УДК 004.021

Труш А. И., к.т.н.,
Мацуева К. А.,
Мацуева К. А.

АЛГОРИТМ РЕШЕНИЯ БЛОЧНО-СИММЕТРИЧНЫХ ЗАДАЧ ПРОЕКТИРОВАНИЯ СИСТЕМ ОБРАБОТКИ ДАННЫХ

Национальный авиационный университет

Предложена постановка блочно-симметричных задач проектирования систем обработки данных. Приведена общая модель и схема её реализации, определены свойства и особенности задач данного класса. Предложен алгоритм итеративных отображений решения блочно-симметричных задач проектирования систем обработки данных полиномиальной вычислительной сложности

Введение

Ряд прикладных задач: проектирования модульного программного обеспечения и массивов базы данных информационных систем, распределение программных модулей и массивов базы данных по узлам вычислительных сетей, выбор проектов в условиях ограниченных ресурсов можно сформулировать в виде нового класса задач – блочно-симметричных моделей дискретного программирования. В отличие от традиционных моделей, модели этого класса позволяют формулировать задачи с несколькими типами переменных различной природы, проводить декомпозицию сложных задач на блоки с единой целевой функцией и разрабатывать эффективные алгоритмы, имеющие полиномиальную вычислительную сложность.

Постановка задачи

Пусть задано множество объектов $A = \{a_i; i = \overline{1,I}\}$ и множество объектов $B = \{b_j; j = \overline{1,J}\}$ с элементами различных типов, а также взаимосвязи между элементами этих множеств, которые определяются матрицей

$$W = \|\omega_{ij}\|,\ i = \overline{1,I},\ j = \overline{1,J},$$

Элементы которой целочисленные и булевы. Необходимо объединить элементы множество A в непересекающиеся подмножества $A_n, n = \overline{1,N}$, а элементы множества B - непересекающейся подмножества $B_m, m = \overline{1,M}$, таким образом, чтобы доставить экстремум целевой функции $F(A_n, B_m)$.

Для формализованной постановки задачи введем следующие переменные. Пусть $X = \|x_{in}\|, i = \overline{1,I}, n = \overline{1,N}$ - булева матрица, где $x_{in} = 1$, если i-й элемент распределяется в n-ю группу, $x_{in} = 0$ в противном случае. Аналогично $Y = \|y_{jm}\|, j = \overline{1,J}, m = \overline{1,M}$, где $y_{im} = 1$, если j-й элемент распределяется в m-ю группу и $y_{jm} = 0$ в противном случае. В общем случае матрицы переменных X и Y могут быть целочисленными.

Определим на множестве $A \times B$ функцию $F(X,Y)$, зависящую от распределения элементов множеств A и B по подмножествам A_n и B_m. Соответственно на множестве A - функции $\varphi_k(X), k = \overline{1,K}$, а на множестве B - функции $\psi_s(Y),\ s = \overline{1,S}$, определяющие ограничения на множествах A и B.

Блочно-симметричная задача дискретного программирования формулируется следующим образом:

$$F(X,Y) \to extr, \tag{1.1}$$

при ограничениях

$$\varphi_k(X) \le \varphi_{k0},\ k = \overline{1,K} \tag{1.2}$$

$$\psi_s(Y) \le \psi_{s0},\ s = \overline{1,S} \tag{1.3}$$

В множестве ограничений (1.2) и (1.3) в зависимости от постановок задач знаки неравенств могут меняться на противоположеные.

В общем случае двухиндексные матрицы – переменных X и Y и заданная матрица W могут быть целочисленными.

Рассмотрим задачу при условии, когда переменные X, Y и W - булевы матрицы. В качестве функции $F(X,Y)$ часто используют функцию вида $F(Z)$, где

$$Z = XWY \qquad (1.4)$$

Рассмотрим выражение (1.4), которое представляет собой произведение матриц переменных X и Y и заданной матрицы W, на которой определена целевая функция. В отличие от традиционных постановок задач дискретного программирования в данной постановке имеются два типа переменных X и Y, переменные X и Y симметричны относительно заданной матрицы W.

В задаче (1.1) -(1.3) можно выделить множество ограничений вида (1.2), которые зависят от переменной X, и множество ограничений вида (1.3), которые зависят от переменной Y.

Функционал вида $F(X,Y)$ можно представить следующим образом:

$$F(p(X), g(Y)) \to extr \qquad (1.5)$$

$$p(X) \to extr \qquad (1.6)$$

$$\varphi_k(X) \le \varphi_{k0},\ k = \overline{1,K} \qquad (1.7)$$

$$g(Y) \to extr \qquad (1.8)$$

$$\psi_s(Y) \le \psi_{s0},\ \ s = \overline{1,S}\,. \qquad (1.9)$$

В постановке задачи (1.5) - (1.9) выделим блок функции (1.6), (1.7), зависящий только от переменной X, и блок функций (1.8), (1.9), зависящий только от переменной Y, объединенных единым функционалом вида (1.5). Заметим, что в ряде постановок задач может быть блок ограничений вида

$f_r(X,Y) \le f_{r0},\ r = \overline{1,R}$ (1.10), зависящий от переменных X и Y.

В этом случае можно выделить блок функционала цели вида (1.5), (1.10).

Определение 1. Блочносимметричной задачей дискретного программирования назовем задачу вида (1.5) - (1.9), где переменные X и Y и значения функций $F(p(X), g(Y))$, $p(X)$, $g(Y)$ - целые, либо булевые [1].

Рассмотрим выражение (1.4). из него следует что переменные X и Y симметричны относительно заданной матрицы W и функция (1.4) может быть определена как слева направо, так и наоборот, т.е.

$$Z = XWY = YWX \qquad (11)$$

При проектировании систем обработки данных (СОД) на этапе предпроектного анализа объектов определяется перечень прикладных задач обработки данных, подлежащих автоматизации, последовательность их решения, исходные документы, используемые для решения прикладных задач, характеристики прикладных задач и документов.

Декомпозиция сложных систем обработки данных

На этапе технического проектирования формируется общая функциональная структура, состав и последовательность решения прикладных задач, структура прикладного программного обеспечения, структура базы данных, определяется общесистемное программное обеспечение проектируемой системы обработки данных. При большом числе прикладных задач и сложном документообороте возникает необходимость декомпозиции системы на кластеры.

Кластер прикладных задач - объединение задач в подмножества, а кластерами документов объединение документов в подмножества и установление взаимосвязей между соответствующими подмножествами. Таким образом, разрабатываемая система может быть представлена в виде двудольного графа. Результатом декомпозиции системы является также двудольный граф, вершинами верхнего уровня которого являются кластеры функциональных задач, вершинами нижнего уровня кластеры исходных документов. Взаимосвязи между ними отражают интегрированные связи между кластерами.

В качестве критерия эффективности процесса декомпозиции исходной системы используем минимум информационных взаимосвязей между кластерами задач и документов. Введём следующие переменные и обозначения. Пусть,

$X = \|x_{mi}\|$, $m = \overline{1,M}$, $i = \overline{1,I}$ - переменная отражающая распределенные i- ой прикладной задачи в m-ой кластер (группу) задач. В данном случае

$$x_{mi} = \begin{cases} 1, \text{если } i - \text{ая прикладная задача} \\ \text{распределяется в } m - \text{й кластер}, \\ 0, \text{ в противном случае}. \end{cases}$$

Аналогично введём переменную

$Y = \|y_{jn}\|$, $n = \overline{1,N}$, $j = \overline{1,J}$, где

$$y_{jn} = \begin{cases} 1, \text{ если } j - \text{й документ распределен} \\ \text{в } n - \text{й кластер документов}, \\ 0, \text{ в противном случае}. \end{cases}$$

В ряде случаев на данном этапе определяются характеристики задач и документов.

Введем t_i - время разработки i-ой задачи, v_j - объем j-ого документа, c_{ij} - общая стоимость разработки i-ой задачи и j-го документа, τ_j - время разработки и подготовки j-го документа, c_i - стоимость разработки i-ой задачи, s_j - стоимость подготовки j-го документа.

Пусть, $A = \{a_i, i = \overline{1,I}\}$ - множество прикладных задач обработки данных, подлежащие автоматизации; $B = \{b_j, j = \overline{1,J}\}$ - множество исходных документов, используемое для решения прикладных задач. Задана, матрица $W = \|\omega_{ij}\|$, $i = \overline{1,I}$, $j = \overline{1,J}$, где $\omega_{ij} = 1$, если j-й исходный документ используется для решения i-ой прикладной задачи системы и $\omega_{ij} = 0$, в противном случае.

Определим дополнительные переменные следующим образом:

$$\alpha_{mj} = \begin{cases} 1, \text{ если } \sum_{i=1}^{I} x_{mi}\omega_{ij} \ge 1, \\ 0, \sum_{i=1}^{I} x_{mi}\omega_{ij} = 0. \end{cases}$$

Данная переменная отражает использование j-го документа для решения задач m-го кластера.

$$\beta_{in} = \begin{cases} 1, \text{ если } \sum_{j=1}^{J} \omega_{ij} y_{jn} \ge 1, \\ 0, \text{ если } \sum_{j=1}^{J} \omega_{ij} y_{jn} = 0. \end{cases}$$

Переменная β_{in} отражает использование в процессе решения i-ой задачи n-го кластера документов.

Взаимосвязи между кластерами прикладных задач и документов определяется из выражения:

$$\gamma_{mn} = \begin{cases} 1, \text{ если } \sum_{i=1}^{I}\sum_{j=1}^{J} \alpha_{mj}\beta_{in} \ge 1, \\ 0, \text{ если } \sum_{i=1}^{I}\sum_{j=1}^{J} \alpha_{mj}\beta_{in} = 0. \end{cases}$$

Задачу декомпозиции СОД сформулируем следующим образом.

Необходимо минимизировать функцию вида

$$\min \sum_{m=1}^{M}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N} x_{mi}\omega_{ij} y_{jn} . \qquad (1.12)$$

При ограничениях на:

– включение каждой прикладной задачи только в один кластер

$$\sum_{m=1}^{M} x_{mi} = 1,\ i = \overline{1,I}; \qquad (1.13)$$

– включение документа только в один кластер документов

$$\sum_{n=1}^{N} x_{ni} = 1,\ j = \overline{1,J}; \qquad (1.14)$$

– время разработки каждого кластера задач

$$\sum_{i=1}^{I} t_i x_{mi} + \sum_{j=1}^{J} \tau_j \alpha_{mj} \le T_m,\ m = \overline{1,M}; \qquad (1.15)$$

– стоимость проектирования каждого кластера задач

$$\sum_{i=1}^{I} c_i x_{mi} + \sum_{j=1}^{J} s_j \alpha_{mj} \le R_m, m = \overline{1,M}; \qquad (1.16)$$

– число прикладных задач в кластере

$$\sum_{i=1}^{I} x_{mi} \le \rho_0,\ m = \overline{1,M}; \qquad (1.17)$$

– число исходных документов в кластере

$$\sum_{j=1}^{J} y_{jn} \le g_0,\ n = \overline{1,N}. \qquad (1.18)$$

В результатах декомпозиции сложных систем обработки данных на кластеры на этапе технического проектирования необходимо для каждого кластера разработать модульную блок-схему прикладного программного обеспечения и базы данных.

Определение 2. Модульной блок-схемой обработки данных будем называть совокупность процедур, объединенных в модули и множество информационных элементов, объединенных в массивы (таблица) данных с отображением интегрированных связей между модулями и массивами[1].

При синтезе блок-схемы должны быть учтены основные характеристики и ограничения систем управления базами данных и вычислительных средств, на которых предполагается эксплуатация создаваемого программного и информационного обеспечения.

Рассмотрим задачу синтеза модульной блок-схемы системы обработки данных, минимизирующей общее число связей между модулями и массивами базы данных.

Для постановки задачи введем следующие обозначения. Пусть, $P=\{p_r; r=\overline{1,R}\}$ - множество процедур обработки данных для решения прикладных задач системы; $D=\{d_l;\overline{1,L}\}$ - множество информационных элементов, необходимых для реализации процедур из множеств P. На множестве $P\times D$ введем отношение W, определяемое матрицей $\|w_{rl}\|$, где:

$$w_{rl}=\begin{cases}1, \textit{если } l-\textit{ый информационный элемент} \\ \textit{необходим для реализации } r-\textit{й} \\ \textit{процедуры}; \\ 0, \textit{в противном случае}\end{cases}$$

Необходимо синтезировать модульную блок-схему СОД путем распределения множества процедур по модулям обработки данных, множества информационных элементов – в логическую структуру базы данных и установить оптимальные взаимосвязи между модулями и логической структурой базы данных, минимизирующих число взаимосвязей между компонентами блок-схемы.

Введем следующие переменные:

$$x_{vr}=\begin{cases}1, \textit{ если } r-\textit{ая процедура обработки} \\ \textit{данных распределяется} \\ \textit{в состав } v-\textit{го модуля}; \\ 0, \textit{иначе}.\end{cases}$$

$$y_{lf}=\begin{cases}1, \textit{если } l-\textit{ый информационный элемент} \\ \textit{распределяется в } f-\textit{й логический} \\ \textit{массив базы данных}; \\ 0, \textit{иначе}.\end{cases}$$

Введем вспомогательные переменные:

$$\pi_{vl}=\begin{cases}1, \textit{если } \sum_{r=1}^{R} x_{vr}\omega_{rl}\geq 1, \\ 0, \textit{если } \sum_{r=1}^{R} x_{vr}\omega_{rl}=0.\end{cases}$$

Переменная отражает использование l-го информационного элемента v-м модулем, т.е. если хотя бы один информационный элемент обрабатывается r-ой процедурой, включенный в состав v-го модуля, то данный элемент также обрабатывается этим модулем.

$$q_{rf}=\begin{cases}1, \textit{если } \sum_{l=1}^{L}\omega_{rl}y_{lf}\geq 1, \\ 0, \textit{если } \sum_{l=1}^{L}\omega_{rl}y_{lf}=0.\end{cases}$$

Переменная отражает использование f-го массива данных r-ой процедурой, т.е. если процедура использует хотя бы один информационный элемент, включенной в состав f-го массива данных, то данная процедура использует этот массив.

Переменную отражающую взаимосвязь между модулями блок-схемы и массивами базы данных можно определить следующим образом:

$$z_{vf} = \begin{cases} 1, \textit{если} \sum_{r=1}^{R} x_{vr} q_{rf} \geq 1, \\ 0, \textit{если} \sum_{r=1}^{R} x_{vr} q_{rf} = 0; \end{cases}$$

либо,

$$z_{vf} = \begin{cases} 1, \textit{если} \sum_{l=1}^{L} \pi_{vl} y_{lf} \geq 1, \\ 0, \textit{если} \sum_{l=1}^{L} \pi_{vl} y_{lf} = 0. \end{cases}$$

Определение указанных переменных вытекает из свойства симметричности блочно-симметричных задач.

Задача проектирования модульных блок-схем систем обработки данных (МСОД) формулируется следующим образом.

Необходимо синтезировать модульную блок схему путем распределения множества процедур по модулям обработки данных, множества информационных элементов – в логическую структуру базы данных и установить оптимальные взаимосвязи между модулями и логической структурой базы данных, минимизирующих число взаимосвязей между компонентами блок-схем.

При этом должны быть учтены такие требования, как ограниченность размеров модулей и логических массивов базы данных, отсутствие дублирования процедур в модулях и информационных элементов в логических массивах.

Математическая постановка задачи имеет вид:

$$\Phi(X,Y) = \sum_{v=1}^{V}\sum_{r=1}^{R}\sum_{l=1}^{L}\sum_{f=1}^{F} x_{vr} w_{rl} y_{lf} \to \min \quad (2.1.)$$

Целевую функцию (2.1) блочно-симметричной задачи разработки модульной блок-схемы удобно представить в матричной форме.

$$\Phi(X,Y) = \sum_{v=1}^{V}\sum_{f=1}^{F} z_{vf} \qquad (2.8)$$

или,

$$Z = XWY. \qquad (2.9)$$

Решением задач (2.1)-(2.7) являются множества булевых матриц $\{X,Y,Z\}$, в котором X - состав модулей блок-схемы, Y - состав массивов базы данных блок-схемы, Z - взаимосвязи между модулями и массивами базы данных блок-схемы, а также оптимальные значение целевой функции $\Phi(X,Y)$.

Алгоритм итеративных отображений

Рассмотрим алгоритм решения блочно-симметричных задач вида (1.12)-(1.16).

Для описания алгоритма введем следующие понятия.

В случае, если в процессе проектирования модульных блок-схем не заданы число разрабатываемых модулей V и массивов базы данных F, они могуть быть определены и следующих соотношений $V = \left]\frac{R}{M}\right[$, $F = \left]\frac{L}{N}\right[$, где M и N соответственно максимальное число процедур в модуле и максимальное число информационных элементов в массивах базы данных. Определим понятие базиса решения задачи.

Определение 3. Подматрицу $B = \left\|b_{i^* j^*}\right\|$, где $i^* = \overline{1,V}$; $i < r$; $j^* = \overline{1,F}$; $j < l$, определенную на исходной матрице $W = \left\|w_{rl}\right\|$, назовем исходным базисом решения задачи.

В качестве базиса используются ключевые информационные элементы и используемые ими процедуры обработки данных. Если ключевые информационные элементы не определены, то элементы (строки и столбцы матрицы B) задаются исходя из технологических требований проекта.

Определение 4. Величины

$$d_{ri} = \sum_{l=1}^{L}(w_{rl} - b_{il}) \qquad (2.10)$$

и,

$$\overline{d}_{lj} = \sum_{r=1}^{R}(w_{rl} - b_{rj}) \qquad (2.11)$$

назовём расстоянием между строками (столбцами) не вошедшими в базис и строками (столбцами), которые вошли в базис.

Вычисленные значения величин d_{ri} и $\overline{d}_{lj}$ составляют матрицу $D = \|d_{ri}\|$ и $\overline{D} = \|\overline{d}_{lj}\|$. Минимальные значения элементов d_{ri} и $\overline{d}_{lj}$ определяют оптимальное однозначное отображение процедур в модули и информационных элементов в массивы базы данных.

В процессе отображения с матрицами D и $\overline{D}$ тесно связаны матрицы состояний соответственно $\pi = \|\pi_{vl}\|$ и $Q = \|q_{rf}\|$, указывающие текущее состояние исходной матрицы после операции отображения, которые заключаются в логическом сложении небазисных строк (столбцов) с базисными.

Алгоритм состоит из ряда итераций. Поэтому определим его как алгоритм итеративных отображений (АИО). Алгоритм состоит из следующих операций:

1) ввод матрицы W. Выделение базиса в матрице W;

2) вычислить величины d_{ri} и составить матрицу $D = \|d_{ri}\|$. Зафиксировать состояние матрицы $\pi = \|\pi_{vl}\|$;

3) k-я итерация:

3.1. в матрице найти k- й минимальный элемент $d_{r^*i^*}$. При наличии нескольких минимальных элементов, среди них выберем такой элемент, для которого значение суммы элементов по строке максимально. Таким образом, выбирая минимальный элемент, избавляемся от большого число связей. Если элементов такого свойства несколько, то среди этих минимальных элементов выберем элемент расположенный первым от начало отсчета строк;

3.2. определить элементы $x_{r^*i^*} = 1$ матрицы X. Проверить ограничения на число процедур в составе каждого модуля. Если оно неудовлетворительно, то перейти к 3.3, иначе к 3.1;

3.3. исключить из рассмотрения элемент $d_{r^*i^*}$. Установить;

3.4. вычислить состояние матрицы;

3.5. исключить из рассмотрения строку с номером i^*. Пересчитать величины d_{ri} относительно i^* столбца с учетом нового состояния π;

3.6. проверить условие: все ли процедуры распределены? Если нет, то перейти к следующей итерации, приняв $k = k + 1$. Иначе переход к 4.

4) запомнить содержание матриц X и π;

5) вычислить $\overline{d}_{lj}$ относительно π и составить матрицу $\overline{D} = \|\overline{d}_{lj}\|$;

6) n-я итерация:

6.1. в матрице D найти n-й минимальный элемент. При наличии нескольких минимальных элементов, среди них выберем такой элемент, для которого значение суммы по строкам минимально. Если элементов такого свойства несколько, то среди этих минимальных элементов выберем элемент расположенный первым от начало отсчета строк;

6.2. определить элементы $y_{l^*j^*} = 1$ матрицы Y. Проверить ограничения на число информационных элементов в логическом массиве. Если оно неудовлетворительно, то перейти к 6.3;

6.3. исключить из рассмотрения элемент $\overline{d}_{l^*j^*}$. Установить;

6.4. вычислить состояние матрицы π;

6.5. исключить из рассмотрения строку с номером l^*. Пересчитать величины $\overline{d}_{lj}$ относительно j^* столбца с учетом нового состояния;

6.6. проверить условие: все ли информационные элементы распределены? Если нет, то перейти к следующей итерации, приняв $n = n + 1$. Иначе переход к 7;

7) вывод решения задачи: матриц X, Y, $\pi = Z$ и значение целевой функции $\sigma(Z)$.

Сравним сложность для получения решения с использованием данного алгоритма, оцениваемую общим количеством шагов, с известным методом «ветвей и границ» для решения дискретных задач комбинаторного типа.

Необходимые количество шагов в процессе решения задачи с использованием алгоритма итеративных отображений равно:

$$\beta_u = \sum_{k=0}^{R-V} V(R-V-k) + \sum_{n=0}^{L-F} F(L-F-n), \quad (2.12)$$

где k, n - количество итераций в процессе формирования соответствующих решений X и Y. Число шагов с использованием метода «ветвей и границ»:

$$\beta_{rl} = \sum_{r=1}^{R} r! + \sum_{l=1}^{L} l!. \quad (2.13)$$

Сравнение соотношений (2.10), (2.11) показывает эффективность и полиномиальную сложность разработанных алгоритмов для решения поставленных задач большой размерности, в отличие от метода «ветвей и границ».

Выводы

Рассмотрена общая постановка блочно-симметричной задачи дискретного программирования, её особенности и свойства. Представлен общий подход решения задач данного класса.

Сформулирована постановка задачи декомпозиции функциональных задач обработки данных и исходных документов в виде блочно-симметричной задачи дискретного программирования.

Указанная задача решается на этапе технического проектирования систем обработки данных. С использованием результата этой задачи поставлена задача проектирования модульных блок-схем обработки данных, которая обеспечивает разработку прикладного программного обеспечения и базы данных.

С целью решения данной задачи предложен алгоритм и произведена оценка вычислительной сложности алгоритма. Сравнив результаты, можно сделать вывод, что данный алгоритм обеспечивает корректную работу с выполнением всех потавленых условий.

Список литературы

1. Казиев Г.З., Набиева Г.С., Шукатаев А. Программная реализация многокритериальных блочно-симметричных задач дискретного программирования. // Научный журнал Министерства образования и науки «Поиск». №4. – М: Мир, 2006. – С. 191 – 196.

2. Емеличев В.А., Перепелица В.А. Полные задачи многокритериальной дискретной оптимизации. – 1998. – Т. 131. - № 3. – С. 501 – 504.

3. Сергиенко И.В., Перепелица В.А. К проблеме нахождения множеств альтернатив в дискретных многокритериальных задачах // Кибернетика. – 1987. - № 5. – С.85 – 93.

4. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: 2-е изд., испр. и доп. – М.: ФИЗМАТЛИТ, 2007. – 304 с.