

УДК 681.32

В.В. Кирюхин, проф., канд. физ.-мат. наук, Е.В. Жук

Севастопольский национальный технический университет

Студгородок, г. Севастополь, Украина, 99053

kvt@sevgtu.sebastopol.ua

ОПТИМИЗАЦИЯ СТРУКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ОПЕРАТИВНОЙ ОБРАБОТКИ МЕТОДОМ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Рассматриваются прямая и обратная задачи синтеза вычислительной системы оперативной обработки (СОО) на заданном множестве устройств. Прямая задача синтеза заключается в минимизации времени реакции при заданной общей стоимости системы. Обратная задача состоит в минимизации стоимости СОО при заданном времени реакции. Причём время реакции представляется более точным, нежели в известном решении [1], выражением. Задачи формулируются в терминах целочисленных переменных и решаются, в отличие от [1], непосредственно в этих переменных методами динамического программирования.

Постановка проблемы

Прямая задача синтеза вычислительной системы оперативной обработки (СОО) заключается [1] в нахождении вектора $(K_1, \dots, K_n)$, минимизирующего время $u(K_1, \dots, K_n)$ реакции системы

$$u(K_1, \dots, K_n) = \sum_{i=1}^n \alpha_i u_i(K_i) \quad (1)$$

при ограничениях

$$\sum_{i=1}^n K_i c_i \leq C^*, \quad (2)$$

$$K_i = 1, 2, \dots, i = 1, \dots, n, \quad (3)$$

где $u_i(K_i)$ - время реакции i -го агрегата системы при числе K_i каналов в нём; n - число типов агрегатов в СОО; α_i - коэффициент передачи агрегата i -го типа; c_i - стоимость одного канала в агрегате i -го типа; C^* - заданная общая стоимость СОО; $i = 1, \dots, n$.

Обратная задача синтеза СОО представляется в следующем виде [1].

Необходимо найти вектор $(K_1, \dots, K_n)$, минимизирующий стоимость $C(K_1, \dots, K_n)$ системы

$$C(K_1, \dots, K_n) = \sum_{i=1}^n K_i c_i, \quad (4)$$

при ограничениях

$$\sum_{i=1}^n \alpha_i u_i(K_i) U^*, \quad (5)$$

$$K_i = 1, 2, \dots, i = 1, \dots, n, \quad (6)$$

где U^* - максимально допустимое время реакции СОО.

С целью получения аналитических (формульных) результатов решения задач в [1] в качестве функций $u_i(K_i)$ используются функции вида

$$u_i(K_i) = \frac{K_i \vartheta_i}{K_i - \lambda_i \vartheta_i}, \quad i=1, \dots, n, \quad (7)$$

где ϑ_i - среднее время обработки задания агрегатом i -го типа; λ_i - интенсивность потока заданий, обрабатываемых этим агрегатом, $R_i = \lambda_i \vartheta_i$. Функции вида (7) является следствием допущения, что агрегат в соответствующей сетевой модели СОО представляется группой одноканальных систем массового обслуживания (СМО).

Более точной моделью СОО является модель, в которой совокупность K_i однотипных агрегатов представлена K_i - канальной СМО. Тогда

$$u(K_i) = \vartheta_i + \frac{\lambda_i^{K_i} \vartheta_i^{K_i+1}}{(K_i - 1)!(K_i - \lambda_i \vartheta_i)^2} \cdot \frac{1}{\sum_{s=0}^{K_i} \frac{\lambda_i^s \vartheta_i^s}{s!} + \frac{\lambda_i^{K_i+1} \vartheta_i^{K_i+1}}{K_i! (K_i - \lambda_i \vartheta_i)}} , \quad i=1, \dots, n, \quad (8)$$

Этот случай и рассматривается в данной статье. Поскольку функции вида (8) не являются аналитическими, получение даже приближённого решения задач в аналитическом виде (как это сделано в [1]), невозможно. Задачи необходимо решать непосредственно в целых числах.

В дальнейшем вектор $K = (K_1, \dots, K_n)$, удовлетворяющий ограничениям и минимизирующий функцию цели (1) либо (4), будет именоваться оптимальной структурой СОО.

Алгоритмы синтеза

Решение прямой задачи (1)-(3) здесь основано на использовании метода динамического программирования [2]. Он включает две основные фазы. В первой фазе получаем n таблиц, каждая из которых содержит время реакции и начальное поведение для определённого числа устройств и значений ресурсов. На втором этапе, используя множество таблиц поведения, находим оптимальное поведение для каждого способа; n -я таблица, полученная в первой фазе, содержит оптимальное начальное распределение для задачи, включающей n способов. Она, в свою очередь, определяет начальное состояние для задач, включающих $n-1$ способ; $n-1$ -я таблица определяет вто-

рое оптимальное решение. Таким образом, все выходные таблицы поведения обрабатываются в обратном порядке.

Ниже следует описание алгоритма решения прямой задачи синтеза.

Начало

1. Вводим параметры:

n - число типов агрегатов в СОО, α_i - коэффициент передачи агрегата i -го типа, ϑ_i - среднее время обработки задания агрегатом i -го типа, λ_0 - суммарная интенсивность потока заявок от пользователей СОО, c_i - стоимость одного агрегата i -го типа, C^* - выделенная общая стоимость СОО; $i=1, \dots, n$.

2. Вычисляем интенсивность потока заявок к агрегатам i -го типа:

$$\lambda_i = \alpha_i \cdot \lambda_0, \quad i = 1, \dots, n.$$

3. Вычисляем величину

$$\Delta C = C^* - \sum_{i=1}^n \lambda_i \vartheta_i [c_i.$$

Здесь $\sum_{i=1}^n \lambda_i \vartheta_i [c_i$ - средства, которые необходимо затратить на СОО для обеспечения стационарного режима. Если $\Delta C \leq 0$ то задача синтеза СОО не имеет решения; переходим на конец.

4. Полагаем $i=1$.

5. Полагаем $z=0$

6. Определяем $g = \left\lceil \frac{z}{c_i} \right\rceil$.

7. Минимизируем $u_i(K_i) + u_{i-1}(z - K_i c_i)$, где $K_i \leq g$; Запоминаем $u_i(z)$ и минимизирующее $K_i(z)$.

8. Если $z < \Delta C$, то полагаем $z = z + 1$, и переходим к пункту 9.

9. Если $i < n$, то полагаем $i = i + 1$ и переходим к пункту 5.

10. На этом пути используется множество $\{K_i(z)\}$ таблиц. Начинаем с задачи, включающей n способов и начальное количество ресурсов ΔC : $i = n$, $z = \Delta C$.

11. Определяем поведение, соответствующее вышеуказанной ситуации, это повлечёт за собой просмотр $K_i(z)$; соответствующее поведение сохраняем в a : $a = K_i(z)$

12. Сохраняем номер способа и соответствующее поведение: i и a .

13. Ресурсы уменьшаем на уже распределённое количество a : $z = z - a c_i$.

14. Количество рассматриваемых способов уменьшаем на 1: $i = i - 1$.

15. Если все способы рассмотрены, т.е. $i = 0$, то переходим к пункту 16, иначе - к пункту 11.
16. Полагаем $K_i = K_i + \lambda_i \vartheta_i$ $i = 1, \dots, n$ и вычисляем стоимость $C_{\min}$ по формуле (4).
17. Выводим результаты: $C_{\min}, (K_1, \dots, K_n)$.

Конец.

Решение обратной задачи (4)-(6) по структуре алгоритма аналогично прямой и основано на использовании таблиц распределения времени реакции СОО и таблиц поведения, получаемых при решении прямой задачи. Описание соответствующего алгоритма ввиду этой аналогичности здесь опускается.

Оба алгоритма были запрограммированы и реализованы на ПЭВМ класса Pentium.

Примеры решения и их анализ

В качестве входных были взяты типовые структуры и данные СОО на базе современных рабочих станций [3]. Первый агрегат - процессор и оперативная память; второй агрегат - накопитель на жёстком магнитном диске и селекторный канал (контроллер); третий агрегат - накопитель на гибком магнитном диске и селекторный канал (контроллер). Входные данные и соответствующие решения приведены в таблице 1.

Таблица 1 - Входные данные и решения прямой задачи

№	i	a_i	ϑ_i, c	$c_i, y.e.$	Точное	Приближённое
1	1	60	0,0002	200	1	1
2	2	50	0,01	70	1	1
3	3	5	0,7	10	1	1
4	$C^*, y.e.$				280	280
5	10				0,1	0,1
6	$U_{\min}, c$				2,166237	2,166237
7	$C, y.e.$				280	280
8	1	60	0,0002	200	1	1
9	2	50	0,01	70	2	2
10	3	5	0,7	10	4	4
11	$C^*, y.e.$				380	380
12	10				0,1	0,1
13	$U_{\min}, c$				1,912333	1,975612
14	$C, y.e.$				380	380
15	1	60	0,0002	200	1	1

16	2	50	0,01	70	3	2
17	3	5	0,7	10	4	11
18	$C^*, \text{y.e.}$				450	450
19	10				0,1	0,1
20	$U_{\min}, \text{c}$				1,912023	1,942883
21	$C, \text{y.e.}$				450	450

Столбец i содержит тип агрегата, в столбце a_i приведены коэффициенты передачи агрегатов i -го типа, столбец v_i содержит среднее время обработки задания в агрегате i -го типа, в столбце c_i - стоимость i -го агрегата. В столбце «Точное» показано точное решение задачи синтеза СОО, в столбце «Приближённое» - приближённое решение. Под последним имеется в виду решение задачи с функциями вида (1) методами, изложенными в [1]. Первые три строки содержат количество агрегатов каждого типа, строка C^* - выделенные средства, строка 10 - интенсивность поступления заданий, строка $U_{\min}$ - минимальное время реакции и строка C - затраченные средства.

При точном и приближённом решении задачи синтеза СОО в ряде тестовых примеров была получена одинаковая структура СОО. Время реакции при точном решении меньше времени реакции, полученном при приближённом, но разница между ними незначительна, в среднем точный метод позволяет получить решение на 3...7% лучше, чем приближённый. Таким образом, для решения задачи синтеза СОО на заданном множестве устройств можно использовать приближённый метод, но если необходимо получить решение с высокой точностью, то рекомендуется использовать точный метод.

Примеры решения обратной задачи приведены в таблице 2.

Назначение столбцов этой таблицы аналогично таблице 1. Строка U^* - заданное время реакции СОО, строка 10 - интенсивность поступления заданий, строка $C_{\min}$ - минимальная стоимость СОО и строка U - время реакции СОО.

Таблица 2 - Входные данные и решения обратной задачи

№	i	a_i	$\vartheta_{i, c}$	$c_i, \text{y.e.}$	Точное	Приближённое
1	1	60	0,0002	200	1	1
2	2	50	0,01	70	1	1
3	3	5	0,7	10	1	1
4	U^*, c				2,5	2,5
5	$10, \text{c}^{-1}$				0,1	0,1
6	$C_{\min}, \text{y.e.}$				280	280
7	U, c				2,15484	2,166237
8	1	60	0,0002	200	1	1
9	2	50	0,01	70	1	1

10	3	5	0,7	10	2	4
11	U^*, c				2,0	2,0
12	$10, c^{-1}$				0,1	0,1
13	$C_{min}, y.e.$				290	310
14	U, c				1,945207	1,989107
15	1	60	0,0002	200	1	1
16	2	50	0,01	70	1	2
17	3	5	0,7	10	3	8
18	U^*, c				2,14	2,14
19	$10, c^{-1}$				0,4	0,4
20	$C_{min}, y.e.$				300	420
21	U, c				2,04884	2,073

При точном и приближённом решении задачи синтеза СОО в ряде тестовых примеров была получена одинаковая структура СОО. Стоимость системы, полученная при точном решении, меньше или совпадает со стоимостью, полученной при приближённом решении. Но чем ближе заданное время реакции к минимально возможному, тем значительно отличаются решения при точном и приближённом методах. Таким образом, в этих случаях рекомендуется использовать точное решение.

В заключении отметим, что сформулированные задачи синтеза можно обобщить на случай ограничений не только по стоимости или времени реакции, но и по другим характеристикам (например, по массе, потребляемой энергии и т.п.). Однако, такие задачи – предмет дальнейших исследований.

Библиографический список

1. Майоров С.А. Основы теории вычислительных систем / С.А. Майоров. — М.: Высш. шк., 1978. — 408 с.
2. Беллман Р. Прикладные задачи динамического программирования/ Р. Беллман, С. Дрейфус. — М.: Наука, 1965. — 457с.
3. Бабенков М. Тестирование высокопроизводительных рабочих станций российских производителей/ М. Бабенков, А. Шобанов// Компьютер пресс. — 2002. — №6. — С.95 – 108.

Поступила в редакцию 23.12.2002 г.