31. 手机基站优化布局-1stOpt 非线性规划问题案例演示

30.1 问题描述

手机基站站址合理选址对区域内手机信号接收质量和范围无疑有着巨大的影响，因此优化布局手机基站具有相当重要的现实意义。

某地计划在一个边长为 L=12 的正方形区域内建设 n=10个手机基站，各手机基站有效范围是半径不等的圆域，其半径如表30.1.

表30.1 基站及对应的范围半径

基站	1	2	3	4	5	6	7	8	9	10
半径	1	1	1.5	1.5	2.0	2.0	2.5	2.5	3.0	3.0

基站站址布局定位的原则是在指定的区域内具有最大的信号覆盖面积，同时要求信号覆盖范围不超过该指定区域，在此前提下求各基站站址的具体位置。

30.2 问题分析

手机基站信号覆盖有效范围是半径不等的圆，并且半径均已给定，同时由于信号覆盖范围不超出指定正方形区域，因此所有基站覆盖的圆面积之和减去它们两两相互重叠的部分之后面积之和，即为手机基站覆盖到的时间面积。因此当基站两两重叠面积之和达到最小时，对应的覆盖面积就达到最大，因此本问题即转变为目标函数为求解基站最小重叠面积的优化问题。

30.3 模型建立及求解

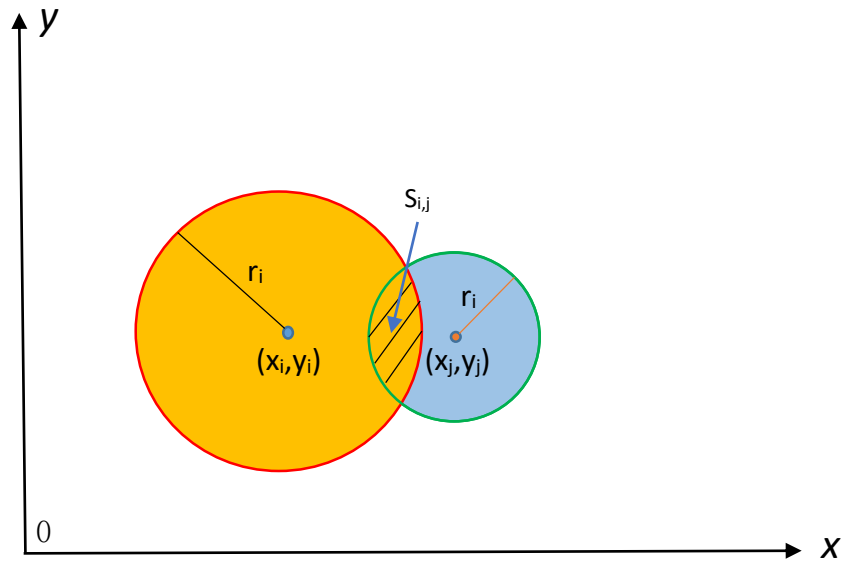

图30-1 基站信号覆盖示意图

如图.1所示平面直角坐标系，设第i和j个手机基站的圆心坐标分别为(xi,yi)和(xj,yj) ，用S_{ij}表示第i个基站和第j个基站的重叠面积，则可建立如下非线性规划模型：

$$\text{Min.} \sum_{i=1}^{n-1}\left(\sum_{j=i+1}^{n}(S_{i,j})\right) \tag{1}$$

$$\text{S.T.} \begin{cases} r_i \le x_i \le L - r_i \\ r_i \le y_i \le L - r_i \end{cases} \text{ i=1,2,3…10}$$

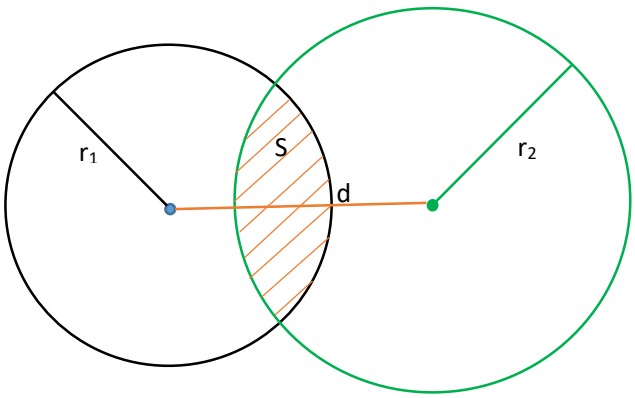

图30-2 相交两圆重叠部分面积计算

相交两圆重叠部分面积计算公式，参照图.2，可通过公式（2）或（3）计算得出，其中r1、r2为两圆各自半径，d为两圆圆心距离：

$$S = A_1 \cdot r_1^2 + A_2 \cdot r_2^2 - r_1 \cdot d \cdot \sin(A_1) \tag{2}$$

$$S = A_1 \cdot r_1^2 + A_2 \cdot r_2^2 - 0.5 \cdot \sqrt{(-d + r_1 + r_2) \cdot (d + r_1 - r_2) \cdot (d - r_1 + r_2) \cdot (d + r_1 + r_2)} \tag{3}$$

其中：

$$A_1 = Arccos\left(\frac{r_1^2 + d^2 - r_2^2}{2 \cdot r_1 \cdot d}\right)$$

$$A_2 = Arccos\left(\frac{r_2^2 + d^2 - r_1^2}{2 \cdot r_2 \cdot d}\right)$$

具体进行计算时还需考虑两圆不相交及两圆相套这两种情况，前者很明显，相交面积为0，后者为较小园的面积：

$$\begin{cases} S = 0, & d \geq r_i + r_j \\ S = \pi \cdot \left(min(r_i, r_j)\right)^2, & d \leq abs(r_i - r_i) \end{cases}$$

模型目的是求出每个基站的具体x与y坐标值，10个基站，对应的求解参数共有20个。该模型获取最优解的难度较大，缺省的UGO算法效果不理想的话亦可尝试其它算法如差分进化算法等。快捷模式、Pascal编程模式及Fortran编程模式三种代码实现模式如下。注意代码中使用了“LowBound”和“UpBound”批量定义参数上下界范围，此外快捷模式中使用反余弦函数“Arccos”时，为防止出现大于1的余弦函数计算的系统错误，使用了是否大于1的判断。

快捷模式代码：

```
Constant L = 12,                                        // 正方形区域的边长
         n = 10,                                       //手机基站的个数
         r(n)=[1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];   //各基站覆盖园半径
Algorithm = DE1;
Parameter x(n), y(n);
LowBound = [1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0,1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];
UpBound = [L-1,L-1,L-1.5,L-1.5,L-2.0,L-2.0,L-2.5,L-2.5,L-3.0,L-3.0,L-1,L-1,L-1.5,L-1.5,L-2.0,L-2.0,L-2.5,L-2.5,L-
3.0,L-3.0];
ConstStr d2 = (x[i]-x[j])^2+(y[i]-y[j])^2, d = sqrt(d2),
           ang1 = arccos(min(1,(r[i]^2+d2-r[j]^2)/(2*r[i]*d))),
           ang2 = arccos(min(1,(r[j]^2+d2-r[i]^2)/(2*r[j]*d))),
           s1 = ang1*r[i]^2 + ang2*r[j]^2 - r[i]*d*sin(ang1);
ConstStr f= if(d >= r[i]+r[j],0,if(d <= abs(r[i]-r[j]),pi*(min(r[i],r[j]))^2,s1));
MinFunction Sum(i=1:n-1)(Sum(j=i+1:n)(f));
```

Pascal 编程模式代码：

```
Constant L = 12,                                        //正方形区域边长
         n = 10,                                       //基站个数
         r(n)=[1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];   //各基站覆盖园半径
Algorithm = DE1;
Parameter x(n), y(n);
LowBound = [1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0,1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];
UpBound = [L-1,L-1,L-1.5,L-1.5,L-2.0,L-2.0,L-2.5,L-2.5,L-3.0,L-3.0,L-1,L-1,L-1.5,L-1.5,L-2.0,L-2.0,L-2.5,L-2.5,L-
3.0,L-3.0];
StartProgram [Pascal];
Procedure MainModel;
var i, j: integer;
    s, aij, dd: double;

    //(xx1,yy1)和(xx2,yy2)为两圆的圆心坐标，rr1 和 rr2 为两圆半径, d:两圆圆心距离
    function AreaIntersection(rr1,rr2,d: double): double;
    var s1, ang1, ang2: double;
    begin
        if d >= rr1+rr2 then s1 := 0                                    //两圆相离
        else if d <= abs(rr1-rr2) then s1 := pi*(min(rr1,rr2))^2        //两圆相套
        else begin                                                      //两圆相交
            ang1 := arccos((rr1^2+d^2-rr2^2)/(2*rr1*d));
            ang2 := arccos((rr2^2+d^2-rr1^2)/(2*rr2*d));
            s1 := ang1*rr1^2 + ang2*rr2^2 - rr1*d*sin(ang1);
        end;
        result := s1;
```

```
    end;

Begin
    //通过循环计算所有圆两两相交的面积总和
    s := 0;
    for i := 1 to N-1 do begin
        for j := i+1 to N do begin
            dd := sqrt((x[i]-x[j])^2 + (y[i]-y[j])^2);
            aij := AreaIntersection(R[i],R[j],dd);
            s := s + aij;
        end;
    end;

    ObjectiveResult := s;
End;
EndProgram;
```

Fortran 编程模式代码：

```
Constant L = 12,                                          //  正方形区域的边长
         n = 10,                                          //  手机基站的个数
         r(n)=[1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];      //各基站覆盖园半径
Algorithm = DE1;
Parameter x(n), y(n);
LowBound = [1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0,1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];
UpBound = [L-1,L-1,L-1.5,L-1.5,L-2.0,L-2.0,L-2.5,L-2.5,L-3.0,L-3.0,L-1,L-1,L-1.5,L-1.5,L-2.0,L-2.0,L-2.5,L-2.5,L-
3.0,L-3.0];
StartProgram [Fortran];

    !(xx1,yy1)和(xx2,yy2)为两圆的圆心坐标，rr1 和 rr2 为两圆半径, d:两圆圆心距离
    function AreaIntersection(rr1,rr2,d)
        real*8 s1, ang1, ang2
        real*8 rr1,rr2,d
        real*8, parameter :: pi = 3.1415926535

        if (d >= rr1+rr2) then
            s1 = 0                                        !两圆相离
        else if (d <= abs(rr1-rr2)) then                  !两圆相套
            s1 = pi*(min(rr1,rr2))**2
        else                                              !两圆相交
            ang1 = acos((rr1**2+d^2-rr2**2)/(2*rr1*d))
            ang2 = acos((rr2**2+d^2-rr1**2)/(2*rr2*d))
            s1 = ang1*rr1**2 + ang2*rr2**2 - rr1*d*sin(ang1)
        end if
        AreaIntersection = s1
    end

Subroutine MainModel
    integer i, j
    real*8 s, aij, dd

    !通过循环计算所有圆两两相交的面积总和
    s = 0
    do i = 1, N-1
        do j = i+1, N
            dd = sqrt((x(i)-x(j))**2 + (y(i)-y(j))**2)
            aij = AreaIntersection(R(i),R(j),dd)
            s = s + aij
        end do
    end do

    ObjectiveResult = s
End Subroutine
EndProgram;
```

模型求解难度大，可多次运行选取最好结果，也可在代码前加一句“MultiRun=5;”，这样代码可自动运行5次并可将每次计算结果保留。计算得到的最优及次优解如下。

结果：

最优解	次优解
Objective Function (Min.): 11.9207115755283 x1: 6.6036053878679 x2: 1.00000025987365 x3: 6.37055288082364 x4: 6.88526954532202 x5: 9.9999785190248 x6: 9.99996924330421 x7: 9.49996516361247 x8: 6.90201555272049 x9: 3.00000137921705 x10: 3.0000037728135 y1: 3.84079192419028 y2: 6.00166135210249 y3: 1.5 y4: 10.4999940779374 y5: 6.41969935784269 y6: 9.99996804127134 y7: 2.50000364237384 y8: 6.74559606245781 y9: 8.99998333605153 y10: 3.00000292283372	Objective Function (Min.): 12.0579470858386 x1: 5.29835637397478 x2: 11 x3: 5.62629035419212 x4: 5.58927774760338 x5: 5.56350934323857 x6: 2 x7: 2.5 x8: 2.5 x9: 9 x10: 9 y1: 8.08614278543384 y2: 5.99999999294408 y3: 10.5 y4: 1.5 y5: 5.52263657157237 y6: 6.11740919669991 y7: 2.5 y8: 9.5 y9: 9 y10: 3

次优解和最优解的目标函数值相差甚微，虽然从理论上最优解为好，但基站实际选址时还受其它众多因素影响，因此给出更多的备选方案亦是很有必要的的。

规划区域为正方形，理论上而言上述最优解和次优解对应的参数都不是唯一的，而是均还有另外三组。虽然从数学角度看各自四组解的目标函数值都一样，但因为对应的具体坐标值不同，在实际基站选址时有着重要的实际意义，因此获取全部最优和次优解对应的基站坐标值非常必要。如何获取这些多解值？最直接的方式就是多次运行，另外更简单的方式是通过坐标旋转变换仅需基于一组解就可以得到其它三组解。坐标变换公式如下：

$$\begin{aligned} x_1 &= \cos(\text{angle}) \cdot x - \sin(\text{angle}) \cdot y \\ y_1 &= \cos(\text{angle}) \cdot y + \sin(\text{angle}) \cdot x \end{aligned} \tag{4}$$

其中 x，y 表示物体相对于旋转点旋转 angle（弧度）的角度之前的坐标，x1，y1 表示物体旋转 angle（弧度）后相对于旋转点的坐标.

基于上述一组最优解和一组次优解，以及坐标旋转变换公式（4），可以很容易得到如表30-2的所有最优解和次优解对应的基站坐标值，坐标旋转变换时分别取旋转90、180和270度三种情况，变换计算时有两点需要注意，一是要将角度先转换为弧度，二是旋转90度时，原区域从第一象限变换至第二象限，此时需要将变换后的x坐标值加上正方形边长值（L=12）从而再平移回原区域的第一象限，同理旋转180度时x和y坐标值，旋转270度的y坐标值，均需加L=12。

表30.2 四组最优解及四组次优解对应坐标值（取三位小数点）

序号	坐标	最优解	次优解
1	x	6.604,1.0,6.371,6.885,10.0,10.0,9.5,6.90,3.0,3.0	5.298,11,5.626,5.589,5.564,2,2.5,2.5,9,9
	y	3.841,6.0,1.5,10.5,6.42,10.0,2.5,6.746,9.0,3.0	8.086,6.000,10.5,1.5,5.523,6.117,2.5,9.5,9,3
2	x	8.159,5.998,10.5,1.5,5.580,2.0,9.5,5.254,3.0,9.0	3.914,6.000,1.5,10.5,6.477,5.883,9.5,2.5,3,9
	y	6.604,1.0,6.371,6.885,10.0,10.0,9.5,6.902,3.0,3.0	5.298,11,5.626,5.589,5.564,2,2.5,2.5,9,9
3	x	5.396,11.0,5.629,5.115,2.0,2.0,2.5,5.098,9.0,9.0	6.702,1,6.374,6.411,6.436,10,9.5,9.5,3,3

	y	8.159,5.998,10.5,1.5,5.580,2.0,9.5,5.254,3.0,9.0	3.914,6.000,1.5,10.5,6.477,5.883,9.5,2.5,3,9
4	x	3.841,6.002,1.5,10.5,6.420,10.0,2.5,6.746,9.0,3.0	8.086,6.000,10.5,1.5,5.523,6.117,2.5,9.5,9,3
	y	5.396,11.0,5.629,5.115,2.0,2.0,2.5,5.098,9.0,9.0	6.702,1.000,6.374,6.411,6.436,10,9.5,9.5,3,3

求出各基站点坐标及对应的覆盖园半径后，还可以进行绘图展示，代码如下：

参数方程绘图代码：

```
Constant n=10;
Constant
x(n)=[6.6036053878679,1.00000025987365,6.37055288082364,6.88526954532202,9.9999785190248,9.9999
6924330421,9.49996516361247,6.90201555272049,3.00000137921705,3.0000037728135],

y(n)=[3.84079192419028,6.00166135210249,1.5,10.4999940779374,6.41969935784269,9.99996804127134,2
.50000364237384,6.74559606245781,8.99998333605153,3.00000292283372],
        r(n)=[1,1,1.5,1.5,2.0,2.0,2.5,2.5,3.0,3.0];
LineWidth = 1;
PointerSize = 2;
StepX = 50;
Variable t=[0,2*pi],xx,yy;
PlotParaFunction For(i=1:n)(xx=x[i]+r[i]*cos(t),yy=y[i]+r[i]*sin(t));
```

上述代码替换不同的基站坐标组值即可得到不同的规划布局图，见图30-3和图30-4.

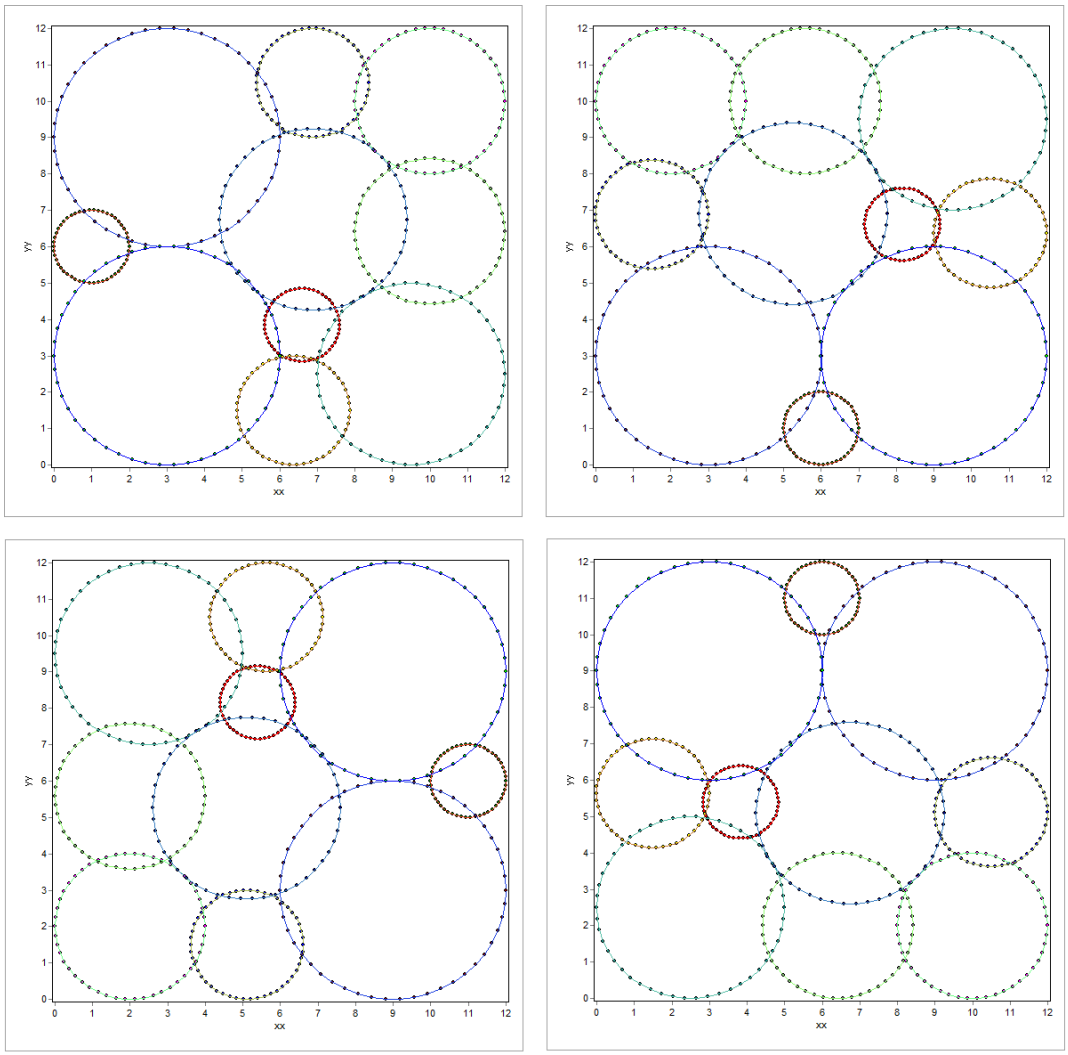

图30-3 基站布局四种最优解模式

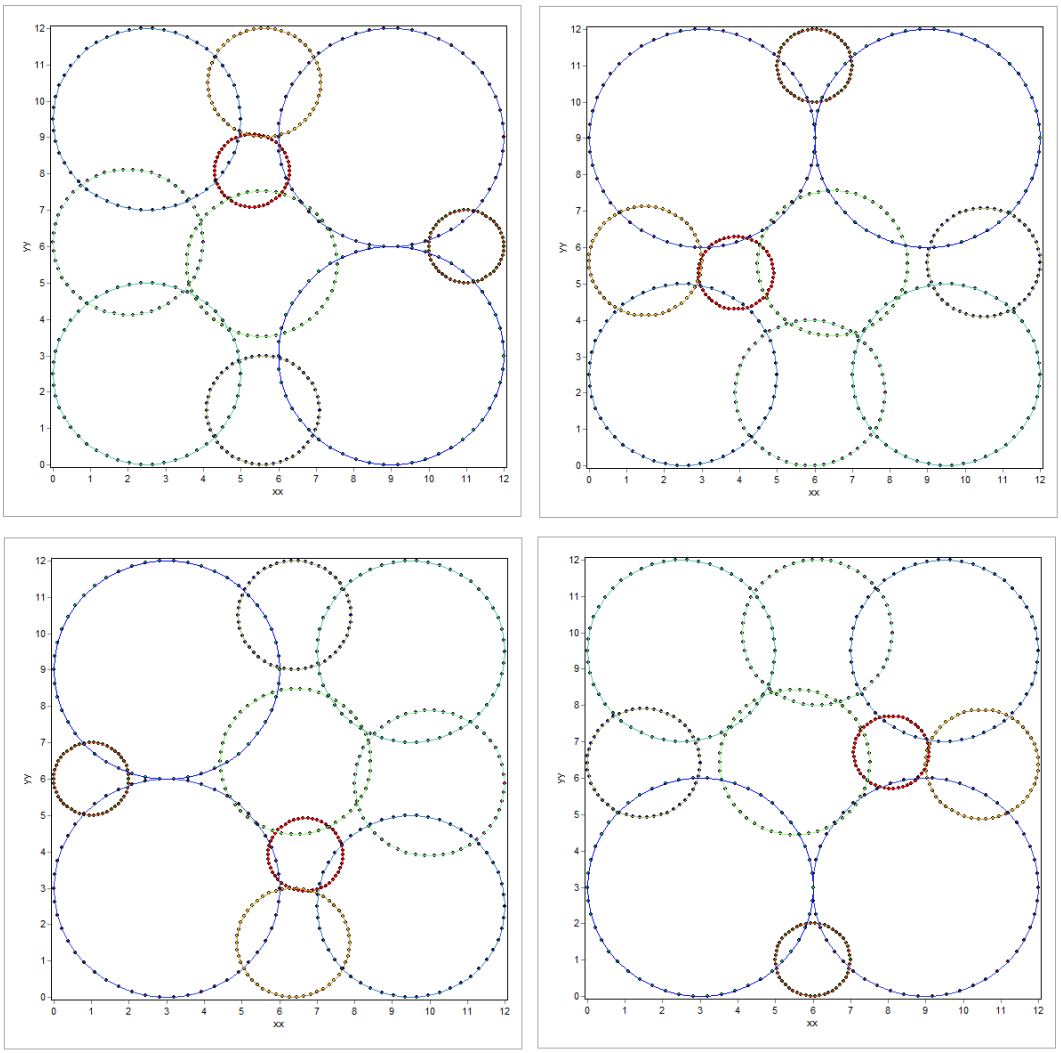

图30-4 基站布局四种次优解模式

30.4 小结

手机基站优化选址布局问题是很有现实意义的非线性优化求解问题，通过建立相关数学模型，基于自身的模型描述语言，1stOpt可以方便处理这类问题。