В Ы Ч И С Л И Т Е Л Ь Н Ы Е   С И С Т Е М Ы
1972 год   Выпуск 51

УДК 681.142.2

СТОХАСТИЧЕСКИЕ АЛГОРИТМЫ ФУНКЦИОНИРОВАНИЯ ОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В.Г. Хорошевский, Л.А. Седухина

Рассматриваются два алгоритма распределения конечного множества задач различной сложности[*] по элементарным машинам (ЭМ) однородной вычислительной системы (ОВС) [1] и один алгоритм подготовки задач к распределению (алгоритм формирования пакетов задач). Приводятся результаты статистической обработки экспериментов по моделированию алгоритмов. В приложении приводятся АЛГОЛ-программы алгоритмов.

§ 1. Алгоритм формирования пакетов задач

Постановка задачи

Имеется ОВС из n ЭМ и множество $\mathcal{I}=\{I_i\}, i=\overline{1,N}$, независимых задач. Задача $I_i \in \mathcal{I}$ имеет ранг r_i и время решения t_i (то есть для ее решения требуется ровно r_i ЭМ и $t_i \geqslant 0$ единиц времени [2]).

Требуется множество всех задач ранга r, $1 \leqslant r \leqslant n$, раз-

[*] Сложность задачи, например [1] характеризуют количеством операций, которые нужно выполнить при ее решении.

бить на подмножества так,чтобы в каждое из них входили задачи, суммарное время решения которых было бы близко к заданной величине θ.

Основные операции алгоритма

Построим подмножества $\mathcal{J}^z \subseteq \mathcal{J}$, $z = 1,2,\dots,n$. В подмножество $\mathcal{J}^z$ входят все задачи $I_i^z \in \mathcal{J}$, $i = 1,2,\dots,a$, которые имеют ранг $z_i = z$. Время решения этих задач

$$T^z = \sum_{i=1}^{a} t_i^z .$$

Пусть $\hat{\mathcal{J}} = \{I_1^z, I_2^z, \dots, I_i^z, \dots, \mathcal{J}_a^z\}$ – некоторая последовательность, членами которой являются элементы $I_i^z \in \mathcal{J}^z$. Подмножество $\mathcal{J}_j \subseteq \hat{\mathcal{J}}$ включает в себя κ_j задач, причем

$$\mathcal{J}_j = \bigcup_{i=K_j+1}^{K_j+\kappa_j} I_i^z , \qquad K_j = \sum_{s=0}^{j-1} \kappa_s ,$$

$$j = 1,2,\dots,L_z,\ L_z = \left]\frac{T^z}{\theta}\right[$$

(L_z –ближайшее к $\frac{T^z}{\theta}$ целое число, $L_z \geq \frac{T^z}{\theta}$), $\kappa_0 = 0$. Каждое подмножество $\mathcal{J}_j \subset \hat{\mathcal{J}}$, $j = 1,2,\dots,L_z$, будем называть укрупненной задачей $\mathcal{J}_j$ ранга z. Время решения таких задач будет равно

$$T_j = \sum_{i=K_j+1}^{K_j+\kappa_j} t_i^z .$$

Будем считать, что пакет укрупненных задач ранга z сформирован, если выполняется условие:

$$|\hat{T} - \theta| = O(\hat{T}), \quad \hat{T} = \max_{\mathcal{J}_j \subset \mathcal{J}} \{T_j\}, \qquad \text{(I)}$$

где $O(\hat{T})$ – бесконечно малая более высокого порядка малости, чем $\hat{T}$. Это условие можно удовлетворить, если за единицу времени взять величину θ такую,что $\theta \gg t_i^z$ для каждой задачи $I_i^z \subset \mathcal{J}^z$. Подмножества $\mathcal{J}_j \subset \hat{\mathcal{J}}$ выбираются следующим образом.

Пусть построены подмножества $\mathcal{J}_s \subset \hat{\mathcal{J}}$, $s = 1,2,\dots,j-1$; тогда подмножество

$$\mathcal{J}_j := \begin{cases} \hat{\mathcal{J}}_j', \ \kappa_j := \kappa_j', \ \text{если} \quad (\theta_б)_j = \dfrac{T^j - (\theta_н)_j}{L_z - j}, \\ \hat{\mathcal{J}}_j' \cup I_{K_j+\kappa_j'+1}, \ \kappa_j := \kappa_j'+1 \ \text{– в противном случае,} \end{cases}$$

где $\hat{\mathcal{J}}_j'$ – часть последовательности $\hat{\mathcal{J}}$ такая, что

$$\hat{\mathcal{J}}_j' = \{I_{K_j+1}, I_{K_j+2}, \dots, I_{K_j+\kappa_j'}\}, \quad T^j = \sum_{i=K_j+1}^{L_z} t_i^z ,$$

а для величин $(\theta_б)_j$ и $(\theta_н)_j$ справедливо:

$$\sum_{i=K_j+1}^{K_j+\kappa_j'+1} t_i^z = (\theta_б)_j > \theta, \quad \sum_{i=K_j+1}^{K_j+\kappa_j'} t_i^z = (\theta_н)_j \leq \theta .$$

Требуется найти последовательность $\hat{\mathcal{J}}^*$ задач, которая обеспечит выполнение условия (I).

Такая последовательность отыскивается с помощью метода цепей Монте-Карло [3,4] по следующей схеме.

Последовательность $\hat{\mathcal{J}}$ принимаем за базовую. Затем рассматриваются перестановки на расстоянии не больше κ, $\kappa \leq a$, от базовой. В качестве расстояния между двумя последовательностями $\hat{\mathcal{J}}$ и $\hat{\mathcal{J}}'$ принимают число индексов в $\hat{\mathcal{J}}'$, которые не следуют за теми же индексами, что и в базовой $\hat{\mathcal{J}}$. Метод получения последовательности $\hat{\mathcal{J}}'$ с расстоянием не больше κ основан на псевдослучайных числах.

Сначала получаем (κ –1) независимых переменных x_i в непрерывном интервале $(0, a)$:

$$x_i = a\,\xi_i , \quad i = 1,2,\dots,\kappa-1 ,$$

где ξ_i – переменные в интервале (0,1),полученные генератором псевдослучайных чисел. Если $0 = x_0 \leq x_1 \leq \dots \leq x_i \leq \dots \leq x_\kappa = a$, то x_i делят последовательность $\hat{\mathcal{J}}$ на κ частей $\hat{\mathcal{J}}^s \subseteq \hat{\mathcal{J}}$, $s = 1,2,\dots,\kappa$, содержащих такие задачи, номера которых являются целыми числами между $(x_{i-1}, x_i]$. Некоторые части могут оказаться пустыми. Случайная перестановка этих частей дает новую последовательность $\hat{\mathcal{J}}'$ с расстоянием не больше κ от базовой.

Если $\hat{T}'$ новой последовательности $\hat{\mathcal{J}}'$ меньше $\hat{T}$, то есть наилучшим образом удовлетворяет условию (I), то $\hat{\mathcal{J}}'$ берется в качестве базовой.

Если сделано d попыток моделирования последовательностей с расстоянием k от базовой без изменения базовой, то рассматриваются последовательности с расстоянием $k/2$ и т.д. до тех пор, пока не будут смоделированы последовательности с расстоянием 2.

Операторная схема алгоритма

Введем операторы:

F_1 : формирование подмножеств $\mathcal{J}^z \subseteq \mathcal{J}$, z = I,2,..., n ;

$\mathcal{L}_2 : z := 1$;

$\mathcal{L}_3 : L_z := \left] \frac{T^z}{\Theta} \right[$;

F_4 : формирование базовой последовательности $\hat{\mathcal{J}}$;

F_5 : формирование подмножеств $\mathcal{J}_\ell \subset \hat{\mathcal{J}}$, ℓ = I,2,..., L_z ;

A_6 : вычисление $\hat{T}$;

$\mathcal{L}_7 : g := 1$;

Φ_8 : формирование последовательности $0 = x_0 \leq x_1 \leq \ldots \leq x_k = Q$;

F_9 : формирование частей $\hat{\mathcal{J}}^s \subseteq \hat{\mathcal{J}}$, s = I,2,..., k ;

Φ_{10} : формирование новой последовательности $\hat{\mathcal{J}}'$;

F_{11} : формирование подмножества $\mathcal{J}'_\ell \subset \hat{\mathcal{J}}$, ℓ = I,2,..., L_z ;

A_{12} : вычисление $\hat{T}'$;

P_{13} : $P\{\hat{T}' < \hat{T}\}$ – проверка выполнения условия $\{\hat{T}' < T\}$,

$P = 0 \longmapsto \mathcal{L}_{17}$ – переход на $\mathcal{L}_{17}$ при невыполнении условия;

$\mathcal{L}_{14} : \hat{T} := \hat{T}'$;

F_{15} : формирование базовой последовательности $\hat{\mathcal{J}} := \hat{\mathcal{J}}'$;

$\Pi_{16} : \longmapsto \mathcal{L}_7$;

$\mathcal{L}_{17} : g := g + 1$;

$P_{18} : P\{g > d\}$, $P = 0 \longmapsto \Phi_8$;

$\mathcal{L}_{19} : k := \frac{1}{2} k$;

$P_{20} : P\{k < 2\}$, $P = 0 \longmapsto \mathcal{L}_7$;

$F_{21} : \mathcal{J}_\ell := \hat{\mathcal{J}}_\ell \in \mathcal{J}^z$, $\ell = 1, 2, \ldots, L_z$;

$\mathcal{L}_{22} : z := z + 1$;

$P_{23} : P\{z > n\}$, $P = 0 \longmapsto \mathcal{L}_3$; $Я_{24}$: конец.

Операторная схема имеет вид:

$$F_1 \mathcal{L}_2 \; {}^{2,23}\mathcal{L}_3 F_4 F_5 A_6 \; {}^{6,16,20}\mathcal{L}_7 \; {}^{7,18}\Phi_8 F_9 \Phi_{10} F_{11} A_{12} P_{13 \uparrow 17} ;$$

$$\mathcal{L}_{14} F_{15} \Pi_{16}^{7} \; {}^{13}\mathcal{L}_{17} P_{18 \uparrow 8} \mathcal{L}_{19} P_{20 \uparrow 7} F_{21} \mathcal{L}_{22} P_{23 \uparrow 3} Я_{24} .$$

Результаты моделирования алгоритма

Пусть N = 300, n = I0, k = 8, d = 5, t , z –псевдослучайные числа, распределенные в интервалах (0,5), (0,I0], соответственно. Условная единица времени Θ = 20.

Результаты по моделированию одного пакета укрупненных задач приведены в табл. I.

Таблица I

Ранг z задач $\mathcal{J}_j$, входящих в пакет	j	Последовательность задач из $\hat{\mathcal{J}}$ в укрупненной задаче $\mathcal{J}_j$.	Время T_j решения $\mathcal{J}_j$
8	I	$I_{76}, I_{90}, I_{97}, I_{102}, I_{103}, I_{108}, I_{131}, I_{147}, I_{150}$	20,67
	2	$I_{156}, I_{170}, I_{171}, I_{175}, I_{183}, I_{195}$	I9,59
	3	$I_{197}, I_{204}, I_{213}, I_{225}, I_{227}, I_{229}, I_{12}$	20,95
	4	$I_{23}, I_{38}, I_{41}, I_{51}, I_{70}, I_{40}, I_{235}, I_{236}, I_{243}$	20,05
	5	$I_{245}, I_{246}, I_{252}, I_{262}, I_{280}, I_{290}, I_{294}, I_{295}, I_{300}$	20,I6

Из табл.I видно, что для ранга z =8 значение $\overset{1}{T} = \max_j \{T_j\}$ = 20,95, и распределение удовлетворяет условию(I).

Эффективность алгоритма можно оценить по результатам статистической обработки экспериментов. Качество формирования укрупненных задач будем характеризовать показателем $\tau_j = \frac{1}{\Theta} |T_j - \Theta|$, j = I,2,..., L , где L –общее число укрупненных задач, образованных из $\mathcal{J}$ ($L = \sum_{z=1}^{n} L_z$, L_z – число укрупненных задач в пакете ранга z = I,2,..., n). Оценки математического ожи –

дания и дисперсии величины τ соответственно равны:

$$M\tau=\frac{1}{L}\sum_{j=1}^{L}\tau_j=6\cdot10^{-2};\qquad D\tau=\frac{1}{L-1}\sum_{j=1}^{L}(\tau_j-M\tau)^2=8\cdot10^{-2}.$$

Другой подход к решению поставленной задачи содержится в [5].

§ 2. Алгоритмы распределения набора сложных задач

Постановка задачи

На ОВС, состоящую из n ЭМ, поступает совокупность $\mathcal{J}=\{I_i\}$, $i=\overline{1,N}$, задач I_i, каждая из которых представлена параллельным алгоритмом и имеет ранг $r_i\in\{1,2,\ldots,m\}$, $m\le n$, время решения t_i, штраф за задержку решения на единицу времени c_i.

Требуется построить алгоритмы распределения, при помощи которых можно было бы получить субоптимальные значения целевых функций, характеризующих эффективность ОВС.

Алгоритм I

Пусть штраф за задержку решения любой задачи $I_i\in\mathcal{J}$, $i=\overline{1,N}$, равен нулю. Исходный набор задач на основе алгоритма формирования пакетов задач преобразуется в набор укрупненных задач $\mathcal{J}'=\{I'_i\}$, $i=\overline{1,L}$ (см. §1). Каждая задача $I'_i\in\mathcal{J}'$, $i=\overline{1,L}$, характеризуется временем решения $t'_i=\Theta$ и имеет ранг $1\le r'_i\le m$.

В основу алгоритма положены следующие операции.

Из элементов множества $\mathcal{J}'$ строится базовая последовательность $\hat{\mathcal{J}}$. Как и для алгоритма формирования пакетов задач, строятся подмножества $\mathcal{J}_j\subset\hat{\mathcal{J}}$. Однако здесь в подмножество $\mathcal{J}_j$ (в случае, когда уже построены подмножества $\mathcal{J}_s\subset\hat{\mathcal{J}}$, $s=1,2,\ldots,j-1$) включается K'_j задач, для которых справедливо:

$$\sum_{i=K_j+1}^{K_j+K'_j} r_i\le n,\qquad \sum_{i=K_j+1}^{K_j+K'_j+1} r_i>n;$$

в последнее подмножество $\mathcal{J}_M\subset\hat{\mathcal{J}}$ включается $K'_M=L-K_{M-1}$ задач.

Подмножество $\mathcal{J}_j\subset\hat{\mathcal{J}}$ реализуется за время Θ на шаге j, число шагов равно M.

Очевидно, что нижние границы числа шагов и времени, кото-

рые требуются для реализации всей совокупности $\mathcal{J}'$ укрупненных задач, соответственно равны

$$M^0=\left]\frac{1}{n}\sum_{i=1}^{L} r_i\right[,\quad \Theta\cdot M^0.$$

Необходимо найти последовательность $\hat{\mathcal{J}}^*$, для которой число шагов M^* минимально или

$$(M^*-M^0)=O(M^*).$$

Следовательно, здесь в качестве целевой функции может быть взято время реализации задач множества $\mathcal{J}'$ или число шагов M.

Последовательность $\hat{\mathcal{J}}^*$ находится с помощью метода цепей Монте-Карло [3]. Следовательно, операторная схема алгоритма I подобна схеме алгоритма формирования пакетов задач.

Приведем результаты машинного моделирования.

Считаем, что исходный набор задач преобразован и получен набор из $L=100$ задач, каждая из которых имеет время решения Θ. ОВС состоит из 20 ЭМ. Пусть $\kappa=8$, $d=5$, r_i — целые псевдослучайные числа, распределенные в промежутке (0,5]. Частично результаты распределения задач приведены в табл.2.

Таблица 2

Номер шага j	Последовательность задач, решаемая на шаге j	Сумма рангов W_j
1	$I_{32}, I_{33}, I_{34}, I_{35}, I_{36}, I_{37}, I_{38}$	19
2	$I_{39}, I_{40}, I_{41}, I_{42}, I_{43}, I_{20}$	19
3	$I_1, I_2, I_3, I_4, I_5, I_6$	18
4	$I_7, I_8, I_9, I_{10}, I_{11}, I_{12}, I_{13}$	18
5	$I_{29}, I_{30}, I_{31}, I_{44}, I_{45}, I_{46}, I_{47}$	20

Для оценки эффективности алгоритма было исследовано 9 различных наборов задач. Для каждого набора и для каждого шага вычислялись относительные погрешности $w_j=\frac{1}{n}(n-W_j)$. Оценки математического ожидания и дисперсии величины w оказались равными:

$$Mw=10^{-1},\qquad Dw=2\cdot10^{-3}.$$

Алгоритм II

Считаем, что штраф за задержку решения задачи $I_i \in \mathcal{J}$ составляет $c_i \neq 0$ денежных единиц.

Так же, как и в алгоритме I, исходный набор $\mathcal{J}$ преобразуется в новый набор $\mathcal{J}'$.

В качестве целевой функции используется общая сумма штрафов F за задержку решения задач совокупности $\mathcal{J}'=\{I_i'\}$, $i=\overline{1,L}$.

Требуется так распределить задачи набора $\mathcal{J}'$ для решения на системе, чтобы получить минимум целевой функции.

Пусть I_i' представляет собой некоторую последовательность задач

$$\hat{I}_i' = \{I_{i1}, I_{i2}, \ldots, I_{i\ell}, \ldots, I_{ik_i}\}, \tag{2}$$

характеризуемую функцией штрафа

$$\varPhi_i = t_{i1}\sum_{\ell=2}^{k_i} c_{i\ell} + t_{i2}\sum_{\ell=3}^{k_i} c_{i\ell} + \ldots + t_{i,k_i-1}\, c_{k_i}. \tag{3}$$

Доказано [4], что $\varPhi_i$, $i=\overline{1,L}$, принимает минимальное значение, если для последовательности (2) справедливо неравенство:

$$\frac{t_{i1}}{c_{i1}} \leqslant \frac{t_{i2}}{c_{i2}} \leqslant \ldots \leqslant \frac{t_{i\ell}}{c_{i\ell}} \leqslant \ldots \leqslant \frac{t_{ik_i}}{c_{ik_i}}. \tag{4}$$

Пусть (2) есть последовательность задач, упорядоченная в соответствии с (4).

Далее, если на каждом шаге $i=\overline{1,L}$ решать только укрупненную задачу I_i', принадлежащую последовательности

$$\hat{\mathcal{J}}' = \{I_1', I_2', \ldots, I_i', \ldots, I_L'\}, \tag{5}$$

то функция штрафа будет иметь вид:

$$F = \varPhi_1 + (\theta\sum_{\ell=1}^{k_2} c_{2\ell} + \varPhi_2) + (2\theta\sum_{\ell=1}^{k_3} c_{3\ell} + \varPhi_3) + \ldots + [(i-1)\theta\sum_{\ell=1}^{k_i} c_{i\ell} +$$

$$+ \varPhi_i] + \ldots + (L-1)\theta\sum_{\ell=1}^{k_L} c_{L\ell} + \varPhi_L = \sum_{\ell=1}^{L} \varPhi_\ell +$$

$$+ \theta[0\cdot\alpha_1 + 1\cdot\alpha_2 + \ldots + (i-1)\alpha_i + \ldots + (L-1)\alpha_L], \tag{6}$$

где величины $\varPhi_\ell$, $\ell=\overline{1,L}$, вычисляются по формуле (3), а

$$\alpha_i = \sum_{\ell=1}^{k_i} c_{i\ell}.$$

Функция штрафа (6) будет принимать субминимальное значение, если для последовательности (5) выполняется следующее неравенство:

$$\alpha_1 \geqslant \alpha_2 \geqslant \ldots \geqslant \alpha_i \geqslant \ldots \geqslant \alpha_L. \tag{7}$$

Пусть произведено упорядочение так, что для (5) справедливо (7).

В основу алгоритма II положен следующий принцип: на шаге z назначаются укрупненные задачи из последовательности (5), не назначавшиеся на предшествующих шагах $z'<z$ и сумма рангов которых не более n, то есть $W_z \leqslant n$.

Заметим, что формирование наборов задач из (5) для каждого шага может быть осуществлено также при помощи метода цепей Монте-Карло.

Введем операторы:

F_1 : формирование последователей $\hat{I}_i'$, $i=1,2,\ldots,L$;

A_2 : вычисление $\varPhi_i$, $i=1,2,\ldots,L$;

A_3 : вычисление α_i, $i=1,2,\ldots,L$;

F_4 : формирование последовательности $\hat{\mathcal{J}}'$;

$\mathcal{L}_5$: $z:=1$; $\mathcal{L}_6 : T_j := F_j := 0$, $j=1,2,\ldots,n$;

F_7 : формирование для шага z подмножества $\mathcal{J}_z=\{I_{z1}',\ldots,I_{zs}',\ldots,I_{zp}'\}$ из задач последовательности $\hat{\mathcal{J}}'$;

H_8 : назначение $\mathcal{J}_z$ для решения на шаге z ;

$\mathcal{L}_9$: $\tau_{zs}:=0$, $s=1,2,\ldots,p$ (τ_{zs} – ранг $I_{zs}' \in \mathcal{J}_z$);

A_{10} : вычисление для ЭМ j штрафа на шаге z за задержку решения назначенной на неё задачи I_{zs}' :

$$f_j = (z-1)\theta\alpha_{zs} + \varPhi_{zs}, \quad s=\overline{1,p}, \quad j=\overline{1,n};$$

A_{11} : вычисление $F_j := F_j + f_j$, $j=\overline{1,n}$;

: вычисление $T_j := T_j + T$, $j=\overline{1,n}$;

A_{13} : вычисление $W = \sum_{j=1}^{L} \tau_j$; $P_{14} : P\{W \neq 0\}$, $P=0 \mapsto \text{Я}_{17}$;

$\mathcal{L}_{15}$: $z := z+1$; $\text{П}_{16} : \mapsto F_7$; Я_{17} : конец.

Операторная схема имеет следующий вид:

$$F_1\ A_2\ A_3\ F_4\ \mathcal{L}_5\ \mathcal{L}_6\ {}^{6,16}F_7\ H_8\ \mathcal{L}_9$$

$$A_{10}\ A_{11}\ A_{12}\ A_{13}\ P_{14\uparrow 17}\ \mathcal{L}_{15}\ \text{П}_{16}^{7}\ \text{Я}_{17}.$$

Приведем результаты моделирования алгоритма II.

Пусть n = I0, L = I00, α_i и Φ_i – псевдослучайные числа, распределенные в промежутках (0,I5) и (0,20) соответственно, i = = I,...,I00. Результаты распределения задач по ЭМ ОВС при помощи алгоритма II и случайным образом приведены в табл. 3, где T_j –время решения, F_j –штраф за задержку решения задач для ЭМ с номером j, j = I,...,I0, ϑ – нижняя грань T_j, $\vartheta = \frac{\theta}{n}\sum_{i=1}^{L} z_i$.

Из табл.3 видно, что время решения T_j одно и то же для алгоритма II и случайного распределения; функция штрафа, найденная по алгоритму II, имеет значение, по крайней мере, меньшее в I,5 раза, чем при случайном назначении.

Таблица 3

	j	I	2	3	4	5	6	7	8	9	I0	j / $T\vee F$	ϑ
T_j	Алгоритм II	295	295	3I0	3I0	320	335	295	280	250	I85	335	267
	Случайное	295	295	3I0	3I0	320	335	295	280	250	I85	335	
F_j	Алгоритм II	35695	35686	37007	37007	37I62	37229	32076	29089	23I8I	I5849	3I998I	
	Случайное	6063I	63074	63304	63488	632I8	62672	54856	49202	38438	30637	579520	

Приведем результаты статистической обработки экспериментов.

Пусть H –число наборов укрупненных задач. Обозначим через

$$\hat{T}_h = \max_j T_j^h, \quad \hat{F}_h = \max_j F_j^h, \quad h=\overline{1,H}, \quad j=\overline{1,n},$$

где T_j^h и F_j^h –соответственно время решения и штраф за задержку решения задач для ЭМ с номером j при применении алгоритма II к набору задач h. Аналогично вычисляется $\tilde{F}_h = \max_j \tilde{F}_j^h$, $h=\overline{1,H}$, $j=\overline{1,n}$, для случайного распределения.

Найдено, что оценки математических ожиданий и дисперсий величин

$$\tau_h = \frac{1}{\vartheta_h}\,|\hat{T}_h - \vartheta_h|, \quad \varphi_h = \frac{1}{\hat{F}_h}\,|\hat{F}_h - \tilde{F}_h|$$

соответственно равны

$$M\tau = 2\cdot 10^{-1}, \qquad D\tau = 2\cdot 10^{-3},$$

$$M\varphi = 5\cdot 10^{-1}, \qquad D\varphi = 14\cdot 10^{-3},$$

где ϑ_h – нижняя грань для T_j^h, $h = \overline{1,H}$, $j = \overline{1,n}$.

Выводы

I. Алгоритм формирования пакетов задач преобразует набор задач с различными временами решения в набор укрупненных задач, которые имеют заданное время решения.

2. Алгоритм I позволяет стохастически оптимально загружать ОВС задачами различных рангов, но имеющими одинаковую ценность.

3. Алгоритм II обеспечивает субминимум функции штрафа при решении множества задач, представленных параллельными программами с различным числом ветвей.

4. Алгоритмы просты в реализации не только на ЭВМ, но и на ОВС и не связаны с большими вычислительными трудностями. Для рассматриваемых примеров время, затрачиваемое на реализацию алгоритмов на ЭВМ "М-220", не превышает минуты.

5. Результаты статистической обработки экспериментов показывают, что алгоритмы достаточно эффективны и могут быть практически использованы при диспетчеризации ОВС.

Литература

I. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, "Наука" СО, I966.

2. ХОРОШЕВСКИЙ В.Г. Об алгоритмах функционирования однородных универсальных вычислительных систем.-"Вычислительные системы", Новосибирск, I970, вып.39, стр.3-I4.

3. PAGE E.S. On Monte-Carlo methods in congestion problems: I. Searching for an optimum in discrete situations.-"Operation Research, The journal of the operation research society of America, Virginia, 1965, vol.13, N 2, p.291-299.

4. ГОЛОСКОКОВА Т.М., ХОРОШЕВСКИЙ В.Г. Алгоритмы функционирования универсальных вычислительных систем в простейших ситуациях.-"Вычислительные системы", Новосибирск, I970, вып. 39, стр.I5-28.

5. МИРЕНКОВ Н.Н. Алгоритмы планирования для диспетчера однородной вычислительной системы. -"Вычислительные системы", Новосибирск, 1970, вып.42, стр.34-46.

Поступила в ред.-изд.отд.
5. IX. 1971 г.

ПРИЛОЖЕНИЕ

Приведем программы алгоритмов, записанные на языке АЛГОЛ-60.

Для программы алгоритма формирования пакетов задач необходимо задать следующие величины: n - число ЭМ; N - числа задач; $u0$, $u1$ - псевдослучайные числа; k -расстояние между последовательностями; d - число попыток; Y - условная единица времени; $t[1:N]$ - массив времени решений задач; $r[1:N]$ - массив рангов.

На печать выдаются значения: l - ранг укрупненной задачи $\mathcal{J}$; L - число задач $\mathcal{J}$ с рангом l; $K[i,j]$ - массив номеров задач $I_i \in \mathcal{J}$, входящих в $\mathcal{J}_j$, $w[1:L]$- массив времени решений $\mathcal{J}_j$, $j=\overline{1,L}$.

```
begin integer  n,N,i,j,p,z,a3,l,b1,L,k,d,e,k1;  real u0,u1,m1,
      Y;  read  (n,N,u0,u1,k,d,Y);  begin iteger array r[1:N],
      R,m[1:n], A[1:n,1:N-n]; real array x,t,t2[1:N], G[1:n];
      read (t,r); for i:=1  step 1 until n do G[i]:=0; p:=1  ;
      j:=1 ; BC: z:=1 ; for i:=1  step 1 until N do  begin  if
      r[i]:= r[j] then begin A[p,z]:= i;   G[p]:= G[p] + t[i];
      t2[i]:= t[i]; t[i]:= 0; z:= z+1  end end;  m[p]:= z- 1  ;
      R[p]:= r[j]; M: j:= j+1 ; if t[j] ≠ 0 then begin p:=p+1;
      go to BC end; if j< N then go to M;  a3:= p; l:= 1 ; D1 :
      l:= R[e]; b1:= m[e]; m1:= G[e]; L:= if m1 /Y =    entier
      (m1/Y) then entier (m1/Y) else entier (m1/Y+1); k1:=k;
begin integer array K,b[1:L,1:b1],h,y[0:k1],a,a1[1:b1],S[1:k1,
      1:b1]; real array t3,T[1:b1],r1[1:k1],x1,v[0:k1],S1[1:k1,
      1:b1],w,w1[1:L]; integer g; realta,W,W1:procedure H (f,
      b1,L,fi,Y,k,v,a2); array a2,f,k; real fi,v;  begin  real
      de1, de2,F1; F1:= fi; for i:=1  step 1 until L do w[i]:=
      0; v:= Y; j:=1 ; for i:=1  step 1 until L-1   do  begin
      p:=1 ; D:  w[i]:= w[i] + f[j]: if w[i]< Y    then  begin
      K[i,p]:= a2[j]; p:= p+1 ; if j< b1  then begin j:= j+1 ;
      go to D end end else begin de1 := w[i]; de2:= (F1- w[i]+
      f[j])/(L-1 ); if de1 ≤ de2  then begin   F1:=F1 - w[i];
      K[i,p]:= a2 [j]; if w[i]> v then v:= w[i]; if j< b1 then
```

```
j:= j+1 end else begin F1:= F1 - w[i]+f[j]; w[i]:=w[i]-
f[j] end end end; p:= b1 -j +1 ; for i:=1 step 1 until
p do begin K[L,i]:= a2 [j]; w[L]:= w[L] + f[j]; j:= j+1
end; if v< w[L] then v:= w[L] end;for i:= 1 step 1 until
L do for j:=1 step 1 until b1 do K[i,j]:= b[i,j] := 0 ;
for j:= 1 step 1 until b1 do begin t3 [j]:= t2 [A[e,j]];
a[j]:= A[e,j] end; H(t3,b1,L,m1,Y,K,W,a);   g:= 1 ;  R1 :
v[0]:= 0; for i:= 1 step 1 until k1 - 1 do begin ta:= u0+
u1 ; u0:= u1 ; if ta>4 then ta:= ta -4 ; u1:=ta; v[i]:=
ta/4  end; v[k1]:= 1 ; for i:= 0 step 1 until k1 do x1[j]
:= b1 x v[i]; p:= 1 ;  A1 : z:= 1 ; for i:= 1 step 1 until
k1 - 1 do begin if x1[p] = x1[i] then begin if p> i then
z:= z+1  end else begin if x1 [p]> x1[i] then  z:= z +1
end end; r1[z]:= x1[p]; if p< k1 - 1 then begin p:=p+ 1 ;
go to A1  end; for z:=1  step 1 until k1 - 1 do   begin
x1[z]:= r1[z]  end; for i:= 0 step 1 until k1  do y[i]:=
entier (x1[i]) for i:= 1 step 1 until k1  do  begin ta:=
u0 + u1 ; u0:= u1 ; if ta> 4 then ta:= ta -4 ; u1 := ta;
v[i]:= ta/4  end;  p:= 1 ; B: z:= 1 ; for i:= 1 step 1 un-
til k 1 do begin if v[p] = v[i] then begin if p > i then
z:= z+1  end else begin if v[p] > v[i] then z:= z+1  end
end; r1[p]:= z; h[z]:= y[p] - y[p-1 ]; if p < k then be-
gin p:= p+1 ; go to B end; for i:=1  step 1 until k do if
y[i] >  y[i - 1] then begin for j:= (y[i-1 ]+1 ) step 1
until y[i] do begin s[r1[i], j-y[i- 1]]:= t3 [j];S[r1[i],
j-y[i- 1 ]]:= a[j] end end end; for j:= 1  step 1 until b1
do a 1 [j]:= T[j]:= 0; j:= 1 ; z:= 1 ;  Q: if h[z]> 0  then
begin for i:= 1  step 1 until h[z] do begin T[j]:=s[z,i];
a 1 [j]:= S[z,i]; j:= j+1  end end; if z< k then begin z:=
z+1 ; go to Q end; H(T,b1,L,m1,Y,b,W1,a, w1); if W1 <  W
then begin W:= W1 ; for i:= 1  step 1 until b1  do  begin
t3 [i]:= T[i]; a[i]:= a1[i] end; g:= 0; for i:= 1 step 1
until L do begin w[i]:= w1 [i]; for j:= 1  step 1 until b1
do begin K[i,j]:= b[i,j]; b[i,j]:= 0 end end end; g:= g+
1 ; if g<  d then go to R1  else begin k:= k/2 : g:= 1 ;
if k≥ 2  then go to R1  end;  print (l,b1,K,W,w) end;e:=
e+1 ; if e ≤ a3  then go to D1  end end
```


Для программы алгоритма I необходимо задать следующие величины: n – число задач; L – число ЭМ в ОВС; k – расстояние между последовательностями; d – число попыток; $u0$, $u1$ – псевдослучайные числа; $r[1:n]$ – массив рангов.

На печать выдаются значения: $a[1:N;1:n]$ – массив номеров задач, назначенных на j-ом шаге, $j=\overline{1,N}$, $w[1:N]$ – массив сумм рангов задач; N – число шагов.

```
begin integer  n, N,L, k, d, i,j,z,p,g,F,W,W1; real uo,u1,ta;
      read (n,L,k,d,uo,u1);
begin real array x1[1:n], x, v[0:k], r2[1:k];  integer array
      r1[1:k], R1, r[1:n], h, y[0:k], s, S[1:k, 1:n-k],  A3,
      A1[1:n]; read (r);
      F:= 0; for i:=1 step 1 until n do F:=F+r[i]; N:=entier
      (F/L+1); begin integer array w,w1[1:N], a,b[1:N, 1:n];
procedure H(f,n,N,fi,Y,a,V,A2,w);
begin integer de1, de2, F1;
      F1:=fi; for i:=1 step 1 until N do w[i]:=0; V:=Y;j:=1;
      for i:=1 step 1 until N-1 do begin p:=1;D: w[i]:=w[i]+
      + f[j]; if w[i]< Y then begin a[i,p]:=A2[j];  p:= p+1;
      if j< n then begin j:=j+1; go to D end end else  begin
      de1:= w[i]; de2:= (F1-w[i]+ f[j])/(N-i); if  de1 ≤ de2
      then begin F1:=F1-w[i]; a[i,p]:= A2[j]; if w[i]>V then
      V:= w[i]; if j< n then j:=j+1 end else begin F1:= F1 -
      w[i]+ f[j]; w[i]:=w[i]- f[j] end end end; p:=n- j + 1;
      for i:=1 step 1 until p do begin a[N,i]:= A2[j];w[N]:=
      w[N]+ f[j]; j:=j+1 end; if V< w[N] then V:= w[N] end;
      for i:= 1 step 1 until n do A3[i]:=i; N(r,n,N,F,L,a,W,
      A3,w); g:=1; R: v[0]:=0; for i:=1 step 1 until k-1  do
      begin ta:=uo+u1; uo:=u1; if ta >4 then ta:=ta-4;u1:=ta;
      v[i]:= ta/4 end; v[k]:= 1; for i:= 0 step 1 until k do
      x[i]:=n x v[i]; p:= 1;  A: z:=1; for i:=1 step 1 until
      k-1  do  begin if x[p]= x[i] then  begin  if p >i then
      z:=z+1 end else begin if x[p]> x[i] then z:= z+1   end
      end; r2[z]:=x[p]; if p< k-1 then begin p:=p+1; go to A
      end; for z:=1 step 1 until k-1 do begin x[z]:= r2[z]
      end; for i:=0 step 1 until k do y[i]:= entier ( x[i]);
```

```
for i:=1 step 1 until k do begin ta:= uo+1; uo:=u1; if
ta >4 then ta:=ta-4; u1:=ta; v[i]:=ta/4 end; p:=1;  B:
z:=1; for i:=1 step 1 until k do begin if  v[p]=  v[i]
then begin if p>i then z:=z+1 end else begin if v[p]>
v[i] then z:=z+1 end end; r1[p]:=z; h[z]:=y[p]- y[p-1];
if p<k then begin p:=p+1; go to B end; for i:=1  step
1 until k do begin if y[i]>y[i-1] then begin for  j:=
(y[i-1]+1)  step 1 until  y[i] do  begin  s[r1[i], j -
y[i-1]]:= r[j]; S[r1[i], j-y[i-1]]:= A3[j] end end end;
for j:=1 step 1 until n do R1[j]:= A1[j]:=0;    j: = 1;
z:=1;  Q: if h[z]> 0 then begin for i:=1 step 1  until
h[z] do begin R1[j]:=s[z,i]; A1[j]:=S[z,i]; j:= j + 1;
end end; if z<k then begin z:=z+1; go to Q end; H(R1,
n,N,F,L,b,W1,A1,w1);  if W1<W then begin W:=W1;  for
i:=1 step 1 until n do begin r[i]:=R1[i]; A3[i]:=A1[i]
end;  g:=0;  for i:= 1 step 1 until N do  begin w[i]:=
w1[i]; for j:=1 step 1 until n do begin a[i,j]:=b[i,j];
b[i,j]:=0 end end end; g:=g+1; if g<d then  go  to  R
else begin k:=k/2; g:=1; if k≥2 then go to R end;
print (N,a,w) end end end
```

Для программы алгоритма II необходимо задать следующие величины: t – время решения задач; n – число ЭМ, L – число задач; $AL[1:L]$ – массив суммарного штрафа; $Fi[1:L]$ – массив значений функции штрафа для каждой задачи, $r[1:L]$ – массив рангов.

На печать выдаются значения: z – номер шага; $K[1:n]$ – массив номеров задач, назначенных на z – ом шаге; $R[0:n]$ – массив рангов назначенных задач; $g=n-W_z$ – число простаивающих машин на z – ом шаге; $F[1:n]$ – массив функций штрафа для ЭМ $j=\overline{1,n}$; $T[1:n]$ – массив времени решения на ЭМj; Q – нижняя грань T_j .

```
begin integer  p,z,i,j,k1 n,L,g,l,s,u; real t,Q,s ; read (L,
      t,n);
begin integer array r[0:L], k[1:n], R[0:n]; real array F1, T,
      F[1:n], Fi, AL[1:L], X,v,y[1:L]; read (r,Fi,AL);  for
      i:=1 step 1 until L do v[i]:=1/AL[i]; p:=1; W: z:=1;
```



```
for i:=1 step 1 until L do begin if v[p] = v[i]  then
begin if p > i then z:=z+1 end else begin if  v[p] >
v[i] then z:=z+1 end end;  x[z]:=AL[p]; x1[z]:= r[p];
y[z]:=Fi[p]; if p < L then begin p:=p+1; go to W end;
for z:=1 step 1 until L do begin AL[z]:=x[z]; Fi[z]:=
y[z]; r[z]:=x1[z] end; s1:=0; for i:=1  step 1 until L
do s1:=s1 +t x r[i]; Q:=s1/L; for i:=1 step 1 until n
do F[i]:=T[i]:=0; z:=1;  A: for i:=1 step 1 until n
do R[i]:=K[i]:=F1[i]:=0;  R[0]:=0; i:=0; k1:=r[0]:=0;
j:=0;  D:  ifr[i]=0 then go to B; if k  ≤ n then begin
l:=k1; j:=j+1; R[j]:=r[i]; K[j]:=i; r[i]:=0;  F1 [j]:=
(z-1) x t x AL[i]+Fi[i]; for u:=1 +R[j-1] step 1 un-
til k1  do begin F[u]:=F[u]+F1[j]; T[u]:=T[u]+t end;go
to B end else go to AB; AB: g:=n-l; if g=o then  go to
BC else begin k1:=k1 -r[i]; go to B end; B: i:=i+1 ;if
i≤ L then begin k1:=k1 +r[i]; go to D end; BC: g:=n-l;
print (z,K,R,g); z:=z+1 ; s:=0; for p:=1  step 1 until
L do s:=s+r[p]; if s≠0 then go to A else print (F,T,Q)
end end
```