ВОПРОСЫ ТЕОРИИ И ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
(Вычислительные системы)
1977 год Выпуск 70

УДК 681.3I:681.3.06

ОБ ОПТИМАЛЬНОМ РАСПРЕДЕЛЕНИИ МАШИН ПО ТЕРМИНАЛАМ РАСПРЕДЕЛЕННОЙ ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

В.И.Жиратков, Э.Г.Хорошевская

Предложен метод стохастически оптимального распределения машин по терминалам, учитывающий спрос на подсистемы различных рангов с терминалов для распределенных однородных вычислительных систем (ОВС) различных конфигураций: кольцевой, линейной, радиальной и произвольной.

Использование потенциальных возможностей распределенных ОВС в большой степени зависит от того, как организовано их функционирование, т.е. как осуществляется процесс решения задач на системе. В качестве показателей эффективности функционирования будем использовать целевые функции, характеризующие потери при эксплуатации систем. Пусть распределенная ОВС коллективного пользования [I] состоит из n элементарных машин (ЭМ), размещенных на T вычислительных центрах (ВЦ), и обслуживает ℓ терминалов, причем $n_i\ (i=1,2,\dots,T)$ – число ЭМ на i-м ВЦ, $\sum_{i=1}^{T} n_i = n$, и i-у ВЦ принадлежит ℓ_i терминалов, $\sum_{i=1}^{T} \ell_i = \ell$.

На терминалы поступают потоки задач различных рангов. Спрос на подсистемы различных рангов подчинен пуассоновскому распределению со средними значениями $\mu_{j\,min}$ и $\mu_{j\,max}$, $j=1,\dots,\ell$, соответственно при минимальном и максимальном требуемых рангах. Стоимость пользования одной ЭМ с j-го терминала – c_j, $j=1,\dots,\ell$. Матрицей $\| c_{ik} \|$ заданы минимальные стоимости каналов между i-м и k-м ВЦ, $i,k=1,2,\dots,T$.

Пусть $x_j\ (j=1,\dots,\ell)$ – ранг подсистемы, используемой j-м терминалом. Обозначим через $S_{ij}\ (i=1,\dots,T,\ j=1,\dots,\ell)$ число ЭМ i-го ВЦ, используемых j-м терминалом, и введем функцию $\delta(S_{ij})$ следующим образом:

$$\delta(S_{ij}) = \begin{cases} 0, & \text{если } S_{ij} = 0, \\ 1, & \text{если } S_{ij} > 0. \end{cases}$$

Тогда потери, связанные с использованием каналов, составят

$$\sum_{j=1}^{\ell} \sum_{i=1}^{T} \delta(S_{ij}) \cdot c_{ik_j}, \quad k_j = 1,\dots,T,$$

где k_j – номер ВЦ, содержащего j-й терминал.

Считаем, что избыточные ЭМ на j-м терминале простаивают, а при необеспечении j-го терминала минимально необходимым числом ЭМ все машины, выделенные этому терминалу, простаивают, и за нерешенную задачу выплачивается штраф, пропорциональный ее сложности.

Задача, которую требуется решить, имеет следующий вид: найти

$$\min \mathcal{Z} = \sum_{j=1}^{\ell} \Big\{ \sum_{i=1}^{T} \delta(S_{ji}) \cdot c_{ik_j} + \varphi_j(x_j, \mu_{j\,min}) (c_j x_j + K \mu_{j\,max}) +$$

$$+ c_j (x_j - \mu_{j\,max}) + c_j [\mu_{j\,max} P_j(x_j - 1, \mu_{j\,max}) - x_j P_j(x_j, \mu_{j\,max})] \Big\}, \quad \text{(I)}$$

при условиях:

$$\left.\begin{aligned} & x_j, S_{ij} \text{ – неотрицательные целые числа}, \\ & \sum_{j=1}^{\ell} S_{ij} = n_i, \quad i = 1,\dots,T, \\ & \sum_{i=1}^{T} S_{ij} - x_j = 0, \ j = 1,\dots,\ell. \end{aligned}\right\} \quad (2)$$

Здесь

$$P_j(x_j, \mu_{j\,max}) = \sum_{y_j = x_j}^{\infty} \frac{y_j}{y_j!} \mu_{j\,max}^{y_j} e^{-\mu_{j\,max}},$$

$$\varphi_j(x_j, \mu_{j\,min}) = \begin{cases} 1, & \text{если } \mu_{j\,min} P_j(x_j - 1, \mu_{j\,min}) - \\ & \quad - x_j P_j(x_j, \mu_{j\,min}) \geq 0{,}5, \\ 0, & \text{если } \mu_{j\,min} P_j(x_j - 1, \mu_{j\,min}) - \\ & \quad - x_j P_j(x_j, \mu_{j\,min}) < 0{,}5, \end{cases}$$

$K\mu_{j max}$ – штраф за нерешение задачи на j-м терминале, считаем, что штраф за нерешение задачи пропорционален ее сложности, K – коэффициент штрафа.

Задачу (I)-(2) стохастического программирования с требованием целочисленности и большим числом ограничений не представляется возможным точно решить известными методами математического программирования [2]. Кроме того, вероятностный характер задачи (I)-(2) не оправдывает затрат на поиск точного оптимума. Тем более, что при диспетчировании более ценным является быстрое и гарантированное получение хотя бы приближенного решения, нежели получение точного решения через большой промежуток времени без полной уверенности в успехе.

Задача (I)-(2) решается методом цепей Монте-Карло [3]. Он состоит в выборе случайным образом распределений, находящихся на некотором расстоянии от базового, пока не будет найдено лучшее; оно затем становится базовым, и моделирование продолжается до тех пор, пока в течение некоторого времени не будет наблюдаться улучшения. Полученное таким образом распределение является хорошим, поэтому случайный поиск продолжается в его окрестности.

Примем за базовое следующее распределение

$$N_1 = \{q_{11}, \dots, q_{1x_1}, 0, q_{21}, \dots, q_{2x_2}, 0, \dots, 0, q_{j1}, \dots, q_{jx_j}, 0, \dots, 0\},$$

где x_j – число ЭМ, выделенных j-у терминалу,

$$x_j = \begin{cases}]\mu_{j min}[, & i > j \geq 1, \\ n - \sum_{z=1}^{i-1} x_z, & j = i, \\ 0, & \ell \geq j > i, \end{cases}$$

i – номер терминала, начиная с которого $\sum_{j=1}^{i} x_j = n$; $q_{jx_j}\ (j = 1, \dots, \ell)$ – номер ВЦ, которым пользуется j-й терминал, $]\alpha[$ – целое число, ближайшее к α.

Таким образом, последовательность N_1 состоит из ℓ числовых интервалов, разделенных $\ell - 1$ нулем. Интервал между двумя соседними нулями соответствует определенному терминалу и несёт в себе информацию о том, сколько ЭМ и с каких ВЦ выделено данному терминалу.

ПРИМЕЧАНИЕ. Следует заметить, что последовательность N_1 получена при условии, что спрос на терминалах превышает возможности

систем, т.е. $\sum_{j=1}^{\ell}]\mu_{j min}[\geq n$. Это наиболее вероятный случай. Если же $\sum_{j=1}^{\ell}]\mu_{j min}[< n$, а $\sum_{j=1}^{\ell}]\mu_{j max}[\geq n$, то x_j примет вид:

$$x_j = \begin{cases}]\mu_{j max}[, & i > j \geq 1, \\ n - \sum_{z=1}^{i-1} x_z, & j = i, \\ 0, & \ell \geq j > i, \end{cases}$$

наконец, если $\sum_{j=1}^{\ell}]\mu_{j max}[< n$, то

$$x_j = \begin{cases}]\mu_{j max}[, & j = 1, \dots, \ell - 1, \\ n - \sum_{z=1}^{\ell-1} x_z, & j = \ell. \end{cases}$$

Далее, для полученного базового распределения N_1 вычисляется целевая функция Z_1. Затем получаем новое распределение на расстоянии h от базового, $n + \ell - 1 > h > 1$. (Расстояние между новым распределением и базовым есть число порядковых номеров новой последовательности, которые не следуют за тем же номером, что и в базовой последовательности.) Всякий раз, когда найдено меньшее значение целевой функции, полученное распределение берется в качестве нового базового. Если сделано g попыток без уменьшения целевой функции, то берутся распределения с расстоянием $h/2$ от базового, и так далее, пока не будут смоделированы распределения с расстоянием 2 [3].

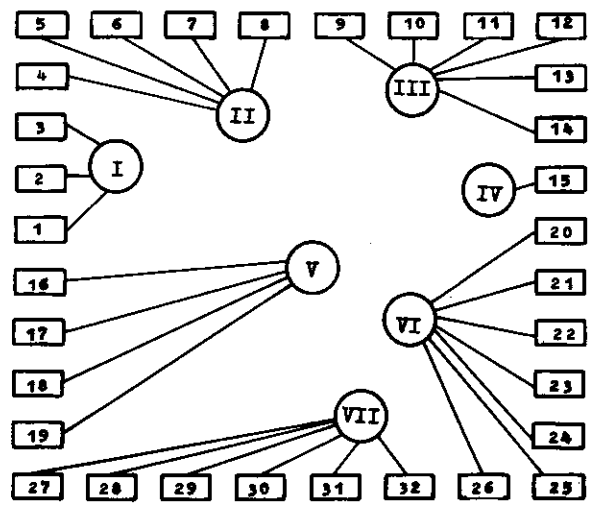

Рис. I

Пользуясь обозначением, приня-

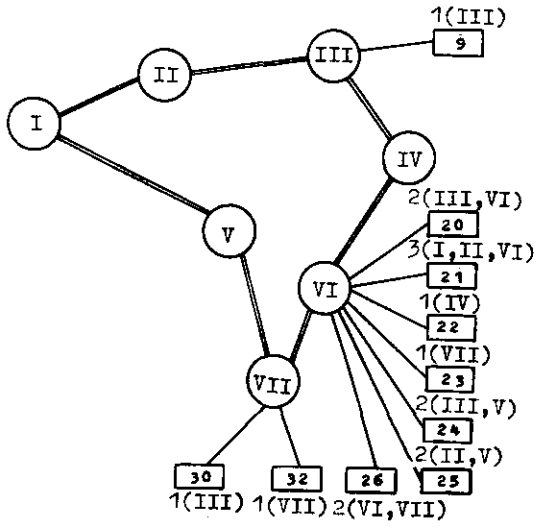

Рис. 2

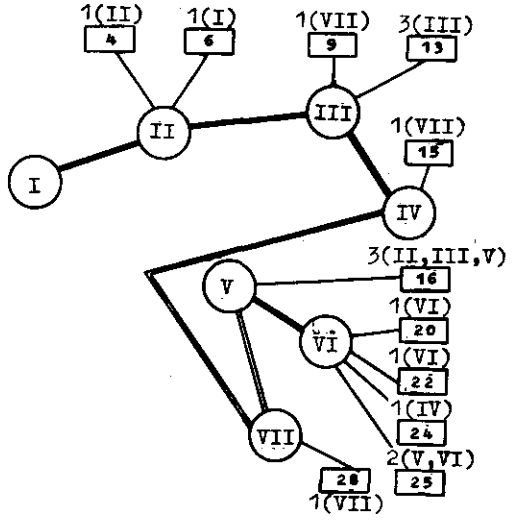

Рис. 3

тым в [4], введем следующие операторы:

F_1 : формирование базового распределения N_1;

A_2 : вычисление z_1 для распределения N_1;

F_3 : $d := 1$;

Φ_4 : формирование нового распределения N_2 как случайной перестановки h частей распределения N_1;

A_5 : вычисление z_2 для распределения N_2;

P_6 : проверка условия $z_2 < z_1$;

A_7 : принятие за базовое распределения N_2;

F_8 : $d := 0$;

K_9 : $d := d + 1$, счетчик числа испытаний, не уменьшающих значения z_1;

P_{10} : проверка условия $d < q$;

F_{11} : $h := h/2$;

P_{12} : проверка условия $h > 1$;

Ψ_{13} : окончание счета; выдача z_1, N_1.

Тогда операторная схема модели рущего алгоритма будет иметь вид:

$$F_1 A_2 {}^{12}F_3 \Phi_4 A_5 P_{6\,9} A_7 F_8 {}^{6}K_9 P_{10}{}^{4} F_{11} P_{12}{}^{13} \Psi_{13}.$$

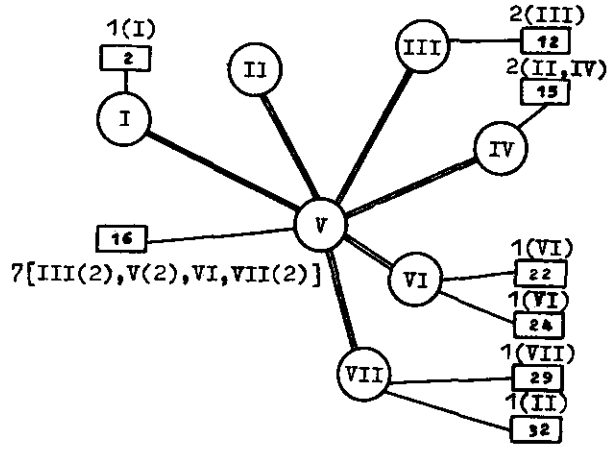

Рис. 4

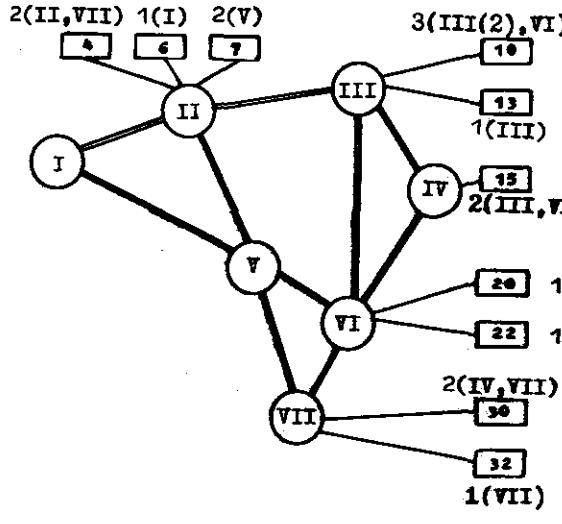

Рис. 5

В приложении приведена программа, написанная на языке АЛГОЛ. Алгоритм реализован на вычислительной машине БЭСМ-6. Ясно, что время счета существенно зависит от величин n, l, q, h. По сравнению с методом Монте-Карло, где время для выбора случайной перестановки возрастает примерно линейно с увеличением $(n+l)$, для метода цепей Монте-Карло оно пропорционально h. В работе [3] показано, что любое распределение может быть получено из базового за конечное число шагов, поэтому, увеличивая q, можно получить минимальное значение целевой функции, естественно, за счет увеличения числа вычислений.

ПРИМЕР. Имеется 7 сосредоточенных ОВС коллек-

тивного пользования (рис.I), организованных в распределенные

I) систему с кольцевой структурой (рис.2),

2) одномерную систему (рис.3),

3) систему с радиальной структурой (рис.4),

4) систему с произвольной структурой (рис.5).

Общее число ЭМ в распределенной системе будет: $n = 16$ ($n_1 = 1$, $n_2 = 2$, $n_3 = 4$, $n_4 = 1$, $n_5 = 2$, $n_6 = 3$, $n_7 = 3$). Она обслуживает 32 терминала ($\ell = 32$, $\ell_1 = 3$, $\ell_2 = 5$, $\ell_3 = 6$, $\ell_4 = 1$, $\ell_5 = 4$, $\ell_6 = 7$, $\ell_7 = 6$). Соответствующие значения c_j, $\mu_{j\,max}$, $\mu_{j\,min}$ приведены в табл.I; элементы c_{ik}, $i, k = 1, \ldots, 7$, матрицы $\|c_{ik}\|$ определяются для каждой конкретной системы как минимальные стоимости каналов между i-м и k-м ВЦ. Матрицу $\|c_{ik}\|$ легко получить из матрицы C, в которой определены стоимости всех возможных каналов:

$$C = \begin{Vmatrix} 0 & 3{,}5 & 6 & 7{,}2 & 4{,}3 & 5{,}2 & 5{,}7 \\ 3{,}5 & 0 & 3 & 5 & 2{,}5 & 4{,}3 & 5{,}5 \\ 6 & 3 & 0 & 3{,}3 & 2{,}7 & 4 & 5{,}5 \\ 7{,}2 & 5 & 3{,}3 & 0 & 3 & 2{,}8 & 3{,}5 \\ 4{,}3 & 2{,}5 & 2{,}7 & 3 & 0 & I{,}8 & 3 \\ 5{,}2 & 4{,}3 & 4 & 2{,}8 & I{,}8 & 0 & I{,}5 \\ 5{,}7 & 5{,}5 & 5{,}5 & 3{,}5 & 3 & I{,}5 & 0 \end{Vmatrix}$$

В табл.I отражен результат назначения ЭМ распределенной ОВС по терминалам при $h = 16$, $q = 500$. Здесь же приведены соответствующие значения целевой функции Z^*, а также $\Delta Z = Z_{баз} - Z^*$, где $Z_{баз}$ вычислено для исходного базового распределения. Время счета одного варианта системы на ЭМ БЭСМ-6 составляет I5 мин. На рис.2–5 указаны только те терминалы, которым выделены ЭМ; в скобках указаны номера ВЦ, которыми пользуются соответствующие терминалы, например, на рис.5 [IO] 3(Ш(2),УI) означает, что IO-у терминалу выделено 3 ЭМ (две ЭМ с третьего ВЦ, одна с шестого); римскими цифрами указаны номера ВЦ.

По табл.2 можно проследить, как изменяются значения целевой функции от числа испытаний q и расстояния h (и соответственно время счета) для кольцевой распределенной ОВС. Очевидно, брать $q > 500$ не имеет смысла, а при хорошем базовом распределении h может быть достаточно мало.

Достоинства метода цепей Монте-Карло станут очевидными, если учесть, что при полном переборе требуется вычисление целевой функции более чем для $2 \cdot 10^{15}$ различных комбинаций.

Таблица I

Распределение машин по терминалам распределений ОВС

№ ВЦ	j	c_j	$\mu_{j\,max}$	$\mu_{j\,min}$	Кольцевая ОВС x_j	Кольцевая ОВС № ВЦ 1	2	3	4	5	6	7	Линейная ОВС x_j	Линейная ОВС № ВЦ 1	2	3	4	5	6	7	Радиальная ОВС x_j	Радиальная ОВС № ВЦ 1	2	3	4	5	6	7	ОВС произвольной структуры x_j	ОВС произвольной структуры № ВЦ 1	2	3	4	5	6	7
1	1	1,2	1	1																	1	1														
	2	1,7	1,5	1																																
	3	2	2	1,5																																
2	4	1,9	4	1,5									1	1																						
	5	2,5	4	2,5	1	1							1		1																					
	6	3,3	3	1																																
	7	5	4	1,2																																
	8	4	2,5	2																																
3	9	2,7	3	1	1	1							1	1																						
	10	3,5	2,4	2,2																																
	11	4	2,3	1,2																																
	12	2	3	1,5																																
	13	3	3	2																																
	14	1,5	3	3									1			1																				
4	15	1,6	2	1									1	1							2	1	1						1	1						
5	16	2,2	3,3	2,3									3	1	1						7		2		2	1	2									
	17	3,4	4	4																																
	18	5,7	3	2,1																																
	19	2	4	3,5																																
6	20	7	4	1	2	1					1		1	1					1		1			1					1	1						
	21	2	4	1,5	3	1	1				1		1						1																
	22	1,1	3	1	1			1																												
	23	3,7	1	1	1					1			1				1	1			1				1										
	24	2,5	2	1	2				1	1			2				1	1			1				1										
	25	8,0	3	1	2				1			1	2					1		1															
	26	4,3	3	1,2																																
7	27	5,4	2	2									1						1	1																
	28	7,1	4	3,5																																
	29	2,3	3,2	2,3																																
	30	1,5	2	1,7	1			1																					1							1
	31	2,4	3	1,2																																
	32	3	3	1	1							1									1						1									
$Z_{баз} - Z^* = \Delta Z$					149,45–81,64=67,81								188,15–127,83=60,32								153,65–102,79=50,86								143,95–101,47=42,48							

Таблица 2

Целевая функция для кольцевой ОВС

Расстояние от базовой последовательности (h)	Число испытаний (g)													
	50		100		200		300		400		500			
	$\mathcal{Z}$	t	$\mathcal{Z}$	t	$\mathcal{Z}$	t	$\mathcal{Z}$	t	$\mathcal{Z}$	t	$\mathcal{Z}$	t		
2	104,1	0,3	104,1	0,4	104,1	0,7	104,1	1,1	104,1	1,3	104,1	2,5		
4	101,5	0,7	95,8	1,6	90,0	2,6	90,0	4,0	90,0	3,9	90,0	4,3		
8	102,7	0,8	97,6	1,8	94,7	2,2	89,0	5,6	83,9	8,9	90,9	8,2		
16	95,7	1,1	92,5	1,7	98,6	5,3	96,7	6,8	75,3	9,5	81,5	10,8		
32	100,7	1,7	88,0	2,5	90,0	5,3	87,2	7,5	87,8	12	81,6	15		

Таким образом, задача распределения машин по терминалам распределенной ОВС может быть поставлена в терминах стохастического программирования. Показано, что метод цепей Монте-Карло эффективен при решении подобных задач. Данный подход достаточно просто позволяет организовать стохастически оптимальное функционирование распределенных ОВС.

Литература

1. ЕВРЕИНОВ Э.В., ХИРАТКОВ В.И. Распределенные вычислительные системы и особенности их построения. -В кн.: Вычислительные системы. Вып. 63. (Теория однородных вычислительных систем). Новосибирск, 1975, с. 109-120.

2. КОРБУТ А.А., ФИНКЕЛЬШТЕЙН Ю.Ю. Дискретное программирование. М., "Наука", 1969.

3. PAGE E.S. On Monte-Carlo methods in congestion problems : I Searching for an optimum in discrete Satuations. -"Operation Research", 1965, v.13, p.291-299.

4. БУСЛЕНКО Н.П. Моделирование сложных систем. М., "Наука", 1968.

Поступила в ред.изд.отд.
28 января 1977 года

П р о г р а м м а

для расчета стохастически оптимального распределения $\{x_1,\dots,x_j,\dots,x_l\}$ ЭМ по терминалам и соответствующих потерь Z.

Программа написана на языке АЛГОЛ. Вводятся следующие исходные данные:

n – число ЭМ в распределенной ОВС,
l – число терминалов,
T – количество ВЦ,
K – коэффициент штрафов,
h – максимальное расстояние от базовой последовательности,
g – допустимое число испытаний,
u0, u1 – случайные числа, нормально распределенные в промежутке (0,4),
c_j – стоимость эксплуатации одной ЭМ с j-го терминала,
$\mu_{j\max}$ – средний спрос на j-м терминале при максимально требуемом ранге,
$\mu_{j\min}$ – средний спрос на j-м терминале при минимально требуемом ранге.
e_{ik} – стоимость канала между i-м и k-м ВЦ (i,k=1,...,T),
l_i – количество терминалов на i-м ВЦ (i = 1,...,T),
n_i – количество ЭМ на i-м ВЦ (i = 1,...,T).

На печать выдаются значения:

j – номер терминала,
x_j – количество ЭМ выделенных j-у терминалу,
s_{ij} – количество ЭМ, выделенных i-м ВЦ j-у терминалу.

```
begin integer n,l,h,g,i,j,p,q,x,T,S,D,m;
      real d,u0,u1,K,Z1,Z2,r0,tau;
      read (n,l,T,K,h,g,u0,u1);
begin real array c1,mi1,mi2[1:l],c,N[1:T,1:T],P1,P2,П1,П2[0:n],X,
      v,r[1:h]; integer array M1,M2[1:n+l-1],l1,n1[1:T],L, lambda
      [1:l],kappa[0:h],ksi[1:h],s[1:h,1:n+l-1],PA[1:T];
      read (c1,mi1,mi2,c,l1,n1);
begin procedure ЦЕЛЬ(f,n,l,c,c1,mi1,mi2,l1,T,L,Z);
      array f,c,mi1,mi2,l1,L,c1; real Z;
begin Z:=0; p:=1; i:=q:=0;
ДБ  : S:=0; j:=1;
 0: S:= S+l1[j]; if p> S then {j:= j+1; go to 0}; for x:= 1
    step 1 until T do PA[x]:= 0;
KSI: if F[i+1]> 0 then
begin q:= q+1; i:= i+1; if PA[f[i]]=0 then
    {PA[f[i]]:= f[i]; Z:= Z+c[j,f[i]]}; if i< n+l-1 then go to
    KSI
end;
    L[p]:= q; i:= i+1; if  i< n+l-1 then
    {p:= p+1; q:= 0; go to ДБ }; if f[n+l-1]=0 then L[1]:= 0;
    j:= 1;
    P1[1]:= P1[0]:= P2[0]:=П1 [1]:= П1 [0]:= П2 [0]:= 1;
ПУАС: if L[j]= 0 then go to Ф ; x:=0; r0:= exp(-mi1[j]);
    d:= exp(-mi2[j]); P2[1]:= 1-r0; П2 [1]:= 1-d;
    if L[j] = 1 then go to Ф; for x:= 1 step 1 until L[j]-1 do
begin r0:= r0 x mi1[j]/x; d:= d x mi2[j]/x;
    P1[x+1]:= P2[x]; П1 [x+1]:= П2 [x];
    P2[x+1]:= P2[x] - r0; П2 [x+1]:= П2 [x] - d
end;
  Ф: Z:=Z+c1[j] x L[j] + c1[j] x (mi1[j] x P1[L[j]] -
    L[j] x P2[L[j]]); if (mi2[j] x П1 [L[j]] -
    L[j] x П2 [L[j]]> 0.5  then Z:= Z+K x mi1[j]+c1[j] x L[j];
    j:= j+1;  if j< l then go to ПУАС;
    for j:= 1 step 1 until l do Z:=Z - mi1[j] x c1[j]
end;
    for j:= 1 step 1 until T do
begin p:= 1;
 AA: q:= 1; for i:= 1 step 1 until T do
begin if c[p,j] = c[i,j] then {if p> 1 then q:= q+1}
    else  {if c[p,j] > c[i,j] then q:= q+1}
end;
    N[q,j]:= p; if p < T then {p:= p+1;  go to AA}
end;
    for i:= 1 step 1 until T do PA[i]:= n1[i];
    for i:= 1 step 1 until n+l-1 do M1[i]:= 0;
    m:= 0; for i:= 1 step 1 until L do m:= m+entier(mi2[i] +
    0.5); if m < n then go to ВК;
 AK: D:= S:=0; p:= q:= j:= 1;
  Ф: S:= S+l1[j];
```

```
  ТУ: if p>S then {j:= j+1; go to Ф};
      i:= 1; m:= entier (mi2[p] + 0.5);  x:= q;
 ИСК: if PA[N[i,j]] > 0 then
begin N1[x]:= N[i,j]; PA[N[i,j]]:= PA[N[i,j]] - 1;
      D:= D+1; if D = n then go to ПАВ
end   else{i:= i+1; if  i ≤ T then go to  ИСК };
      x:= x+1; if x ≤ q + m - 1 then go to ИСК ;
      N1[x]:= 0; p:= p+1; q:= x+1; go to ТУ;
  ВК: r0:= 0; for i:= 1 step 1 until l do r0:=r0 +
      entier(mi1[i] + 0.5); S:= 0; p:= q:= j:= 1;
  ФФ: S:= S+l1[j];
 НЕВ: if p > S then{j:= j+1; go to ФФ};
      i:= 1; if r0 ≤ n then m:= entier (mi1[p] + 0.5)
      else m:= entier (mi2[p] + 0.5); x:= q;
  NN: if PA[N[i,j]] > 0 then
      (N1[x]:= N[i,j]; PA[N[i,j]]:= PA[N[i,j]] - 1)
      else {i:= i+1; go to NN}; x:= x+1;
      if x ≤ q+m-1 then go to NN;
      N1[x]:= 0; p:= p+1; q:= x+1;
      if p < l then go to НЕВ else
begin i:= 1;
  ПЧ: if PA[i]> 0 then for x:= q step 1 until q+PA[i]-1 do
      N1[x]:= i else {i:= i+1; if i< T then go to  ПЧ }
end;
 ПАВ: ЦЕЛЬ (N1,n,l,c,c1,mi1,mi2,l1,T,L,Z1); D:= 1;
ПЕЙДЖ: for i:= 1 step 1 until h-1 do
begin tau:= u0+u1; u0:= u1;  if tau> 4 then tau:= tau-4;
      u1:= tau; v[1]:= tau/4
end;
      for i:= 1 step 1 until h-1 do X[i]:= (n+1-i) x v[i]; p:= 1;
УПОР: x:= 1; for i:= 1,...,h-1 do
begin if X[p] = X[i] then {if p>1 then x:= x+1} else
      {if X[p]>X[i] then x:= x+1}
end;
      r[x]:= X[p]: if p<h-1 then
      {p:= p+1; go to  УПОР};
      for x:= 1 step 1 until h-1 do kappa [x]:= entier (r[x]);
      kappa[0]:= 0; kappa[h]:= n+1-1;for i:= 1 step 1 until h do
```



```
begin tau:= u0+u1; u0:= u1; if tau >4 then
      tau:= tau-4; u1:= tau; v[1]:= tau/4
end;
      p:= 1;
 ПЕР : x:= 1; for i:= 1 step h do
begin if v[p]=v[i] then {if p> 1 then x:= x+1}
      else {if v[p]> v[i] then x:= x+1}
end;
      r[p]:= x; ksi[x]:= kappa[p]-kappa[p-1];
      if p<h then { p:= p+1; go to ПЕР };
      for i:= 1 step 1 until h do
begin if kappa[i] > kappa[i-1] then
begin for j:= kappa[i-1]+1 step 1 until kappa[i]
      do s[r[i],j-kappa[i-1]]:= N1[j]
end
end;
      j:= 1; x:= 1;
   R: if ksi[x]>0 then
      for i:= 1 step 1 until ksi[x] do
      {N2[j]:= s[x,i]; j:= j+1};
      if x<h then {x:= x+1; go to R};
      ЦЕЛЬ (N2,n,l,c,c1,mi1,mi2,l1,T,lambda,L2);
      if Z2< Z1 then
begin Z1:= Z2; for i:= 1 step 1 until n+1-1 do N1[i]:=N2[i];
      D:= 0; for i:= 1 step 1 until l do L[i]:= lambda[i]
end;
      D:= D+1; if D <g then go to ПЕЙДЖ  else {h:= h/2; D:= 1;
      if h>1 then go to ПЕЙДЖ }; q:= 1; for j:= 1 step 1 until l
      do
begin for i:= 1 step 1 until T do PA[i]:= 0;
      if N1[q]=0 then print (j,L[j],PA) else
begin for p:= q step 1 until q+L[j]-1 do
begin for x:= 1 step 1 until T do
      if N1[p]=x then PA[x]:= PA[x]+1
end;
      print (j,L[j],PA)
end;
      q:= q+L[j]+1
end end end end
```