УДК 512.25 + 519.3

С.Г. ЕСИПЕНКО

АЛГОРИФМ СДВИГОВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ СЕТЕВОГО ПЛАНИРОВАНИЯ

Постановка задачи

Пусть рассматриваемый проект содержит n работ и m событий. Считается для простоты, что на всех работах потребляется один вид ресурса. Для каждой j-й работы определены числа $t_j > 0$ и $r_j > 0$ (t_j - продолжительность j-й работы, r_j - интенсивность потребления ресурса на j-й работе). Время выполнения проекта $T \in [T^*, T^* + E]$, где T^* - критическое время, E - заданное целое число, $E > 0$.

Требуется найти план (указать сроки α_j начала работ, $j = 1, \ldots, n$), выполняемый за время T и минимизирующий функционал:

$$F(\alpha, T) = \frac{1}{T}\int_0^T [R(t) - R_{cp}]^2 dt = \frac{1}{T}\int_0^T R^2(t)dt - R_{cp}^2 .$$

$R(t)$ - суммарное потребление ресурса в момент времени t,

$$R_{cp} = \frac{1}{T}\sum_{j=1}^{n} t_j r_j .$$

Минимизация функционала $F(\alpha, T)$ при фиксированном T, очевидно, сводится к минимизации функционала:

$$\bar{F}(\alpha, T) = \int_0^T R^2(t)\, dt$$

Рассматривается дискретный вариант задачи. Тогда минимизируемый функционал имеет вид: $R_1^2 + R_2^2 + R_3^2 + \ldots + R_K^2 + \ldots + R_T^2$, (1)

где R_K - суммарное потребление ресурса в K-й день.

Алгорифм сдвигов

Для приближенного решения задачи применяется алгорифм сдвигов, несколько модифицированный по сравнению с алгорифмом, изложенным в [I].

Моделью при решении задачи будет сетевой график с n дугами (работами) и m вершинами (событиями).

Информация о проекте задается в следующем виде:

m, n -размерность,

E -целое, величина интервала изменения T, $E \geqslant 0$;

n- мерные векторы x, y, t, r - информация о работах. (Для j -ой работы: x_j и y_j - номера событий начала и окончания j - й работы, $t_j, t_j \geqslant 0$, и $r_j, r_j \geqslant 0$, - соответственно продолжительность j - ой работы и интенсивность потребления ресурса на j -й работе).

Полагаем T равным $T^* + E$.

Предварительный шаг

В качестве допустимого плана принимаем план

$$\Pi = \{\alpha / \alpha_j = A x_j, \quad j = 1, \dots, n\}$$, где

$A x_j$ - наиболее ранний допустимый срок начала j - й работы.

Общий шаг

Классы работ просматриваются последовательно, начиная с последнего. План улучшается покоординатно: считаются фиксированными все αj, кроме координаты ведущей работы. Если j_0 -номер ведущей работы, а $B y_{j_0}$ - наиболее поздний допустимый срок окончания ведущей работы, то αj_0 выбирается из интервала $[A x_{j_0}, B y_{j_0} - t_{j_0}]$ так, чтобы функционал (I) стал оптимальным. В работе [I] интервал изменения αj_0 есть $[\alpha j_0^*, B y_{j_0} - t_{j_0}]$ (αj_0^* - j_0 - ая координата плана на предыдущем шаге), т.е. сдвиг начала работы возможен только вправо. В модифицированном алгорифме возможен сдвиг начала работы и влево и вправо.

Порядок просмотра списка классов и порядок просмотра списка работ ведущего класса, вообще говоря, могут быть любыми, но они существенно влияют на результативность алгорифма.

Из всех возможных порядков просмотра списка работ ведущего класса выбираем два: прямой (по убыванию индекса j) и обратный (по возрастанию индекса j). Лучшее расписание работ класса определим по результатам этих просмотров путем сравнения суммарных приращений функционала. Обозначим через ψ , $\psi \leq 0$,приращение функционала после полного просмотра списка классов. Общий шаг на этом закончен.

Общий шаг повторяется до тех пор, пока $(-\psi)$ не станет меньше $\varepsilon, \varepsilon > 0$. (При решении задач ε полагали равным $10^{-3} \times F_0$, где F_0 - значение функционала при допустимом плане). Если же $(-\psi) < \varepsilon$, то задача решена для T . Если затем выполняется неравенство $T > T^*$, то T уменьшается на единицу и алгорифм применяется для решения задачи при новом T .

Таким образом, получаем $E+1$ планов. Модификацией алгорифма сдвигов, изложенного в [I] , являются двойной просмотр списка работ классов, а также допустимый сдвиг начала ведущей работы в обе стороны.

Ниже приводится запись модифицированного алгорифма на АЛГОЛе.

```
СХЕМА   :begin integer m,n; ввод (m,n);
БЛОК    :begin integer a,b,ba,c,d,g,h,ha,i,ii,j,jj,k,ka,kk,l,u,
                    ua,αa,E,T,TE;
            integer array α,x,y,t[1:n],A,B,C[1:m];
            real f,fa,fb,ε,s,sa,z,ψ,P;
            real array r[1:n];
 procedure Просмотр ведущего класса (β,δ,γ,q,Λ,P );
            integer β,δ,γ;
            integer array Λ;
            real q;
            real array P;
 begin q:=0.0; for j:=β step δ until γ do
      begin u:=1; ua:=-1; h:=ha:=k:=ka:=0;
            s:=sa:=z:=0.0; c:=x[j]; d:=y[j];
              kk:=if γ>β then j-β+ else j-γ+1;
                 a:=B[d]-A[c]-t[j];
                   if a=0 then go to Z13; g:=α[j];
                     l:=g+t[j]+1; ba:=B[d]- l+2 ;
```

```
                         b:=g-A[c]+1; g:=g+1;
Z1: ba:=ba-1; if ba=0 then go to Z4; ua:=ua+1;
    sa:=sa+P[l+ua]-P[g+ua]+r[j];
       if z ≥ sa then go to Z2 else go to Z3;
Z2: ha:=ha+ka+1; ka:=0; z:=sa; go to Z1;
Z3: ka:=ka+1; go to Z1;
Z4: sa:=z; z:=0.0; l:=l-1; g:=g-1;
Z5: b:=b-1; if b=0 then go to Z8; u:=u-1;
    s:=s+P[g+u]-P[l+u]+r[j];
    if z ≥ s then go to Z6 else go to Z7;
Z6: h:=h+k-1; k:=0; z:=s; go to Z5;
Z7: k:=k-1; go to Z5;
Z8: if h=0 then go to Z9 else go to Z10;
Z9: if ha=0 then go to Z13;
Z10:s:=z;comment исследуется возможность сдвига работы
влево и вправо путем сравнения возможных приращений функционала;
    if sa ≥ s then go to Z11 else go to Z12;
comment в массиве Λ производится запоминание величины
сдвига соответствующей работы ведущего класса, суммируется зна-
чение приращения функционала и перевычисляется распределение
ресурса;
Z11: Λ[kk]:=h; q:=q+2×s×r[j]; h:=-h; g:=α[j]+1;
   l:=g+t[j];for in:=1,...,h do {P[g-in]:=P[g-in]+r[j];
          P[l-in]:=P[l-in]-r[j]} ; go to Z14;
Z12: Λ[kk]:=ha; q:=q+2×sa×r[j]; g:=α[j]; l:=g+t[j];
     for in:=1,...,ha do {P[g+in]:=P[g+in]-r[j];
          P[l+in]:=P[l+in]+r[j]} ; go to Z14;
Z13: Λ[kk]:=0;
Z14: end; end;
```

```
U1  : предварительные вычисления: ввод (   x,y,t,r,E      ) ;
            for i:=1,...,m do  { A[i]:=0; B[i]:=1} ;
            for j:=1,...,n do  {α[j]:=0 } ;
    Y1  : просмотр списка:
            i:=0; for j:=1,...,n do  { k:=B[x[j]] ;
            if B[y[j]] < k then {B[y[j]] :=k+1; i:=1}} ;
               if i=1 then go to Y1;
вычисление размеров классов:
        for j:=1,...,n do  { i:=B[y[j]] ;A[i]:=A[i]+1};ii:=i;i:=1;
  K1  : вычисление границ классов:
      i:=i+1; if i:=m+1 then go to K3; A[i]:=A[i]+A[i-1] ;
             if A[i+1]>0 then go to K2 else go to K3;
  K2  : вычисление числа границ классов:
            ii:=ii+1; go to K1;
  K3 : begin integer array M[1:ii] ;
            for i:=1,...,ii do  { M[i]:=A[i]} ;
перестановка работ:

            for j:=1,...,n do

      begin
  Y2 :  if α[j]=1 then go to Y3; i:=B[y[j]] -1; k:=A[i]+1;
         a:=x[j]; b:=y[j]; d:=t[j]; f:=r[j]; x[j]:=x[k];
          y[j]:=y[k]; t[j]:=t[k]; r[j]:=r[k]; x[k]:=a;
               y[k]:=b; t[k]:=d; r[k]:=f; α[k]:=1;
                    A[i]:=k; go to Y2;
Y3:end; вывод     (x,y,t,r,M);
           for i:=1,...,m do{ A[i]:=B[i]:=C[i]:=0} ;
     comment   следующий оператор вычисляет самые ранние до-
пустимые сроки начала работ;
           for j:=I,...,n do   c:=x[j];d:=y[j];
               A[d]:=if C[d]≥C[c]+t[j]  then C[d]
                  else C[c]+t[j]; C[d]:=A[d]} ;
предварительные вычисления для нахождения самых поздних допусти-
мых сроков окончания работ:
           for i:=1,...,m do  {C[i]:=0} ;
```

```
for j:=n step-1 until 1 do {c:=x[j]; d:=y[j];
    B[c] :=if C[c] ≥ C[d]+t[j] then C[c]
      else C[d]+t[j]; C[c] :=B[c] } ;
```

comment следующий оператор вычисляет критическое время выполнения проекта TK ;

```
for j:=1,...,n do { TK:=if A[y[j]] ≥ A[x[j]]
    then A [y[j]] else A[x[j]] } ;  T:=TK+E;
```

вычисление самых поздних допустимых сроков окончания работ :

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

    вывод  (A,B,TK,T);
```

A1: begin comment в блоке A1 ищется план, минимизирующий функционал $F=\sum_{i=1}^{T} R_i^2$ при фиксированном T ;

```
real array R[1:T];
```

вычисление среднего уровня потребления ресурса:

```
f:=0.0; for j:=1,...,n do { f:=f+r[j]× t[j] } ;
    f:=f/T;        вывод ( f  );
```

выбор допустимого плана и вычисление соответствующего этому плану распределения ресурса:

```
for k:=1,...,T do { R[k] :=0.0} ;
for j:=1,...,n do {α[j] :=A[x[j]] ;
    for k:=α[j]+1,...,α[j]+t[j] do{R[k] :=R[k]+r[j]}};
```

вычисление среднеквадратичного уклонения потребления ресурса от среднего:

```
F:=0.0; for k:=1,...,T do { F:=F+(R[k]-f)↑2 } ;
    F:=F/T; вывод (R,F);
```

вычисление минимизируемого функционала :

```
F:=0.0; for k:=1,...,T do { F:=F+R[k]×R[k]} ;
                     вывод (F); jj:=0; ε :=0.001 × F;
```

```
L0: ψ:=0.0;  jj:=jj+1;
```

просмотр классов, начиная с последнего:

```
               for i:=ii-1 step-1 until 1 do
      begin α a:=M[i+1]-M[i] ;
         begin integer array L,D[1:αa];
     Просмотр ведущего класса    (M[i]+1,1,M[i+1],fa,L,R);
            for k:=1,...,T do {R[k]:=0.0} ;
               for j:=1,...,n do {
                  for k:=α[j]+1,...,α[j]+t[j] do {R[k]:=R[k]+r[j]}}
   Просмотр ведущего класса   (M[i+1],-1,M[i]+1,fb,D,R);
сравнение двух последовательностей просмотра по результатам:
               if fa≥fb then go to L1;
исправление расписания работ ведущего класса и перевычисление
распределения ресурса:
               for j:=M[i+1] step-1 until M[i]+1 do {
                  kk:=j-M[i]; α[j]:=α[j]+L[kk]} ;
               for k:=1,...,T do { R[k]:=0.0} ; ψ:=ψ+fa;
               for j:=1,...,n do { for k:=α[j]+1,...,α[j] +
                    t[j] do {R[k]:=R[k]+r[j]}} ; go to L2;
L1:            for j:=M[i]+1,...,M[i+1] do {kk:=j-M[i] ;
                  α [j]:=α[j]+D[kk]} ; ψ :=ψ+fb;
L2:   end; end;
               ψ:=abs(ψ); F:=F-ψ; вывод (jj,F,ψ);
                  F:=0.0; for k:=1,...,T do {F:=F+(R[k]-f)↑2},
         F:=F/T; вывод (F);
            if ψ ≥ ε then go to L0;        вывод (α );
end A1;  for i:=1,...,m do { B[i]:=B[i]-1} ; T:=T-1;
               if T ≥ TK then go to A1;
        end K3;
           end БЛОКА
              end СХЕМЫ ; *
```

Поступила в редакцию

15.XI.1967 г.