

张剑如组成员分工：

张剑如：论文 辅助建模

黄智宇：建模 辅助编程

王嘉豪：编程

## 摘要

警察在人们的生产生活一直担负着重要的责任，但是警务资源是有限的，所以合理的安排交巡警服务平台，分配好警力资源对社会的安定、人们的安全都起着至关重要的作用。本文接合图论的相关知识，建立数学模型，对相关问题进行分析与讨论。

对于问题一的第一个子问题：分配交巡警服务平台管辖范围。运用图论中的 Floyd 算法建立最短路径模型，得到最短路径矩阵，并且在 3 分钟的时间约束条件下，按照就近分配原则给出了交巡警服务平台管理范围的最优分配方案。针对第二个子问题：快速封锁 A 区的平台调度问题，我们采用了最小代价的最大匹配的思想，根据图论中基本原理，建立二分图匹配，使用匈牙利算法解得所有可行解，最后利用二分搜索，找到最优调度方案，得出最短封锁时间为 8.015 分钟。考虑最后一个子问题需要增设交巡警服务平台，我们以 3 分钟内有交巡警到达事发地为约束条件建立 0-1 规划模型，使得每一个服务平台的案发率尽量接近总的案发率平均值，即与案发率平均值之间的差距总和最小化，循环实验，最终确定增加标号为 29、40、48、89 的节点作为新设立平台。

针对问题二，本文从全市整体出发，分别对六个地区做了一个简单的模糊综合评价，评价的三个指标分别确立为交巡警三分钟能够到的路口数最多、每个交巡警台平均服务人数、每个交巡警台平均每天处理案件数。通过计算每一个地区的分数，发现 D 区分数最低，原因是交巡警服务平台无法及时到达的节点过多。针对这一不合理现象，通过与问题一的第三小问相似的模型得出新增平台的位置，最终确定 330、333、370、344、362 为新增设平台，增设后交巡警服务平台三分钟的可到达率为 96%，相较优化前的 72.22%，有了显著的提升，使得 D 区平台设置趋于合理。对于怎样选择最佳围堵方案，采用图论中的广度优先算法，找到嫌疑犯所有可能的逃跑路线，比较该节点所辖交巡警服务平台与嫌疑犯到达该节点所用时间，若警车先到，则可以完成对线路的封锁，否则继续重复上述步骤进行封锁。直到得到最优围堵方案：在假设嫌疑犯的逃逸速度为 60km/h 的前提下，求得最优围堵时间为 8.979 分钟。

**关键词：**Floyd 算法，广度优先算法，0-1 规划模型，匈牙利算法，二分搜索

## 1 问题重述

“有困难找警察”，是家喻户晓的一句流行语。警察肩负着刑事执法、治安管理、交通管理、服务群众四大职能。为了更有效地贯彻实施这些职能，需要在市区的一些交通要道和重要部位设置交巡警服务平台。每个交巡警服务平台的职能和警力配备基本相同。由于警务资源是有限的，如何根据城市的实际情况与需求合理地设置交巡警服务平台、分配各平台的管辖范围、调度警务资源是警务部门面临的一个实际课题。

试就某市设置交巡警服务平台的相关情况，建立数学模型分析研究下面的问题：

(1) 附件 1 中的附图 1 给出了该市中心城区 A 的交通网络和现有的 20 个交巡警服务平台的设置情况示意图，相关的数据信息见附件 2。请为各交巡警服务平台分配管辖范围，使其在所管辖的范围内出现突发事件时，尽量能在 3 分钟内有交巡警（警车的时速为 60km/h）到达事发地。

对于重大突发事件，需要调度全区 20 个交巡警服务平台的警力资源，对进出该区的 13 条交通要道实现快速全封锁。实际中一个平台的警力最多封锁一个路口，请给出该区交巡警服务平台警力合理的调度方案。

根据现有交巡警服务平台的工作量不均衡和有些地方出警时间过长的实际情况，拟在该区内再增加 2 至 5 个平台，请确定需要增加平台的具体个数和位置。

(2) 针对全市(主城六区 A, B, C, D, E, F)的具体情况，按照设置交巡警服务平台的原则和任务，分析研究该市现有交巡警服务平台设置方案（详见附件）的合理性。如果有明显不合理，请给出解决方案。

如果该市地点 P(第 32 个节点)处发生了重大刑事案件，在案发 3 分钟后接到报警，犯罪嫌疑人已驾车逃跑。为了快速搜捕嫌疑犯，请给出调度全市交巡警服务平台警力资源的最佳围堵方案。

## 2 问题分析

### 2.1 问题一的分析

#### 2.1.1 交巡警服务平台管辖范围分配问题的分析

根据题意，以现有的 20 个交巡警服务平台覆盖整个城区 A，并尽量要在出现突发事件的 3min 内赶到事发地，因为警车的速度为 60km/h，即警车距离事发地的最远距离不能超过 3km。根据题目提供的数据，采用 Floyd 算法使用 MatLab 软件进行编程，得到 A 区 92 个路口节点任意两节点之间的最短距离，并且筛选出距离 20 个交巡警服务平台 3km 之内的所有点。对于交叉隶属的节点，我们采用就近原则进行管辖分配。

#### 2.1.2 快速封锁 A 区的问题分析

为了解决封锁 A 区 13 个的路口的问题，我们采用了最小代价的最大匹配的思想，其中代价指封锁最后一个路口所用的时间，根据图论中基本原理，建立二

分图匹配，使用匈牙利算法解得所有可行解，最后利用二分搜索，找到最优调度方案。

### 2.1.3 交巡警服务平台增设分配的问题分析

在问题 1 的求解中，我们发现 6 个节点到交巡警服务平台的时间会超过 3 分钟。当这些节点发生案件时，极易容易出现出警时间过长的情况。因此需要增加交巡警服务平台的数量，对所需出警时间过长的节点进行有效覆盖，并且起到均衡工作量较大平台的作用为了解决出警时间过长的问题，我们需要增设交巡警服务平台使得以上 6 个点到服务平台的时间小于 3 分钟。通过查找 5.1 生成的 Floyd 的矩阵，得到到以上 6 个节点的距离小于 3km 的节点，将其划分为四个区域，选取涵盖点多的点作为服务平台

使得每一个服务平台的案发率尽量接近案发率平均值，即与案发率平均值之间的差距总和最小化，同时必须满足与交巡警服务平台之间的距离小于 3000 米（即 3 分钟内能够到达案发现场），从而建立了以下 0-1 规划模型。

## 2.2 问题二的分析

### 2.2.1 交巡警服务平台设置合理性的问题分析

本文分别对六个地区做了一个简单的模糊综合评价，交巡警三分钟能够到的路口数最多、每个交巡警台平均服务人数、每个交巡警台平均每天处理案件数作为标准。来分别计算每一个地区的分数，并根据分数判断出不合理的地区。针对该地区的得分最低项进行分析，得出所需增加服务平台数量。通过与问题三相似的模型得出新增平台的位置，并对结果进行评价。

再判断出需要对一个地区的时，仍然是对一个地区的合理性修改，所以考虑与第一问建立类似模型。求得一个较为合理的方案。

### 2.2.2 确立最佳围堵方案问题的分析

通过广度优先搜索，找到嫌疑犯所有可能的逃跑路线，比较该节点所辖交巡警服务平台与嫌疑犯到达该节点所用时间，若警车先到，则可以完成对线路的封锁，否则继续重复上述步骤进行封锁。

## 3 模型假设

- (1) 交通状况足够通畅即无阻拦，且警车和嫌疑犯的车辆可以一直保持匀速行驶；
- (2) 交巡警服务平台在接到报警后立即知道嫌疑犯位置；
- (3) 假设单一管辖范围内某一时刻无并发事件；
- (4) 假设出警时警车不会发生故障；
- (5) 假设地图中所有路线均为直线；
- (6) 假设所有道路均为双行道；
- (7) 假设所有突发事件均发生在城区路口节点处。

4 符号说明

符号	定义
$p_i$	第 <i>i</i> 个交巡警服务平台的案发率
$x_{ij}$	第 <i>i</i> 个交巡警服务平台所管辖的第 <i>j</i> 个节点
$d_{ij}$	第 <i>i</i> 个交巡警服务平台到第 <i>j</i> 个节点的距离
$\alpha$	三分钟到达率
$\beta$	每个交巡警台平均服务人数
$\eta$	每个交训警台平均每天处理案件数
$u_i(x)$	第 <i>i</i> 个指标的隶属函数

5 问题一模型的建立与求解

5.1 交巡警服务平台管辖范围的分配

针对交巡警服务平台的管辖范围的分配问题,在不考虑各服务平台的工作量因素时,我们可以采用最短路径的分配方案,建立最短路径的模型进行求解。

5.1.1 基于 Floyd 算法的最短路径模型的建立

根据附件 2 中提供的 A 区 92 个节点的坐标,使用 MatLab 软件绘制出具体节点图,并且根据提供的交巡警服务平台的坐标在节点图中标示出相应服务平台的序号。

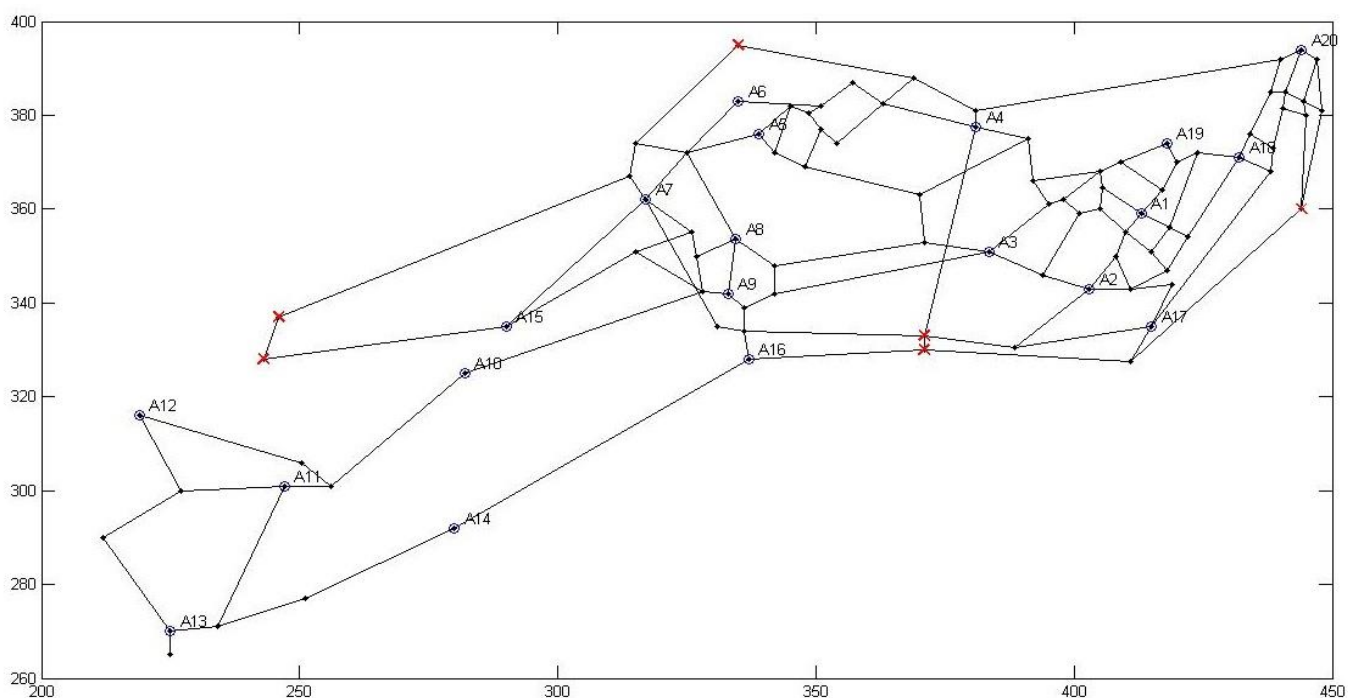


图 1 各交巡警服务平台分布图

根据表格中所给的节点点坐标，应用距离公式

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

算出任意两点间线段的距离，即任意两节点的路程。

#### 5.1.2 利用 Floyd 算法确定各服务平台覆盖的节点

利用 Floyd 最短路径算法的基本原理，使用 MatLab 软件进行编程，得到 A 区 92 个路口节点任意两节点之间的最短距离，并且筛选出距离 20 个交巡警服务平台 3km 之内的所有点。

注：针对有重复的节点采取就近分配原则

表 1 各服务台覆盖节点表

服务台编号	节点名称
A1	1、67、68、69、71、73、74、75、76、78
A2	2、40、43、44、54、55、70、72
A3	3、65、66
A4	4、57、60、62、63、64
A5	5、49、50、51、52、53、56、58、59
A6	6
A7	7、30、32、47
A8	8、33、34、35、46

A9	9、31、45
A10	10
A11	11、26、27
A12	12、25
A13	13、21、22、23、24
A14	14
A15	15
A16	16、36、37
A17	17、41、42
A18	18、80、81、82、83
A19	19、77、79
A20	20、84、85、86、87、88、89、90、91
无法到达	28、29、38、39、61、92

## 5.2 交巡警服务平台封锁各交通要道的调度方案

根据问题的描述：对于重大突发事件，需要调度全区 20 个交巡警服务平台的警力资源，对进出该区的 13 条交通要道实现快速全封锁。实际中一个平台的警力最多封锁一个路口，请给出该区交巡警服务平台警力合理的调度方案。为解决此问题，我们引进二部图匹配模型进行解决。具体步骤如下：

- (1) 分别将 13 条交通要道与 20 各巡警服务台作为二部图的两部分。
- (2) 根据匈牙利算法使用二分搜索的方法找出最大匹配即找到最优解

$p$ 。

<1>当  $p = 13$  时,  $p$  就为最优解,

<2>当  $p < 13$  时, 则说明还有路口没有被及时封闭, 则  $p$  不是最优解。

通过二部图匹配模型我们解出了需要被封锁的 13 个路口所对应的交巡警服务平台及封锁路口所需时间, 详见下表。

表 2 各平台封锁对应路口所需时间

平台与对应的封锁路口	所需时间 (min)	平台与对应的封锁路口	所需时间 (min)
A1-38	5.881	A11-21	5.072
A2-16	7.388	A12-22	6.882
A3-30	4.393	A13-24	2.385
A4-48	7.395	A14-23	6.473
A5-62	3.182	A15-28	4.751
A7-29	8.015	A16-14	6.741
A10-12	7.586		

5.3 交巡警服务平台的增设分配方案

在 5.1 的求解中，不难发现共有 6 个点到服务平台的时间会超过 3 分钟。当这些点发生案件时，极其容易出现出警时间过长的情况。为了解决出警时间过长的问題，我们需要增设交巡警服务平台使得以上 6 个点到服务平台的时间小于 3 分钟。通过查找 5.1 生成的 Floyd 的矩阵，不难得到到以上 6 个点的距离小于 3km 的点，也即到达时间小于 3 分钟的点。如下表：

表 3 出警时间过长节点与邻接节点

出警时间时间过长的点	与该点距离小于 3km 的点
29	28
28	29
38	39、40
39	38、40
61	48
92	87、88、89、90、91

上表中的点可以分为四块区域，如下表：

表 4 不同区域包含节点

区域	该区域包含的点
区域 A	28、29
区域 B	38、39、40
区域 C	61、48
区域 D	92、87、88、89、90、91

以上四个区域每个区域中任取一点作为服务平台即可解决出警时间过长的问題。

为了能够分担工作量，我们显然应该选择能涵盖点最多的点作为服务平台，这里的涵盖指能在 3 分钟以内到达。以上每个点作为平台时能涵盖的节点如下表：

表 5 不同区域涵盖的节点

区域	可作服务平台的点	该平台能涵盖的节点
区域 A	28	28、29
	29	28、29
区域 B	38	38、39、40
	39	38、39、40
	40	38、39、40、43、44、70、2、17
区域 C	61	61、48
	48	61、48、5、6、7、30、32、33、47、48、49
区域 D	87	18、20、81、82、83、84、85、86、87、88、89、90、91、92
	88	18、20、81、82、83、84、85、86、87、88、89、90、91、92



89	18、20、81、82、83、84、85、86、87、88、 89、90、91、92、80
90	18、20、81、82、83、84、85、86、87、88、 89、90、91、92、80
91	18、20、81、82、83、84、85、86、87、88、 89、90、91、92
92	87、88、89、90、91

通过上表，我们选取涵盖点多的点作为服务平台，当覆盖的点完全一样时，我们选择案发率较高的点作为服务平台。不难得到选择的点应该为：29、40、48、89。

加入以上 4 点后，为了平均分配工作量，需要重新调整各个交巡警服务平台所管辖的节点或节点个数，使得每一个服务平台的案发率尽量接近案发率平均值，即与案发率平均值之间的差距总和最小化，同时必须满足与交巡警服务平台之间的距离小于 3000 米（即 3 分钟内能够到达案发现场），从而建立了以下 0-1 规划模型。

$$\left\{ \begin{array}{l} \min \sum_{i=1}^{24} \left| \overline{p_i} - \sum_{i=1}^{92} p_i x_{ij} \right| \\ s.t. \quad x_{ij} d_{ij} \leq 3000 \\ \sum_{i=1}^{24} x_{ij} = 1 \\ x_{ij} = 0 \text{ 或 } 1 \end{array} \right. \quad (j=1, 2, \dots, 92)$$

该模型等价于

$$\left\{ \begin{array}{l} \min \sum_{i=1}^{24} \left( \overline{p_i} - \sum_{i=1}^{92} p_i x_{ij} \right)^2 \\ s.t. \quad x_{ij} d_{ij} \leq 3000 \\ \sum_{i=1}^{24} x_{ij} = 1 \\ x_{ij} = 0 \text{ 或 } 1 \end{array} \right. \quad (j=1, 2, \dots, 92)$$

对于以上模型的求解，采用枚举法较为复杂且效率极低，难以接近最优解。在本题中，去掉一些不可调配的点，剩下的点可以通过人工调整的方式得到较优解，我们通过表 1 以及案发频率合理调配，得到以下表格。

表 6 各交巡警服务平台所辖节点与总案发率

交巡警平台序号	所管辖的节点	总案发率
1	1、71、73、74、75、78	6.4
2	2、43、66、76	5.7
3	3、54、55、64、65	5.6
4	4、57、60、62、63	6.4
5	5、50、56、58、59	5.7
6	6、51、52、53	5.3
7	7、30、32	6
8	8、33、46	5
9	9、34、45	5.2
10	10	1.6
11	11、26、27	4.6
12	12、25	4
13	13、21、22、23、24	8.5
14	14	2.5
15	15、31	3.7
16	16、35、36、37	5.2
17	17、41、42、72	6.1
18	18、80、81、82、83	6.1
19	19、77、67、68、69、79	6.2
20	20、84、85、86、88	6.4
29	28、29	2.7
40	38、39、40、44、70	6.3
48	47、48、49、61	4.8
89	91、87、89、90、92	5.1

该近似最优解与平均发案率 5.2 之间的距离总和为 25.9，如果从表中去掉不可优化的服务平台 10，13，14，28 则与平均发案率之间的距离总和为 13.7，方差已经较为接近 0 了。即各平台的工作负荷较为平均。

6 问题二的模型建立与求解

6.1 现有交巡警服务平台设置方案的评价与改进

6.1.1 现有平台设置的合理性分析

对于合理性的分析，本文利用模糊综和评价法，给出对每个地区的综合评分，得到得分最低的地区，即交巡警服务平台方案设置最不合理的地区。

本文给出的模糊综合评价模型的三个评判指标如下：

(1) 交巡警服务台三分钟能够的到达率。记为  $\alpha$ 。

(2)每个交巡警台平均服务人数 $\beta$ （万人）。

(3)每个交训警台平均每天处理案件数 $\eta$ （件）。

利用第一问中对 A 区进行的优化前的简单的最短路模型，分别对另外五个区 B、C、D、E、F 的四个评判指标进行求解，得到的六个区的三个指标如下图所示：

表 7 各地区三项指标的具体数值

地区	三分钟到达率 $\alpha$	每个交巡警台平均服务人数 $\beta$ （万人）	每个交训警台平均每天处理案件数 $\eta$ （件）
A	93.48%	3	6.23
B	91.78%	3.16	8.3
C	74.83%	2.88	11.01
D	77%	8.11	7.53
E	67.60%	5.07	7.96
F	72.22%	4.82	9.92

针对上述评价指标，定义隶属函数：

定义 1 指标(1)为三分钟的到达率：此指标越大，得分越高，这里考虑应用其自身作为隶属函数：

$$u_1(x)=x$$

定义 2 指标(2)为每个交巡警台的平均服务人数：平均服务人数越少，即人均分配尽力高，易于管理，亦即目标越小，得分越高，由此定义隶属函数为：

$$u_2(x)=1-\frac{1}{10}x$$

定义 3 指标(3)为每个交训警台平均每天处理案件数：每天处理案件数少，地区安全性高，由此定义隶属函数：

$$u_3(x)=\frac{1}{x}$$

模型的求解：

根据以上的隶属函数的定义，可以求得 A、B、C、D、E、F 六个地区相对应的隶属度，见下表。

表 8 各地区三项指标评价隶属表

地区	三分钟到达率 $\alpha$	每个交巡警台平均服务人数 $\beta$ （万人）	每个交训警台平均每天处理案件数 $\eta$ （件）
A	93.48%	0.7	0.161
B	91.78%	0.684	0.121

C	74.83%	0.712	0.091
D	77%	0.189	0.133
E	67.60%	0.493	0.126
F	72.22%	0.518	0.101

根据上述隶属度表，得到模糊关系矩阵  $R$ ，如下：

$$R = \begin{pmatrix} 0.935 & 0.700 & 0.161 \\ 0.918 & 0.684 & 0.121 \\ 0.748 & 0.712 & 0.091 \\ 0.770 & 0.189 & 0.133 \\ 0.676 & 0.493 & 0.126 \\ 0.722 & 0.518 & 0.101 \end{pmatrix}$$

通过分析，确定与之对应每项指标的权重  $W = (0.4, 0.3, 0.3)$ ；得到每一个地区的综合得分  $D = WR$ 。即

$$D = (0.7332, 0.7118, 0.6094, 0.5264, 0.4561, 0.557) ;$$

由此，我们可以知道 D 区的得分最少。

#### 6.1.2 D 区交巡警服务平台设置合理化改进方案

综上所述，我们将考虑对 D 区的优化，由表一可知，D 区交警平均服务人数过多，显然是由于服务平台数量不足所致，因此我们通过增设 5 个服务平台将平均服务人数降至一个较低的水准。由计算可得，在优化之前，三分钟的可到达率仅为 72.22%。

建议提出：

该模型与 5.1.3 相似，在区域 D 内建立 Floyd 矩阵，找出与服务平台距离超过 3km 的点，共 12 个，分别为 329、330、331、332、336、337、339、362、344、369、370、371，并在 MatLab 中画出其在 D 区图像中的位置。

图 2 D 区的交通网络与交巡警服务平台设置示意图

当这些点发生案件时，极其容易出现出警时间过长的情况。通过查找生成的 Floyd 的矩阵，不难得到到以上 12 个点的距离小于 3km 的点，也即到达时间小于 3 分钟的点，如下表：

表 9 出警时间过长节点与邻接节点

出警时间时间过长的点	与该点距离小于 3km 的点
330	331 329
331	330 329
329	330 331
336	333 334 335
337	338
339	333
369	368
371	370
370	371
344	无
362	无
332	无

为了能够分担工作量，我们显然应该选择能涵盖点最多的点作为服务平台，这里的涵盖指能在 3 分钟以内到达。以上每个点作为平台时能涵盖的点如下表：

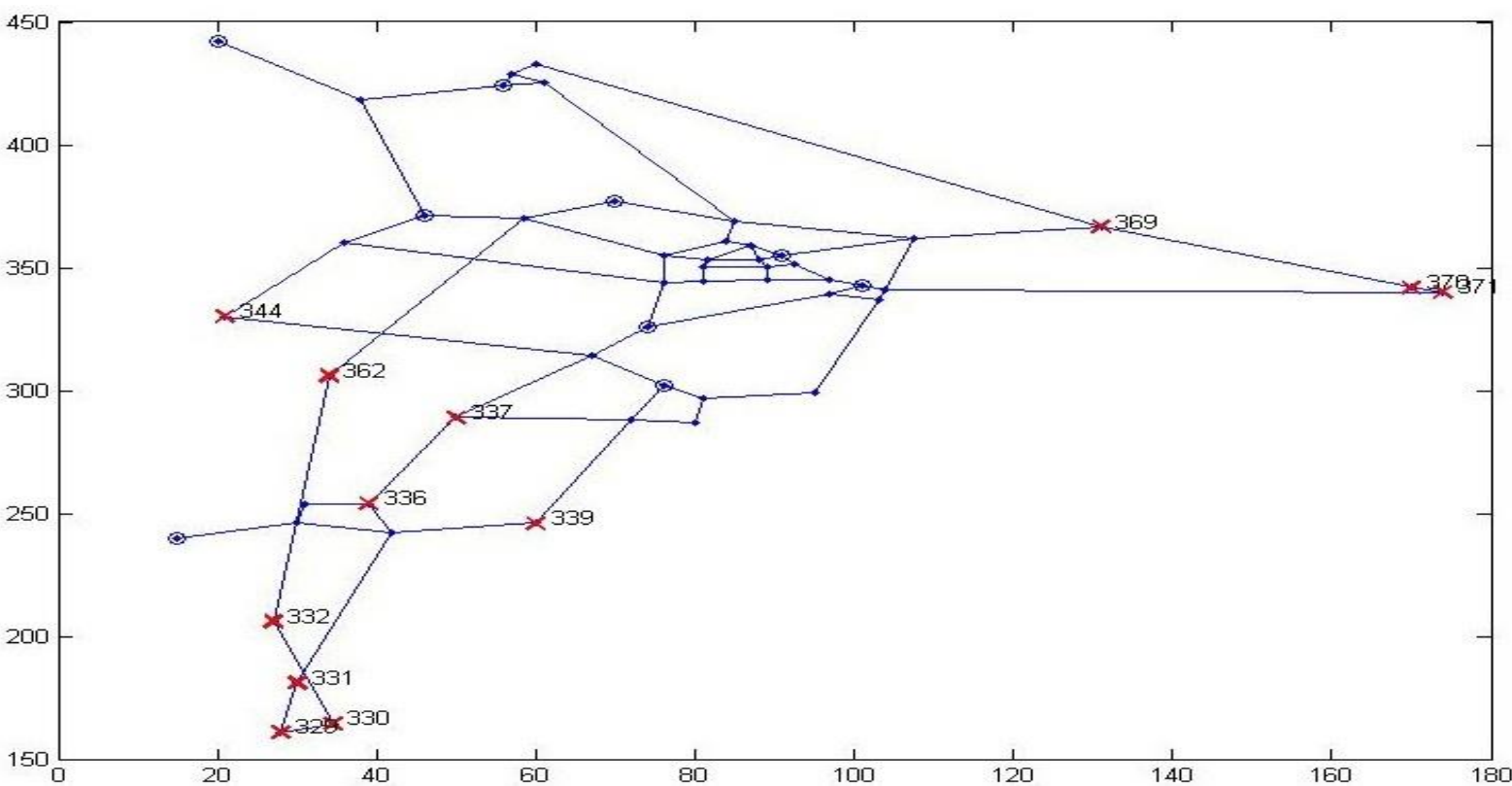


表 10 各区域可做服务平台节点及能涵盖的节点

区域	能作为服务平台的节点	该平台能涵盖的节点
区域 1	330	331 329 330
	331	330 329 331
	329	330 331 329
	332	332
区域 2	336	338 336
	337	333 337
	339	368 339
区域 3	369	368 369
	371	370 371
	370	371 370
散点	362	362
	344	344

通过上表，我们选取涵盖点多的点作为该区域的服务平台，两个散点单独作为服务平台，当覆盖的点完全一样时，我们选取到达周围点距离之和最短的点设为平台，不难得到选择的点为：330、333、370、344、362。

在增加上述 5 个点后三分钟的可达率为 96%，相较优化之前，三分钟的可达率为 72.22%，有显著提升。

6.2 最佳围堵方案的制定

6.2.1 模型的建立

根据题意，若在该市地点 P(A 区第 32 个节点)处发生案件，由于案发 3min 后才能接到报警，因此在这 3min 内，嫌疑犯驾车逃跑的路线存在多种可能，因此针对每种不同的情况采用不同的围堵方案并且得到与之相对应的围堵时间。为此，本题采用了图论中的广度优先搜索原理进行解决。具体步骤如下：

- (1) 以 P 点作为起点，搜索与 P 点相连接的所有路径及其对应的节点。
- (2) 计算嫌疑犯到达各邻接节点的时间与该管辖区域的交巡警服务平台到邻接节点的时间。
- (3) 若交巡警服务平台到达的时间小于嫌疑犯到达的时间，则该路线被封堵 否则，以未封堵成功的节点作为新的起点重复前两步，直到逃跑路线被全部封锁。

6.2.2 模型的求解

假设嫌疑犯的行驶速度为 60km/h,并且在报警的同时知道嫌疑犯的位置，警车立即出发开始围堵嫌疑犯，根据图的广度优先搜索原理使用 MatLab 编程，求解可得在不同路口处完成封堵所用的时间和最终封堵他的属于哪个区的交巡警服务平台，结果见表

表 11 不同路口成功封堵所用时间

成功封堵的 交巡警平台	成功封堵的 路口标号	封堵时 间/min	成功封堵的 交巡警平台	成功封堵的 路口标号	封堵时 间/min
A1	41	4.441	C167	248	3.678

A2	3	2.111	C168	4	3.699
A3	55	1.265	C169	240	7.047
A4	60	1.732	C171	232	1.906
A5	5	0	C172	246	4.719
A6	6	0	C173	236	0.632
A10	10	0	C175	168	4.977
A15	15	0	D320	370	7.808
A16	16	0	D321	371	8.979
A17	40	2.687	F475	561	4.355

对表的结果进行分析，发现围堵嫌疑犯所需的最长时间为 8.979min，此时嫌疑犯已经逃到了 D 区。

## 7 模型的优点与缺点

### 7.1 模型的优点

(1) 在求解快速封锁 A 区的问题时，采用了最小代价的最大匹配的思想，根据图论中基本原理，建立二分图匹配，使用匈牙利算法解得所有可行解，最后利用二分搜索，找到最优调度方案。

(2) 在讨论增设交巡警服务平台的问题时，通过使用 Floyd 距离矩阵对目标节点进行有效覆盖，并且通过建立 0-1 规划模型，使得每一个服务平台的案发率尽量接近案发率平均值，从而找到最佳的平台增设方案。

(3) 在解决最佳围堵问题时，本文采用了广度优先算法，通过循环模拟可行路径和可行围堵方案最终找到了最佳围堵方案。

### 7.2 模型的缺点

(1) 在分配管辖范围时，有的节点可能同时满足好几个平台的管辖范围，本文以最短路径作为选择依据使用就近分配的原则对这些特殊节点进行划分，这样造成了部分平台负担较轻，另一部分又相对过重。

(2) 在实际问题中，接警通信、处理案件的时间、道路转弯路程等不可避免的因素都会延长交巡警到达案发地的时间。

## 参考文献

- [1]袁新生，邵大宏，郁时炼.LINGO 和 Excel 在数学建模中的应用[M]. 北京：科学出版社，2007.
- [2]严蔚敏，吴伟民.数据结构（C 语言版）[M]. 北京：清华大学出版社，2009.
- [3]姜启源.数学模型[M]. 北京：高等教育出版社，1993.

## 附录

```
clear,clc
%加载数据
file='../\cumcm2011B 附件 2_全市六区交通网路 and 平台设置的数据表.xls';
kiosk=xlsread(file, 3, 'A2:B81');
p=xlsread(file, 1, 'A2:E583');
e=xlsread(file, 2, 'A2:B929');
entryA=xlsread(file, 4, 'C2:C14');
n=582;
limit=30;
%plot graph
A=zeros(n);
xy=zeros(n, 2);
for i=1:size(e, 1)
    x=e(i, 1);
    y=e(i, 2);
    A(x, y)=1;
    A(y, x)=1;
end
xy= p(:, 2:3);
gplot(A, xy, 'k.-');
hold on
for i=1:size(p, 1)
```

### 绘制图表



```

        text(p(i, 2)+2, p(i, 3)+2, num2str(i));
    end
%floyd 算法实现
    for i=1:size(e, 1)
        x=e(i, 1);
        y=e(i, 2);
        A(x, y)=1;
        A(y, x)=1;
    end
    adj=ones(n)*Inf;
    for i=1:n
        adj(i, i)=0;
    end
    for i=1:n
        for j=1:n
            if A(i, j)==1
                p1=[p(i, 2);p(i, 3)];
                p2=[p(j, 2);p(j, 3)];
                adj(i, j)=sqrt((p1-p2)'*(p1-p2));
            end
        end
    end
    adjbak=adj;
    for k=1:n
        for i=1:n
            for j=1:n
                if adj(i, k)+adj(k, j)<adj(i, j)
                    adj(i, j)=adj(i, k)+adj(k, j);
                end
            end
        end
    end
    end
%初始最短分配图
    Bad=zeros(92, 1);
    for i=1:size(kiosk)
        plot(p(kiosk(i), 2), p(kiosk(i), 3), 'bo')
    end
    assign=zeros(1, 92);
    for i=1:92
        [Y, l]=min(adj(i, 1:20));
        if Y>limit
            Bad(i)=1;
        end
        assign(i)=l;
    end

```

```

end
for i=1:92
    if Bad(i)==1
        plot(p(i,2),p(i,3),'rx','MarkerSize',10,'LineWidth',2)
    end
end
hold off
%匈牙利分配
low=0.0;
high=max(max(adj));
    %20*13 警亭与节点距离矩阵
L=zeros(20,size(entryA,1));
for i=1:20
    for j=1:size(entryA,1)
        L(i,j)=adj(i,entryA(j));
    end
end
%二分搜索
while high-low>0.001
    mid=(low+high)/2;
    Metrix=1./(L<=mid);
    [at,Cost]=munkres(Metrix);
    if Cost<13
        low=mid;
    else
        high=mid;
    end
end
Metrix=1./(L<=high);
[at,Cost]=munkres(Metrix);
at
Cost
high
M=zeros(20,13);
for i=1:20
    if at(i)~=0 M(i,at(i))=1; end
end

[X,Y]=find(M==1);
Y=entryA(Y);
I=[X,Y];
for i=1:13
    I(i,3)=adj(I(i,1),I(i,2));
end

```

```

%分配均匀度计算
load=zeros(20,1);
for i=1:92
    if assign(i)~=0
        load(assign(i))=load(assign(i))+p(i,5);
    end
end
end
%bfs 广度优先搜索
clc
m=size(kiosk,1);
canuse=ones(m,1);
catchpoint=zeros(m,1);
touchtime=min(adj(kiosk,:));
point=ones(n,1);
    %dijkstra
us=ones(n,1);
us(32)=0;
d=adjbak(32,:)./us'; %到集合的最短距离
w=ones(n,1)*Inf;
w(32)=0;
for i=1:n-1
    [Y I]=min(d);
    if Y==Inf
        break;
    end
    w(I)=Y;
    us(I)=0;
    d(I)=Inf;
    if Y-30>touchtime(I) && Y>=30
        jt=kiosk(find(canuse==1));
        K=find(kiosk==jt(find(adj(jt,I)==touchtime(I))));
        canuse(K)=0;
        catchpoint(K)=I;
        point(I)=0;
        touchtime=min(adj(kiosk(find(canuse==1)),:));
    else
        d=min(d,adjbak(I,:)./us'./point'+Y);
    end
end
end
temp=find(point==0);
hold on
plot(p(temp,2),p(temp,3),'r^','MarkerSize',10)
hold off
%show answer

```

```

tp=canuse==0;
sum(tp)
tp=[kiosk(find(catchpoint~=0)) catchpoint(tp)];
AA=zeros(n,n);
for i=1:size(tp,1)
    AA(tp(i,1),tp(i,2))=1;
end
hold on
gplot(AA,xy,'r*-')
hold off
for i=1:size(tp,1)
    tp(i,3)=adj(tp(i,1),tp(i,2));
end
%D 区域改善分析
errorD=find(min(adj(kiosk(46:54),320:371))>30)+319
figure(2)
gplot(A(320:371,320:371),xy(320:371,:), 'b.-');
hold on
plot(p(kiosk(46:54),2),p(kiosk(46:54),3),'bo');
plot(p(errorD,2),p(errorD,3),'rx','MarkerSize',10,'LineWidth',2)
for i=errorD
    text(p(i,2)+2,p(i,3)+2,num2str(i));
end
hold off
[X Y]=find(adj(errorD,320:371)<=30)

```