l_5 l_6$, $c_2 = (l_1 + l_2) l_4 l_5 l_6$, $c_3 = \ldots = c_7 = 1/T$. Имеем следующие матрицы A, B_0, B:

$$A = \left[\begin{array}{ccccc} -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & -1 & -1 & -1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \qquad B_0 = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{array}\right] \qquad B = \left[\begin{array}{cc} 1 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{array}\right]$$

Получаем разложение двойственных переменных по базису: $\delta_1 = 1 - \tau$, $\delta_2 = \tau$, $\delta_3 = 1$, $\delta_4 = 1 - \tau$, $\delta_5 = \tau$, $\delta_6 = \delta_7 = 1$. Вычисляем

$$\lambda_1(\tau) = \sum_{i=3}^{7} \delta_i = 4 \ , \quad \lambda_1^{(1)} = 0 .$$

Максимизирующее уравнение в этой задаче получается такое же, как и в предыдущей. Поэтому сразу можем выписать оптимальное решение нашей задачи (при $l_3 = l_4$):

$$t_1^* = t_2^* = T \frac{\delta_3}{\lambda_1(\tau)} = \frac{T}{4} ; \quad t_3^* = T \frac{\delta_4}{\lambda_1(\tau)} = \frac{T}{8} ;$$

$$t_4^* = T\frac{\delta_5}{\lambda_1(r)} = \frac{T}{8};\ t_5^* = T\frac{\delta_6}{\lambda_1(r)} = \frac{T}{4};\ t_6^* = T\frac{\delta_7}{\lambda_1(r)} = \frac{T}{4}.$$

ПРИМЕР 3. Построим для схемы (3.I) из первого примера другую оптимизационную задачу – о минимизации времени обмена информацией. Предположим, что исходные массивы $a_i\ (i=1,\dots,4)$ размещаются на четырех различных уровнях памяти, имеющих соответствующие характеристики $x_i, y_i\ (i=1,\dots,4)$. Ограничения на переменные C^a остаются теми же, что и в примере I, характеристики $g_0(t)$ тоже остаются прежними. Новый функционал $\tau(t)$ вычисляется по правилам (I.I)-(I.4):

$$\tau^a = \tau^b + g_0^b \tau^c;$$

$$\tau^b = \tau_1 + \tau_2 = l_1 x_1 + \frac{l_1 y_1}{t_1} + l_2 x_2 + \frac{l_2 y_2}{t_2};$$

$$\tau^c = \tau_3 + \tau_4 = l_3 x_3 + \frac{l_3 y_3}{t_3} + l_4 x_4 + \frac{l_4 y_4}{t_4}.$$

Подставляя в выражение для τ^a соответствующие представления входящих в него величин и учитывая соотношение $t_1 = t_2$, получаем

$$\tau^a = c_0 + \frac{c_1}{t_1} + \frac{c_2}{t_1 t_3} + \frac{c_3}{t_1 t_4},$$

где $c_0 = l_1 x_1 + l_2 x_2,\ c_1 = (l_1 + l_2)(l_3 x_3 + l_4 x_4) + l_1 y_1 + l_2 y_2,$
$c_2 = (l_1 + l_2)\, l_3 x_3,\ c_3 = (l_1 + l_2)\, l_4 y_4.$

Таким образом, сформулирована задача

$$\min\left\{\frac{c_1}{t_1} + \frac{c_2}{t_1 t_3} + \frac{c_3}{t_1 t_4} \,\middle|\, \frac{1}{T}(t_1 + t_3 + t_4) \le 1\right\}. \quad (3.4)$$

В этой задаче имеем следующую матрицу экспонент:

$$A = \begin{bmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Степень трудности $d = 6 - 3 - 1 = 2$, $s = 1$, $c_4 = c_5 = c_6 = \frac{1}{T}$,
$I(0) = \{1, 2, 3\}$, $I(1) = \{4, 5, 6\}$.

Перейдем к решению задачи (3.4). Построим базис B двойственного пространства. Матрицы A, B_0, B имеют соответственно

вид

$$
\begin{array}{ccc}
A & B_0 & B \\
\left[\begin{array}{ccc} -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & 0 & -1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] &
\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] &
\left[\begin{array}{ccc} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]
\end{array}
$$

Выпишем разложение двойственных переменных по найденному базису: $\delta_1 = 1 - r_1 - r_2$, $\delta_2 = \delta_5 = r_1$, $\delta_3 = \delta_6 = r_2$, $\delta_4 = 1$. Составим максимизирующие уравнения для этой задачи. Сначала вычисляем базисные постоянные K_1 и K_2:

$$K_1 = \prod_{i=1}^{n} c_i^{b_i^{(1)}} = c_1^{-1} c_2 c_5 = \frac{(l_1 + l_2) l_3 y_3}{T[(l_1 + l_2)(l_3 x_3 + l_4 x_4) + l_1 y_1 + l_2 y_2]},$$

$$K_2 = \prod_{i=1}^{n} c_i^{b_i^{(2)}} = c_1^{-1} c_3 c_6 = \frac{(l_1 + l_2) l_4 y_4}{T[(l_1 + l_2)(l_3 x_3 + l_4 x_4) + l_1 y_1 + l_2 y_2]}.$$

Затем вычисляем величины

$$\lambda_1(r) = \sum_{i \in I(1)} \delta_i = 1 + r_1 + r_2, \quad \lambda_1^{(1)} = \lambda_1^{(2)} = 1.$$

Максимизирующая система в этой задаче состоит из двух уравнений:

$$K_j = \Big[\prod_{i=1}^{6} \delta_i(r)^{b_i^{(j)}}\Big] \lambda_1(r)^{-\lambda_1^{(j)}}, \quad j = 1, 2.$$

Выпишем эти уравнения:

$$
\begin{aligned}
K_1 &= (1 - r_1 - r_2)^{-1} r_1^2 (1 + r_1 + r_2)^{-1}, \\
K_2 &= (1 - r_1 - r_2)^{-1} r_2^2 (1 + r_1 + r_2)^{-1}.
\end{aligned} \qquad (3.5)
$$

Перепишем систему (3.5) в виде

$$r_1^2 = K_1(1 - (r_1 + r_2)^2), \quad r_2^2 = K_2(1 - (r_1 + r_2)^2),$$

откуда $r_1^2 / r_2^2 = K_1 / K_2 = l_3 y_3 / l_4 y_4$. Найдем решение для некоторых конкретных наборов параметров. Пусть $K_1 = K_2$. Тогда $r_1 = r_2$ есть решение уравнения $r^2 = K_1(1 - 4r^2)$, т.е. $r_1 = r_2 = \sqrt{K_1/(1 + 4K_1)}$. При условиях $l_i = 10^4$, $x_i = 10^{-3}$, $y_i = 10^{-1}$ $(i = 1, \dots, 4)$, $T = 10^2$ получаем решение $r = 0{,}4$, т.е.

$$t_1^* = t_2^* = T^{\delta_4/\lambda_1(\tau)} = 0{,}56T,$$

$$t_3^* = t_4^* = T^{\delta_5/\lambda_1(\tau)} = 0{,}22T.$$

При $T=10^3$ получаем $\tau = 0{,}2$, т.е.

$$t_1^* = t_2^* = 0{,}71T; \quad t_3^* = t_4^* = 0{,}14T.$$

ЛИТЕРАТУРА

1. ПЕТРОВА Л.Т. Об оптимальном управлении данными. - "Программирование", 1977, № 6, с. 29-38.
2. ДАФФИН Р., ПИТЕРСОН Э., ЗЕНЕР К. Геометрическое программирование. М., "Мир", 1972.
3. ECKER J.G., ZORACKI M.J. An easy primal method for geometric programming. - "Management Sci.", 1976, v. 23, N 1, p. 71-77.

Поступила в ред.-изд. отдел
18.X.1978 г.