2016 年软件学院

信息安全专业数学建模论文

建模题目：智能地雷问题

团队成员：段 晨 姜善棋 薛 旗

姓名	学号
段 晨	20155360
姜善棋	20155326
薛 旗	20155362

Catalog

1.Abstract

Intelligent mine is a new type of landmine, can take the initiative to track and target, and can through autonomic regulation arrangement range as far as possible do no dead fire and control. According to given intelligent mine fire control coverage area. According to the steps of division, model construction, reasonable assumptions, looping through, solver, insufficient analysis, etc. this problem for detailed analysis method. According to the methods and steps of solving this problem specific functions and the corresponding and complete the modeling and solution of this problem is put forward.

For problem 1, the given rectangle region is divided, and the attack range of mine is converted into the circle covering problem. Through the establishment of space rectangular coordinate system, mine is abstracted into a point, landmine attack range is abstracted as to point as the center of the circle, the number of mines and location coordinates. Respectively by triangle inscribed in a circle and quadrilateral shape on a given area are divided, and the circular to cover the rectangular area, full coverage of the landmine attack range simulation. For mine redistribution problems, by changing the inscribed polygon shape of the regions were recovered, reduce mine attack range overlap region, improve the coverage of the mine, in order to achieve full coverage.

For question 2, in mine mobile speed constant to make mine movement with the least amount of time, which should be met in order to ensure the full coverage of the case, all the mines in mobile and mobile's largest mines in all scenarios in mobile distance is the shortest distance, thereby transforming to the moving distance and traversal function problem. We calculate and determine the regional division of the standard, then the loop program to coordinate mathematical model corresponding traversal, traversal process using detection function for each vertex of mine fire detection coverage, and then use the adjustment function of the corresponding adjustment and change the variable covering the location of the circle, so repeatedly, until all vertices were intelligent mine fire cover, suggesting that problem of the rectangular area was completely intelligent mine fire control coverage, calculated amount of movement of the points of maximum in all scenarios minimum moving distance, complete to the problem of solving the mathematical model.

Key words: regional division, from point to area, coordinate, cyclic traversal, detection and adjustment function

2. Restatement Of the Problem

Now there is a weapon called intelligent mine. Intelligent mine is "with legs mine", not only to jump, but also can jump, but also to fly, can be active and accurate detection and tracking of tanks, armored vehicles, the vertical attack tank top or abdomen. These mines have been able to get smart, because it and the detection technology, sensor technology, microprocessor technology and other high-tech combination, so show an unprecedented vitality.

In a rectangular area of a uniform distribution of intelligent mine, fire control covering an area of overlap, according to the current distribution density of the 80% distribution of mine, fire control is still able to cover all areas. When there is a part of the mine after the fire, the fire control of the dead, the mine will automatically adjust the position to re distribution as much as possible, do not appear fire control dead.

The problem we have now is:

1 to adjust the position to re distribution, how to move as little as possible (that is not necessarily all mobile, re distribution is no longer uniform distribution)?

2 how to use the shortest time to complete the move?

3 if you have time, you can use the computer to simulate.

3. Analysis Of The Problem

3.1 problem 1 Analysis

First given region to rectangle region and need more pieces of the intelligent mine can do fire control region complete coverage of a given area, so we should divided into regions corresponding to a given rectangle, a gold intelligent mine fire control area can cover a segmentation area.

For each part of the area of segmentation is still to a surface, is not easy to establish mathematical model and discussed and solved, and under the precondition of reasonable assumptions, the split after four vertices of the small rectangular area instead of the small rectangular areas, namely as long as four vertices are intelligent mine fire completely covered, think rectangular area of the four vertices are intelligent mine fire, which completely covered, the 2D problem is transformed to one-dimensional problems.

In order to establish a complete mathematical model and the given rectangle establish Cartesian coordinate system, and will be discussed in the area and vertex position and the number of intelligent mine and place the coordinate, to build the mathematical model of reasonable and perfect.

On the basis of the mathematical model, the solution of the problem is solved by using the method of loop traversal, function detection and adjustment.

3.2 problem 2 Analysis

In problem, through the analysis and calculation, obtained to ensure complete coverage of the mobile distance to the point of maximum in all scenarios of minimum moving distance to represent to the minimum time required to complete a moving target.

4. Model Assumptions

We address different issues and propose the following assumptions:

1, every mine explosion has no effect on other intelligent mine;

2, intelligent fire control range coverage detection function based on specific issues and determine;

3, in a mine fire control area of the rectangular area, whether it is fire coverage can be attributed to the region's four vertices are mine fire coverage;

4, given a reasonable number of rectangular area and the initial number of intelligent mine;

The coordinate of center 5, covering circle can be accurately positioning, covering the same radius of circle;

5. Definition and symbol description

A: the length of the rectangular area.

B: the width of the rectangular region

R: intelligent mine explosion radius

A: the length of each small rectangle

B: the width of each small rectangle

I: unit of length (i.e. X axis unit)

J: width unit (i.e. Y axis unit)

M: length range, i.e. 0 = I = m (m=A/a)

N: the width range, i.e. 0 = J = n (n=B/b)

K: start with a number of intelligent mine

K: the first K function of mine

X[K]: a one-dimensional array of X coordinate values for all mine mines

Y[K]: a one-dimensional array of Y coordinate values for all mine mines

Coordinates (I, J) points

ρ : repeat coverage area

s1 : Intersection area

s2 : Unit circle area

ρ=s1/s2

Contribution: Unit circle inscribed regular polygon in the horizontal (vertical) direction of the length.

6. The establishment and solution of the model

6.1 region division

According to the requirements of the subject, the rectangular area is divided into a number of small rectangular shapes according to the following principles.

The principle is divided into: the length of the rectangular area B and width a in accordance with the B and A unit span for segmentation, and at the same time to meet the relationship; after the division of the region as shown in figure 1:

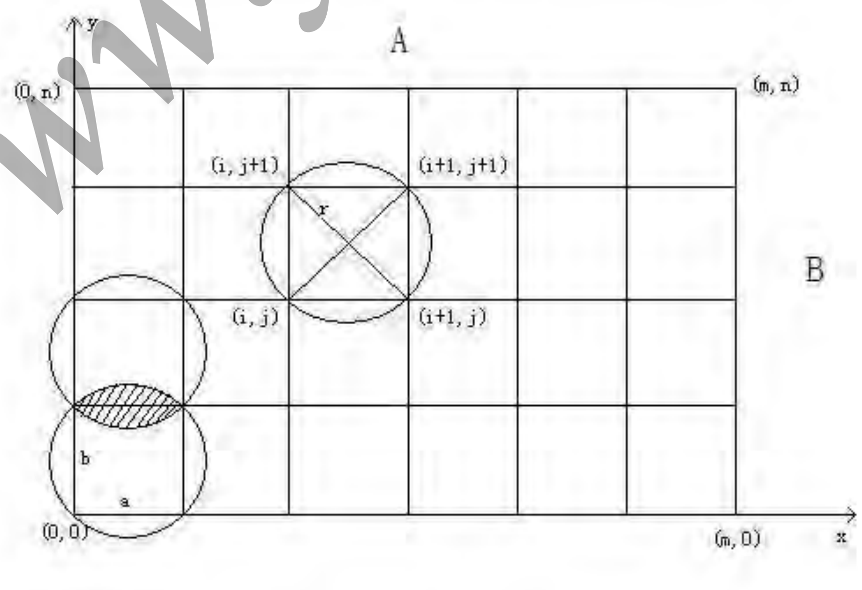

Figure 1

Note: the center of each small rectangle (the intersection point of the diagonal) is the control center of the rectangle. The circular area is the fire coverage area of the control center.

6.2 model analysis and assumptions

By diagram shows the firepower of a whole piece of rectangular area covering problem can be simplified sum for a small rectangle of each fire cover problem, namely as long as it meets the intelligent mine fire control area covering a small rectangle of each, which meet the requirements of the subject, the intelligent mine fire control covering the whole piece of rectangular area, no dead fire control.

And according to classification principles and schema of rectangular, each small rectangular area is intelligent mine completely firepower coverage can boils down to this four vertices of a rectangular area is completely covered with a mine fire, namely: to verify design rectangular region is completely intelligent mine fire coverage, only need to verify that the division of small rectangular all vertices are full power can be covered.

So but the use of the two for cycle to complete the traversal of these vertices, in order to test which vertices are not covered by mine fire, and make the appropriate adjustments.

```
note：for(i=0;i<=m;i++)
          {for(j=0;j<=n;j++){……}
……}
```

6.3 answers to questions

First of all, the use of a given array Y[K], X[K] are stored in the existing intelligent mine x, y coordinates, and after each explosion of a mine to the corresponding coordinate data in X, Y array deleted, into X[K-1], Y[K-1].

Use two for loops iterate through all the vertex coordinates (i , j) , and $0 < i < m$, $0 < j < n$, use the check function (i , j) to test whether coordinates (i , j) is mine fire, if so, the function returns a value of true, otherwise it returns false. Then test whether of the four vertices of a small rectangular region are fire. If so,

check(i，j)&&check(i+1，j)&&check(i，j+1)&&check(i+1，j+1)=true,

the area needn’t be adjusted. If not,

check(i，j)&&check(i+1，j)&&check(i，j+1)&&check(i+1，j+1)=false,

that is to say that (i+1/2, j+1/2) as the center of the rectangular region is not fully covered, and by assuming that know a is more than or equal to B, you need the abscissa in between I and I + 1 (i.e. I is less than or equal to X[k] is less than or equal to 1. The following diagram---- The shaded part of Figure 2 shows) mine to adjust.

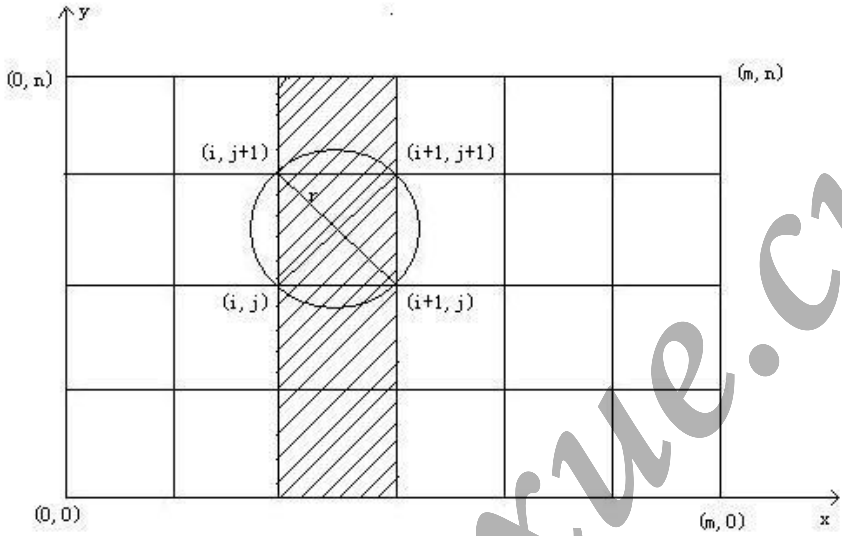

Figure 2

Use function distance (p , q) to calculate the p mine and the Q mines' distance ,finding out in the above image shown in the shaded area in the distance from the point of recent mine point K, and to make the corresponding adjustment by moving the move function (k) mine mobile point-to-point i+1/2 j + 1 / 2, then the use of for the circulation and the check function test rectangular region contains all vertices if there is not intelligent mine fire cover, if, in accordance with the above steps to adjust; otherwise, the completion of the inspection and adjustment, the program continues to, until the (M + 1) * (n + 1) vertices are mine fire covered so far.

6.4 limited condition

By the principle of region division and figure, problem given rectangle area division is M-by-N a small rectangular area, so apparently available in this way to arrange intelligent mine, at least m * n intelligent mine (and this Mn Mine Distribution in the center of each small rectangular area) to make the fire completely cover a given area, do in the region without dead angle of fire control. That is to say when the number of intelligent mine is less than m*n, no matter how to adjust the mine fire control can not be fully covered with a given rectangular area.

6.5 model promotion and sample

The discussion is inscribed rectangle partition of the coverage problem, for inscribed in a circle is triangular and hexagonal coverage problem solving ibid.

Theorem: some radius r circle completely cover the sufficient conditions for square area is the round inscribed regular polygon is completely covered by the square area.

Proof: Assume that the positive n shape A covers the square region of the B, A all A (i=1,2,3...... M), the area covered by the square area B, that is, all the B coverage area contains the square area B. Because of the circumcircle formed by A O, will include the regional B, so all O (i=1,2,3...... M) coverage of the area must contain a square area B.

For two intersecting circles(Figure 3 shows):

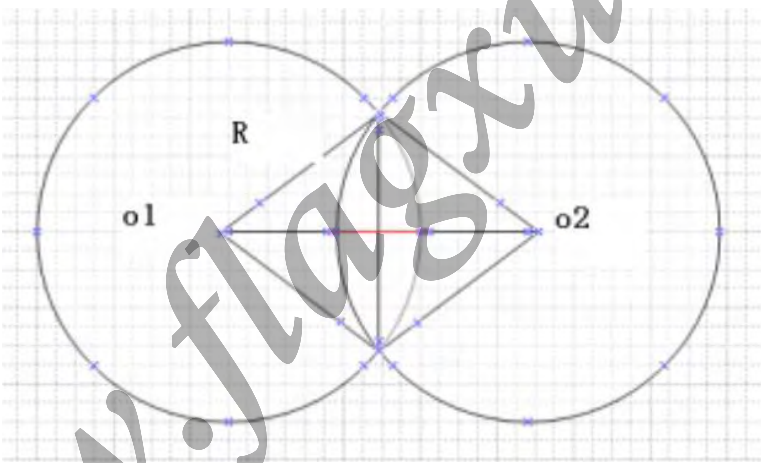

Figure 3

Intersection area $s_1 = 2[\frac{\theta}{180}\pi R^2 - \frac{1}{2}R^2\sin\theta]$

Assume $\rho = \frac{s_1}{s_2} = \frac{\frac{\theta\pi R^2}{180} - R^2\sin\theta}{\pi R^2} = \frac{\theta}{180} - \frac{\sin\theta}{\pi}$

So the value of ρ has no relation with radius of a circle R

Then transform the θ in $\sin\theta$ into value $\sin\frac{\theta\pi}{180}$

$$\frac{dk}{d\theta} = \frac{d}{d\theta}\left(\frac{\theta}{180} - \frac{\sin\frac{\theta\pi}{180}}{\pi}\right) = \frac{1}{180}\left(1-\cos\frac{\theta\pi}{180}\right) > 0$$

So with the increasing of θ, the value of ρ is monotonically increasing and the growth rate to meet the sine form (Figure 4 shows)

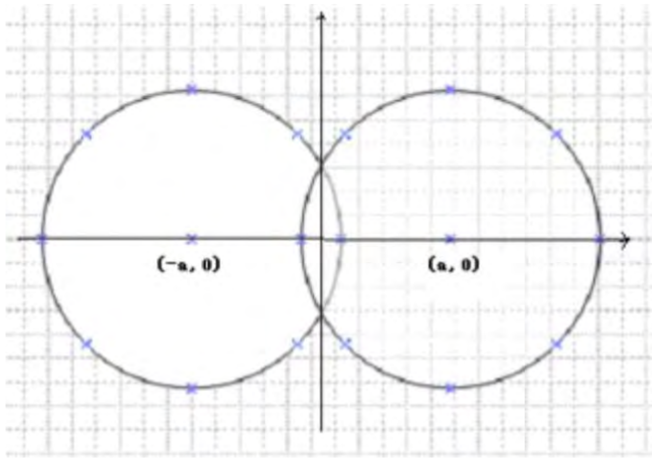

Figure 4

For a rather large rectangular area, we divide it into a number of regular polygons. Then these normal polygons should be covered with a full rectangular area which is not exactly coincident (we don't have to be covered by a regular polygon on the border)

Lemma 1: Is divided into regular polygon, the positive polygon can and can only be a positive triangle, positive quadrilateral and regular hexagon.

Now we come to the proof of lemma 1:

Prove : Assume the rectangle which is divided is a regular polygon which has n slides

And n $\geq$3, the value of each interior angle in a regular polygon is a, obviously you can conclude

$$a=\frac{180^{\circ}(n-2)}{n}$$

then we assume

$$x=\frac{360}{a}\ (x\geq 3, x\in N^{*}),$$

So

$$x=\frac{2n}{n-2}\Rightarrow n=\frac{2x}{x-2}=2+\frac{4}{x-2}$$

By solving the equation:

$$x=3,\quad n=6;\ x=4, n=4;\ x=6, n=3$$

That is to say, In dividing the rectangle into a regular triangle, it can only be the positive triangle, the positive quadrilateral and the regular hexagon. Therefore, we should choose the rectangular area of the rectangular area, which is the positive triangle, the positive quadrilateral or the regular hexagon.

Now we will talk about the critical value of k when n has different value.

Plan one: When using positive triangle to divide rectangle, n=3,

$$\frac{2\pi}{3} - Sin\frac{2\pi}{3} \geq k\% \cdot \pi$$

Then get the critical value, **k_1=39.1002**

Plan two: When using the positive quadrilateral to divide rectangle, n=4

$$\frac{2\pi}{4} - Sin\frac{2\pi}{4} \geq k\% \cdot \pi$$

Then get the critical value, **k_2=18.1690**

Plan three: When using the regular hexagon to divide rectangle, n=6

$$\frac{2\pi}{6} - Sin\frac{2\pi}{6} \geq k\% \cdot \pi$$

Then get the critical value, **k_3=5.7669**

Therefore,

1. When $k_2 \leq k \leq k_1$, we choose positive triangle to divide rectangle, the amount of positive triangle is the least amount of circles to cover the rectangle.

2. When $k_3 \leq k \leq k_2$, we choose positive quadrilateral to divide rectangle, the amount of positive quadrilateral is the least amount of circles to cover the rectangle.

3. When $0 \leq k \leq k_3$, we choose regular hexagon to divide rectangle, the amount of regular hexagon is the least amount of circles to cover the rectangle.

Table 1 shows:

k	**18.169＜k⩽39.1002**	**5.7669＜k⩽18.169**	**0⩽k⩽5.7669**
regular polygon which has n slides	3	4	6

Table 1

In addition, intersection of a given circle and circle area is K, we can also use the following inequality to calculated the value of n

$$\frac{2\pi}{n} - Sin\frac{2\pi}{n} \geq k\% \cdot \pi \geq \frac{2\pi}{n+1} - Sin\frac{2\pi}{n+1} \quad (n \in N^*)$$

In this problem, due to the assumption that the rectangular area of the length is a, far greater than the mine explosion radius of r, so the formula derived n=6, that is, with a regular hexagon to completely partition the original rectangular coverage area.

Below we bring into the actual value, to check.

Assume A=1600，B=1600，r=$100\sqrt{2}$

According to the initial uniform distribution (inscribed Square) as shown in Figure5:

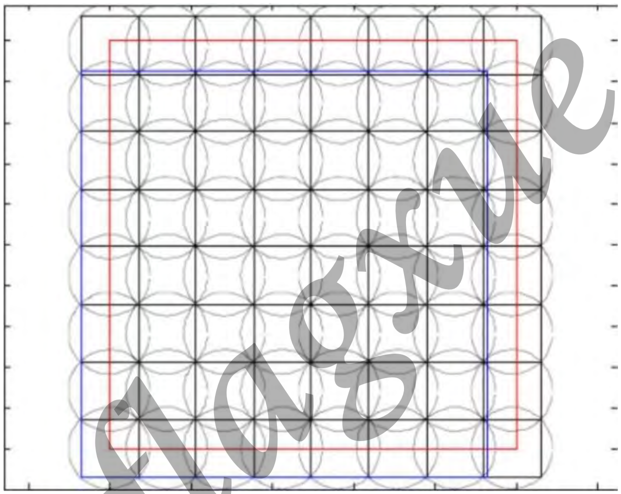

Figure 5

The angle is 90°, length is 200. The contribution extending on two directions is 200, the number is $\frac{1600}{200} = 8$, total number is 64.

When using regular hexagon to cover as the way in Figure 6 ,

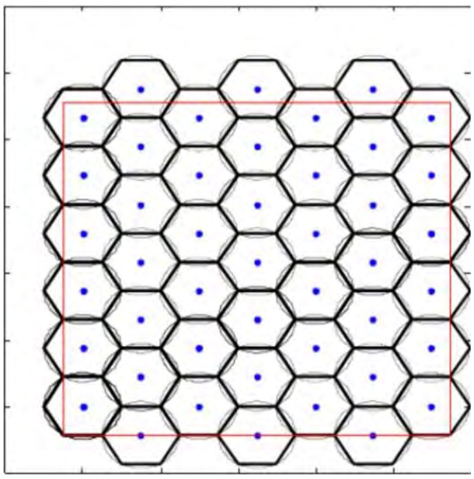

Figure 6

The angles of 120°, length is $100\sqrt{2}$, the contribution of odd numbered rows in perpendicular direction is $100\sqrt{6}$, the even lines of the first two hexagonal effective contribution is $50\sqrt{6}$, the horizontal direction contribution is $150\sqrt{2}$ (which is the boundary of a hexagonal contribution is $100\sqrt{2}$).

Therefore, the vertical direction of the hexagon is $\frac{1600}{100\sqrt{6}}+1=7$, the horizontal direction of a total of 7 lines, odd lines of each line needs 6, even lines of each line needs $\frac{1600\text{-}100\sqrt{2}}{150\sqrt{2}}+1=7$, so the number of regular hexagon is 45.

If the number of the original distribution of mines is 64, after blasting according to inscribed hexagon covering only 45 to achieve full coverage. Because 64*80%=51 is greater than 45, so the simulation results fit the purpose.

For question 2, only need to each mine blasting, adjacent to the recent ray in accordance with the inner hexagonal arrangement can achieve full coverage, and through cyclic algorithm in 5.3 we can get moved furthest mine in all scenarios the minimum value, then obtain the optimal solution.

7. model evaluation and promotion

Segmentation of the use of this method on the given topic of rectangular area treatment and through inspection division four vertices of the small rectangular area fire coverage to represent the corresponding small rectangular areas of the mine fire coverage, while the use of arrays and the for loop to split each vertex traversal, and the function for the corresponding detection and adjustment, has reached at any time to adjust the intelligent mine distribution to make fire control cover the rectangular area, no dead fire control.

8. Appendix (part of the procedure of the code)

8.1 check function

check (i , j) //The detection point is (I, J), the first input parameter is the horizontal unit value of the point to be detected. Second input parameters for the detection point of the vertical coordinates of the unit value, the return value of true or false: when the detection point is intelligent control of intelligent mine fire coverage, then return to true; otherwise return false.

8.2 distance function

distance(p，q) // The distance between point p and point q is calculated, and the input parameters are the horizontal and vertical coordinates of p points and q points. The return value is the distance between p and q points.

Formula is : distance(p，q)= $\sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$

8.3 move function

move(k， $R_j^{\ i}$) // move the number k of mine to the specified location $R_j^{\ i}$, input parameters one for the number of specified Mines, and call the coordinate value. Then input parameter two for the specified position, at the same time call the position coordinates. In the process of function implementation, first complete the call to the two sets of coordinates and will be moved mine coordinate values to specify the location of the corresponding coordinate values, in order to complete moving corresponding mine.

8.4 Program loop detection adjustment section

```
for(i=0;i<=m;i++)
{
    for(j=0;j<=n;)
    {
        if(check(i，j)&&check(i+1，j)&&check(i，j+1)&&check(i+1，j+1)=false)
        // These four points are not fully covered by the fire, or at least one of
    the mine is not covered by the fire.
            {
                int p,q,min=0;
    for(k=1;k<=K;k++)
                {
```

```
                    if(X[k]>=i&&X[k]<=i+1){
                        q=distance(k， $R_{j+1/2}^{i+1/2}$);
                    }   //calculate distance
                     If (q<min) {
                        min = q; p = k;
                    } // Find the minimum distance of mine number
                        and location, and record
             }
             move(p， $R_{j+1/2}^{i+1/2}$) // Adjust the nearest mine to the specified location
(i+1/2，j+1/2)
            j=0; //After setting j as zero, circulate again
          }
            else
                j ++;   //the value of j continues to increase and iterate over
    }
}
```

8.5 The code of Minimum circle cover

```
#include<stdio.h>
#include<math.h>
struct    TPoint
{
          double x,y;
};
TPoint a[1005],d;
double r;

double   distance(TPoint   p1,   TPoint   p2)   //the distance between
two points
{
    return (sqrt((p1.x-p2.x)*(p1.x -p2.x)+(p1.y-p2.y)*(p1.y-p2.y)));
}
double multiply(TPoint   p1,   TPoint   p2,   TPoint   p0)
```

```
{
    return  ((p1.x-p0.x)*(p2.y-p0.y)-(p2.x-p0.x)*(p1.y-p0.y));
}
void MiniDiscWith2Point(TPoint  p,TPoint  q,int  n)
{
    d.x=(p.x+q.x)/2.0;
    d.y=(p.y+q.y)/2.0;
    r=distance(p,q)/2;
    int k;
    double c1,c2,t1,t2,t3;
    for(k=1;k<=n;k++)
    {
        if(distance(d,a[k])<=r)continue;
        if(multiply(p,q,a[k])!=0.0)
        {
            c1=(p.x*p.x+p.y*p.y-q.x*q.x-q.y*q.y)/2.0;
            c2=(p.x*p.x+p.y*p.y-a[k].x*a[k].x-a[k].y*a[k].y)/2.0;

            d.x=(c1*(p.y-a[k].y)-c2*(p.y-q.y))/((p.x-q.x)*(p.y-a[k].y)-(p.x-
a[k].x)*(p.y-q.y));
            d.y=(c1*(p.x-a[k].x)-c2*(p.x-q.x))/((p.y-q.y)*(p.x-a[k].x)-(p.y-
a[k].y)*(p.x-q.x));
            r=distance(d,a[k]);
        }
        else
        {
            t1=distance(p,q);
            t2=distance(q,a[k]);
            t3=distance(p,a[k]);
            if(t1>=t2&&t1>=t3)
            {d.x=(p.x+q.x)/2.0;d.y=(p.y+q.y)/2.0;r=distance(p,q)/2.0;}
            else if(t2>=t1&&t2>=t3)

{d.x=(a[k].x+q.x)/2.0;d.y=(a[k].y+q.y)/2.0;r=distance(a[k],q)/2.0;}
            else
```

```
{d.x=(a[k].x+p.x)/2.0;d.y=(a[k].y+p.y)/2.0;r=distance(a[k],p)/2.0;}
            }
        }
    }

    void MiniDiscWithPoint(TPoint    pi,int    n)
    {
        d.x=(pi.x+a[1].x)/2.0;
        d.y=(pi.y+a[1].y)/2.0;
        r=distance(pi,a[1])/2.0;
        int j;
        for(j=2;j<=n;j++)
        {
        if(distance(d,a[j])<=r)continue;
            else
            {
                MiniDiscWith2Point(pi,a[j],j-1);
            }
        }
    }
    int main()
    {
        int i,n;
        while(scanf("%d",&n)&&n)
        {
            for(i=1;i<=n;i++)
            {
                scanf("%lf %lf",&a[i].x,&a[i].y);
            }
            if(n==1)
            { printf("%.2lf %.2lf 0.00\n",a[1].x,a[1].y);continue;}
            r=distance(a[1],a[2])/2.0;
            d.x=(a[1].x+a[2].x)/2.0;
            d.y=(a[1].y+a[2].y)/2.0;
```

```
        for(i=3;i<=n;i++)
        {
              if(distance(d,a[i])<=r)continue;
              else
              MiniDiscWithPoint(a[i],i-1);
        }
        printf("%.2lf %.2lf %.2lf\n",d.x,d.y,r);
}
return 0;}
```