СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

Р11 - 10159

Д.Д.Арнаудов, М.Д.Иванчев

МЕТОД МИНИМИЗАЦИИ ЗАТРАТ ОПЕРАТИВНОЙ ПАМЯТИ ЭВМ ПРИ РАБОТЕ ОБСЛУЖИВАЮЩИХ ПРОГРАММ ИПС

1976

P11 - 10159

Д.Д.Арнаудов, М.Д.Иванчев

МЕТОД МИНИМИЗАЦИИ ЗАТРАТ ОПЕРАТИВНОЙ ПАМЯТИ ЭВМ ПРИ РАБОТЕ ОБСЛУЖИВАЮЩИХ ПРОГРАММ ИПС

Арнаудов Д.Д., Иванчев М.Д. Р11 - 10159

Метод минимизации затрат оперативной памяти ЭВМ при работе обслуживающих программ ИПС

Рассматривается комбинаторная нелинейная задача, связанная с минимизацией затрат оперативной памяти ЭВМ, работающей в мультипрограммном режиме. Она возникает при решении нескольких задач, требующих в разные дискретные моменты времени различные объемы оперативной памяти. Задача особенно актуальна для вспомогательных обслуживающих программ больших ИПС.

Описана математическая модель задачи, алгоритм типа ветвей и границ для ее решения, приведено краткое описание программы и результаты машинного эксперимента.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований

Дубна 1976

Arnaudov D.D., Ivanchev M.D. P11 - 10159

A Method of Minimizing the Operative Memory of the Computer during the Work of the Service Programs of the IRS

A nonlinear problem, connected with the minimization of the operative memory in a multiprogram regime is solved. It is of great importance when several programs simultaneously are being run in the operative memory of the computer. A mathematical model of the problem, the branch and bound type is built, and the results of the machine experiment on the CDC-6500 are described.

Communication of the Joint Institute for Nuclear Research

Dubna 1976

Теоретико-множественная модель и основные характеристики информационно-поисковой системы подробно описаны в работах [1,2]. Там же приведены методы построения алгоритмов и программ основных модулей системы. Однако при работе любой ИПС часто выполняются и некоторые добавочные работы, которые связаны с накоплением статистики о запросах потребителей, обработкой статистической информации о частоте использования дескрипторов основного информационного фонда, с определением величины свободной списковой памяти и т.п. Все эти работы связаны с решением задач статистического характера и играют важную роль при подготовке и сборе информации для оптимизации работы информационно-поисковой системы. Однако они выполняются в большинстве случаев вне рабочего цикла информационной системы и совместно с другими программами, использующими данную ЭВМ. Поэтому особенно актуальным является вопрос о минимизации оперативной памяти, занимаемой этими программами.

I. Постановка задачи и ее математические модели

Пример. На одной из ЭВМ, работающей в мультипрограммном режиме, надо решить несколько задач, каждая из которых решается при помощи групп программ, требующих в разные дискретные моменты времени (минуты, секунды) разные (но известные) объемы опе-

ративной памяти. Если решение задачи началось, то желательно, чтобы оно не прерывалось. Наша цель - найти такие начальные моменты пуска отдельных задач, при которых в определенные дискретные единицы времени не превышался бы объем памяти ЭВМ, а главное, чтобы максимально используемая всеми задачами память была минимальной.

В /3/ указан ряд других прикладных примеров, которые попадают в рассматриваемый класс задач.

Математическая модель этого класса может быть описана следующим образом.

Даны целые числа: s_i, $i = 1,2,...,n$, где $1 \le s_i \le m$, и числа $a_{ij} \ge 0$, где $i = 1,2,..., n$; $j = 1,2,...., s_i$. Определяем функции

$$y_i(x_i, t) = \begin{cases} 0, & \text{если} \quad t < x_i; \\ a_{i,t-x_i}, & \text{если} \quad x_i \le t < x_i + s_i; \\ 0, & \text{если} \quad t \ge x_i + s_i, \end{cases} \quad (1)$$

где $t \in N = \{1,2,..., m\}$;

$$x_i \in S_i = \{1,2,..., m - s_i\}.$$

При этих условиях надо найти минимум функции

$$L(X) = L(x_1, x_2, ..., x_n) = \max_{t \in N} \sum_{i=1}^{n} y_i(x_i, t). \quad (2)$$

В этой математической модели m - число дискретных единиц времени в рассматриваемом (плановом) периоде, n - число обслуживаемых групп программ

(задач), s_i - количество дискретных единиц времени обслуживания данного предприятия, a_{ij} - потребность каждой группы программ в каждом дискретном периоде времени. Переменной x_i задается начальный момент запуска i-ой группы программ. Очевидно, что сформулированная так задача эквивалентна следующей.

Пусть $S_i (i=1,2,...,n)$ - конечные подмножества m-мерного пространства с нормой $(||.||)\max(b_i)$, где b_i $(i = 1,2,..., m)$ - компоненты одного m-мерного вектора. Пусть конечные подмножества евклидова пространства имеют следующие свойства: для каждой пары векторов $X_i = (x_{i1}, x_{i2}, ..., x_{im})$ и $X_j = (x_{j1}, x_{j2}, ..., x_{jm})$, принадлежащих S_k, существует целое число p такое, что $x_{i,s} = x_{j,s+p}$. Тогда задача состоит в нахождении минимума функции

$$L(X) = ||X_1 + X_2 + ... + X_n|| \quad \text{для} \quad X_i \in S_i. \quad (3)$$

Эти формулировки не дают никаких преимуществ для ее решения из-за особого вида целевой функции $L(X)$, являющейся нелинейной. Ниже будет дана другая формулировка той же самой задачи, из которой лучше видна как сама математическая проблема, так и основная идея алгоритма.

Дана матрица, имеющая n строк и m столбцов, в которой i-ая строка $(i = 1,2,..., n)$ содержит набор упорядоченных чисел a_{ij}, $j = 1,2,..., s_i$ (j - номер числа в наборе), а остальные элементы в строке равны нулю. Цель задачи - расположить эти наборы таким образом, чтобы максимальная сумма по столбцам матрицы была минимальной. Если обозначим через x_i номер столбца, с которого начинается набор в i-ой строке, то надо найти те значения $x_i \in S_i$, для которых минимизируется самая большая сумма по столбцам (при этом $x_i + s_i \le m$, , где $i = 1,2,..., n$). Требование, чтобы x_i $(i = 1,2,..., n)$ были целыми и положительными, следует из смысла x_i (номера столбца). Иными словами, в каждой строке (с номером i) матрицы набор a_{ij},

$j = 1,2,..., s_i$ можно "двигать" в горизонтальном направлении, и начало его может находиться в любом столбце, если весь набор умещается в i-ой строке матрицы. Все остальные элементы в строке, кроме элементов набора, равны нулю, при этом элементы набора не должны разделяться. Наша цель состоит в обнаружении такого "движения" наборов во всех строках матрицы, чтобы максимальная сумма по столбцам полученной матрицы была минимальной.

В /3/ показано, что последняя формулировка является эквивалентной задаче (1) - (2).

II. Алгоритмы решения задачи

Для решения задачи предлагается алгоритм типа ветвей и границ с односторонним ветвлением /5/. В основе описываемого ниже алгоритма лежат работы /3,4/. Усовершенствование его состоит в улучшении оценок отдельных вершин дерева решений. При описании алгоритма будем пользоваться следующими обозначениями:

n - число групп программ (задач);

m - число дискретных единиц времени в плановом периоде;

N - множество $N = \{1,2,...; m\}$; Z ;

s_i , $i = 1,2,..., n$, n - целые числа, для которых $1 \le s_i \le m$. Они представляют собой количество дискретных единиц времени обслуживания i-ой группы программ;

a_{ij} , $i = 1,2,..., n$, $j = 1,2,..., s_i$ - неотрицательные числа, представляющие собой потребности памяти i-ой группы программ в j-ый дискретный период времени;

S_i , $i = 1,2,..., n$ - множества, каждое из которых определяется как $S_i = \{1,2,..., m - s_i\}$;

x_i, $i = 1,2,..., n$ - начальные моменты удовлетворения потребностей i-ой группы программ;

$y_i(x_i,t)$, $i = 1,2,..., n$ - функции, определенные следующим образом:

$$y_i(x_i,t) = \begin{cases} 0, & \text{если } t < x_i ; \\ a_{i;\, t-x_i}, & \text{если } x \le t < x_i + s_i ; \\ 0, & \text{если } t \ge x_i + s_i , \end{cases}$$

где $t \in N$, а $x_i \in S_i$;

k - уровень в дереве решений;

R_i , $i = 1,2,..., m$ - числа, представляющие собой сумму по столбцам матрицы A_{nxm} для этих строк (групп программ), которые уже фиксированы ($x_i = \text{const}$);

p_k , $k = 1,2,..., n$ - числа, представляющие собой нижние границы и часть целевой функции для добавления к ней еще нефиксированных $(n+1-k)$ строк;

F - верхняя граница целевой функции (рекорд).

Надо найти минимум функции:

$$L(X) = L(x_1, x_2, ..., x_n) = \max_{t \in N} \sum_{i=1}^{n} y_i(x_i,t).$$

При этих обозначениях предлагаемый алгоритм может быть описан следующим образом:

1. Перенумеровать множества S_i $(i = 1,2,...,n)$ в порядке убывания числа элементов в них. Положим $R_i = 0$ $(i = 1,2,...,n)$, $k = 0$ и зададим первоначальное значение верхней границы целевой функции (рекорд) F.

2. Перейти к следующему уровню разветвления $(k = k + 1)$. Если $k > n$, переходим к 8, если нет, то вычисляем p_k и переходим к 3.

3. Если $S_k = \Phi$, переходим к 6, в противном случае полагаем $x_k = z$, где $z \in S_k$, множество S_k редуцируется $S_k = S_k \setminus z$, где знаком "$\setminus$" обозначена операция вычитания в теоретико-множественном смысле.

4. Вычисляем $R = \max_{i \in N} (R_i + y_k(z,i))$. Если $R \ge F$, то переходим к 3. Если $P + P_k \ge F$ - тоже переходим к 3, если нет, то переходим к 5.

5. Вычисляем $R_i = R_i + y_k(z,i)$, где $i = 1,2,..., m$, и переходим ко 2.

6. Восстанавливаем первоначальное содержание множества S_k. Возвращаемся к предыдущему уровню разветвления $(k = k - 1)$.

7. Если $k=0$, переходим к 11, если нет, то переходим к 3.

8. Найдено возможное решение с целевой функцией $L(X)=R$.

Решение запоминается, заменяется значение рекорда $F = R$ и происходит возврат к предыдущему уровню ветвления $(k = k - 1)$.

9. Вычисляем $R_i = R_i - Y_n(z,n)$ для $i = z, z+1, \ldots, z+s_n$, где s_n – число элементов множества S_n.

10. Если $F = \max\limits_{i \in N} R_i$, переходим к 6, если нет, то к 3.

11. Если число найденных возможных решений равно нулю, переходим к 12, в противном случае перейдем к 13.

12. При первоначальном значении рекорда F задача не имеет решения. Переходим к 14.

13. Последнее из найденных возможных решений является оптимальным.

14. Конец.

Дадим краткие пояснения к некоторым основным пунктам алгоритма.

В пункте 1 перенумерация множества $S_i (i = 1,2,\ldots,n)$ связана с выбором переменной, подлежащей фиксации. Удачный выбор такой переменной существенно влияет на время решения задачи. Пока неизвестен критерий выбора переменной, подлежащей фиксации, в пункте 1 используется эвристический критерий, обеспечивающий минимальное число промежуточных вершин (в дереве, образованном всеми возможными решениями). Подходящий выбор значения рекорда F существенно влияет на время решения задачи. При отсутствии априорной информации о рекорде F, для верхней границы задается достаточно большое значение.

В пункте 3 наилучшие результаты дает процедура, при которой x_k принимает такое значение $z \in S_k$, при котором с наибольшей вероятностью достигается оптимальное решение. Подходящим критерием для этого

является выбор такого значения z для x_k, при котором дисперсия чисел $R_i (i=1,2,\ldots,m)$ минимальна. Реализация такого критерия связана с большим объемом вычислений, поэтому в рассмотренном алгоритме выбран один эвристический критерий, требующий минимального количества вычислений и обеспечивающий хорошую дисперсию.

В пункте 8 лучше каждое найденное возможное решение не запоминать, а сразу выдавать на печать. Достаточно запомнить лишь значение целевой функции и заменить рекорд, благодаря чему отбрасываются "бесперспективные" ветви (если на определенном уровне значение целевой функции уже превышает рекорд).

В пунктах 9 и 10, если после нахождения возможного решения и замены рекорда (см. пункт 8) при вычитании $Y_n(z,n)$ из R_i (см. пункт 9) величина $R = \max\limits_{i \in N} R_i$ равняется F, то для каждого другого значения $X_n \in S_n$ и $X_n \neq z$ значение целевой функции будет не меньше, чем при $X_n = z$. Поэтому исследования для $X_n \neq z$ беспредметны.

III. Структура программы и результаты проведенного машинного эксперимента на ЭВМ CDC-6400

По описанному алгоритму была создана программа на языке ФОРТРАН- IV. Исходные данные можно вводить с любого вводного устройства. Если, предположим, ввод осуществляется с перфокарт, то вводимые данные будут состоять из следующих групп перфокарт:

1) одна карта для размерности задачи – число групп программ n и число дискретных единиц времени m. Карта имеет FORMAT (2I5);

2) карты (одна или несколько) для чисел $S_i (i = 1,2,\ldots,n)$ с FORMAT (8I10). Следовательно, число карт равно $[\frac{n}{8} - \epsilon] + 1$;

3) карты для коэффициентов a_{ij}. Каждая строка начинается с новой карты с FORMAT(8F10.0) и, следовательно, число карт для i -ой строки равно $[\frac{s_i}{8} - \epsilon] + 1$;

4) одна карта для задания начального значения рекорда с FORMAT(F10.0).

5) одна карта для конца данных - стандартная.

Результаты решения задачи могут быть введены на любое внешнее устройство, указанное потребителем. Все они сопровождаются пояснительным текстом. Необходимая оперативная память равна $nxm + 5n + m$, что позволяет решать задачи сравнительно больших размеров. Для реализации алгоритма характерно сравнительно быстрое нахождение допустимого решения, которое впоследствии заменяется найденными лучшими решениями. Время для доказательства оптимальности решения значительно превышает время нахождения оптимального решения. Вообще говоря, все замечания, относящиеся к программным реализациям метода ветвей и границ, действительны и для этой реализации.

С помощью созданной программы было решено много примеров как прикладных задач, так и полученных с помощью генератора псевдослучайных чисел. В таблице 1 даны результаты машинного эксперимента решений прикладных задач, связанных с работой сервисных программ ИПС ОИЯИ. Соответствующие столбцы в таблице означают:

1 - номер по порядку;

2 - размеры задачи (n x m);

3 - время решения (в с.);

4 - установлена ли оптимальность;

5 - число найденных возможных решений;

6 - процент исследованных возможных решений.

ТАБЛИЦА 1

1	2	3	4	5	6
1	4x20	0,218	да	4	14,25
2	5x20	0,703	да	9	7,86
3	6x20	0,547	да	7	0,18
4	8x20	1,564	да	13	0,0053
5	4x30	0,522	да	7	3,79
6	5x30	0,815	да	10	1,03
7	7x30	2,918	да	20	0,008
8	4x50	1,249	да	10	4,17
9	6x50	6,980	да	14	0,033
10	8x50	35,889	да	19	0,015

ТАБЛИЦА 2

1	2	3	4	5	6
1	15x10	10	52.08	10	11
2	15x15	10	100,68	10	14
3	20x15	10	258,40	10	17
4	25x15	5	387,21	5	20
5	30x15	5	525,32	5	26
6	35x15	5	719,46	5	21
7	20x30	10	578,41	10	24
8	25x31	10	632,74	9	25
9	30x52	5	928,43	4	19
10	40x52	3	1021,32	3	28
11	50x150	3	2130,32	3	31
12	100x100	3	3256,34	1	26

В таблице 2 даны результаты машинного эксперимента генерируемых задач. Соответствующие столбцы таблицы означают:

1 - номер по порядку;

2 - размеры задачи (nxm);

3 - число решенных задач;

4 - среднее время для решения одной задачи (в с);

5 - число задач, в которых установлена оптимальность;

6 - среднее число найденных возможных решений.

ЛИТЕРАТУРА

1. Д.Д.Арнаудов. "Автоматизирани информационно-търсещи системи". изд. Техника, София, 1975.
2. Д.Д.Арнаудов. "Обработка списочной информации при программировании основных алгоритмов ИПС на КОБОЛе". Сообщение ОИЯИ, 10-7587, Дубна, 1973.
3. М.Д.Иванчев, Н.И.Янев, Й.Г.Митев. "Реализация на метода разклоняване и ограничаване за една комбинаторна оптимизационна задача", "Системи и управление", София, №2, 1974.
4. М.Д.Иванчев. "Алгоритми и подготовката им за машинна реализация". Изд. "Народная просвета", София, 1974.
5. A.M.Geofrrion, R.E.Marsten. "Integer programming algorithms: a frammework and state of the art survey", Mahag. Sci., 18, 9, 1972.

Рукопись поступила в издательский отдел
6 октября 1976 года.