УДК 512.25 + 519.3

Д.А. ЗИЯУДИНОВА

АЛГОРИФМ СЛУЧАЙНОГО ПОИСКА С ИСПОЛЬЗОВАНИЕМ УСРЕДНЕННЫХ РАСПИСАНИЙ ДЛЯ РЕШЕНИЯ РЕСУРСНОЙ ЗАДАЧИ

Дан сетевой график $G(X,U)$ с одной начальной вершиной $a \in X$ и одной конечной вершиной $z \in X$. Для каждой работы $(x,y) \in U$ заданы неотрицательные величины: $\tau(x,y)$ - длительность данной работы и $s(x,y)$ - интенсивность потребления некоторого ресурса на данной работе. Для простоты считается, что все работы рассматриваемого графика используют ресурс одного вида. Иногда может быть задана функция $R(t)$ наличия этого ресурса как функция времени. Задан максимальный срок завершения процесса $T \geq T^*$. T^* - длина критического пути в сетевом графике.

Задача (I) ставится так:

Найти целочисленное расписание $\{\alpha(x,y),\ (x,y) \in U\}$, удовлетворяющее ограничениям:

$$\alpha(x,y) + \tau(x,y) \leq \alpha(y,v), \quad (x,y) \in U, \quad v \gg y,$$

$$\alpha(a,x) \geq 0, \qquad x \gg a,$$

$$\alpha(x,z) + \tau(x,z) \leq T, \qquad x \ll z \quad (T \geq T^*)$$

и доставляющее минимум функционалу

$$F_1(\alpha) = \frac{1}{T}\sum_{K=1}^{T}\left[\sigma(K,\alpha) - \sigma_0\right]^2.$$

Следовательно, в качестве меры неравномерности потребления ресурса в данной задаче принимается среднее квадратичное уклонение потребляемого в момент K ресурса от его среднего ежедневного потребления.

В случае, когда вместо средней интенсивности σ_0 задана функция $R(t)$ наличия ресурса, мы имеем другой вариант ресурсной задачи. В нем минимизируется функционал

$$F_2(\alpha) = \frac{1}{T}\sum_{K=1}^{T}\left[\dot{\sigma}(K,\alpha) - R(K)\right]^2$$

при тех же технологических ограничениях на последовательность работ. В качестве меры неравномерности потребления ресурса в этом варианте принимается среднеквадратичное уклонение потребляемого в момент K ресурса от его наличия в данный момент. Оба варианта задачи несущественно отличаются друг от друга, поэтому изложение алгорифма и запись его на АЛГОЛе приводятся только для варианта (I).

В данной работе рассматривается следующий метод решения поставленной задачи. Он имеет две характерные особенности. Во-первых, метод основан на том, что множество дуг U разбивается на классы эквивалентности

$$U = U_1 \cup U_2 \cup \ldots \cup U_L ,$$

где L - число классов эквивалентности. Внутри каждого класса дуги (работы) независимы в том смысле, что никакие две дуги из одного класса не связаны между собою путём. Это обстоятельство позволяет для построения расписания воспользоваться методом статистических испытаний, а именно выбирать начала работ внутри каждого класса случайным образом независимо друг от друга. На каждом шаге алгорифма для дуг данного класса строится N различных расписаний, выбираемых случайным образом. Второй особенностью метода является то, что для оценки полученных на каждом шаге расписаний используются усредненные расписания, введённые в [I] и [2].

Задача решается в L шагов. На ℓ-м шаге выделяется класс U_ℓ, и расписание строится сразу для всех дуг ℓ - го класса процедурой Монте-Карло.

К началу ℓ-го шага имеем:

U^* - множество дуг, для которых уже зафиксировано расписание α^*;

U_ℓ - дуги ℓ - го класса, для которых на данном шаге строится расписание;

$\bar{U}$ - все остальные дуги, для них строится усреднённое расписание.

Можно рассмотреть подробнее один шаг алгорифма.

Пусть $\ell = 1$. К началу первого шага имеем:

$$U^* = \Phi ,$$
$$U_\ell = U_1 ,$$

$$\bar{U} = U_2 \cup U_3 \dots \cup U_\ell .$$

Прежде всего решается задача о критическом пути для данного графика и определяются

$$A(x), \; B(x), \quad x \in X,$$

самые ранние и самые поздние сроки наступления событий. Затем выделяется множество U_1 дуг I-го класса, а для всех остальных дуг, то есть дуг из $\bar{U}$, строится некоторая усредненная характеристика $\bar{\sigma}(\kappa)$, которая представляет усредненную суммарную интенсивность потребления ресурса в каждый момент времени всеми дугами из $\bar{U}$ в предположении, что каждая работа $(x, y) \in \bar{U}$ с равной вероятностью может выполняться на любом отрезке из $[A(x), B(y)]$ независимо от других работ; то есть

$$\bar{\sigma}(\kappa) = \sum_{(x,y) \in \bar{U}} p_\kappa(x, y) \cdot s(x, y),$$

где величина $p_\kappa(x, y)$ есть вероятность выполнения работы (x, y) на интервале $[\kappa, \kappa+1]$ при усредненном расписании $\bar{\alpha}$. Тем самым для $\bar{U}$ построено так называемое усредненное расписание $\bar{\alpha}$.

Для построения расписания из U_1 пользуемся методом статистических испытаний (методом Монте-Карло). Проводится N испытаний. Каждое из этих испытаний состоит в том, что для каждой дуги $(x, y) \in U_1$ случайным образом выбирается начало её из промежутка $[A(x), B(y) - \tau(x, y)]$. Считаем, что $\alpha(x, y)$ является независимой случайной величиной, равномерно распределённой на этом интервале. Таким образом, в результате i-го испытания начала работ выбираются одновременно для всех работ $(x, y) \in U_1$ независимо друг от друга и получается комбинированное расписание

$$\tilde{\alpha}_i = \begin{cases} \alpha^* & \text{для } (x, y) \in U^*, \\ \alpha_i - & \text{случайное расписание для } (x, y) \in U_1, \\ \bar{\alpha} - & \text{усредненное расписание для } (x, y) \in \bar{U}, \end{cases}$$

$i := 1, \dots, N.$

Для каждого из полученных комбинированных расписаний вычисляется значение функционала $F(\tilde{\alpha}_i)$, и в качестве расписания для U_1 фиксируется случайное расписание $\{\alpha_{i_*}(x, y), (x, y) \in U_1\}$, доставляющее минимум функционалу F.

Итак, в результате одного шага процесса фиксируется распи-

сание сразу для всех дуг первого класса. После этого переходим ко второму шагу. Теперь

$$U^* = U_1 ,$$

$$U_\ell = U_2 ,$$

$$\bar{U} = U_3 \cup U_4 \cup \dots \cup U_L .$$

После построения расписания для дуг из U_L процесс заканчивается. Полученное расписание

$$\{ \alpha^*(x,y), (x,y) \in U \}$$

принимается в качестве приближенного решения задачи.
Число испытаний N можно задавать по-разному. В данном случае число испытаний N для каждого класса определяется следующим образом. Пусть N' есть суммарный резерв времени всех работ рассматриваемого класса. Тогда полагаем $N = K \times N'$, где K задано.

Изложенный алгорифм случайного поиска был применен для решения некоторых экспериментальных задач. Вычисления проводились на машинах М-20 и БЭСМ-6 СО АН СССР.

Моделью при решении задачи являлся сетевой график с m вершинами (событиями) и n дугами (работами) , информация о проекте задавалась в следующем виде:

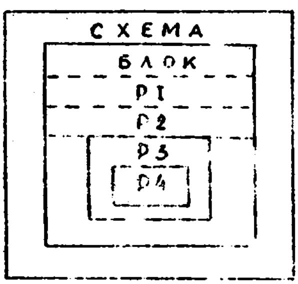
m , n - размерность,

E - целое, величина изменения T, $E \geq 0$. Если вычисления проводятся на критическом пути , $E = 0$;

n -мерные массивы x, y, t, S - информация о работах;

j -е компоненты каждого из массивов содержат информацию об одной и той же j -й работе и заполняются соответственно значениями величин: x предшествующее событие, y -последующее событие, $\tau(x,y)$ и $S(x,y)$.

Структура схемы алгорифма условно может быть изображена следующим образом:



В первой из четырех основных частей схемы прои:ходит обработка исходной информации: разбиение дуг на классы эквивалентности и их упорядочение таким образом, чтобы критический путь находился за один просмотр списка дуг; В Р2 проводятся некоторые предварительные вычисления; в Р3 описаны используемые в программе процедуры, а в Р4 строится расписание. Ниже приводится запись алгорифма на АЛГОЛе.

```
СХЕМА: begin integer m, n; ВВОД (m,n);
БЛОК:  begin integer a,b,d,e,h,i,j,k,l,q, ,tl,E,L,N,T;
              integer array x,y,t,α[1:n] ,A[O:m],B[I:m];
              real f,F,FO,p,sl,6m,6O,Ψ,f;
              real array s [I:n];
         ВВОД (E,x,y,t,s );
```

PI: Обработка исходной информации:

```
    begin
     A[O]:=O; for i:=I,...,m do {A[i]:=B[i]:=O};
                 for j:=I,...,n do α[j]:=O;
```

PII: Разбиение вершин по классам эквивалентности:

comment Для вершин графика вводится понятие ранга. Рангом вершины x назовем число $r(x)$, равное максимальной длине пути, заканчивающегося в данной вершине. Множество вершин графика X разбивается на классы эквивалентности таким образом, что к одному классу относятся вершины с одинаковым рангом;

```
     i:=O; for j:=I,...,n do
    { k:=B[x[j]]; q:=y[j]; if B[q]<k then {B[q]:=k+I; i:=I}};
     if i=1 then go to P11;
```

PI2: Вычисление размеров классов и числа классов эквивалентности:

```
     L:=O; for j:=I,...,n do {i:=B[y[j]]; A[i]:=A[i]+I;
     if i>L then L:=i};
     h:=O; for i:=I,...,L do {k:=A[i]; if k>h then h:=k};
     if L>h then h:=L;
  begin integer array C[-I:L],αI[I:n];
```

comment Массив C вводится для выделения границ классов эквивалентности в упорядоченном массиве дуг, i -я компонента массива $C[i]$ -номер последней дуги i -го класса. Массив $\alpha 1$ используется в предварительных вычислениях как массив размерности L, а также при построении расписания для дуг очередного класса как вспомогательный массив для хранения лучшего из случайных расписаний, поэтому размерность его выбирается следующим образом

$$h = max\ (L, \max_{i:=1,\ldots,L} A[i]),$$

где $A[i]$ -размерность i -го класса;

```
    for i:=I,...,h do αI[i]:=0;
    for i:=I,...,L do αI[i]:=A[i];
вывод (αI,L);
    C[-I]:=C[0]:=0;  C[I]:=A[I];
    for i:=2,...,L do
    {A[i]:=A[i-I]+ A[i]; C[i]:= A[i]};
```

PI3: Сортировка работ по классам эквивалентности:

comment Соответственно разбиению вершин на классы эквивалентности множество дуг U также может быть разбито на классы, притом двумя способами, а именно: две дуги относятся к одному классу, если они заканчиваются (начинаются) в эквивалентных вершинах. В данном алгорифме приводится упорядочение по концам, то есть такое упорядочение при котором к одному классу относятся дуги с эквивалентными конечными вершинами;

```
     for j:=1,...,n do
     begin
PI4: if α[j]=I then go to PI5;
     i:=B[y[j]]-I; k:=A[i]+I;
     a:=x[j]; b:=y[j]; tI:=t[j]; f:=s[j];
     x[j]:=x[k]; y[j]:=y[k]; t[j]:=t[k]; s[j]:=s[k].
     x[k]:=a; y[k]:=b; t[ ]:=tI; s[k]:=f;
    α[k]:=I; A[i]:=k; go to PI4
PI5: end;
```

Определение общего объема ресурса, потребляемого графиком:

```
Ψ :=0;
  for j:=1,...,n  do Ψ :=Ψ+t [j]×s[j];
вывод  ( x,y,t,s,Ψ );
 end P1;
```

P2: Предварительные вычисления:

```
begin

 Нахождение критического пути:
   T:=0; for i:=1,...,m do A[i]:=B[i]:=0;
   for j:=n step -1 until 1 do
   {k:=B[y[j]] +t[j]; i:=x[j]; if B[i]<k then
   {B[i]:=k; if T<k then T:=k}};
```

Определение срока завершения проекта:

```
   T:=T+K;
```

Определение поздних сроков наступления событий:

```
  for i:=1,...,m do B[i]:=T-B[i];
```

Определение средней интенсивности:

```
  σ0:=Ψ/T;
end P2;
```

P3: Процедуры :

```
begin

   real array σ, σ1[1:T];
```

comment В массиве σ вычисляется суммарное потребление ресурса на k -м интервале работами двух видов: работами, для которых расписание зафиксировано, и работами, для которых построено усредненное расписание. Массив σ1 вводится для работ ведущего класса. В этом массиве вычисляется суммарное потребление ресурса всеми работами ведущего класса при случайном расписании α ;

procedure выч A;

comment $A(x)$ –самый ранний допустимый срок наступления события $x \in X$. В процедуре перевычисляются значения $A(x)$ с учетом фиксированного расписания для работ первых ℓ классов эквивалентности, то есть для первых $C[\ell]$ работ. Перевычисление $A(x)$ производится от шага к шагу, то есть при переходе к очередному классу эквивалентности;

```
begin

   for j:=C[l-1]+1,...,C[l]  do
   {k:=α[j]+t[j]; q:=y[j]; if A[q]<k  then A[q]:=k};
```

```
    for j := C[l]+1,...,n do
    {k:=A[x[j]]+t[j]; q:=y[j]; if A[q]<k then A[q]:=k}

end выч A;
    procedure выч σ(φ);
    comment В массив σ[1:T] добавляется (φ = +1)
или вычитается (φ = -1) усредненная компонента ρ_k(x,y)×S(x,y),
отвечающая данной работе (x,y),
 j  -порядковый номер данной работы,
a = A (x),
b = B (y) + I;
  begin
     a:=α[j]; sl:=s[j]×φ; tl:=t[j];
     q:=b-a-tl;
     if q=1 then go to P31;
     f:=sl/q; if tl<q then {q:=tl; sl:=f×q}; p:=0;
     for k:=a+1,...,a+q-1 do {p:=p+f; σ[k]:=σ[k]+p};
     for k:=b-q+1,...,b-1 do {σ[k]:=σ[k]+p; p:=p-f};
   P31: for k:=a+q,...,b-q do σ[k]:=σ[k]+sl
  end выч σ(φ);
     procedure выч e;
comment В процедуре вычисляются границы временного интер-
вала, на котором выполняются работы ℓ - го класса эквивалент-
ности [ℓ,d] ∈ [1:T]
```

$$e = \min_{j \in U_\ell} A[x[j]],$$

$$d = \max_{j \in U_\ell} B[y[j]];$$

```
  begin
     e:=T; d:=0;
     for j:=C[l-1]+1,...,C[l] do
     {k:=x[j]; if A[k]<e then e:=A[k];
     k:=y[j]; if B[k]>d then d:=B[k]};    e:=e+1
  end выч e;
  procedure вых F0;
```

comment Вычисляется значение функционала F на промежутке $[\ell, ...] \in [1 : T]$ для данного расписания и одновременно фиксируется минимальное значение функционала для данного набора расписаний и случайное расписание для дуг рассматриваемого класса, доставляющее этот минимум функционалу;

```
begin F:=0;
    for k · e,...,d do
    {f:=б[k]+б1[k]-б0; F:=F+f×f}; F:=F/T;
    if F<F0 then {F0:=F; for j:=C[l-1]+1,...,C[l] do
    α1[j-C[l-1]] :=α[j]}
end выч F0;
    procedure б ;
```

comment В конце ℓ -го шага вычисляется массив $б$ при данном усредненном расписании, которое определяется уже зафиксированными началами работ ведущего ℓ -го класса. Вычисленные к моменту обращения к процедуре величины исправляются здесь за счёт компонент, для которых $\alpha[j] \neq A[x[j]]$;

```
begin
    for j:=C[l]+1,...,C[l+1] do выч б(-1); выч A ;
    for j:=C[l+1]+1,.. ,n do {if t[j]≠0 then
  {if α[j]≠A[x[j]] then {выч б(-1);α[j]:=A[x[j]] ; выч б(1)}}};
end сч (б);
procedure опр α;
```

comment Данная процедура осуществляет моделирование целых случайных чисел $\alpha[j]$, равномерно распределенных на $[A[x[j]]$, $B[y[j]] - t[j]]$, используя стандартную процедуру RAND, которая осуществляет моделирование случайных чисел, равномерно распределенных на $[0,1]$;

```
begin
    ξ :=RAND;
    if ξ=1 then α[j]:=b else α[j]:=a+entier(t-a+1) × ξ
  end опр α;
procedure опр N;
```

comment В процедуре вычисляются значения N для каждого класса дуг;

```
begin N:=0;
    for j:=C[l-1]+1,...,C[l] do
    {if t[j]≠0 then
    {i:=B[y[j]]-A[x[j]]-t[j]+1; N:=N+i}}
end опр N;
procedure нах σm;
```

comment Вводится показатель σm . Процедура нах σm вычисляет максимальное значение интенсивности потребления ресурса графиком на промежутке [I : T] ;

```
begin
    σm:=0;
    for k:=1,...,T do {i:=σ[k]; if i>σm then σm:=i}
end нах σm;
P4: Вычисление расписания:
begin
Чистка полей:
    for k:=1,...,T do σ[k]:=0;
Построение исходного усредненного расписания:
    l:=0; выч A;
    for j:=C[1]+1,...,n do
    {if t[j]≠0 then {α[j]:=A[x[j]]; выч σ(1)}
    else α[j]:=B[y[j]]};
```

comment ℓ - это номер ведущего класса, для которого на данном шаге строится расписание;

```
P4I: for j:=1,..,h do α1[j]:=0;
    l:=l+1;
    F0:=10↑18; выч e; опр N; N:=k×N;
    for i:=1,...,N do {for k:=e,...,d do σ1[k]:=0;
    for j:=C[l-1]+1,...,C[l] do{a:=A[x[j]];b:=B[y[j]]-t[j];
    if a=b then α[j]:=a else опр α;
    for k:=α[j]+1,...,α[j]+t[j] do σ1[k]:=σ1[k]+s[j]; выч F0};
    for j:=C[l-1]+1,...,C[l] do {α[j]:=α1[j-C[l-1]];
    for k:=α[j]+1,...,α[j]+t[j] do σ[k]:=σ[k]+s[j]}};
    if l+1>l then go to P42; сч σ; go to P41;
```

```
P42: FO:=0; e:=1; d:=T;
     for k:=1,...,T do 1 k :=0; выч FO; нах m;
вывод (T, , m, O, ,F);
     if E 0 then  E:=E-1; go to P2  end;
end P3;
end P4;
end БЛОКА;
end СХЕМЫ;
```

Литература

I. Л.Т. Петрова, Об одном подходе к решению задач распределения ресурсов в сетевом планировании. I-я Всесоюзная конференция по математическим вопросам сетевого планирования и управления. (Тезисы и доклады), Киев, 1967, стр. 86-88.

2. С.М.Анцыз, Л.Т. Петрова, Задачи распределения ресурсов в сетевом планировании . Оптимальное планирование, Изд-во "Наука", вып.7, Сибирское отделение, Новосибирск, 1967, стр. 41-76.

Поступила в редакцию
5 июня 1969 г.