

# 基于网格扫描的目标点覆盖问题的分析与应用

## 摘要

覆盖问题是较基本的优化问题，其在生活、工程应用中的实例很多，本文针对全向覆盖问题和有向覆盖问题，进行了分析、建模和求解。

具体地，本文第一部分对以出租车的分配为实际背景的全向覆盖问题进行讨论，研究以最少的车辆，覆盖所有的热点。具体手段为，对区域进行离散化、网格划分，并将覆盖区域抽象为半径一定的圆形。对于这一抽象模型的求解，鉴于其作为 0-1 整数规划一般具有的 NP-hard 的性质，本文中采用了贪心算法，并在基本覆盖的模型基础上增加了概率感知、规避点两个功能拓展。

本文第二部分，则主要面向以无线传感网络为背景的有向覆盖问题，在离散化的区域基础上，以扇形作为覆盖区域的抽象模型。并对第一部分的贪心算法进行了改进，加入随机算法的思想形成随机贪心法，可以更好地避免陷入局部最优的情况，提高了算法的鲁棒性。以上两部分均给出了程序实现和仿真结果，完整代码在附录中展示。

**关键字：**目标点覆盖，概率感知模型，贪心算法，最优化

# Abstract

Coverage Problem is a basic category of Optimization Problem, with numerous applications. Based on practical cases, this paper would illustrate the analyzing, modeling and solving process for both two branches of Coverage Problems, omnidirectional and directional.

The first half of this paper would focus on the solution to omnidirectional coverage, using the assignment of taxis in city scale as its specific background. This part involves Grid-based mapping and circular region modeling. As the problem defined as 0-1 ILP (integer linear programming), which is often NP-hard, Greedy algorithm would be utilized to seek for a sub-optimal solution. Besides, two functional expansions of this problem, perception and evasion would be introduced.

The second half would ulteriorly provide an analysis and solution to the directional coverage problem, in which fan-shaped regions replace the circular ones. Furthermore, Greedy algorithm would be improved to become random greedy algorithm in order to promote robustness. All the MATLAB programs mentioned in this paper are presented in details in appendix.

**Key words:** Target coverage; Probabilistic sensing model; Greedy algorithms; Optimization

# 目录

## 第一部分 圆形覆盖问题

一、圆形问题引入 .....	4
二、理论分析 .....	5
2.1 出租车圆形覆盖 .....	5
2.2 网格模型 .....	6
2.3 二值感知模型假设 .....	7
2.4 节点部署算法 .....	8
三、模型求解 .....	9
3.1 建立模型 .....	9
3.1.1 参数说明 .....	9
3.1.2 问题的公式化 .....	10
3.2 求解结果及分析 .....	11
四、圆形覆盖问题拓展 .....	14
4.1 考虑覆盖水平问题求解 .....	14
4.1.1 概率感知模型 .....	14
4.1.2 运行结果及分析 .....	15
4.2 含规避点约束求解 .....	18
4.2.1 问题描述 .....	18
4.2.2 结果及分析 .....	19
一、有向覆盖问题的引入 .....	22

二、问题建模.....	24
2.1 参数说明.....	24
2.2 问题的公式化.....	25
三、求解方案讨论.....	27
3.1 直接求解 0-1 规划.....	27
3.2 随机贪心法求解.....	27
四、仿真分析.....	28
参考文献.....	32
小组成员分工.....	33
附录.....	34

# 第一部分 圆形覆盖问题

---

## 一、圆形问题引入

近年来，滴滴、Uber 等打车软件在越来越多的进入我们的生活，打车软件是一种智能手机应用，乘客可以便捷地通过手机发布打车信息，并立即和接单司机直接沟通，大大提高了打车效率。如今各种手机应用软件正实现着对传统服务业和原有消费行为的颠覆。

与此同时，各大打车软件之间的竞争也是如火如荼，为了抢占市场，扩大客源，如何有效的有限的让司机覆盖到亟需顾客的地点，成为了摆在各大打车公司面前的一旦难题。基于这一现状，以及我们在打车时有时会遇到的困境，提出了问题：出租车公司在杭州城区如何派遣最少的出租车实现顾客群的最大覆盖？

考虑到问题的规模和可解性，直接对于整个杭城交通情况进行覆盖和分析显然不现实。这里选取了位于武林广场附近的一部分区域作为代表，选取了人群较为集中的十个上车地点，分析在这一区域如何分布出租车能够实现数量最少的出租车对所有区域指定点的全覆盖。



图 1 武林广场附近客流点统计

## 二、理论分析

### 2.1 出租车圆形覆盖

假设出租车以自身为中心，画一个圆，如下图圆形范围内的区域均为出租车覆盖区域，圆形覆盖区域内的街道该出租车均能到达。



图 2 出租车圆形覆盖模型

## 2.2 网格模型

本次大作业中我们需要解决的问题是如何使用最少的出租车（抽象为节点）实现对呼叫客户（抽象为目标点）的覆盖。

为了方便确定节点和目标点的位置，我们将研究区域划分为若干正方形网格，则整个研究区域离散化为若干网格的集合。网格边长可根据实际情况确定，边长越小，划分越精确，得到的结果也越精确，这里为了方便描述，可设置边长为 1，根据所选区域（武林门广场）实际情况，可确定比例尺为 1: 25，在地图上划分网格结果如下：

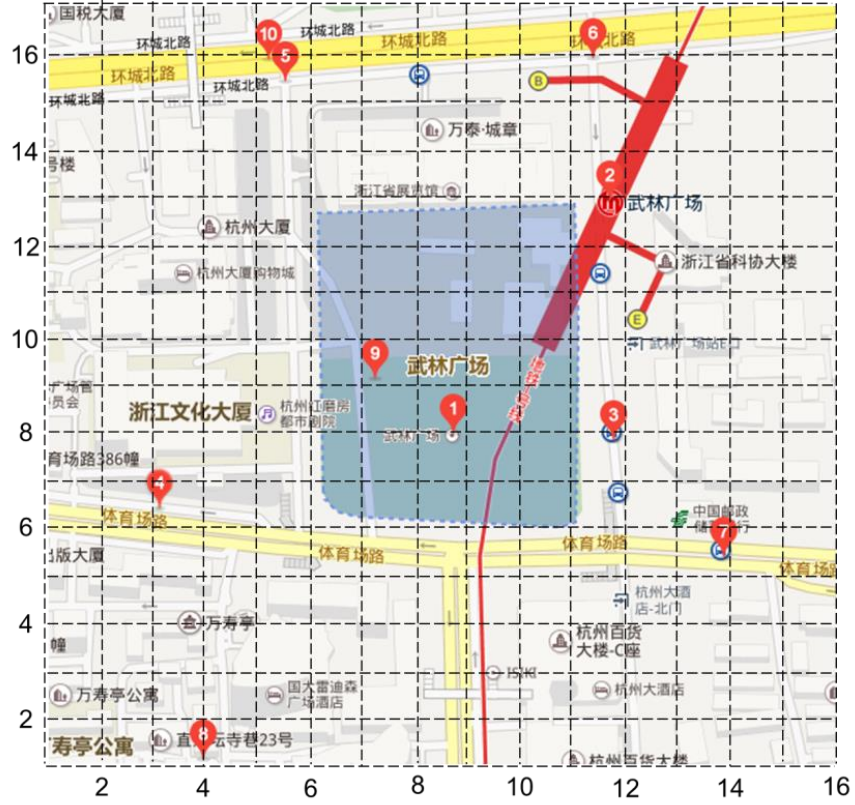


图 3 指定区域（武林广场附近）网格化结果

从地图中信息可得各客源（目标点）坐标如下（靠近目的地的格点取整）：

$$a_1 = (9,8), a_2 = (12,13), a_3 = (12,8), a_4 = (3,7), a_5 = (6,15),$$

$$a_6 = (11,16), a_7 = (14,6), a_8 = (4,1), a_9 = (7,9), a_{10} = (5,16).$$

## 2.3 二值感知模型假设

覆盖问题中的一个基本的问题就是目标点被覆盖，这里引入二值感知模型，当节点到目标点的距离小于或等于感知半径  $r$  时，我们认为该节点能够感知覆盖目标点；否则，不能感知覆盖目标点，得到以下 0-1 分布为：

$$P_{ij} = \begin{cases} 1, & \text{若 } D_{ij} \leq r \\ 0, & \text{其他} \end{cases}$$

其中， $D_{ij}$  为第  $i$  个目标点与第  $j$  个方格点之间的距离。在本例中，我们定义出租车（感知节点）的覆盖半径为 100m，即感知半径为  $r = \frac{100}{25} = 4$ 。

## 2.4 节点部署算法

在一个已知若干目标点的区域中，最保险的情况是在每个目标点处放置一个感知节点，从而实现目标全覆盖。但在本例中，为每一个客源都派遣一辆出租车是不现实并且很耗费资源的，不能实现出租车公司盈利最大化。为了实现效益最大化，这里可以采用一种节点部署算法，实现用最少的节点覆盖所有目标点，大大降低了成本。

首先，读入所有待覆盖目标点的坐标 $a_i (i = 1, 2, \dots, n)$ ；然后逐行扫描所有网格点 $b_j (j = 1, 2, \dots, m)$ ，对每个网格点计算其与各目标点之间的距离 $D_{ij} = |b_j - a_i|$ 。根据二值感知模型，记下每个网格点能够覆盖的目标点数 $c_j (j = 1, 2, \dots, m)$ 。寻找能够覆盖最多目标点的网格点，将其作为一个节点放置；当放置了一个新节点后，将能够被它感知到的目标点从待覆盖的目标点集合中剔除，以防止重复覆盖；进行新一轮的网格点扫描，以同样的方法选择下一个节点放置位置；循环上述过程，直至待覆盖目标点集合为空。

根据描述，绘制算法流程图如下：

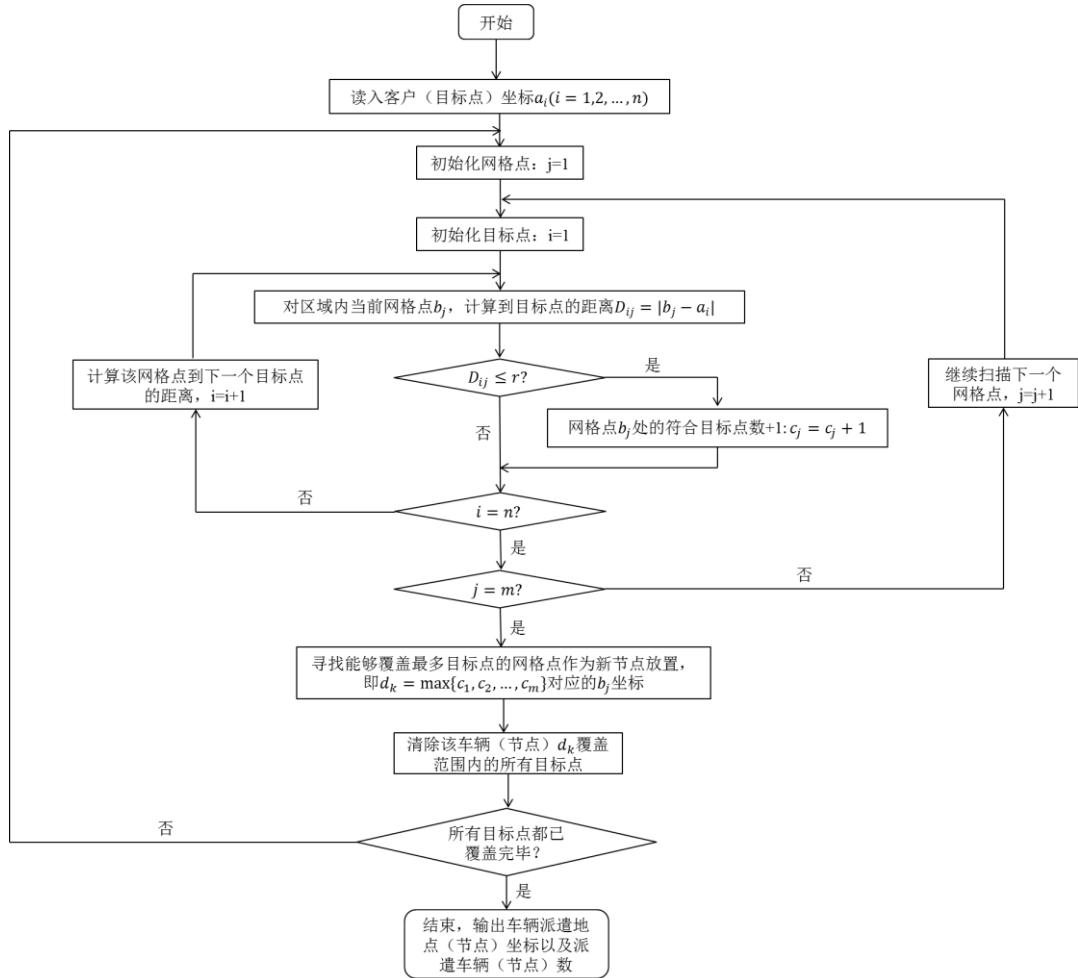


图 4 求解圆形覆盖算法流程图





备注：在本次建模中，为了使模型更加简洁，这里假设所有布置的节点均为同一型号，也即是每一个节点对应的覆盖范围都相同。同时对于目标点，我们假设其相同的重要性，在覆盖时并不因点自身性质加以区分。

### 3.1.2 问题的公式化

基于之前的网格划分，以及我们对于节点和目标点布置的定义，我们的这个布置问题实质上是一个 0-1 规划问题，首先引入决策变量。

- **决策变量**

$x_i$ ：0-1 变量，当且仅当一个节点被安装在位置  $i \in \Pi$  时取值为 1；

$\sigma_{ij}$ ：0-1 变量，当且仅当一个目标点  $j \in \Omega$  被安装在  $i$  的节点覆盖时取值为 1。

进一步我们可以得到问题的目标函数以及约束条件。

- **目标函数**

$$z = \min C_E$$

$$C_E = c \sum_{i \in \Pi} x_i$$

也即是要求我们所布置的节点的总花费最少，考虑到在我们这个问题中假设每一个布置的节点的费用都是相同的，实质上这个费用最小问题也等价于布置的节点的数目最少。

- **约束条件**

首先最重要的一个约束条件就是要判断我们的目标点是否被节点或者使用专业术语 Target In Sector (TIS) test 【引用】，即限制目标点在圆形区域半径以内。

1) 在我们的问题中，目标点  $j$  被节点  $i$  覆盖的条件可以表示为

$$rx_i \geq \|d_{ij}\| \sigma_{ij} (\forall i \in \Pi, j \in \Omega)$$

2) 每个目标点要至少被一个节点覆盖。

$$\sum_{i \in \Pi} \sigma_{ij} \geq 1 (\forall j \in \Omega)$$

3) 目标点不能被不存在的节点覆盖。

$$\sigma_{ij} \leq x_i (\forall i \in \Pi; j \in \Omega)$$

综上所述，可以得到数学模型如下：

$$\begin{aligned} z = \min & c \sum_{i \in \Pi} x_i \\ \text{s.t.} & \begin{cases} rx_i \geq \|d_{ij}\| \sigma_{ij} (\forall i \in \Pi, j \in \Omega) \\ \sum_{\{i \in \Pi\}} \sigma_{ij} \geq 1 (\forall j \in \Omega) \\ \sigma_{ij} \leq x_i (\forall i \in \Pi; j \in \Omega) \end{cases} \end{aligned}$$

### 3.2 求解结果及分析

运行程序（完整程序见附录），得到结果如下：

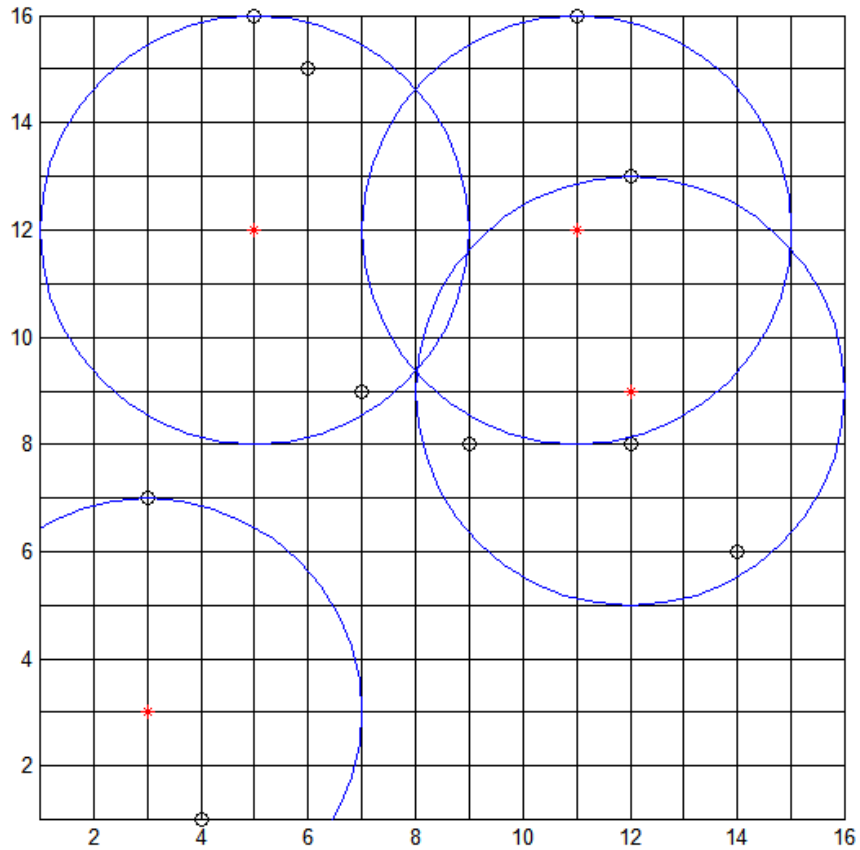


图 5 求解圆形覆盖结果

在该程序运行时, 节点的放置步骤如下:

① 计算  $16 \times 16$  网格各点到各目标点  $a_i (i = 1, 2, \dots, 10)$  的距离。全部计算过程结束后,发现当节点位于  $(12, 9)$  时,能够覆盖目标点  $a_1 = (9, 8), a_2 = (12, 13), a_3 = (12, 8), a_7 = (14, 6)$ , 可覆盖目标点数最多。因此令  $T_x(1) = 12, T_y(1) = 9$ , 即在  $(12, 9)$  处放置第 1 个节点。

② 计算16x16网格各点到剩余目标点 $a_i(i = 4,5,6,8,9,10)$ 的距离。全部计算过程结束后,发现当节点位于(5,12)时,能够覆盖目标点 $a_5 = (6,15), a_9 = (7,9), a_{10} = (5,16)$ ,可覆盖目标点数最多。因此令  $T_x(2) = 5, T_y(2) = 12$ , 即在(5,12)处放置第2个节点。

③ 计算 16x16 网格各点到剩余目标点 $a_i(i = 4,6,8)$ 的距离。全部计算过程结束后,发现当节点位于(3,3)时,能够覆盖目标点 $a_4 = (3,7), a_8 = (4,1)$ ,可覆盖目标点数最多。因此令 $T_x(3) = 3, T_y(3) = 3$ ,即在(3,3)处放置第 3 个节点。

④ 计算 16x16 网格各点到剩余目标点 $a_i(i = 6)$ 的距离。全部计算过程结束后,发现当节点位于(11,12)时,能够覆盖目标点 $a_{10} = (11,16)$ 。因此令  $T\_x(4) = 11, T\_y(4) = 12$ , 即在(11,12)处放置第 4 个节点。

节点放置顺序的表格表示如下:

表 1 基本模型节点放置顺序表

目标点 节点个数	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
1	√	√	√	×	×	×	√	×	×	×
2				×	√	×		×	√	√
3				√		×		√		
4						√				



## 四、圆形覆盖问题拓展

### 4.1 考虑覆盖水平问题求解

之前我们的基本假设里，出租车可以无差别的覆盖到圆形半径中的所有地点，但在实际生活中，由于受到距离、障碍物等各种情况的干扰，实际覆盖圆形范围内的点可到达水平是不一样的。司机显然会倾向于先去到距离自己更近点进行接客（覆盖），因此我们要对我们之前的模型进行修改。

#### 4.1.1 概率感知模型

在上述解决方案中，我们采用了二值概率模型，即只要当节点与目标点的距离小于或等于感知半径  $r$  时，我们就认为能够感知覆盖目标点，即覆盖概率为  $P_{ij} = 1$ 。其覆盖概率与二值感知模型存在一定的偏差，这里我们可引入概率感知模型，将信号强弱转化为感知概率如下：

$$P_{ij \text{ 概率}} = \begin{cases} e^{-\alpha D_{ij}}, & \text{若 } D_{ij} \leq r \\ 0, & \text{其他} \end{cases}$$

其中，参数  $\alpha$  为与车辆覆盖范围特性相关的常数，这里取 0.4； $D_{ij}$  为第  $i$  个目标点与第  $j$  个方格点之间的距离。由上式可知，当客户（目标点）与出租车（感知节点）之间的距离  $D_{ij}$  小于或等于  $r$  时，节点对目标点的感知概率为  $e^{-\alpha D_{ij}}$ ，目标点距离节点越近，被感知的概率越大。当距离大于感知半径  $r$  时，目标节点就不能被感知到了，如下图所示。



图 8 出租车的概率感知模型

与不考虑概率感知模型的算法相比,该算法在求取 16x16 网格中能覆盖最多目标点的节点位置时,不再仅仅考虑“节点覆盖的目标点个数最多”,而是考虑“节点覆盖的目标点概率总和最大”。

#### 4.1.2 运行结果及分析

运行修改后的程序（完整程序见附录“程序二：覆盖范围概率模型”），得到结果如下：

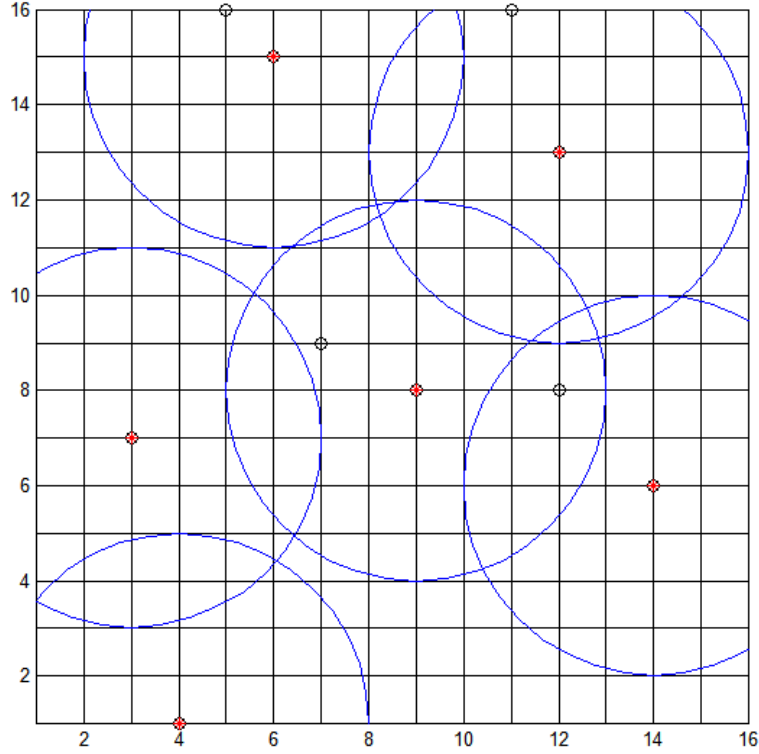


图 9 考虑覆盖水平差异下的覆盖结果

在该程序运行时，节点的放置步骤如下：

① 计算 16x16 网格各点到各目标点 $a_i(i = 1,2, \dots, 10)$ 的距离及概率。全部计算过程结束后，发现当节点位于(9,8)时，到目标点 $a_1 = (9,8), a_3 = (12,8), a_9 = (7,9)$ 的距离分别为： $0, \sqrt{5}, 3$ ，概率分别为： $1, 0.41, 0.30$ ，概率总和为 1.71，覆盖目标点概率总和最大。因此令  $T_x(1) = 9, T_y(1) = 8$ ，即在(9,8)处放置第 1 个节点。

② 计算 16x16 网格各点到各目标点 $a_i(i = 2,4,5,6,7,8,10)$ 的距离及概率。全部计算过程结束后，发现当节点位于(6,15)时，到目标点 $a_5 = (6,15), a_{10} = (5,16)$ 的距离分别为： $0, \sqrt{2}$ ，概率分别为： $1, 0.57$ ，概率总和为 1.57，覆盖目标点概率总和最大。因此令  $T_x(2) = 6, T_y(2) = 15$ ，即在(6,15)处放置第 2 个节点。

③ 计算 16x16 网格各点到各目标点 $a_i(i = 2,4,6,7,8)$ 的距离及概率。全部计算过程结束后,发现当节点位于(12,13)时,到目标点 $a_2 = (12,13), a_6 = (11,16)$ 的距离分别为:  $0, \sqrt{10}$ , 概率分别为: 1,0.28, 概率总和为 1.28, 覆盖目标点概率总和最大。因此令  $T_x(3) = 12, T_y(3) = 13$ , 即在(12,13)处放置第 3 个节点。

④ 计算 16x16 网格各点到各目标点 $a_i(i = 4,7,8)$ 的距离及概率。全部计算过程结束后，发现当节点分别位于(3,7)、(14,6)、(4,1)时，到目标点 $a_4 = (3,7), a_7 = (14,6), a_8 = (4,1)$ 的距离为 1，概率为 1。因此令  $T\_x(4) = 3, T\_y(4) = 7, T\_x(5) = 14, T\_y(5) = 6, T\_x(6) = 4, T\_y(6) = 1$ ，即在(3,7)、(14,6)、(4,1)处分别放置第 4、5、6 个节点。

节点放置顺序的表格表示如下:

表 2 考虑覆盖水平下的节点放置结果

[illegible]



节点放置顺序的图示表示如下：

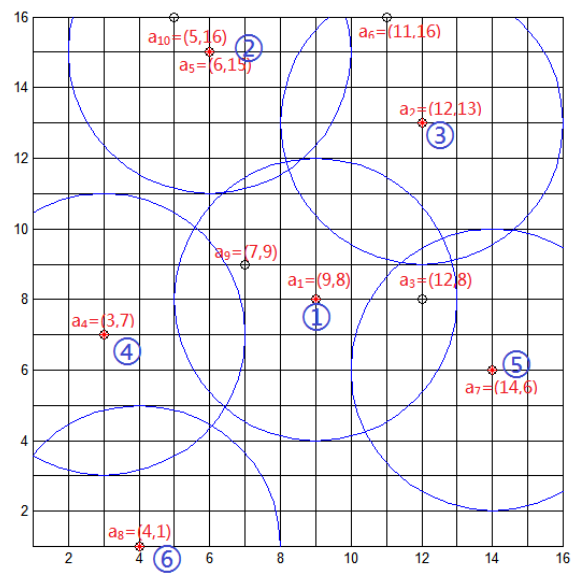


图 10 考虑覆盖水平差异下的节点覆盖图示

则可得到武林广场地区出租车派遣情况如下：

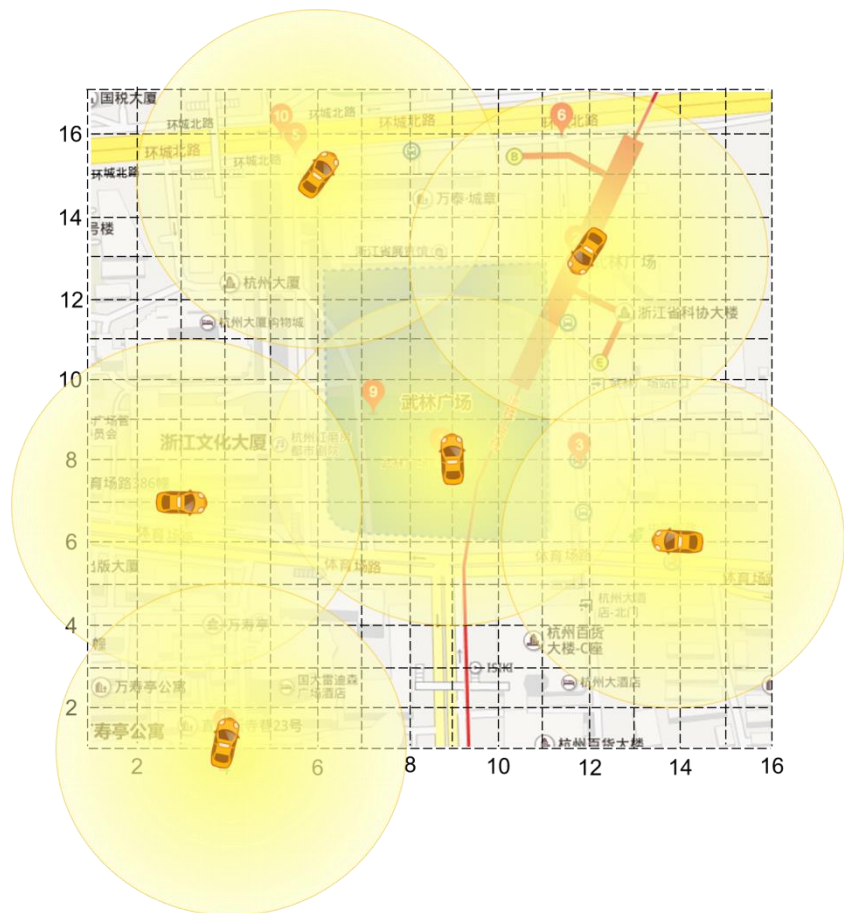


图 11 实际地区目标覆盖情况（概率感知模型）

观察上图可以发现，在该派遣方案下能够实现顾客群（目标点）的全覆盖，最少需派遣出租车 6 辆，且这 6 辆出租车几乎与某些目标点位置重合（如第 1/5/2/4/7/8 目标点），其余的 4 个目标点也距离出租车较近，能够以较大概率实现覆盖。

由此可得，同样的待覆盖目标点，位置也相同的情况下，由于概率模型的引入，需要放置的节点个数增多，并且节点距离目标点更近。

## 4.2 含规避点约束求解

### 4.2.1 问题描述

武林广场位于杭州市的中心地段，在客流高峰期可能会遇到堵车等情况，从而造成道路交通的不便。出租车公司为了盈利最大化，应尽量避免向堵车地段派遣出租车拉客，由此我们可引申出规避点问题，即在感兴趣区域中设置节点时，其覆盖区域应避开某些特定的点，在此约束条件下继续研究出租车公司的最优化派遣策略。

根据资料，武林广场地区主干道分布密度适中，但次干道与支路的网络系统性较为薄弱。道路段、窄且不连贯，不能有效缓解主干道的交通压力。早晚高峰时期武林广场附近的拥堵点包括武林广场西湖湖滨地带、河坊街吴山广场、延安路、环城北路、体育场路等。由此可抽象出规避点坐标为

$$n_1 = (9,5), n_2 = (8,16), n_3 = (15,5)$$

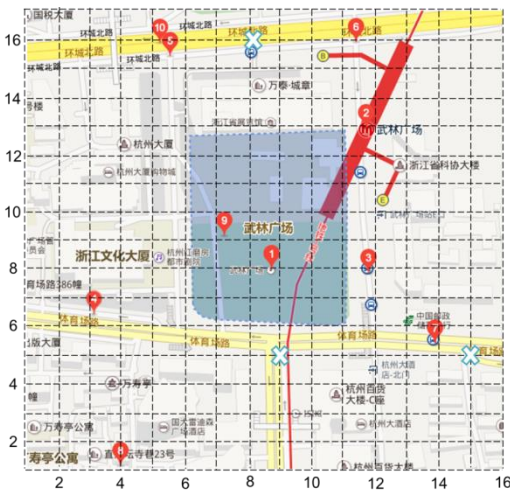


图 12 指定区域（武林门广场附近）规避点标注

与前两个算法相比，该算法在求取 16x16 网格中能覆盖最多目标点的节点位置时，不仅仅考虑“节点覆盖的目标点概率总和最大”，更要考虑前提条件，即节点的覆盖区域不包括 3 个规避点。

### 4.2.2 结果及分析

根据以上分析编程（完整程序见附录“程序三：规避点模型”），求解得到覆盖结果如下：

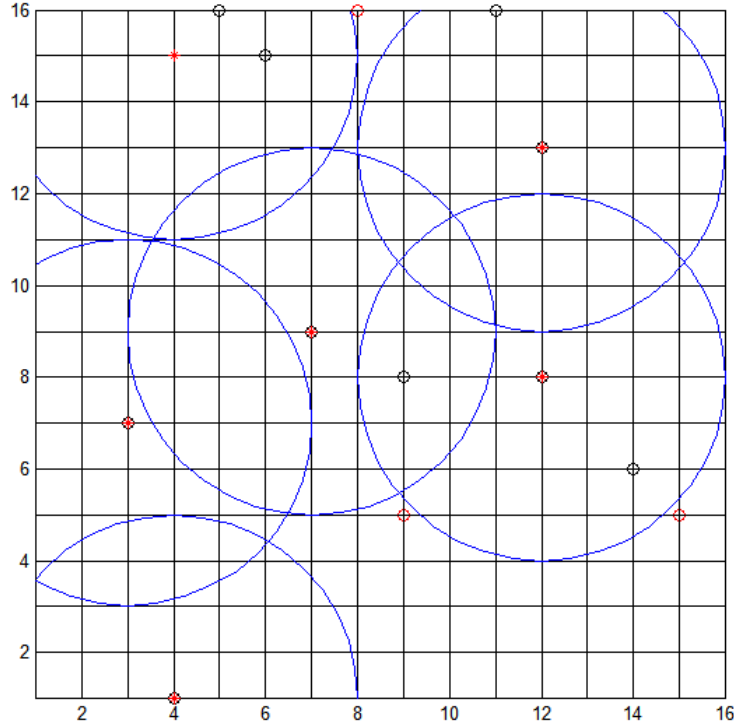


图 13 考虑规避点的覆盖结果

在该程序运行时，节点的放置步骤如下：

① 计算  $16 \times 16$  网格各点到各目标点  $a_i (i = 1, 2, \dots, 10)$  的距离及概率。全部计算过程结束后，发现当节点位于  $(12, 8)$  时，覆盖区域不包括规避点，到目标点  $a_1 = (9, 8)$ ,  $a_3 = (12, 8)$ ,  $a_7 = (14, 5)$  的距离分别为：3,  $0, 2\sqrt{2}$ ，概率分别为：0.30, 1, 0.32，概率总和为 1.62，覆盖目标点概率总和最大。因此令  $T_x(1) = 12$ ,  $T_y(1) = 8$ ，即在  $(12, 8)$  处放置第 1 个节点。

② 计算  $16 \times 16$  网格各点到各目标点  $a_i (i = 2, 4, 5, 6, 8, 9, 10)$  的距离及概率。全部计算过程结束后，发现当节点位于  $(12, 13)$  时，覆盖区域不包括规避点，到目标点  $a_2 = (12, 13)$ ,  $a_6 = (11, 16)$  的距离分别为：0,  $\sqrt{10}$ ，概率分别为：1, 0.28，概率总和为 1.28，覆盖目标点概率总和最大。因此令  $T_x(2) = 12$ ,  $T_y(2) = 13$ ，即在  $(12, 13)$  处放置第 2 个节点。

③ 计算  $16 \times 16$  网格各点到各目标点  $a_i (i = 4, 5, 8, 9, 10)$  的距离及概率。全部计算过程结束后，发现当节点位于  $(4, 15)$  时，覆盖区域不包括规避点，到目标点  $a_5 = (6, 15)$ ,  $a_{10} = (5, 16)$  的距离分别为：2,  $\sqrt{2}$ ，概率分别为：0.45, 0.57，概率总和为 1.02，覆盖目标点概率总和最大。因此令  $T_x(3) = 4$ ,  $T_y(3) = 15$ ，即在  $(4, 15)$  处放置第 3 个节点。

④ 计算 16x16 网格各点到各目标点 $a_i(i = 4,8,9)$ 的距离及概率。全部计算过程结束后，发现当节点分别位于(3,7)、(4,1)、(7,9)时，到目标点 $a_4 = (3,7), a_8 = (4,1), a_9 = (7,9)$ 的距离为 1，概率为 1。因此令  $T_x(4) = 3, T_y(4) = 7$ 、 $T_x(5) = 4, T_y(5) = 1$ 、 $T_x(6) = 7, T_y(6) = 9$ ，即在(3,7)、(4,1)、(7,9)处分别放置第 4、5、6 个节点。

节点放置顺序的表格表示如下：

表 3 考虑规避点情况下的节点放置结果

目标点 节点个数	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
1	√	×	√	×	×	×	√	×	×	×
2		√		×	×	√		×	×	×
3				×	√			×	×	√
4				√				×	×	
5								√	×	
6									√	

节点放置顺序的图示表示如下：

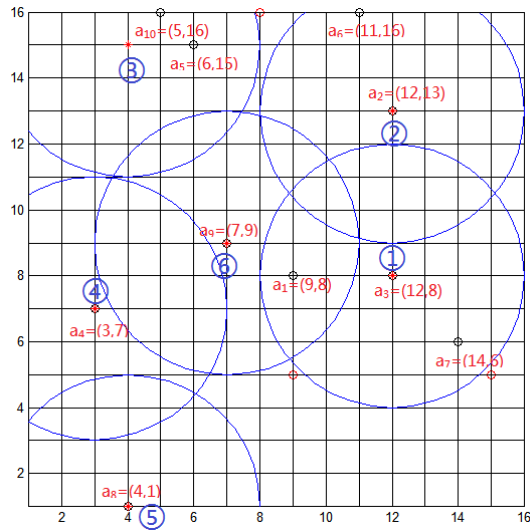


图 14 加入待规避点的节点覆盖结果

则可得到武林广场地区出租车派遣情况如下：

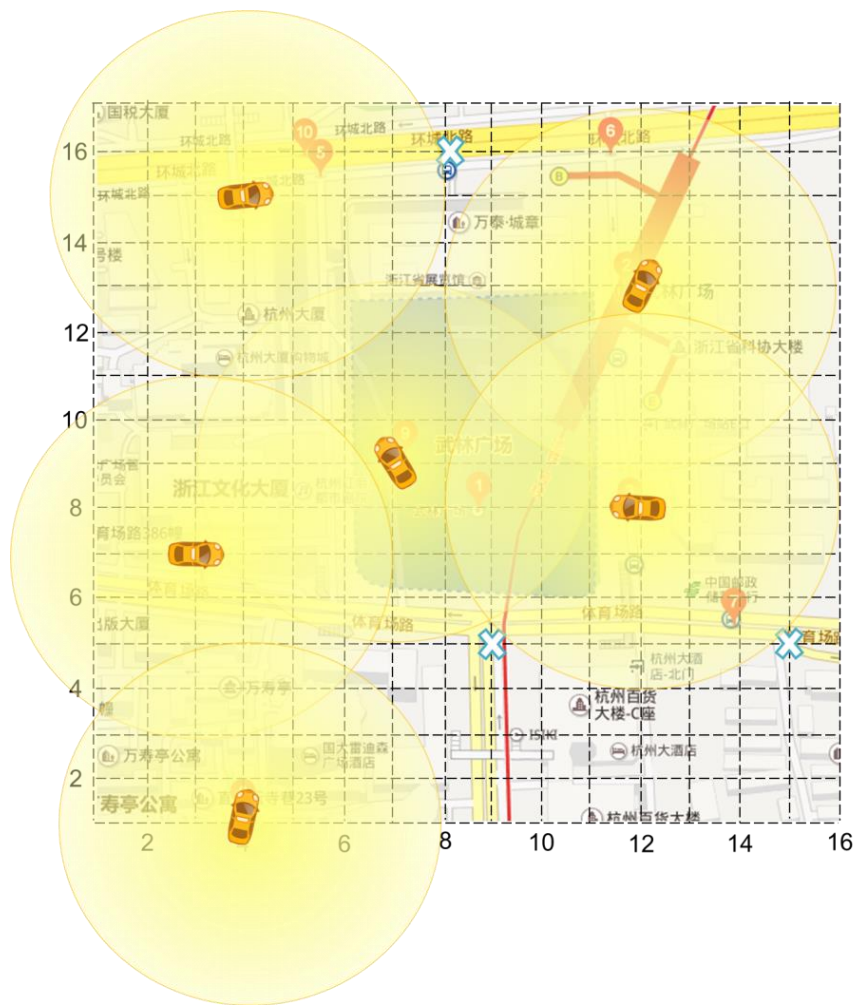


图 15 实际地区目标覆盖情况（规避点+概率感知模型）

观察上图可以发现，在该派遣方案下能够实现顾客群（目标点）的全覆盖，最少需派遣出租车 6 辆，且这 6 辆出租车几乎与某些目标点位置重合（如第 3/2/4/8/9 目标点），其余的 5 个目标点也距离出租车较近，能够以较大概率实现覆盖。另外，对于 3 个拥堵路口（规避点）也实现了较好的规避。

# 第二部分 扇形覆盖问题

---

## 一、有向覆盖问题的引入

在第一部分的问题中,主要讨论了基于圆形覆盖这一基本模型的一些生活实际问题的建模和求解方案。实际上,当更深入地看待这一问题,圆形的覆盖问题在本质上是一个全向覆盖(Omnidirectional Coverage)的问题,在实际的应用场景中,可以找到出租车这一实际的问题与其相匹配,而相似的,全向覆盖模型还可以解决很多诸如喷灌,通信基站的覆盖等实际问题。

然而,全向覆盖模型也有其局限性,在许多实际模型中,方向性是不可避免地存在的,在这类情况下,覆盖区域形式的变化,会带来模型约束上的复杂化,成为有向覆盖问题(Directional Coverage),以下将和第一部分一样,从一些实际模型入手进行分析。

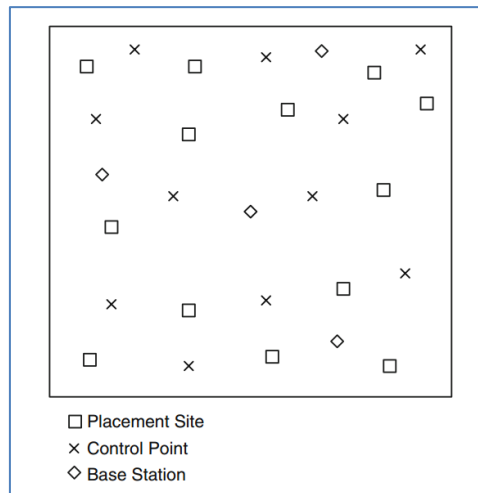
实际工程应用中,摄像头、有向天线、雷达等设备由于构造和其他因素的影响,最终抽象出的模型,一般是一个从中心点出发,半径一定或不等,并且仅限于一定张角内的一个覆盖区域,比较接近于三角或扇形的抽象形式。例如监视器监控一定的区域,监控器镜头所能覆盖的范围,一般假设为一个扇形区域。

另一方面,在一些应用情况下,不仅仅是物理模型的约束造成了有向的覆盖区域形式。有的情况下,需要被覆盖的节点的分布,比较集中或是有一定规律,若构建全向的区域,则会造成浪费,比如在高速公路上常见的测速雷达,每个测速点对各个方向都发射测速信号是显然没有意义的,因为车辆都在道路内定向行驶,只要捕捉固定区域就可以检测所有车辆,覆盖其他区域没有意义。这时就有必要提供有向的覆盖区域,达到同样的效果时,提高覆盖的效率,节约开支。





示例 1 美国导弹远程预警雷达覆盖



示例 2 无线可充电传感网络覆盖问题

综上，对于有向覆盖问题的建模和求解是十分有必要进行研究的，因此我们将基于抽象程度比较高的，相对而言比较基本的扇形覆盖模型进行分析。进一步探究以下问题：

在给定的离散坐标区域内，以扇形作为单位覆盖区域，覆盖一定数量的随机分布的结点，并且尽可能得到一个成本最低的覆盖方案。

以下部分将以参数定义、模型建立、规划问题分析和求解方案讨论和实施的顺序进行阐述。

注：以下的语境为 示例 2“无线可充电传感网络覆盖问题”。

## 二、问题建模

### 2.1 参数说明

符号	含义	备注
$(x_i, y_i)$	发射器的二维平面坐标	由网格点确定坐标点
$v_{id}$	表征发射器覆盖区域方向的单位向量，沿扇形角平分线	本次设为了 4 个离散方向 (1,0),(0,1),(-1,0),(0,-1)
$\varphi_i$	发射器覆盖范围（扇形）的张角	假设所有发射器此参数相同
$r_i$	发射器覆盖范围（扇形）的半径	所有发射器此参数相同
$c_i$	每一个发射器的成本	假设各个发射器节点的布置成本相同均为 c
$C_E$	一次布置结果所有发射器成本的和	
$\Pi$	发射器节点位置集合	一个覆盖的集合
$D$	发射器节点的方向集合	$v_{id}$ 的 4 个可能值组成
$d_{ij}$	从发射点 i 指向待覆盖点 j 的向量	
$\Omega$	待覆盖节点的集合	

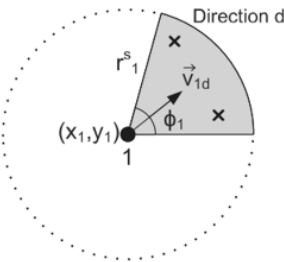
- 前提假设

在本次建模中，为了使模型更加简洁，这里假设：

- 1) 所有布置的发射器节点均为同一型号，也即是每一个发射节点对应的覆盖范围都为相同形式的扇形。
- 2) 同时对于等待覆盖的部署节点，我们假设其具有相同的重要性，在覆盖时不分先后顺序。

- 模型抽象

针对每一个发射器节点的覆盖区域图如下：





## 2.2 问题的公式化

基于之前的网格划分，以及我们对于发射器和接收点布置的定义，我们的这个布置问题实质上是一个 0-1 规划问题，首先引入决策变量。

- 决策变量

- $x_i^d$ :
- 0-1 变量，当且仅当一个发射器被安装在位置  $i \in \Pi$ ，且方向指向  $d$  方向时取值为 1；
- $\sigma_{ij}^d$ :
- 0-1 变量，当且仅当一个待覆盖节点  $j \in \Omega$  被安装在  $i$  且指向为  $d$  的发射器覆盖时取值为 1。

基于以上参数和决策变量，进一步我们可以得到问题的目标函数以及约束条件的公式化描述。

- 目标函数

$$z = \min C_E$$
$$C_E = c \sum_{i \in \Pi} x_i^d$$

也即是要求所布置的发射器节点的总花费最少，考虑到这个问题中假设每一个布置的发射器节点的费用都是相同的，实质上这个费用最小问题也等价于布置的发射器节点的数目最少。

- 约束条件

- 1) 首先最重要的一个约束条件就是要判断我们的待覆盖点是否被部署发射器节点，也即 Target In Sector (TIS) test。

在我们的问题中，待覆盖节点  $j$  被发射器节点  $i$  覆盖的条件可以表示为

$$r \sum_{\{d \in D\}} x_i^d \geq \|d_{ij}\| \sum_{\{d \in D\}} \sigma_{ij}^d (\forall i \in \Pi, j \in \Omega)$$
$$\phi x_i^d \geq 2 \cos^{-1} \left( \frac{\mathbf{v}_{id} \bullet \mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|_2} \right) \sigma_{ij}^d (\forall i \in \Pi; j \in \Omega; d \in D)$$

以上第一个式子，限制待覆盖节点在扇形区域半径以内；第二个式子限制待覆盖节点在扇形角度以内。

- 2) 每个待覆盖节点要至少被一个传感器覆盖,也即是考虑来自待覆盖节点各个方向的各个可能的发射节点,至少有一个覆盖了该节点。

$$\sum_{\{i \in \Pi\}} \sum_{\{d \in D\}} \sigma_{ij}^d \geq 1 \quad (\forall j \in \Omega)$$

- 3) 待覆盖节点不能被不存在的发射器覆盖

$$\sigma_{ij}^d \leq x_i^d \quad (\forall i \in \Pi; j \in \Omega; d \in D)$$

- 4) 考虑到实际空间的有限,在每一个位置,至多只有一个发射器被安装

$$\sum_{\{d \in D\}} x_i^d \leq 1 \quad (\forall i \in \Pi)$$

综上所述,可以得到数学模型如下:

$$\begin{aligned} z = \min c \sum_{i \in \Pi} x_i^d \\ \text{s.t.} \begin{cases} r \sum_{\{d \in D\}} x_i^d \geq \|d_{ij}\| \sum_{\{d \in D\}} \sigma_{ij}^d \quad (\forall i \in \Pi, j \in \Omega) \\ \phi x_i^d \geq 2 \cos^{-1} \left( \frac{\mathbf{v}_{id} \bullet \mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|_2} \right) \sigma_{ij}^d \quad (\forall i \in \Pi; j \in \Omega; d \in D) \\ \sum_{\{i \in \Pi\}} \sum_{\{d \in D\}} \sigma_{ij}^d \geq 1 \quad (\forall j \in \Omega) \\ \sigma_{ij}^d \leq x_i^d \quad (\forall i \in \Pi; j \in \Omega; d \in D) \\ \sum_{\{d \in D\}} x_i^d \leq 1 \quad (\forall i \in \Pi) \end{cases} \end{aligned}$$

## 三、求解方案讨论

### 3.1 直接求解 0-1 规划

对于模型中描述的规划问题，很明显地可以定性为 0-1 整数规划问题，这类问题形式很简单，但是一般来说是 NP-hard 问题，其大规模的问题，当前的认识是不存在精确的算法确保其能求出最优解。

对此，我们先进行了一定的验证，考虑的方法是将，利用标准的模型  $Ax \leq b$  来表达问题，并代入 MATLAB 中现成的求解器进行求解。在这一过程中，重点在于将约束条件转化为系数矩阵 A 的各行，随后代入求解。将约束条件转化为系数矩阵的程序可以在附录中查到，在此不多赘述。实际运行时，发现即使对于一个很小的 7\*7 区域，安置 5 个目标结点，其系数矩阵的维度会达到 2259\*1176 的水平，并且是一个巨大稀疏矩阵。而 MATLAB 中使用的默认算法为 Branch Bound（分支定界法求解），其计算量随着问题规模的上升很明显，进行长时间的运行也无法得到结果。

一方面，直接按照常规的 0-1 规划问题的求解方式无法得到精确解，另一方面，鉴于在圆形覆盖问题中，贪心算法已经能得到比较好的效果，因此考虑使用这一种相对比较简单启发式算法来求解次优解。以下进行简要说明。

### 3.2 随机贪心法求解

在分析圆形覆盖问题的时候，实质上是使用了贪心法作为启发式的求解方案，每次扫描过后总是做出在当前看来是最好的选择，也即是在一次扫描之后首先选取出能够覆盖最多的点的圆形，之后在剩余的点里继续进行搜索，这样的方法，所做出的是次优解而不一定是最优解。

作为一种启发式算法，贪心算法不是对所有问题都能得到整体最优值，首先应用贪心策略的问题若要得到最优值，需要问题本身具备无后效性，即某个状态以前的过程不会影响以后的状态，只与当前状态有关。显然在此次的有向覆盖问题中，这一条件是不满足的，贪心法得到的结果不是最优解，若待覆盖点出现一些病态的分布时，贪心法很可能掉入一个局部最优，最终不可能达到全局最优。

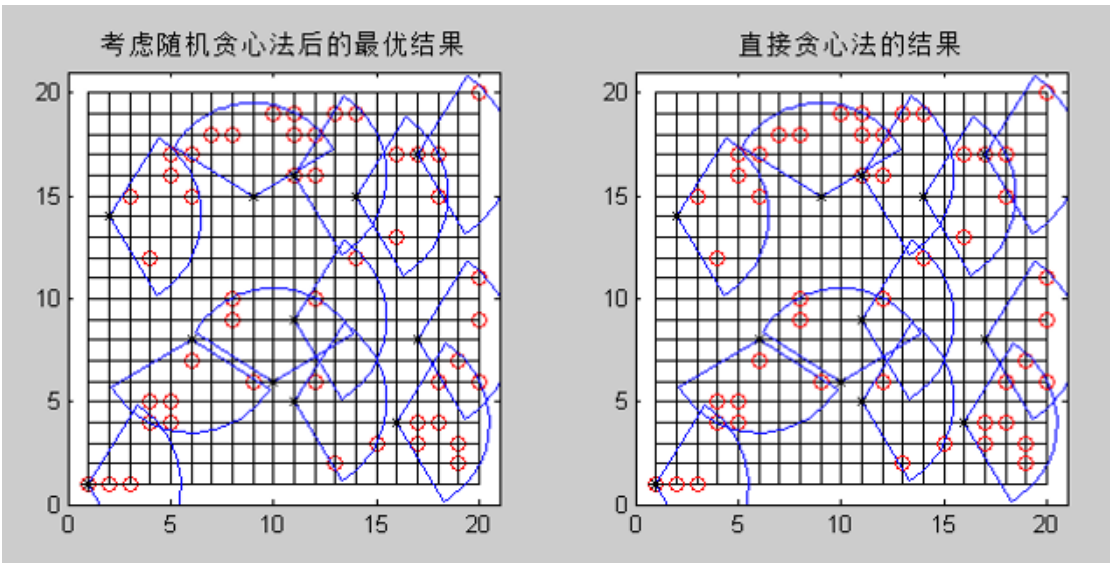
针对这一这一问题，在基础的贪心法上加入了随机算法的改进，也就是随机贪心法，这种随机算法和启发式算法结合的方法也是现存算法中很常见的，可以一定程度上增加在数据结构不理想时的鲁棒性。

随机的改进，具体而言，是每次不直接选取覆盖点数最多的位置安置发射器。根据覆盖点数给每个位置，赋予一个被选中的概率值，覆盖最多结点的扇形区域，在最终的结果中最有可能出现，但不是一定出现。在随机的基础上，要加入多次循环才能起到效果。每一次针对一个具体问题，循环运行多次随机贪心算法（实验中为 20 组），并且其第 1 次直接使用基本贪心算法，而后的情况都与这一基本优化结果相比较，每次都选出使用发射点最少的当前最优覆盖结果。

## 四、仿真分析

以下为对于不同的条件下随机贪心算法的运行结果。（随机贪心算法程序详见附录四）

1. 区域面积：20\*20 随机生成 50 个待覆盖点散布在图中，随机贪心算法循环 20 次，其结果如下：

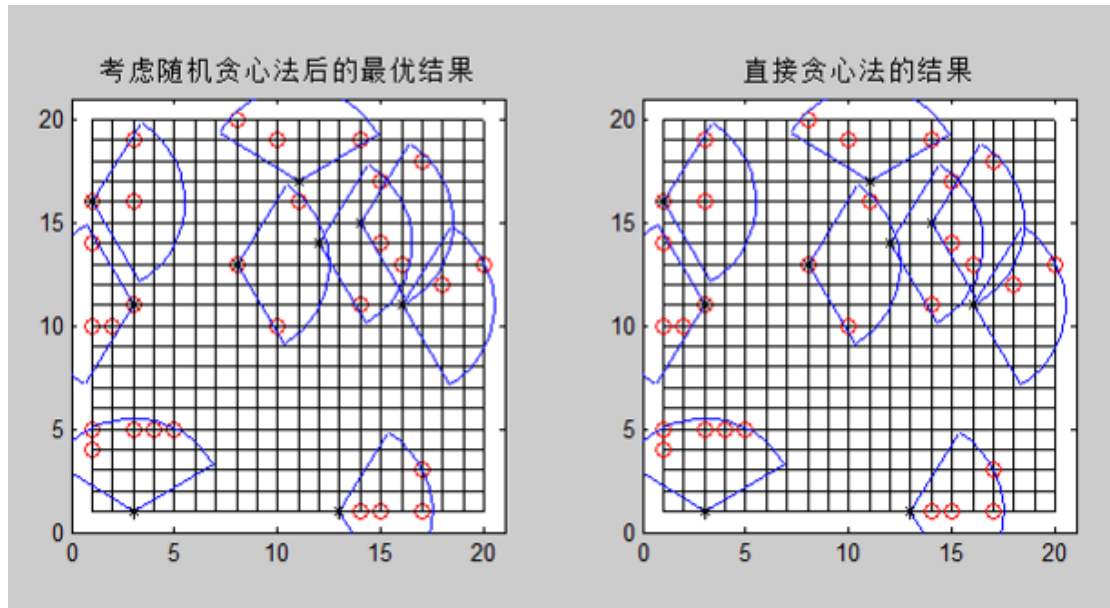


以上可以看出，最终的最佳结果和基本的贪心算法是一致的，其布置方案也相同。说明了两个可能性：

其一，这一数据集比较适合于贪心法的策略，可以达到比较好的一个次优解，随机贪心法对于贪心法没有提升；

其二，随机贪心法的循环次数还不够，没能够达到充足的随机情况，因而没有得到明显的效果。

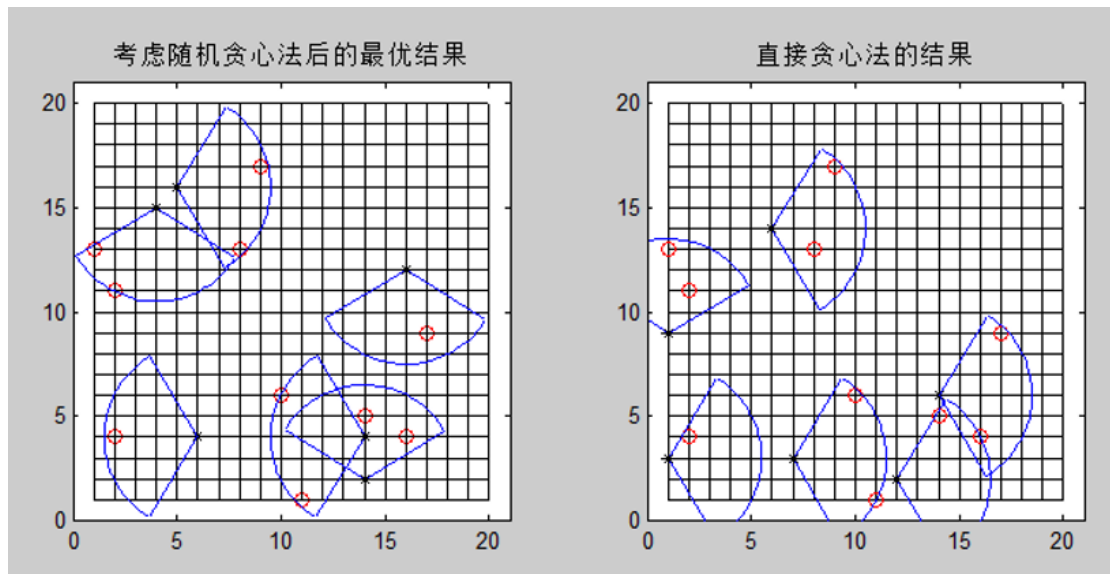
2. 区域面积:  $20 \times 20$  随机布置 30 个待覆盖的节点, 结果如下:



更改了数据集的结构之后的效果, 还是出现循环过程的最佳结果和基本贪心法一致的情况。

类似的数据集结构调整还做了多次测试, 都得到了相近的效果, 说明一方面本次的问题的数据集的形式比较适合于贪心法的求解, 另一方面也说明如果要得到随机贪心法对于贪心法有所优化的情况, 可能需要提高循环的次数。

3. 区域面积：20\*20 随机布置 10 个待覆盖的节点，结果如下：

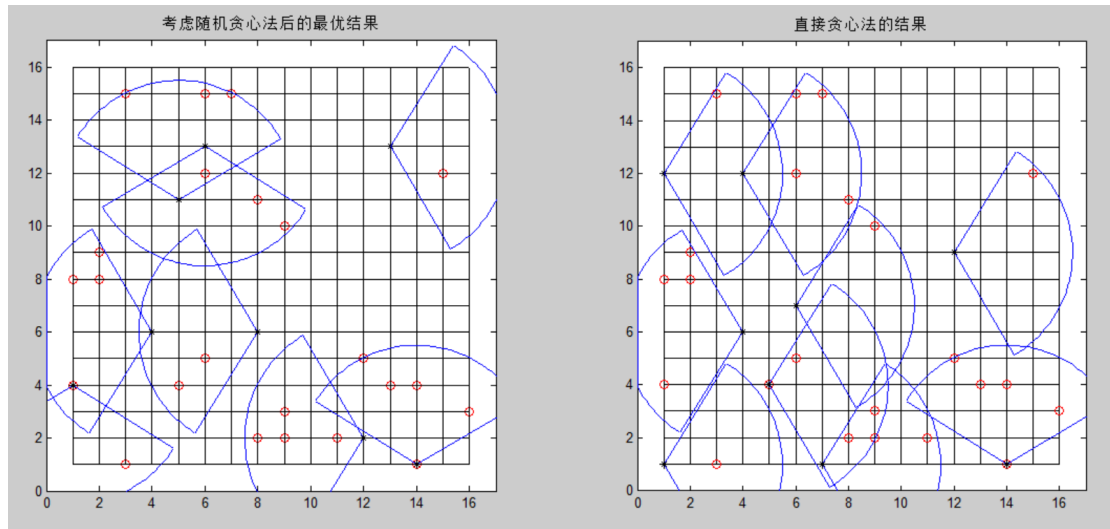


此时考虑随机贪心法后运行的结果需要 6 个扇形才能覆盖，贪心法得到的结果也是 6 个扇形。随机贪心法没有取得优化效果。

但是值得注意的是，在这里，同样的数量下，覆盖方案有了多样性，在随机贪心法得到的结果中，发射器的方向和位置分布明显不同于基本的贪心法，也就是说对于同一个随机分布的序列点，可能有多种最优（次优）的覆盖方案。

可以大致推出点的密度越大，分布的越均匀，贪心法就越可能得到最优解，因为此时点的分布较好，不会陷入病态的局部最优。

4. 区域大小为  $16*16$ ，结点数为 25，循环次数为 50 次。



在以上的结果中，随机贪心法的优化效果有了体现，其中左图的随机贪心法在 50 次循环之后，得到了一个只需要 8 个发射器的覆盖集，而基本贪心法需要 9 个发射器，说明随机贪心法**避免了一次陷入局部最优**的情况，得到了应有的效果。

不过在多次的测试中，提高随机循环的次数在大部分情况下也不会得到优化的结果，说明本次问题的数据分布结构的确比较适合贪心法的求解。

## 参考文献

- [1] 郭秀明,赵春江,杨信廷,孙传恒,李明,李文勇,周超. 基于网格扫描的实现目标点覆盖的确定性传感器节点部署方法[J]. 传感技术学报,2012,01:104-109.
- [2] 张梦樱,唐乾刚,韩小军,张青斌,葛健全. 复杂约束条件下的再入轨迹迭代求解方法[J]. 兵工学报,2015,06:1015-1023.
- [3] 范高俊. 无线传感器网络覆盖性能评估与提高[D].国防科学技术大学,2009.
- [4] 崔笑川. 最小点覆盖近似算法及其应用研究[D].兰州交通大学,2015.
- [5] 戴海鹏, 陈贵海, 徐力杰,等. 一种高效有向无线充电器的布置算法[J]. 软件学报, 2015(7):1711-1729.
- [6] Liao JH, So WT, Jiang JR. Optimized charger deployment for wireless rechargeable sensor networks. 2013. [http://in1.csie.ncu.edu.tw/~jrjiang/publication/wasn2013\\_submission\\_77.pdf](http://in1.csie.ncu.edu.tw/~jrjiang/publication/wasn2013_submission_77.pdf)
- [7] Ai J, Abouzeid AA. Coverage by directional sensors in randomly deployed wireless sensor networks. Journal of Combinatorial Optimization, 2006,11(1):21-41.
- [8] 赵海滨. MATLAB应用大全[M]. 清华大学出版社, 2012.
- [9] “杭州城市道路拥堵成因与对策分析”  
[http://www.360doc.com/content/13/0905/15/3558410\\_312418258.shtml](http://www.360doc.com/content/13/0905/15/3558410_312418258.shtml)
- [10] “杭州市区主拥堵点统计”<http://www.chinanews.com/sh/2013/12-19/5638140.shtml>
- [11] “杭州市武林广场及周边区域分析”  
[http://wenku.baidu.com/link?url=YZBb--WFqPqpuEb5WYGxphkB\\_IQeeI7JNbgC85BrQYuzldgDrPRcjGuZqtXUeh\\_yONrcEwdcf8oCW8cp5bIECpnoW-8hhFp00bCY0iTFZG](http://wenku.baidu.com/link?url=YZBb--WFqPqpuEb5WYGxphkB_IQeeI7JNbgC85BrQYuzldgDrPRcjGuZqtXUeh_yONrcEwdcf8oCW8cp5bIECpnoW-8hhFp00bCY0iTFZG)



## 小组成员分工

任务	完成人
第一部分圆形覆盖建模	
圆形覆盖全部程序实现	
圆形覆盖报告撰写	
第二部分扇形覆盖建模	
扇形覆盖全部程序实现	
扇形覆盖报告撰写	
报告文档整合	
展示 PPT 制作	

## 附录