ВЕКТОРНО ГРАФИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЗАДАЧАХ БЕЗОПАСНОГО ДВИЖЕНИЯ СУДОВ

Болотов В.П., Зеленин А.И.

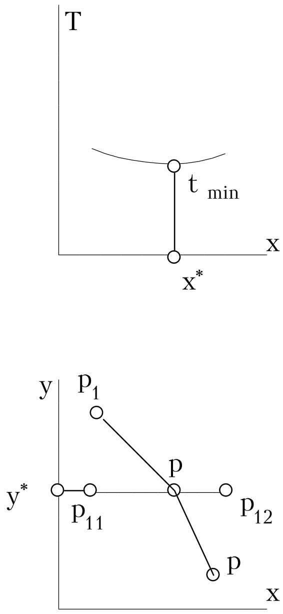

Рис. 1

Движение с заданной скоростью v определяется как расстояние s в зависимости от скорости и времени s=vt. Движение точки p по прямой линии p1-p2 со скоростью s1 от точки p1 в зависимости от времени s определяется:

p =p1 + s1*s p99

где p99 - единичный вектор (p99=|p1-p191| /s99), где s99 - длина отрезка p1=p2 (определяется s99=|p1-p2|).

С использованием операций системы “Вектор” типа “Отложить данное расстояние на прямой или кривой линии от заданной точки” задачи подобного типа решается намного проще.

Пример . Определить маршрут движения путника по скошенному и нескошенному лугам из точки ***А*** в точку ***Б*** так, чтобы маршрут его движения был наиболее коротким и время его прохождения было наименьшим. Скорость движения по скошенному лугу равна ***v***1, по нескошенному - ***v***2.

1. ***Анализ:*** исходя из формул определяющих взаимосвязь скорости, времени и расстояния (***v=s/t***), определяем, что время нам неизвестно и его удобно выбрать в качестве минимизирующего параметра.

2.1. Алгоритмизация**:* общее время движения путника по лугу 1 и лугу 2 определяется суммой ***t1 и ***t***2 :

$$t = t1 + t2 = |p - p1| \,/\, v1 + |p - p2| \,/\, v2,$$

где

$p = (1-u)p11 + up12.$

0<u<1

u - безразмерный параметр (веса). Может обозначаться любой буквой.

2.2. ***Формируем ЦФ***:

$$f(u) = \frac{\sqrt{(x-x1)^2 + (y-y1)^2}}{v1} + \frac{\sqrt{(x-x2)^2 + (y-y2)^2}}{v2}$$

3. ***Решение*** заключается в формировании ЦФ (построении графика), нахождению ее минимального значения (времени, за которое путник пройдет весь путь) и нахождении ему маршрута движения (рис. 2).

Пример 2. Определить на каком наикратчайшем расстоянии пройдут два судна ($\boldsymbol{p}_i$, и $\boldsymbol{p}_j$), при их движении в пересекающихся маршрутах. Известны скорость первого судна (v_1) и второго (v_2). Направление движения судов определены векторами $\boldsymbol{p}_1\boldsymbol{p}_{10}$ и $\boldsymbol{p}_2\boldsymbol{p}_{20}$.

Формализация.

Целевая функция

$$S = |p_i - p_j| \to \min.$$

Ограничения

$$p_i = p_1 + t \cdot v_1 \cdot \frac{p_{10} - p_1}{|p_{10} - p_1|},$$

$$p_j = p_2 + t \cdot v_2 \cdot \frac{p_{20} - p_2}{|p_{20} - p_2|},$$

$$0 \le t.$$

Решение задачи осуществляем с помощью формирования массива точек ЦФ и ее анализа (рис. 2) в системе "Вектор"

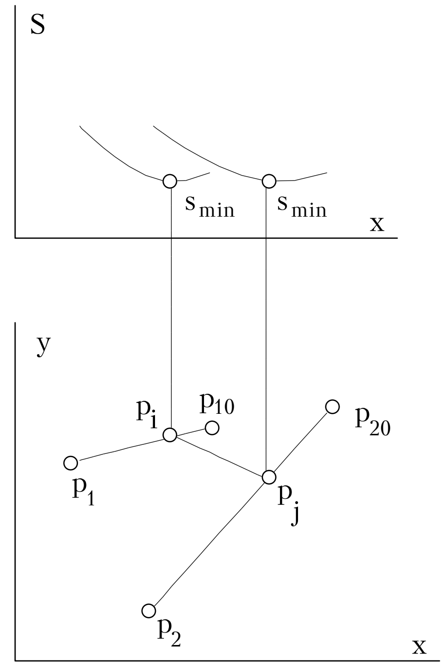

Рис. 2. Графическая интерпретация постановки и решение задачи 13, полученные в модуле Vec_optim системы "Вектор"

```
#include   <math.h>
#include  "geotyps3.h"
   main()
{
   point3 mthkp[11] [11]  ;
   point3 mthkp2[11] [11]  ;
   point3 p,pi,pj,p_ed1,p_ed2;
   float u,v,s,v1,v2,mod1,mod2,t,t_max;
   int k,i, npprx;
   point3
                  p1  = { 10., 30.0 },
                  p10  = { 70., 45.0 },
                  p2 =  { 20., 10.0 },
                  p20  = { 60., 55.0 };
     /* задача 13 "О расхожении двух судов, миним-
ем s от времени t */
     v1=1.1;  v2=1.7;      /* скорость */
    /* ед. вектора направления движения */
    mod1=sqrt((p10.x-p1.x)*(p10.x-p1.x) + (p10.y-
p1.y)*(p10.y-p1.y));
    p_ed1.x=(p10.x-p1.x)/mod1;
    p_ed1.y=(p10.y-p1.y)/mod1;
    mod2 =sqrt((p20.x-p2.x)*(p20.x-p2.x) + (p10.y-
p1.y)*(p20.y-p2.y));
    p_ed2.x=(p20.x-p2.x)/mod2;
    p_ed2.y=(p20.y-p2.y)/mod2;
     t_max=3.;
    npprx=11;
    for ( k=0, u=0; k < npprx; k++, u +=1./(npprx-1))
{
          t=u*t_max;
          pi.x=p1.x + t*v1*p_ed1.x ;
          pi.y=p1.y + t*v1*p_ed1.y ;
          pj.x=p2.x + t*v2*p_ed2.x ;
          pj.y=p2.y + t*v2*p_ed2.y ;
         s = sqrt((pi.x-pj.x)*(pi.x-pj.x)+(pi.y-pj.y)*(pi.y-
pj.y));
     mthkp [0] [k].x = pi.x;
     mthkp [0] [k].y = pi.y;
     mthkp [0] [k].z = s;
     mthkp2 [0] [k].x = pj.x;
     mthkp2 [0] [k].y = pj.y;
     mthkp2 [0] [k].z = s;
      p1=pi; p2=pj;
        }
        vectorcg( "line13", 1, npprx, mthkp );
        vectorcg( "lind13", 1, npprx, mthkp2);
return( 0 );
}
```

Задача 3. Определить на каком наикратчайшем расстоянии пройдут два судна ($\boldsymbol{p}_i$, и $\boldsymbol{p}_j$), если известно, что первое судно идет со скоростью v_1 по вектору $\boldsymbol{p}_{11}\boldsymbol{p}_{12}$, второе (скорость v_2), пытаясь уйти от столкновения, идет, все время сворачивая на угол а.

Формализация.

Целевая функция

$$S = \left| p_i - p_j \right| \to \min.$$

Ограничения

$$p_i = p_1 + t \cdot v_1 \cdot \frac{p_{10} - p_1}{\left| p_{10} - p_1 \right|},$$

$$p_j = p_2 + t \cdot v_2 \cdot \frac{(p_{20} + \Delta x) - p_2}{\left| (p_{20} + \Delta x) - p_2 \right|},$$

$$0 \le t.$$

Примечание. После первой итерации расчетов значения p_1, и p_j) перепресваеваются на p_i,и p_j), соответственно.

Решение задачи осуществляем с помощью формирования массива точек ЦФ и ее анализа (рис. 3) в системе "Вектор" **.**

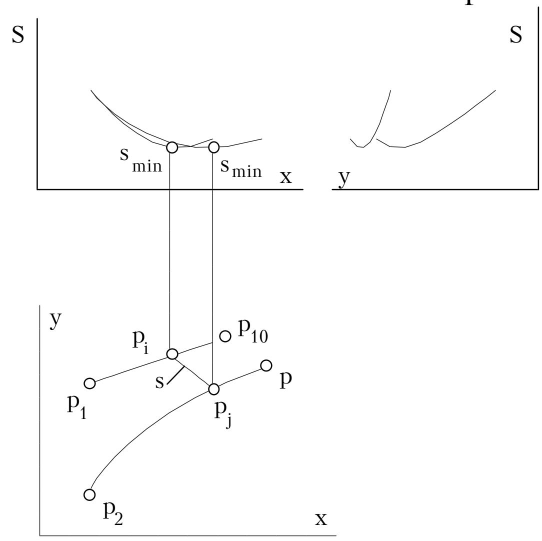

Рис. 3. Графическая интерпретация постановки и решение задачи 14, полученные в модуле Vec_optim системы "Вектор"

```
#include   <math.h>
#include  "geotyps3.h"
   main()
   point3 mthkp[11] [11]  ;
   point3 mthkp2[11] [11]  ;
   point3 p,pi,pj,p_ed1,p_ed2,pd,p3;
   float
u,v,s,v1,v2,mod1,mod2,t,t_max,alfa,d_alfa;
   int k,i;
   int npprx;
   point3
                 p1  = { 10., 30.0 },
                 p10  = { 70., 50.0 },
                 p2 =  { 10., 10.0 },
                 p20  = { 20., 50.0 };
     /* задача 14 "О расхожении двух судов,
миним-ем s от времени t */
     v1=3.4;  v2=5.5;      /* скорость */
    /* ед. вектора направления движения */
    mod1=sqrt((p10.x-p1.x)*(p10.x-p1.x) +
(p10.y-p1.y)*(p10.y-p1.y));
    p_ed1.x=(p10.x-p1.x)/mod1;
    p_ed1.y=(p10.y-p1.y)/mod1;
    mod2 =sqrt((p20.x-p2.x)*(p20.x-p2.x) +
(p20.y-p1.y)*(p20.y-p2.y));
    p_ed2.x=(p20.x-p2.x)/mod2;
        p_ed2.y=(p20.y-p2.y)/mod2;
        alfa=tan((p20.y-p2.y)/(p20.x-p2.x));
        d_alfa=3.14/40.0;
     t_max=1.3;
     npprx=11;
     for ( k=0, u=0; k < npprx; k++, u
+=1./(npprx-1)) {
/*          mthkp [0] [k].x = u;   */
           t=u*t_max;
           pi.x=p1.x + t*v1*p_ed1.x ;
           pi.y=p1.y + t*v1*p_ed1.y ;
           pj.x=p2.x + t*v2*p_ed2.x ;
           pj.y=p2.y + t*v2*p_ed2.y ;
          s = sqrt((pi.x-pj.x)*(pi.x-pj.x)+(pi.y-
pj.y)*(pi.y-pj.y));
      mthkp2 [0] [k].x = pi.x;
      mthkp2 [0] [k].y = pi.y ;
      mthkp2 [0] [k].z = s;
      mthkp [0] [k].x = pj.x;
      mthkp [0] [k].y = pj.y ;
      mthkp [0] [k].z = s;
      p1=pi; p2=pj;
      alfa = alfa - d_alfa;
      p3.x=p2.x+100.;
      p3.y=p2.y+tan(alfa)*(p3.x-p2.x);
     mod2 =sqrt((p3.x-p2.x)*(p3.x-p2.x) + (p3.y-
p1.y)*(p3.y-p2.y));
     p_ed2.x=(p3.x-p2.x)/mod2;
     p_ed2.y=(p3.y-p2.y)/mod2;
        }
        vectorcg( "line14", 1, npprx, mthkp );
        vectorcg( "lind14", 1, npprx, mthkp2 );
return( 0 );
}
```