СИСТЕМНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

УДК 519.874

МНОГОПРОДУКТОВЫЕ ПОТОКИ В ДРЕВОВИДНЫХ СЕТЯХ

 Л. Г. Афраймович, М. Х. Прилуцкий

Нижний Новгород, Нижегородский государственный ун-т

Поступила в редакцию 14.05.07 г.

Рассматривается задача поиска многопродуктовых потоков минимальной стоимости в сетях древовидной структуры. Предлагается алгоритм решения поставленной задачи, основанный на сводимости к задаче поиска однопродуктового потока в сети произвольной структуры. В случае однопродуктовых потоков в древовидных сетях применяется алгоритм, использующий процедуру приведения границ.

Введение. Необходимость поиска многопродуктовых потоков в сетях возникает при решении широкого класса прикладных задач. Примерами являются задачи объемно-календарного планирования [1], распределения мощностей каналов передачи данных провайдерами сети Интернет [2], сбалансированной загрузки распределенной вычислительной сети [3], транспортная задача с промежуточными пунктами [4] и др.

Классическими потоковыми задачами служат поиск максимального потока и потока минимальной стоимости. Здесь наиболее изучены однопродуктовые задачи, которые формализуются как задачи линейного программирования с абсолютно унимодулярной матрицей системы ограничений. Эти задачи являются полиномиально разрешимыми, в том числе и в целочисленном случае. Обзор алгоритмов решения однопродуктовой задачи о максимальном потоке можно найти, например, в [5]; алгоритмы решения задачи об однопродуктовом потоке минимальной стоимости рассматриваются в [6].

Класс задач поиска многопродуктовых потоков менее изучен. Алгоритмы нахождения многопродуктовых (нецелочисленных) потоков приведены, например, в [7, 8]. В [9] показано, что задача поиска максимального целочисленного многопродуктового потока NP-трудна уже в двухпродуктовом случае; там же был предложен полиномиальный алгоритм отыскания максимального двухпродуктового потока, строящий целочисленный, с точностью до одной второй (например, 7/2, 15/2, 16/2), поток. Известны приближенные алгоритмы решения задачи о максимальном целочисленном многопродуктовом потоке [10, 11]. Некоторые обобщения моделей многопродуктовых потоков в сетях обсуждаются в [12, 13]. Особый интерес представляет класс многопродуктовых потоков в сетях древовидной структуры. Из [14] следует, что задача поиска целочисленного многопродуктового потока в древовидной неориентированной сети в общем случае NP-трудна.

В настоящей работе исследуется специальный подкласс задач поиска многопродуктового потока в ориентированных сетях корневой древовидной структуры в случае, когда корень дерева является истоком для потоков различных продуктов, а листья – стоками. Для рассматриваемой сети предлагается алгоритм решения задачи поиска многопродуктового потока минимальной стоимости, основанный на сводимости к задаче получения однопродуктового потока в сети произвольной структуры; доказывается полиномиальная разрешимость подобной задачи; для случая однопродуктовых потоков в сетях древовидной структуры разработан метод, использующий процедуру приведения границ.

1. Древовидные сетевые структуры. Рассматриваемые в работе задачи возникают при планировании и управлении сетевыми древовидными структурами, моделирующими экономические, производственные, технические системы. Примерами могут служить задачи объемного многономенклатурного планирования для подразделений предприятия; многопродуктовая древовидная транспортная задача с промежуточными пунктами; разузлования при планировании производства новых изделий для предприятий с единичным и мелкосерийным характером производства; номенклатурного планирования для предприятий с непрерывным циклом изготовления продукции и др.

Содержательно задачи объемного многономенклатурного планирования для подразделений предприятия формулируются следующим образом. Предприятие состоит из подразделений (цехов), каждое из которых в свою очередь объединяет участки, на которых расположены рабочие места. Требуется на заданный период планирования распределить поступившие заказы в объемных характеристиках (нормо-часы, рубли, условные тонны) по цехам и участкам таким образом, чтобы были удовлетворены заданные показатели и суммарная прибыль предприятия была максимальна.

Пусть I – множество подразделений предприятия; J_i – множество участков подразделения i, $i \in I$; K – множество заказов, которые предприятие должно выполнить. Рассмотрим следующие показатели плана: A – оставшийся незавершенным на заданный период планирования объем работ по всем заказам; B_i, C_i – соответственно минимально возможный и максимально допустимый объемы работ, которые могут быть сделаны в планируемом периоде подразделением i по всем заказам, $i \in I$; D_{ij}, E_{ij} – соответственно минимально возможный и максимально допустимый объемы работ, которые могут быть исполнены в планируемом периоде участком j подразделения i по всем заказам, $j \in J_i$, $i \in I$; G_{ijk} – оставшийся объем работ по заказу k участком j подразделения i, $j \in J_i$, $i \in I$, $k \in K$; v_{ijk} – прибыль, которую получит предприятие за единицу работы, совершенной по заказу k участком j в подразделении i, $j \in J_i$, $i \in I$, $k \in K$. Тогда задача объемного планирования формализуется в виде следующей задачи линейного программирования:

$$
\begin{gathered}
\sum_{i \in I}\sum_{j \in J_i}\sum_{k \in K} x_{ijk} \le A, \\
B_i \le \sum_{k \in K}\sum_{j \in J_i} x_{ijk} \le C_i, \quad i \in I, \\
D_{ij} \le \sum_{k \in K} x_{ijk} \le E_{ij}, \quad j \in J_i,\ i \in I, \\
0 \le x_{ijk} \le G_{ijk}, \quad j \in J_i,\ i \in I, \quad k \in K, \\
\sum_{i \in I}\sum_{j \in J_i}\sum_{k \in K} v_{ijk} x_{ijk} \to \max,
\end{gathered}
\tag{1.1}
$$

где x_{ijk} – объем работ, который будет сделан в планируемом периоде участком j подразделения i по заказу k, $j \in J_i$, $i \in I$, $k \in K$.

В случае, если показателями плана являются: A_k – оставшийся незавершенным на заданный период планирования объем работ по заказу k, $k \in K$; B_i, C_i – соответственно минимально возможный и максимально допустимый объемы работ, которые могут быть выполнены в планируемом периоде подразделением i по всем заказам, $i \in I$; D_{ijk} – максимально возможный объем работ, доступный в планируемом периоде участку j подразделения i по заказу k, $j \in J_i$, $i \in I$, $k \in K$; g_k – доход, который получит предприятие при совершении единицы работы по заказу k, $k \in K$; h_{ij} – затраты предприятия на выполнение участком j подразделения i единицы работы, $j \in J_i$, $i \in I$; тогда задача объемного планирования преобразуется в следующую задачу линейного программирования:

$$
\begin{gathered}
\sum_{i \in I}\sum_{j \in J_i} x_{ijk} \le A_k, \quad k \in K, \\
B_i \le \sum_{k \in K}\sum_{j \in J_i} x_{ijk} \le C_i, \quad i \in I, \\
0 \le x_{ijk} \le D_{ijk}, \quad j \in J_i,\ i \in I,\ k \in K, \\
\sum_{k \in K} g_k \cdot \sum_{i \in I}\sum_{j \in J_i} x_{ijk} - \sum_{i \in I}\sum_{j \in J_i} h_{ij} \cdot \sum_{k \in K} x_{ijk} \to \max.
\end{gathered}
\tag{1.2}
$$

Многопродуктовый характер приведенных оптимизационных задач связан с тем, что на предприятии одновременно осуществляются работы по различным заказам. Сетевая модель рассмотренной производственной системы ставит в соответствие всему предприятию корень ориентированного дерева (исток разнородных продуктов, соответствующих различным заказам), участкам подразделений – листья (стоки разнородных продуктов, отвечающих различным заказам), подразделениям предприятия – остальные вершины ориентированного дерева (элементы, передающие разнородные продукты, связанные с различными заказами).

2. Постановка задачи. Рассмотрим ориентированную q-продуктовую сеть древовидной структуры, корень которой является истоком потоков всех q продуктов, листья – стоками. Пусть $G = (V, A)$ корневое ориентированное дерево, $A \subseteq V^2$, $|V| = n$. Среди множества вершин дерева выделим специальную вершину s – корень дерева, $s \in V$, и подмножество T – множество листьев, $T \subseteq V$. Обозначим через $\pi(i)$ вершину, непосредственно предшествующую вершине i, $i \in V \backslash \{s\}$.

Введем следующие параметры: u_s^k, w_s^k – минимально и максимально возможный суммарный объем исходящих потоков продукта k из истока s, $k = \overline{1, q}$; u_i^k, w_i^k – минимально и максимально возможный объем входящего в вершину i потока продукта k, $i \in V \backslash \{s\}$, $k = \overline{1, q}$; l_{ij}, c_{ij} – нижняя и верхняя пропускные способности дуги (i, j), $(i, j) \in A$; e_{ij}^k – стоимость единицы потока продукта k, проходящего через дугу (i, j), $(i, j) \in A$, $k = \overline{1, q}$.

Тогда задача поиска многопродуктового потока минимальной стоимости (в дальнейшем – задача L_q^{tree}) заключается в определении x_{ij}^k – величин потока продукта k, следующего через дугу (i, j),

$(i, j) \in A$, $k = \overline{1, q}$, для которых выполняются ограничения

$$u_s^k \le \sum_{j|\pi(j) = s} x_{sj}^k \le w_s^k, \quad k = \overline{1, q},$$
$$u_i^k \le x_{\pi(i)i}^k \le w_i^k, \quad k = \overline{1, q}, \; i \in V\backslash\{s\},$$
$$l_{ij} \le \sum_{k=1}^{q} x_{ij}^k \le c_{ij}, \quad (i, j) \in A,$$
$$x_{\pi(i)i}^k - \sum_{j|\pi(j) = i} x_{ij}^k = 0, \quad k = \overline{1, q}, \; i \in V\backslash(T \cup \{s\}),$$
$$x_{ij}^k \ge 0, \quad k = \overline{1, q}, \; (i, j) \in A,$$

и принимает минимальное значение критерий $\sum_{k=1}^{q} \sum_{(i, j) \in A} x_{ij}^k e_{ij}^k$ характеризующий суммарную стоимость потока.

Поставленная задача L_q^{tree} является задачей линейного программирования, для которой могут быть использованы общие методы решения, например симплекс-метод. Транспортная специфика математической модели (коэффициенты матрицы ограничений принимают значения 0.1, –1) позволяет применить к решению сформулированной задачи модифицированный итерационный метод ортогональных проекций Агмона-Моцкина [15]. В работе предлагаются более эффективные алгоритмы решения, основанные на сводимости исходной многопродуктовой ($q \ge 2$) задачи к поиску максимального однопродуктового потока в транспортной сети, а для однопродуктового случая ($q = 1$) – базирующиеся на методе приведенных границ [16]. Интерес здесь также представляет задача поиска целочисленного многопродуктового потока минимальной стоимости (задача $L_{q,Z}^{tree}$), для которой дополнительно предполагается, что $x_{ij}^k \in Z$, $(i, j) \in A$, $k = \overline{1, q}$.

3. Алгоритм решения задачи L_1^{tree}. Для однопродуктового потока система ограничений задачи L_1^{tree} преобразуется к следующему виду:

$$\begin{gathered} u_s \le \sum_{j|\pi(j) = s} x_{sj} \le w_s, \\ u_i \le x_{\pi(i)i} \le w_i, \quad i \in V\backslash\{s\}, \\ l_{ij} \le x_{ij} \le c_{ij}, \quad (i, j) \in A, \\ x_{\pi(i)i} - \sum_{j|\pi(j) = i} x_{ij} = 0, \quad i \in V\backslash(T \cup \{s\}), \\ x_{ij} \ge 0, \quad (i, j) \in A, \end{gathered} \tag{3.1}$$

а критерий, характеризующий суммарную стоимость потока, запишем как

$$\sum_{(i, j) \in A} x_{ij} e_{ij} \to \min. \tag{3.2}$$

Индекс $k = 1$ для удобства изложения здесь опущен.

Преобразуем систему (3.1) следующим образом. Для вершины j определим минимально и максимально возможные суммарные объемы потока продукта соответственно через $U_j = \max(u_j, l_{\pi(j)j})$ и $W_j = \min(w_j, c_{\pi(j)j})$, $j \in V\backslash\{s\}$; $U_s = u_s$, $W_s = w_s$. Пусть

$$y_i = x_{\pi(j)j}, \quad j \in V\backslash\{s\}; \quad y_s = \sum_{i|\pi(i) = s} x_{si}.$$

Тогда (3.1) приводится к виду

$$\begin{gathered} U_i \le y_i \le W_i, \quad i \in V, \\ y_i = \sum_{j|\pi(j) = i} y_i, \quad i \in V\backslash T, \\ y_i \ge 0, \quad i \in V. \end{gathered} \tag{3.3}$$

Критерий оптимальности установим как

$$\sum_{i \in V\backslash\{s\}} y_i e_i' \to \min, \tag{3.4}$$

где $e_i' = e_{\pi(i)i}$, $i \in V\backslash\{s\}$. Очевидно, что задача (3.1), (3.2) эквивалентна постановке (3.3), (3.4).

Проведем процедуру “приведения границ” для системы ограничений (3.3), т.е. с помощью следующих рекуррентных соотношений определим величины U_i^p, W_i^p, $i \in V$:

$$\begin{gathered} U_i^p = U_i, \quad i \in T, \\ W_i^p = W_i, \quad i \in T, \\ U_i^p = \max\Big(U_i, \sum_{j|\pi(j) = i} U_j^p\Big), \quad i \in V\backslash T, \\ W_i^p = \min\Big(W_i, \sum_{j|\pi(j) = i} W_j^p\Big), \quad i \in V\backslash T. \end{gathered} \tag{3.5}$$

Т е о р е м а о п р и в е д е н н ы х г р а н и ц а х. Система (3.3) совместна тогда и только тогда, когда $U_i^p \le W_i^p$, $i \in V$.

Д о к а з а т е л ь с т в о. Необходимость условий теоремы очевидна. Доказательство достаточности проведем конструктивно. Покажем, что при справедливости теоремы специальным образом построенный вектор $\vec{y}^0 \in R^n$ является допустимым решением системы (3.3). Так как по условиям теоремы $U_s^p \le W_s^p$, то найдется такое значение y_s^0, что $y_s^0 \in [U_s^p, W_s^p]$, а тем самым (из-за того, что $[U_s^p, W_s^p] \subseteq [U_s, W_s]$) y_s^0 удовлетворяет соответствующему условию (3.3).

Компоненты y_j^0 вектора $\vec{y}^0$, для которых $\pi(j) = s$ определяются из следующих выражений:

$$\sum_{j|\pi(j) = s} y_j^0 = y_s^0; \quad U_j^p \le y_j^0 \le W_j^p, \quad \pi(j) = s.$$

Эта система совместна, так как $y_s^0 \in [U_s^p, W_s^p]$ и из рекуррентных соотношений (3.5) следует, что

$$U_s^p \ge \sum_{j|\pi(j) = s} U_j^p, \quad W_s^p \le \sum_{j|\pi(j) = s} W_j^p.$$

Продолжая находить аналогичным образом значения компонент вектора $\vec{y}^0$, построим требуемое допустимое решение системы (3.3). *Теорема доказана.*

Из (3.5) получаем, что процедура приведения границ имеет вычислительную сложность $O(n)$. Тогда конструктивная схема доказательства теоремы о приведенных границах порождает ряд следствий.

Следствие 1. Существует алгоритм поиска допустимого решения задачи L_1^{tree} с вычислительной сложностью $O(n)$.

Следствие 2. Имеется алгоритм решения задачи L_1^{tree} с вычислительной сложностью $O(n^2)$.

Доказательство. Схема доказательства теоремы о приведенных границах позволяет строить эффективную процедуру решения задачи L_1^{tree}. Сначала для системы ограничений (3.3) проведем процедуру приведения границ. Если условия теоремы о приведенных границах будут выполнены, то задача L_1^{tree} имеет решение. Найдем все $|T|$ цепей, соединяющих корень дерева с листьями. Для каждой цепи $(s, j_1, j_2, \dots, j_t), j_t \in T$, определим величину $\sum_{i=1}^{t} e'_{j_t}$ – стоимость единицы потока продукта по этой цепи. Начиная с цепи, для которой стоимость единицы потока продукта максимальна по модулю, не нарушая приведенные границы, пропустим минимально возможный поток по этой цепи, если стоимость единицы потока положительна, и максимально возможный поток в противном случае. Уменьшим левые и правые границы на величину потока во всех вершинах, соответствующих выбранной цепи. Исключим эту цепь из рассмотрения, при этом условия теоремы о приведенных границах будут выполнены. Выберем следующую цепь с максимальной по модулю стоимостью единицы потока. По выбранной цепи пропустим минимально (максимально) возможный поток, уменьшим на величину потока левые и правые границы соответствующих вершин. Процесс повторим $|T|$ раз. Найденный суммарный поток определит оптимальное решение задачи L_1^{tree}. Если для рассматриваемой древовидной сети максимальная длина цепи, соединяющей корень с листьями, равна k, то предложенный алгоритм будет просматривать не больше $(k + 1)|T|$ вершин, что и определяет указанную оценку вычислительной сложности. *Следствие доказано.*

Замечание 1. Приведенная оценка $O(n^2)$ алгоритма решения задачи L_1^{tree}, описанного при доказательстве следствия 2, является достижимой. Так, например, данная оценка имеет место для дерева, состоящего из $2n$ вершин и n листьев, соединенных с корнем цепями длины $1, 2, \dots, n$ соответственно. С другой стороны, для широкого класса прикладных задач, для которых сетевая древовидная модель не содержит цепей между корнем и листьями, превышающих по длине некоторую заданную константу, вычислительная сложность алгоритма становится линейной. К таким задачам относятся некоторые многоиндексные задачи с ограничениями транспортного типа. Так, среди приведенных выше задач объемного многономенклатурного планирования (1, 1) и (1, 2), которые относятся к классу L_q^{tree}, постановка (1.1) сводится к задаче L_1^{tree}, для которой алгоритм, описанный в следствии 2, имеет линейную вычислительную сложность.

4. Алгоритм решения задачи L_q^{tree}. Предлагаемый алгоритм решения задачи L_q^{tree} поиска многопродуктового потока минимальной стоимости в ориентированной древовидной сети основан на ее сводимости к задаче поиска (однопродуктовой) циркуляции минимальной стоимости в сети произвольной структуры с двусторонними пропускными способностями дуг, обозначаемой задачей L_1. Будем применять следующую концепцию сводимости.

Определение 1. Матрица A', $A' = \|a'_{ij}\|_{m' \times n'}$, сводится к матрице A'', $A'' = \|a''_{ij}\|_{m'' \times n''}$, если существует отображение $\alpha: \{1, 2, \dots, n'\} \longrightarrow \{1, 2, \dots, n''\}$, такое, что для любой задачи линейного программирования $f'(\vec{x}'^0) = \max\{(\vec{x}', \vec{c}') | A'\vec{x}' \le \vec{b}', \vec{x}' \ge 0\}$, обозначаемой L', можно построить задачу линейного программирования $f''(\vec{x}''^0) = \max\{(\vec{x}'', \vec{c}'') | A''\vec{x}'' \le \vec{b}'', \vec{x}'' \ge 0\}$, именуемую L'', причем:

если $\vec{x}''^0 = (x''_1, x''_2, \ldots, x''_{n''})$ – оптимальное (допустимое) решение L'', то $\vec{x}'^0 = ((x''_{\alpha(1)}, x''_{\alpha(2)}, \ldots, x''_{\alpha(n')})$ – оптимальное (допустимое) решение L';

задача L' не совместна, если не совместна L'';

все компоненты вектора $\vec{b}''$ целочисленны, если целочисленны все компоненты $\vec{b}'$.

Определение 2. Задача линейного программирования L' сводится к задаче линейного программирования L'', если матрица системы ограничений задачи L' сводится к матрице системы ограничений задачи L''.

Применение введенной концепция сводимости задач линейного программирования позволяет гарантировать, что если L' сводится к L'', то

решив задачу L'', можно определить решение задачи L' как подмножество компонент решения задачи L'', которые определяются отображением α;

матрица A'' и отображение α остаются постоянными при любых значениях векторов $\vec{b}'$, $\vec{c}'$ и зависят лишь от вида матрицы A'.

Теорема сводимости. Задача L_q^{tree} сводится к задаче L_1.

Доказательство. Проведем конструктивное доказательство, основанное на схеме построения задачи L_1, соответствующей задаче L_q^{tree}. Пусть L_q^{tree} определяется графом $G = (V, A)$. Используем вспомогательную вершину v и для каждой вершины i графа G введем q дополнительных вершин i^k, $i = V$, $k = \overline{1, q}$. Построим множество вершин и множество дуг, задающих структуру сети задачи L_1. Множество вершин: $V \cup \{i^k | i \in V, k = \overline{1, q}\} \cup \cup \{v\}$. Множество дуг: $A \cup A_q \cup A_{T,q} \cup A_{s,q} \cup \{(v, s)\}$, где $A_q = \{(j^k, i^k) | (i, j) \cup A, k = \overline{1, q}\}$, $A_{T,q} = \{(i, i^k), i \in T, k = \overline{1, q}\}$, $A_{s,q} = \{(s^k, v) | k = \overline{1, q}\}$.

Определим пропускные способности дуг для задачи L_1: нижняя и верхняя пропускные способности дуги (i, j), $(i, j) \in A$, равны l_{ij} и c_{ij} соответственно; пропускные способности дуги (j^k, i^k), $(j^k, i^k) \in A_q$, составляют u_j^k, w_j^k; пропускные способности дуги (s^k, v), $(s^k, v) \in A_{s,q}$, установим u_s^k, w_s^k; остальные дуги имеют нулевую нижнюю и неограниченную верхнюю пропускные способности. Стоимость единицы потока, проходящего через дугу (j^k, i^k), $(j^k, i^k) \in A_q$, равна e_{ij}^k; прочие дуги характеризуются нулевыми стоимостями.

Вводя отображение α, трансформирующее множество переменных задачи L_q^{tree} в множество переменных задачи L_1, каждой переменной x_{ij}^k, $(i, j) \in A$, $k = \overline{1, q}$, поставим в соответствие величину потока через дугу (j^k, i^k), $(j^k, i^k) \in A$, а значит, получим сведение задачи L_q^{tree} к L_1. *Теорема доказана.*

Следствие 3. Существует алгоритм решения задачи L_q^{tree} с вычислительной сложностью $O(n^3q^3\log^2(nq))$.

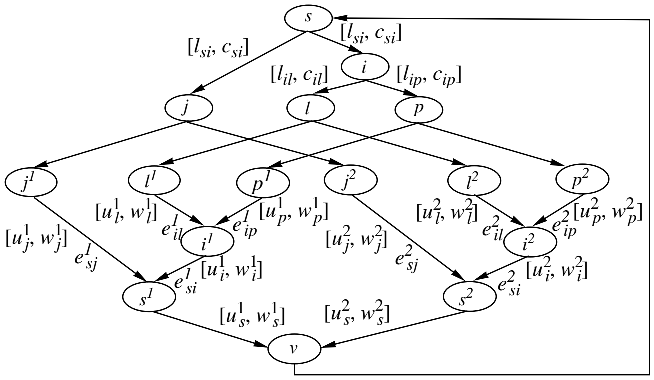
Доказательство. По теореме сводимости решение задачи L_q^{tree} может быть определено как подмножество компонент решения задачи L_1. При этом из конструктивной схемы доказательства следует, что сеть для задачи L_1 содержит $O(nq)$ вершин и $O(nq)$ дуг.

Для решения задачи L_1 можно воспользоваться известными алгоритмами, например сильнополиномиальным алгоритмом Галиля и Тардош [17]. Тогда вычислительная сложность алгоритма решения задачи L_1, а следовательно, и исходной задачи L_q^{tree} будет составлять $O((nq)^2\log(nq)(nq + nq\log(nq)))$, что и определяет указанную оценку вычислительной сложности. *Следствие доказано.*

Утверждение. Если задача линейного программирования сводится к задаче L_1, то матрица системы ограничений исходной задачи абсолютно унимодулярна.

Доказательство. Проведем доказательство от противного. Предположим, что задача линейного программирования сводится к L_1, но соответствующая матрица системы ограничений не является абсолютно унимодулярной. Условия абсолютной унимодулярности в случае целочисленных параметров задачи линейного программирования являются необходимыми и достаточными для целочисленности всех вершин многогранника, соответствующего системе ограничений задачи линейного программирования [18]. Тогда можно построить такую линейную целевую функцию, при которой задача линейного программирования будет иметь единственное оптимальное нецелочисленное решение. Это противоречит теореме о целочисленности потока [19], по которой если задача имеет решение, то всегда существует ее оптимальное целочисленное решение. *Утверждение доказано.*

Из доказанных утверждения и теоремы сводимости непосредственно следует, что предлагаемый алгоритм решения задачи L_q^{tree}, используемый при доказательстве следствия 3, также может быть применен для решения задачи $L_{q,Z}^{tree}$.



Рисунок

С л е д с т в и е 4. Задача $L_{q,Z}^{tree}$ полиномиально разрешима.

З а м е ч а н и е 2. Теорема о приведенных границах может быть использована в качестве необходимых условий совместности задачи L_q^{tree}. Для этого необходимо ограничения многопродуктовой задачи L_q^{tree} свернуть, преобразуя ее в однопродуктовую задачу L_1^{tree}. Процедура сведения заключается в переходе от ограничений задачи L_q^{tree} к ограничениям (3.3) задачи L_1^{tree}, исходные параметры для которых находятся из соотношений

$$U_j = \max\left(\sum_{k=1}^{q} u_j^k, c_{\pi(j)j}\right),$$

$$W_j = \min\left(\sum_{k=1}^{q} w_j^k, c_{\pi(j)j}\right), \quad j \in V\backslash\{s\};$$

$$U_s = \sum_{k=1}^{q} u_s^k, \quad W_s = \sum_{k=1}^{q} w_s^k.$$

5. Примеры решения задач. Рассмотрим численный пример решения задачи L_q^{tree} при $q = 2$ и $q = 1$. Пусть $V = \{s, i, j, l, p\}$, $T = \{j, l, p\}$, $A = \{(s, i), (s, j), (i, l), (i, p)\}$.

Исследуем случай $q = 2$. Определим следующие значения исходных параметров задачи L_2^{tree}:

$$u_s^1 = 5, \ w_s^1 = 20, \ u_s^2 = 7, \ w_s^2 = 18;$$

$$u_i^1 = 3, \ w_i^1 = 7, \ u_i^2 = 6, \ w_i^2 = 15;$$

$$u_j^1 = 2, \ w_j^1 = 4, \ u_j^2 = 3, \ w_j^2 = 6;$$

$$u_l^1 = 1, \ w_l^1 = 4, \ u_l^2 = 4, \ w_l^2 = 6;$$

$$u_p^1 = 0, \ w_p^1 = 2, \ u_p^2 = 2, \ w_p^2 = 5;$$

$$l_{si} = 8, \ c_{si} = 17, \ e_{si}^1 = -2, \ e_{si}^2 = 1;$$

$$l_{sj} = 2, \ c_{sj} = 8, \ e_{sj}^1 = 2, \ e_{sj}^2 = 0;$$

$$l_{il} = 6, \ c_{il} = 10, \ e_{il}^1 = 3, \ e_{il}^2 = -4;$$

$$l_{ip} = 4, \ c_{ip} = 9, \ e_{ip}^1 = -3, \ e_{ip}^2 = 1.$$

Построим соответствующую задачу L_1, описанную при доказательстве теоремы сводимости (рисунок). Циркуляция минимальной стоимости в L_1 определяет в соответствии с построенным отображением α следующее решение L_2^{tree}:

$$x_{si}^1 = 3, \ x_{sj}^1 = 2, \ x_{il}^1 = 1, \ x_{ip}^1 = 2;$$

$$x_{si}^2 = 8, \ x_{sj}^2 = 3, \ x_{il}^2 = 6, \ x_{ip}^2 = 2.$$

Оптимальное значение критерия при этом равно –19.

Пусть $q = 1$. Определим значения параметров задачи L_1^{tree}, свернув ограничения задачи L_2^{tree}:

$$U_s = 12, \ W_s = 38;$$

$$U_i = 9, \ W_i = 17, \ e_i' = -1;$$

$$U_j = 5, \ W_j = 8, \ e_j' = 2;$$

$$U_l = 6, \ W_l = 10, \ e_l' = -1;$$

$$U_p = 4, \ W_p = 7, \ e_p' = -2.$$

После процедуры приведения границ исходные параметры системы будут иметь вид

$$U_s^p = 15, \ W_s^p = 25; \ U_i^p = 10, \ W_i^p = 17;$$

$$U_j^p = 5, \ W_j^p = 8; \ U_l^p = 6, \ W_l^p = 10;$$

$$U_p^p = 4, W_p^p = 7.$$

Система совместна, и ее оптимальное решение: $y_s = 22$, $y_i = 17$, $y_j = 5$, $y_l = 10$, $y_p = 7$. Оптимальное значение критерия при этом равно –31.

Заключение. Разработанные алгоритмы были использованы при создании программной системы “Распределение ограниченных ресурсов в иерархических системах транспортного типа”. Система прошла апробацию на ряде предприятий г. Н.Новгорода. Проведенные вычислительные эксперименты позволяют говорить об эффективности данных алгоритмов при решении большеразмерных прикладных задач. Так, среднее время поиска максимального многопродуктового потока в сети с 520 узлами и 30 продуктами (что соответствует решению задачи объемно-календарного планирования для подразделений опытного производства ФГУП "Опытное конструкторское бюро машиностроения им. И.И.Африкантова" г. Н.Новгород) составило около 6 мин (параметры вычислительной системы – Pentium IV, 1200Mhz, 256 DDR, OS Windows 2000). Отметим, что при этом математическая модель задачи L_{30}^{tree} содержала 15570 переменных, 6090 двусторонних ограничений на подсуммы и 570 уравнений баланса.

СПИСОК ЛИТЕРАТУРЫ

1. *Прилуцкий М.Х.* Многокритериальные многоиндексные задачи объемно-календарного планирования // Изв. РАН. ТиСУ. 2007. № 1. С. 78–82.
2. *Афраймович Л.Г., Прилуцкий М.Х.* Многоиндексные задачи распределения ресурсов в иерархических системах // АиТ. 2006. № 6. С. 194–205.
3. *Deikmann R., Frommer A., Monien B.* Efficient schemes for nearest neighbor load balancing // Parallel Computing. V. 25. 1999. P. 789–812.
4. *Corban A.* Transportation problem with intermediate centers // Rev.Roum. Mat. Pures. Appl. 1971. V. 16. № 9. P. 128–134.
5. *Goldberg A.V., Rao S.* Beyond the flow decomposition barrier // J. ACM. 1998. V. 45. № 5. P. 783–797.
6. *Goldberg A.V., Tarjan R.E.* Solving minimum-cost flow problems by successive approximation // Mathematics of Operations Research. 1990. V. 15. № 3. P. 430–466.
7. *Гольштейн Е.Г., Соколов Н.А.* Новый метод решения многопродуктовой транспортной задачи // Экономика и мат. методы. 1995. Т. 31. Вып. 2. С. 128–136.
8. *Kamath A., Palmon O.* Improved interior point algorithms for exact and approximate solution of multicommodity flow problems // Proc. 6th Annual ACM-SIAM Symp. on Discrete algorithms. San Francisco, 1995. P. 502–511.
9. *Itai A.* Two-Commodity Flow // J. ACM. 1978. V. 25. № 4. P. 596–611.
10. *Kenji O.* Approximate max-integral-flow/min-multicut theorems // Proc. 36th Annual ACM Symp. on Theory of computing. Chicago. 2004. P. 539–545.
11. *Garg N., Konemann J.* Faster and simpler algorithms for multicommodity flow and other fractional packing problems // Proc. 39th Annual Symp. on Foundations of Computer Science. Washington. 1998. P. 300–309.
12. *Малашенко Ю.Е., Новикова Н.М.* Модели неопределенности в многопользовательских сетях. М.: Эдиториал УРСС, 1999.
13. *Ху Т.* Целочисленное программирование и потоки в сетях. М.: Мир, 1974.
14. *Garg N., Vazirani V.V., Yannakakis M.* Primal-dual approximation algorithms for integral flow and multicut in trees // Algorithmica. 1997. V. 18. № 1. P 3–20.
15. *Прилуцкий М.Х.* Многокритериальное распределение однородного ресурса в иерархических системах // АиТ. 1996. № 2. С. 24–29.
16. *Прилуцкий М.Х.* Распределение однородного ресурса в иерархических системах древовидной структуры // Тр. междунар. конф. "Идентификация систем и задачи управления SICPRO'2000". М.: ИПУ им. В.А. Трапезникова РАН. 2000. С. 2038–2049.
17. *Galil Z., Tardos E.* An $O(n^2 \log n(m + n\log n))$ min-cost flow algorithm // J. ACM. 1988. V. 35. № 2. P. 374–386.
18. *Гофман А.Д., Краскал Д.Б.* Целочисленные граничные точки выпуклых многогранников // Линейные неравенства и смежные вопросы. М.: Изд-во иностр. лит., 1959. С. 325-–347.
19. *Форд Л., Фалкерсон Д.* Потоки в сетях. М.: Мир, 1966.