УДК 681.142.2

ОПТИМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ ИЗБЫТОЧНЫХ МАШИН ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ В РЕЖИМЕ КОНВЕЙЕРНОЙ ОБРАБОТКИ ДАННЫХ

В.В.Кирюхин, Е.Р.Ягор

За последние годы в мультипроцессорных системах, функционирующих в реальном масштабе времени, находит распространение конвейерный [1] (или каскадный [2], или поточный [3]) способ обработки данных. Необходимым требованием, предъявляемым к подобным системам, является условие высокой достоверности (безошибочности) результатов вычислений. Это требование особенно жестко должно выполняться для вычислительных систем, установленных в телеавтоматических системах массового обслуживания типа центров управления воздушным движением, резервирования и продажи билетов в крупных аэропортах [2] и т.п.

При современном состоянии надежности вычислительных средств высокая достоверность результатов обычно достигается с помощью контроля методом двойного счета либо с помощью мажоритарного принципа принятия решений при тройном счете. Реализация этих методов на одной электронной вычислительной машине (ЭВМ) требует достаточного быстродействия и надежности. С другой стороны, реализация двойного или тройного счета на параллельно работающих ЭВМ не предъявляет жестких требований к каждой из них. По существу, в условиях ограниченного быстродействия и недостаточной надежности ЭВМ последний способ представляется пока единственно возможным с точки зрения обеспечения как требования высокой достоверности результатов вычислений, так и требования работы в реальном масштабе времени.

Однородные вычислительные системы (ОВС) [4] имеют программно управляемую структуру, и поэтому они наиболее приспособлены для решения подобного рода задач. В ОВС может быть выделена основная подсистема для реализации задач, представленных последовательными программами, в конвейерном режиме [5]. Остальные машины ОВС, образующие вспомогательную подсистему, могут быть использованы для повышения надежности той или иной фазы обработки, причем с изменением параметров фаз можно обеспечить перераспределение машин между фазами конвейера с помощью программной коммутации. Подобная система может быть реализована, в частности, на ОВС МИНИМАКС [6].

Возникает задача оптимального в некотором смысле распределения машин вспомогательной системы по фазам конвейера. Дадим формальную постановку этой задачи.

На ОВС, состоящую из N элементарных машин (ЭМ), поступает пакет из m задач, представленных последовательными программами. Для их решения из состава ОВС выделяется о с н о в н а я п о д с и с т е м а из n ЭМ ($n \leqslant N$), функционирующая в конвейерном режиме. Каждая задача пакета, таким образом, должна пройти n последовательных фаз обработки. Будем считать известными величины ρ_{ij} – вероятности безошибочного решения фазы j задачи i. Далее, остальные $s = N - n$ ЭМ распределяются по фазам конвейера для работы в режиме параллельного счета. Естественно назвать совокупность этих машин в с п о м о г а т е л ь н о й п о д с и с т е м о й. Пусть x_j – число ЭМ вспомогательной подсистемы, распределенное на фазу j, и $P_{ij}(x_j)$ – вероятность безошибочного решения фазы j задачи i при наличии x_j избыточных машин. В частности, $P_{ij}(0) = \rho_{ij}$. В этих обозначениях величина

$$P_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} P_{ij}(x_j), \qquad (1)$$

очевидно, является вероятностью безошибочного решения задачи i. Положим, что качество работы ОВС оценивается следующим образом: система получает "доход" $\alpha_i > 0$, если задача i решена без ошибок; в противном случае доход равен нулю. Тогда ожидаемый доход от решения m задач пакета составит величину

$$A(x) = \sum_{i=1}^{m} \alpha_i \prod_{j=1}^{n} P_{ij}(x_j), \qquad (2)$$

где $x = (x_1, \ldots, x_j, \ldots, x_n)$, x_j – целочисленные неотрицательные компоненты. В частности, если $\alpha_i = 1$ для всех i, то $A(x)$ есть не что иное, как ожидаемое число правильно решенных задач пакета.

Требуется максимизировать функцию (2) в области S, задаваемой равенством

$$S = \left\{ x \,\middle|\, \sum_{j=1}^{n} x_j = s \right\}. \qquad (3)$$

Введем некоторые естественные определения. Очевидно, вектор x задает на ОВС некоторую с т р у к т у р у конвейерной обработки пакета. В частности, структура $x = (0, \ldots, 0)$ соответствует основной подсистеме. Структуру назовем д о п у с т и м о й, если $x \in S$. Допустимую структуру назовем о п т и м а л ь н о й, если она максимизирует (2). Здесь и далее понятия "вектор x" и "структура x" отождествляются.

Алгоритм построения структуры

По своей постановке задача (2), (3) относится к классу задач целочисленного программирования. Мы построим алгоритм ее решения, основываясь на сформулированном ниже условии оптимальности структуры.

ТЕОРЕМА. Н е о б х о д и м ы м п р и з н а к о м о п т и м а л ь н о с т и с т р у к т у р ы x я в л я е т с я у с л о в и е

$$A(x) \geqslant \max_{1 \leqslant k \leqslant n} \max_{\ell \in L} A(x_1, \ldots, x_k + 1, \ldots, x_\ell - 1, \ldots, x_n), \qquad (4)$$

$$L = \{ i \mid x_i > 0 \}.$$

ДОКАЗАТЕЛЬСТВО проведем от противного. Допустим, что структура x оптимальна, т.е.

$$A(x) \geqslant A(\tilde{x}), \qquad \tilde{x} \in S, \qquad (5)$$

но условие (4) для x не имеет места, то есть

$$A(x) < \max_{1\le k\le n} \max_{\ell\in L} A(x_1,\ldots,x_k+1,\ldots,x_\ell-1,\ldots,x_n). \quad (6)$$

Выберем структуру y следующим образом: $y_i = x_i$ для $i \ne \alpha, \beta$, $y_\alpha = x_\alpha + 1$, $y_\beta = x_\beta - 1$ для индексов α и β, соответствующих равенству

$$A(x_1,\ldots,x_\alpha+1,\ldots,x_\beta-1,\ldots,x_n) =$$

$$= \max_{1\le k\le n} \max_{\ell\in L} A(x_1,\ldots,x_k+1,\ldots,x_\ell-1,\ldots,x_n).$$

Очевидно, структура y допустима, так как $y \in S$, и компоненты вектора y целочислены. В то же время, ввиду (6) $A(x) < A(y)$, что противоречит условию (5) теоремы. Требуемое доказано.

Суть алгоритма отыскания решения состоит в том, что, отправляясь от некоторой допустимой структуры, он производит перераспределение машин по фазам до тех пор, пока не выполнится условие (4). Необходимо отметить, что получаемая таким образом структура может и не быть оптимальной, так как условие (4) лишь необходимо и в общем случае не является достаточным. Ниже (стр. 2I) будут выяснены условия, при которых алгоритм приводит к строго оптимальной структуре.

Алгоритм А.

(а) Пусть x - некоторая допустимая структура. Проверяется условие

$$\Delta_{\alpha\beta} \ge 0, \quad (7)$$

где

$$\Delta_{\alpha\beta} = \max_{1\le k\le n} \max_{\ell\in L} \left\{ \Delta_{k\ell} \right\},$$

$$\Delta_{k\ell} = \sum_{i=1}^{n} \alpha_i \, \delta_{ik\ell} \, P_i(x),$$

$$\delta_{ik\ell} = 1 - \frac{P_{ik}(x_k+1)\, P_{i\ell}(x_\ell-1)}{P_{ik}(x_k)\, P_{i\ell}(x_\ell)},$$

$$i = 1,\ldots,m, \quad k = 1,\ldots,n, \quad \ell \in L, \quad L = \left\{ j \mid x_j > 0 \right\}.$$

Если условие (7) выполнено, x - (суб)оптимальная структура и задача решена. В противном случае производится переход к (б).

(б) Выбирается новая допустимая структура следующим образом: $x_j := x_j$ для $j \ne \alpha, \beta$, $x_\alpha := x_\alpha + 1$, $x_\beta := x_\beta - 1$, и осуществляется переход к (а).

Оптимальные структуры высокоэффективных ОВС

Как уже отмечалось, предлагаемый алгоритм (стр. 20) в общем случае может дать лишь структуру, близкую к оптимальной (в смысле значения величины $A(x)$). Под общим понимается случай, когда на величины $P_{ij}(x_j)$ не накладывается никаких специальных ограничений. Однако следует заметить, что речь об организации обработки данных в конвейерном режиме и в реальном времени может идти лишь для ОВС, машины которой обладают достаточно высокой надежностью и, следовательно, вероятность (I) безошибочного решения какой-либо задачи достаточно близка к единице. Воспользуемся этим обстоятельством для формулировки ограничений на вид функции $P_{ij}(x_j)$; эти ограничения, с одной стороны, представляются естественными, а с другой - дают возможность получать с помощью алгоритма А (или его видоизменений) оптимальные структуры ОВС. Нам удобнее будет эти ограничения записать для функции $Q_{ij}(x_j) = 1 - P_{ij}(x_j)$, которая, очевидно, имеет смысл вероятности неверного решения фазы j задачи i.

Итак, пусть

$$q_{ij} \ll 1, \quad q_{ij} = 1 - p_{ij}, \quad (8)$$

$$Q_{ij}(x_j - 1) - Q_{ij}(x_j) \ge Q_{ij}(x_j) - Q_{ij}(x_j + 1). \quad (9)$$

Условия (9) означают, что функции $Q_{ij}(x_j)$ выпуклы. Вообще говоря, ограничения (8) и (9) независимы. Однако интересно отметить, что для существующих методов дублирования счета условия (9) могут быть следствием условий (8), и поэтому можно было бы ограничиться введением (8). Однако мы в целях общности рассмотрения оставим условия (9), так как не исключена возможность "изобретения" метода счета, при котором (9) из (8) не следуют.

Функцию (2) можно представить следующим образом:

$$A(x)=\sum_{i=1}^{m}\alpha_i\,[1-\sum_{j=1}^{n}Q_{ij}(x_j)+\sum_{k<j}Q_{ij}(x)Q_{ik}(x_k)-$$

$$-\dots+(-1)^n\prod_{j=1}^{n}Q_{ij}(x_j)\,].$$

Учитывая (8), всеми членами в квадратных скобках, начиная с третьего, можно пренебречь, и $A(x)$ принимает вид:

$$A(x)=\sum_{i=1}^{m}\alpha_i-B(x),\tag{10}$$

а

$$B(x)=\sum_{i=1}^{m}\alpha_i\,Q_{ij}(x_j)$$

можно рассматривать как величину потерь от неверного решения некоторых задач пакета. Очевидно, максимизация $A(x)$ эквивалентна минимизации $B(x)$. Таким образом, задача (стр.19) сводится к следующей: минимизировать потери

$$B(x)=\sum_{j=1}^{n}\varphi_j(x_j),\tag{11}$$

при условии $x\in S$.

Здесь $\varphi_j(x_j)=\sum_{i=1}^{m}\alpha_i\,Q_{ij}(x_j)$ – выпуклые функции, поскольку каждая из них является взвешенной (с положительными весами α_i) суммой выпуклых функций. Необходимый признак (4) оптимальной структуры, как легко показать, выглядит в случае (10) следующим образом:

$$\min_{1\le k\le n}\,[\varphi_k(x_k+1)-\varphi_k(x_k)]\ge\max_{\ell\in L}\,[\varphi_\ell(x_\ell)-\varphi_\ell(x_\ell-1)],$$

то есть он совпадает с признаком оптимального решения задачи (11), приведенным в [7], и, следовательно, является достаточным. Поэтому алгоритм А в случае функции $A(x)$ вида (10) обеспечивает получение оптимальной структуры. Представляется целесообразным для этого случая записать алгоритм отдельно, поскольку выражение (7) значительно упрощается.

А л г о р и т м В решает задачу минимизации (11) для $x\in S$.

(а) Пусть x – некоторая допустимая структура. Проверяется условие

$$\Delta_\alpha\ge\Delta_\beta,\tag{12}$$

где

$$\Delta_\alpha=\min_{1\le k\le n}\left\{\sum_{i=1}^{m}\alpha_i\,[\,Q_{ik}(x_k+1)-Q_{ik}(x_k)\,]\right\},\tag{13}$$

$$\Delta_\beta=\max_{\ell\in L}\left\{\sum_{i=1}^{m}\alpha_i\,[\,Q_{i\ell}(x_\ell)-Q_{i\ell}(x_\ell-1)\,]\right\}.\tag{14}$$

Если условие (12) выполнено, x – оптимальная структура. В противном случае – переход к (б).

(б) Выбирается новая допустимая структура следующим образом: $x_j:=x_j$ для $j\ne\alpha,\beta$, $x_\alpha:=x_\alpha+1$, $x_\beta:=x_\beta-1$ для α и β, соответствующих равенствам (13), (14), и осуществляется переход к (а).

Заметим, что задача (11), (3) может быть решена и с помощью алгоритма, приведенного, в частности, в [7]. Однако алгоритм В более предпочтителен в случаях, когда известна некоторая допустимая структура, близкая к оптимальной, либо когда необходимо достаточно быстро перестроить структуру при изменяющихся параметрах α_i, q_{ij} задач пакета.

Структуры при известных способах программного контроля

Наиболее распространенными способами контроля в технике вычислений являются методы двойного и тройного счета. При использовании двойного счета, как известно, задача (или некоторая ее часть) просчитывается дважды, результаты счета сравниваются, и при их совпадении полагают, что задача решена верно; в противном случае производится повторный счет и т.д. В данной работе рассматривается только случай двузначных результатов счета. Тогда если фраза j задачи i решается методом двойного счета на двух параллельно работающих машинах ОВС, то вероятность неверного её решения, очевидно, равна

$$Q_{ij}=q_{ij}^2.\tag{15}$$

Для повышения надежности вычислений можно предложить обобщенный метод двойного счета, когда задача решается параллельно не на двух, а на $x_j+1>2$ машинах, причем решение остается верным в том лишь случае, когда результат счета совпадает на всех ЭМ. Тогда

$$Q_{ij}(x_j) = q_{ij}^{x_j+1} . \qquad (16)$$

Легко проверить, что функция (I6) удовлетворяет условию (9). Следовательно, при соблюдении условия (8) оптимальная структура ОВС может быть получена с помощью алгоритма В.

При использовании тройного счета задача решается трижды, и в качестве решения принимают результат, полученный "по большинству". Если этот метод реализуется на трех параллельно работающих машинах ОВС, то совокупность этих ЭМ действует подобно трехвходовому мажоритарному элементу и вероятность неверного решения будет

$$Q_{ij} = 3\,q_{ij}^2 - 2\,q_{ij}^3 . \qquad (17)$$

Можно распространить этот метод на произвольное число машин. В этом случае совокупность машин, обрабатывающих j-ю фазу, действует подобно (x_j+1)-входовому мажоритарному элементу, или, что то же самое, подобно пороговому элементу с $K = x_j+1$ входами и порогом $\rho = (x_j/2)+1$, причем x_j должно быть четным: $x_j = 2,4,6\ldots$. Для использования полученных результатов удобнее иметь дело с x_j, принимающими значения I,2,3,... . Для этого положим $K = 2x_j+1$, $\rho = x_j+1$, а в ограничении (3) будем считать s равным $s/2$ (причем $s/2$ – целое). Применяя формулы надежности пороговых элементов с известными K и β [8], имеем:

$$Q_{ij}(x_j) = \sum_{k=x_j+1}^{2x_j+1} C_{2x_j+1}^{k}\, q_{ij}^{k}\,(1-q_{ij})^{2x_j+1-k} \qquad (18)$$

При выполнении условия (8) в сумме (I8) можно пренебречь всеми членами, кроме первого; тогда

$$Q_{ij}(x_j) \approx C_{2x_j+1}^{x_j+1}\, q_{ij}^{x_j+1} . \qquad (19)$$

Легко показать, что функция (I9) удовлетворяет условию (9). В самом деле, неравенство

$$Q_{ij}(x_j-1) + Q_{ij}(x_j+1) - 2Q_{ij}(x_j) >$$
$$> C_{2x_j-1}^{x_j}\, q_{ij}^{x_j} \left(1 - 4q_{ij}\,\frac{2x_j+1}{x_j+1}\right) > 0,$$

выполняется при любых x_j, если только $q_{ij} < 1/8$, что не противоречит (8). Таким образом, условие (8) влечет за собой для функции (I8) выполнение условия (9); на эту возможность уже было указано выше. Поскольку функция (I8) выпукла, здесь также же, как и в случае (I6), для получения оптимальной структуры можно применить алгоритм В.

АЛГОЛ-программы алгоритмов и результаты эксперимента

Ниже приводятся программы алгоритмов А и В на языке АЛГОЛ-60. Для решения задачи оптимального распределения необходимо ввести следующие параметры:

m – число задач пакета;

n – число ЭМ основной подсистемы;

s – число ЭМ вспомогательной подсистемы;

$q[1{:}m, 1{:}n]$ – массив вероятностей неправильного решения j-й фразы i-й задачи;

$a[1{:}m]$ – массив "доходов" от правильного решения задачи.

На печать выдаются начальная допустимая структура x, оптимальная структура x и величина дохода от реализации пакета задач $A(x)$.

Программа алгоритма А для метода двойного счета:

```
Begin integer j,i,m,n,s,S,e,f,r,t,u,v,U,V,Б ; real P,A,D,
DMIN,DELTA; read (m,n,S); print (m,n,S); begin integer array K,
L,X[1:n]; array q[1:m,1:n], a, B[1:m]; read (q,a); print (q,a);
Б :=entier (s/n); S:=0; for j:=1 step 1 until n do begin X[j]:=
Б ; S:=S+X[j]; end; j:=1; M1: if S ≥ s then go to M2 else begin
X[j]:=X[j]+1; j:=j+1; S:=S+1; go to M1; end; M2: print (X); M3:
for i:=1 step 1 until m do begin p:=1; for j:=1 step 1 until n
do p*:=p x (1-q[i,j] ↑ (X[j]+1)); B[i]:=a[i] x p; end; r:=0; t:=
0; for j:=1 step 1 until n do begin  M4:  t:=t+1;  K[t]:=j;  if
```

```
X[j]> 0 then begin r:=r+1; L[r]:=j; end; end; DMIN:=1;   l:=r;
f:=t; r:=1; M5: t:=1;  M6: if L[r]=K[t] then go to M8 else  go
to M7; M7: u:=K[t]; v:=L[l]; D:=0; for i:=1 step 1 until m  do
begin DELTA*:=1-((1-q[i,u] ↑ ((X[u]+1)+1)) x (1-q[i,v] ↑  (X[v]+
1))); D:=D+DELTA x B[i]; end; if DMIN≥ 0 then begin U:=u; V:=v;
DMIN:=0; end; M8: if t< f then begin t:=t+1; go to M6; end else
if r<l then begin r:=r+1; go to M5; end else if DMIN ≥ 0 then
go to M10 else go to M9; M9: for j:=1 step 1 until n do if j=U
then X[j]:=X[j]-1 else X[j]:=X[j]; go to M3; M10: A:=0; for  i
:=1 step 1 until m do A:=A+B[i]; print (X,A); end; end;
```

Для метода тройного счета следует ввести функцию процедуры ОТКАЗ, вычисляющую $Q_{ij}(x_j)$ по формуле (I9):

```
real procedure ОТКАЗ  (А1, Б1); value A1, Б1; integer A1; real
Б 1; begin integer k,l,M,N,C,F; integer array Д[ 0:2]; switch
B:=B1,B2,B3; l:=0; M:=2 x A1+1;  L1:  F:=1;  for  N:=1  step 1
until M do F:=F x N; Д [l]:=F; l:=l+1; go to B[l]; B1: M:=A1;
go to L1; B2: M:=A1+1; go to L1; B3: C:= Д[0]/( Д[1] x Д[2]);
ОТКАЗ := C x Б 1 ↑ (A1+1); end;
```

Описание процедуры помещается перед вводом q и a, а в ходе решения функции, отмеченные звездочкой (*), вычисляются по формулам:

```
p:=p x (1-ОТКАЗ (X[j], q[i,j]));
DELTA:=1-(((1-ОТКАЗ  (X[u]+1,q[i,u])) x (1-ОТКАЗ  (X[v]-1, q
[i,v])))/((1-ОТКАЗ  (X[u], q[i,u])) x (1-ОТКАЗ  (X[v] ,  Q[i,
v]))));
```

Программа алгоритма В запишется следующим образом:

```
Begin integer j,i,n,m,l,k,U,V,P,S,s, Б,t,r,K2,L2; real A,G,F,
R,E,D; read (m,n,s); print (m,n,s); begin integer array X[1:n],
K1,L1[1:n]; array q[1:m,1:n], a[1:m]; read (q,a); print (q,a);
Б := entier (s/n); S:=0; for j:=1 until n do begin X[j]
:= Б; S:=S+X[j]; end; j:=1;  M1: if S > s then go to M2 else
begin X[j]:=X[j]+1; j:=j+1; S:=S+1; go to M1; end;  M2:  print
(X);  M3: E:=0; D:=-1; t:=0; r:=0; for j:=1 step 1 until n do
begin t:=t+1; K1[t]:=j; if X[j]> 0 then begin r:=r+1; L1[r]:=
```



```
j; end; end; K2:=t; L2:=r; r:=1;  M4: t:=1;  M5: if K1[t] = L1
[r] then go to M7 else go to M6;  M6: l:=L1[r]; F:=0;  for T:=
1  step 1 until m do F:=F+a[i] x (q[i,l] ↑ (X[j]+1)-q[i,l] ↑ ((X
[l]+1)-1)); if D< F then begin V:=l; D:=F; end; k:=K1[t]; R:=Q;
for i:=1 step 1 until m do R:=R+a[i] x (q[i,k] ↑ ((X[k]+1)+1)-q
[i,k] ↑ (X[k]+1)); if E> R then begin U:=k; E:=R; end;  M7:  if
t< K2 then begin t:=t+1; go to M5; end else if r < L2 then begin
r:=r+1; go to M4; end else go to M8;  M8: if E> D then go to
M10 else go to M9;  M9: for j:=1 step 1 until n do if j=U then
X[j]:=X[j]+1 else if j=V then X[j]:=X[j]-1 else X[j]:=X[j]; go
to M3;  M10: print (X); A:=0; for i:=1 step 1 until m do begin
G:=1; for j:=1 step 1 until n do G:=G x (1-q[i,j] ↑ (X[j]+1)) ;
A:=A+a[i] x G; end; print (A); end; end;
```

При использовании метода тройного счета вводится функция процедуры ОТКАЗ, и вычисление R, F и G производится по формулам:

```
R:=R+a[i] x (ОТКАЗ(X[k]+1,q[i,k])- ОТКАЗ(X[k],q[i,k]));
F:=F+a[i] x (ОТКАЗ(X[l],q[i,l])- ОТКАЗ(X[l]-1,q[i,l]));
G:=G-(1- ОТКАЗ(X[j],q[i,j]));
```

Все программы отлажены на ЭВМ М-222. Эксперименты показали хорошее совпадение результатов счета по алгоритмам А и В для высоконадежных ОВС. Далее, если в качестве исходной допустимой структуры выбирать такую, в которой количества ЭМ, назначаемых на различные фазы, отличаются друг от друга не более чем на единицу, то оптимальное решение находится в среднем за минимальное время. Для систем с m = 5, n = I0, s = 40–50 время решения на ЭВМ М-222, включая трансляцию (транслятор ТА-IМ), составляло 4-5 мин при методе двойного счета и 6-8 мин – при методе тройного счета. На трансляцию тратится 3-4 мин, на одну итерацию алгоритма – 0,5 мин. Затраты времени на решение по алгоритмам А и В существенно не отличались. Следует заметить, что алгоритм В допускает распараллеливание при вычислениях величин D и E (соответствуют Δ_β и Δ_α в описании алгоритма) и, следовательно, может быть реализован за существенно меньшее время.

Литература

1. ФОСТЕР. Направление развития архитектуры ЭВМ. -"Зарубежная радиоэлектроника". 1973, № 8, с. 42-52.

2. ВИНОКУРОВ В.Г., ЗАСТЕЛА В.В., КОСТЕЛЯНСКИЙ В.М., НОВОХАТНИЙ А.А. Применение мини-ЭВМ в центрах обработки данных телеавтоматических систем массового обслуживания. -В кн.: Вычислительные системы. Вып. 51. Новосибирск, 1972, с. 146-156.

3. РАЙЛИ. Сеть малых вычислительных машин вместо большой ЭЦВМ. -"Электроника", 1971, № 29, с. 34-42.

4. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, "Наука", 1966.

5. МИРЕНКОВ Н.Н. Параллельные алгоритмы для решения задач на однородных вычислительных системах. -В кн.: Вычислительные системы. Вып. 57. Новосибирск, 1973, с. 3-32.

6. ВИНОКУРОВ В.Г., ДИМИТРИЕВ Ю.К., ЕВРЕИНОВ Э.В., КОСТЕЛЯНСКИЙ В.М., ЛЕХНОВА Г.М., МИРЕНКОВ Н.Н., РЕЗАНОВ В.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы из мини-машин. -В кн.: Вычислительные системы. Вып. 51. Новосибирск, 1972, с. 127-145.

7. СААТИ Т.Л. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. М., "Мир", 1973.

8. ДОМАНИЦКИЙ С.М. Построение надежных логических устройств. М., "Энергия", 1971.

Поступила в ред.-изд.отд.
25 сентября 1974 года