

УДК 004.8.023

ЭВРИСТИЧЕСКИЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ СЕГМЕНТАЦИИ ПРОГРАММ

THE HEURISTIC APPROACH TO THE DECISION OF THE PROBLEM OF SEGMENTATION OF PROGRAMS

Рамазанов Е. Т.

Е. Т. Ramazanov

Международная академия бизнеса, кафедра информационных технологий

В данной статье рассмотрен алгоритм нахождения первого приближения для решения задачи сегментации программ. Описана математическая модель задачи сегментации и для этой постановки задачи предлагается процедура, существенной частью которой является, метод перестановок элементов строк для бинарной матрицы. Даётся обоснование используемых эвристических соображений, которые лежат в основе приведенного алгоритма.

Одной из интересных задач в области управления вычислительным процессом в системах с виртуальной памятью является задача сегментации программ (задача НР-полна).

Напомним, что под задачей сегментации обычно принято понимать задачу разбиения последовательной программы на взаимозависимые по управлению и информационной части (блоки, секции, сегменты и. т. д) в соответствии с той или иной целью[2.]. Задача сегментации определяется как задача разбиение программы на части по страницам виртуальной памяти. Проблема заключается в том, что при размещении программы по сегментам виртуальной памяти каждый элемент программы получает свой адрес. Операционная система выделяет каждой программе некоторый участок основной памяти. Причем объем выделенной памяти меньше чем сама программа. По мере выполнения программы в памяти находятся копии страниц программ. Обмен между вспомогательной памятью и основной осуществляется целыми страницами, во время обмена центральный процессор переключается на выполнение команд другого сегмента, если во время выполнения программы происходит ссылка на сегмент программы, которая отсутствует в основной памяти, то происходит страницочное прерывание. Выполнение программы прерывается. Из-за программ, в которые генерируют избыточное число страницочных прерываний. При снижается производительность самой вычислительной системы. Как известно существуют различные подходы разрешение проблемы избыточной генерации числа страницочных отказов. Алгоритмы основанные на понятии рабочего множества сегментов предложенные Питером Деннингом обеспечивают уменьшение числа страницочных отказов. Если имеется структура программы т.е. программа состоит из некоторого числа блоков, то за счет переструктурирования программы можно улучшить поведение и локальность самой программы. В связи с этим интересен математический подход к решению задачи сегментации в классической графовой постановке. Идея графового подхода заключается в представлении программы в виде полного взвешенного графа. Вершинам этого графа соответствует блоки программ, ребрам - передачи управления или информации между блоками программы. Вес вершины определяет размер блока программы, а вес ребра число передач управления или информации между блоками. Задача состоит в разбиении вершин на множества так чтобы суммарный вес вершин попавших в одно множество не превосходил веса множества т.е. страницы. А

суммарный вес ребер между разбитыми множествами вершин был бы минимален. На основе графового определение задачи сегментации была построена модель задачи сегментации дающую возможность использовать методы кластерного анализа. Принципиальную возможность применение кластерного алгоритма обосновано в работе [2]. Приведем формальную постановку задачи.

Пусть исполняемая программа разбита на m блоков. Размеры заданных блоков соответственно равны l_1, l_2, \dots, l_m . Память выделенной операционной системой имеет n страниц. Допустим, что число блоков программы больше числа страниц основной памяти $m \geq n$. Причем, выполняется неравенство:

$$l_1 + l_2 + \dots + l_m \leq s_1 + s_2 + \dots + s_n \quad (1)$$

где s_i размер i страницы. Страницы имеют одинаковый размер, типичный размер, как правило, степень двойки. Также пусть задана матрица передач управления или информации между блоками $R = [r_{ij}]$, r_{ij} - суммарное число передач между блоками b_i и b_j . Очевидно, что матрица передач R симметричная матрица т. е. $r_{ij} = r_{ji}$. Страницы обладают свойством:

$$S_i \cap S_j = \emptyset \text{ для всех } \forall i, j, i \neq j \quad (2)$$

Из свойства (2) следует, что при разбиении блоков по страницам каждый блок программы может принадлежать только одной странице. При этом суммарная длина всех блоков программы принадлежащих странице не превышает длины самой страницы. Решением задачи является матрица разбиения $\pi = [\pi_{ij}]$:

$$\pi = [\pi_{ij}], i = 1, 2, \dots, m, j = 1, 2, \dots, n, \pi_{ij} \in \{0, 1\} \quad (3)$$

Элементы матрицы $\|\pi_{ij}\|_{m \times n}$: $\pi_{ij} = 1$ тогда, когда алгоритм A решения задачи сегментации Z ($A(Z) = \|\pi_{ij}\|_{m \times n}$) отнесет i блок программы к странице j , $\pi_{ij} = 0$ в противном случае.

Тогда основные ограничения модели можно представить следующим образом:

$$\sum_{i=1}^m l_i \times \pi_{ik} \leq V_k, \quad k = 1, 2, \dots, n \quad (4)$$

$$\sum_{j=1}^n \pi_{ij} = 1, \quad i = 1, 2, \dots, m \quad (5)$$

Как известно, задача сегментации является задачей дискретной оптимизации. Часто алгоритмы решения задач данного класса представляют суперпозицию двух алгоритмов, т.е. $A = A^1 \circ A^2$ где алгоритм A^1 определяет первое приближение решения, а алгоритм A^2 является алгоритмом дискретной оптимизации. Удачное определение первого приближения влияет на эффективность поиска и точность решения алгоритма дискретной оптимизации.

Рассмотрим алгоритм первого приближения. Алгоритм основан на эвристических предположениях. Результат алгоритма A^1 это допустимое разбиение или опорное решение задачи Z , которое доставляет функционалу задачи $f = f(\pi)$ меньшее значение, чем уже имеющиеся. Предложенный эвристический алгоритм A^1 получения 1-го приближения решает свою задачу в два этапа. На первом этапе формируется допустимое (опорное) решение задачи на основе заданных размеров блоков программы и страниц, на втором этапе полученное допустимое решение улучшается, если это возможно.

Рассмотрим реализацию первого этапа вычислений. Генерация допустимого решения сводится к формированию бинарной матрицы разбиение $\pi = [\pi_{ij}]$. Для формирования строк матрицы используется процедура перестановок элементов строк. На каждой итерации в результате работы процедуры перестановок в каждой i строке матрицы элементы циклически сдвигаются вправо на определенное число позиций. Данный способ генерации перестановки с применением циклических сдвигов элементов строки существующей перестановки состоит в следующем: начиная с первой строки бинарной матрицы $\pi = [\pi_{ij}]$ последовательно сдвигается по циклу на одну позицию все элементы данной строки. Полученная перестановка запоминается, затем заменяется соответствующая строка новой. Затем после каждой работы процедуры перестановки элементов строк формируется новое разбиение. Так поступаем со всеми строками матрицы. Если сдвиг по циклу n элементов строки возвращает нас к ранее порожденной перестановке, то сдвигаем первые $n-1$ элемент строки. Если и этот шаг возвращает нас к ранее порожденной перестановке, то сдвигаем по циклу первые $n-2$ элемента строки и так далее. Затем, получив новую перестановку, снова сдвигаем по циклу все n элементов строки. Вычисления прекращаются, если все попытки построить новую перестановку безуспешны. Таким образом, сгенерирував бинарную матрицу, алгоритм на следующем шаге проверяет матрицу на допустимость. Процедура проверки допустимости матрицы использует m логических переменных. Если хотя бы одна из m переменных при проверке столбцов матрицы принимает значение *False*, то данная матрица не может быть допустимой и алгоритм возвращается к процедуре перестановок элементов строк матрицы. Если при проверке столбцов матрицы все m переменных принимают значение *True*, то матрица является допустимой и алгоритм выполняет следующий этап вычислений. Полученная матрица на данном этапе становится матрицей разбиение, строкам матрицы соответствуют блоки b_i , $i = 1, 2, \dots, m$ программы, а столбцам страницы S_j , $j = 1, 2, \dots, n$. Причем значение элемента матрицы $\pi_{ij} = 1$ определяет предикату « i блок принадлежит j странице». На втором этапе процедура использует в качестве входных данных матрицу передач управления между блоками программы.

В результате приведем ниже псевдокод алгоритма начального приближения. Определим формально множество Q допустимых (опорных) решений.

Вход: бинарная матрица π (в общем случае $\pi \notin Q$).

1. {формируем бинарную матрицу} *for* $i \leftarrow 1$ to m *do begin* Процедура перестановки элементов i строки матрицы π ; *end*;
2. {проверяем полученную новую матрицу на допустимость.} *for* $j \leftarrow 1$ to n *do begin for* $i \leftarrow 1$ to m *do* $\sum_{i=1}^m l_i \times \pi_{ij}$;
3. *if* $\sum_{i=1}^m l_i \times \pi_{ij} \leq V_j$ *then* $p_j = true$;
4. $\eta \leftarrow p_1 \& p_2 \& \dots \& p_n$;
5. *if* $\eta = true$ *then* $\pi \in Q$, *go to L*;
6. *else* $\pi \notin Q$ {возвращаемся к шаг 1};
7. *L*: {вычисляем функционал} $\forall i, k \Rightarrow$ *if* $b_i, b_k \in S_j$ *then* $\epsilon_{ik} \leftarrow 0$;
8. *else* $\epsilon_{ik} \leftarrow r_{ik}$;

9. *for* $i \leftarrow 1$ *to* m *do* *for* $j \leftarrow 1$ *to* m *do begin* $f \leftarrow \sum_{i=1}^m \sum_{j=1}^m \epsilon_j$;
10. *Сохраняем полученный результат* π_1, f_1 ;
11. *{улучшаем первое приближение}* *While* $f > f_1$ *do begin* *{формируем новое опорное решение* π, f }
12. *if* $f < f_1$ *then begin* *Return* $\leftarrow \pi$ *end;*

Выход: бинарная матрица π , $\pi \in Q : f(\pi) < f(\pi_1) \forall \pi_1 \in Q$.

1. Журавлев Ю.И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов // Проблемы кибернетики. Вып.2. 1978. С. 35-42.
2. Дюсембаев А.Е. Математические модели сегментации программ. -М.: Физматлит, 2001. 207с.

В данной статье рассмотрен алгоритм первого приближения для решения задачи сегментации программ. Описана математическая модель задачи сегментации и приведены формулы вычисления для процедур алгоритма, также в статье приведена описание метода перестановок элементов строки для бинарной матрицы. Приведены некоторые эвристические утверждения, связанные с построением алгоритма.

Мақалада программаларды сегменттеу есебінің алғашқы жуық шешімін құру алгоритмі қарастырылған. Есептің койылымының математикалық моделі көлтірілген. Алгоритмді құрайтын процедураларын есептейтін формулаалар қарастырылып, алгоритмге қатысты кейбір эвристикалық түжырымдардың дәлелдемелері жазылған. Оған қоса мақалада бинарлы матрицаға арналған матрица жолдарының элементтерін алмастыру әдісі қарастырылған.

This article describes the first approximation algorithm for solving the problem of segmentation programs. Describes the mathematical model of the segmentation, and a formula for calculating the procedure for the algorithm, as in article describes a method of permutations of a string to a binary matrix. There are some heuristic allegations relating to the construction of the algorithm.