

# 交巡警服务平台的设置与调度

## 摘要

本文针对问题，建立了回溯算法等多种模型，解决了交巡警服务平台的设置与调度问题。

针对问题一，对三公里路程内进行搜索，

针对问题二，

针对问题三，

针对问题四，

针对问题五，

关键字：

# 1 问题背景与重述

## 1.1 问题背景

“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：

## 1.2 问题表述

- (1) 问题一：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分钟内由交巡警（警车的时速为60km/h）到达事发地。
- (2) 问题二：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。
- (3) 问题三：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。
- (4) 问题四：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。
- (5) 问题五：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 问题一分析

对于问题一，为解决划分辖区这一问题，需满足交巡警平台管辖的范围内出现突发事件时，尽量能在3分钟内到达事发地这一条件，可从三分钟内抵达入手，将其作为约束条件，进行搜索，由此根据附图1中的交通网络，现有的二十个交巡警服务平台的设置情况示意图和附件2中的相关数据信息，划分管辖区域。

## 2.2 问题二分析

首先，这个是一个指派问题，为解决需对进出该区的13条交通要道实现快速全封锁这一问题，需要通过调度全区警力资源以最快速度完成封锁。为此，我们需建立最优调度方案使这一目标最快实现。

2.3 问题三分析

基于问题一已建立的扩散搜索模型，引入工作量判定函数，综合考虑A区每个交巡警平台范围内的路径长度与节点发案率，为使平台工作量均衡并且尽量保证出警时间在三分分钟内，最终确定增加平台的位置与数目，以达到最优设置方案。

2.4 问题四分析

2.5 问题五分析

对于该问题

3 模型假设

- (1)
- (2) 交巡警服务平台管辖范围不能重叠。
- (3) 不考虑交通拥堵，假设各节点，路段通畅。
- (4) 假设

4 符号说明

符号	意义	单位
符号	意义	单位
符号	意义	单位

5 模型建立与求解

5.1 问题一模型的建立与求解

5.1.1 问题一模型的建立

首先，读取文件并进行数据预处理，为解决划分辖区这一问题，需满足交巡警平台管辖的范围内出现突发事件时尽量能在3分钟内抵达事发地这一条件，可将交巡警平台在三分钟内抵达该路口节点作为约束条件。以交巡警平台作为起始点向外延伸三公里进行搜索，由此根据附图1中的交通网络所给出的二十个现有的交巡警服务平台的设置情况示意图和附件2中的相关数据信息，划分管辖区域。

5.1.2 数据预处理

首先读取文件，根据区域名称将节点坐标，节点间连接关系，平台所在节点和路口所在节点从附件二中提取出，并格式化和建立数据库，进而计算连接节点间的欧几里得距离，考虑到点对间存在一定关系，即点与点之间存在连接，并包含点之间的距离信息，构建节点网络地图。

### 5.1.3 Dijkstra算法的原理

设有一个图  $G = (V, E)$ ，其中  $V$  是节点集合， $E$  是边集合，权重函数  $w : E \rightarrow \mathbb{R}^+$  定义每条边的非负权重。起始节点为  $s$ ，目标是找到从  $s$  到所有其他节点的最短路径。

1. 初始化：

$$\forall v \in V, \text{dist}[v] = \infty$$

$$\text{dist}[s] = 0$$

2. 每次选择距离最小的未访问节点  $u$ ：

$$u = \arg \min_{v \in Q} \text{dist}[v]$$

3. 更新邻接节点的距离：对于每个邻接节点  $v$ ：

$$\text{alt} = \text{dist}[u] + w(u, v)$$

如果  $\text{alt} < \text{dist}[v]$ ，则：

$$\text{dist}[v] = \text{alt}$$

并更新前驱节点：

$$\text{prev}[v] = u$$

4. 重复上述过程，直到所有节点都被访问过。

Dijkstra算法通过上述步骤有效地找到从源点到所有其他节点的最短路径，其时间复杂度在使用二叉堆实现优先队列时为  $O((|E| + |V|) \log |V|)$ 。

### 5.1.4 指定深度的DFS模型的建立

为了得出平台3km距离内所有的路口，引入深度优先搜索模型：

给定一个图  $G = (V, E)$ ，其中  $V$  是顶点集合， $E$  是边集合。DFS 算法的步骤如下：

1. 初始化：标记所有节点为未访问状态。
2. 选择起始节点：从起始节点  $v \in V$  开始搜索。
3. 递归搜索：

$$\text{DFS}(v) = \begin{cases} \text{visited}(v) = \text{true}, & \text{if } v \text{ is the starting node} \\ \text{DFS}(u), & \text{for each neighbor } u \text{ of } v \text{ and } \text{visited}(u) = \text{false} \end{cases}$$

4. 利用递归，搜索出3km路程内的点集，并将其标示在图上。

### 5.1.5 去除重叠区域与优化

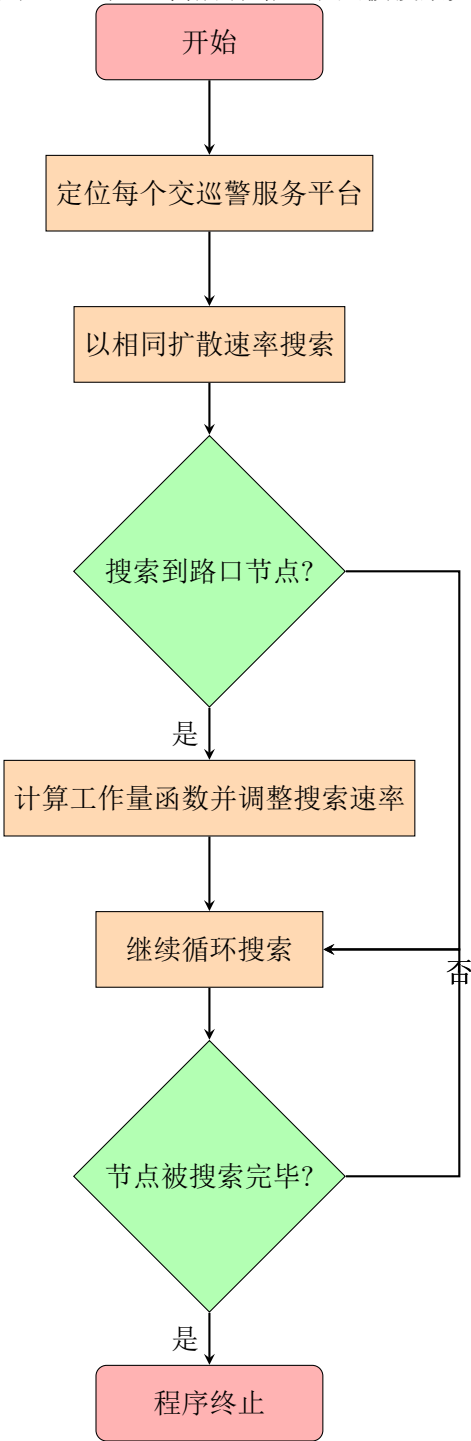
由于深度优先搜索算法有可能使某些路口无人管辖或邻近交巡警服务平台的管辖区域存在重叠，而在现实中，不同警局的执法范围一般不会重叠，为避免此现象发生，需要对该模型进行优化。

但是深度优先搜索算法会搜索某一深度内的所有节点，进而导致同一路口节点被不同交巡警平台多次覆盖。

为解决这一问题，从各平台同时进行向外扩散搜索，当一个节点被某一平台纳入管辖范围，就不能被其他平台搜索到，为此引入扩散搜索模型。

### 5.1.6 扩散模型的引入

定位每一个交巡警服务平台，对所有平台，以相同扩散速率进行搜索，若搜索到的节点未被其他平台添加到管辖区域内，则将其加入自己的管辖区。在每一轮增加搜索距离之前，计算各个平台工作量之和，根据计算结果，调整该交巡警平台的扩散搜索速率。重复此循环，直到3km内所有路口节点被搜索完毕，程序终止。



1. 定位每一个交巡警服务平台:

设交巡警服务平台集合为  $P = \{p_1, p_2, \dots, p_m\}$ ，路口节点集合为  $N = \{n_1, n_2, \dots, n_n\}$ 。

2. 以相同速率进行扩散搜索，直到搜索到第一个路口节点：

对每一个交巡警服务平台  $p_i \in P$ ，设初始搜索速率为  $v_0$ ，在时间  $t$  内的搜索半径为  $r_i(t) = v_0 \cdot t$ 。搜索半径  $r_i$  满足：

$$r_i(t) = \min\{r \mid \exists n_j \in N, \text{dist}(p_i, n_j) \leq r\}$$

3. 继续重复此循环，直到3km内所有路口节点被搜索完毕，程序终止：

设搜索终止距离为  $R = 3 \text{ km}$ ，程序在满足以下条件时终止：

$$\forall n_j \in N, \text{dist}(p_i, n_j) \leq R \text{ for some } p_i \in P$$

优点在于无需去除重叠区域且到案时间更短。

放张图！

### 5.1.7 结果的检验

虽然任意平台无法在3分钟内抵达28, 29, 38, 39, 61, 92这六个节点，但可通过延长搜索时间覆盖所有节点。

路口节点	28	29	38	39	61	92
警局节点	A15	A15	A16	A2	A7	A20
抵达时间	4.8	5.7	3.4	3.7	4.2	3.6

### 5.1.8 问题的结论

交巡警平台	管辖节点
A1	1、67、68、69、71、72、73、74、75、76、78
A2	2、39、40、43、44、70
A3	3、54、55、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、48、61
A8	8、33、46
A9	9、31、34、35、45
A10	10
A11	11、26、27
A12	12、25
A13	13、21、22、23、24
A14	14
A15	15、28、29
A16	16、36、37、38
A17	17、41、42
A18	18、80、81、82、83
A19	19、77、79
A20	20、84、85、86、87、88、89、90、91、92

## 5.2 问题二模型的建立与求解

### 5.2.1 问题二的模型建立

为解决这一指派问题，需对进出该区的13条交通要道实现快速全封锁，通过调度全区警力资源以最快速度完成封锁。为此，我们需建立最优调度方案使这一目标最快实现。

为实现对进出该区的13条交通要道快速全封锁，即指派交巡警服务平台到交通要道，目标为封锁完成的时间最短。据此建立最大流二分查找模型，规划出最佳封锁方案。

### 5.2.2 最大流算法的模型建立

最大流问题是在给定一个源点（source）和汇点（sink）的流网络中，找到从源点到汇点的最大流量。网络中每条边都有一个容量限制，表示该边能通过的最大流量。最大流算法的目标是找到一种流量分配方式，使得从源点到汇点的总流量最大。

#### 1. 流量守恒约束

在网络的每个节点  $v$ （除了源点和汇点）处，流量守恒原则要求：

$$\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$$

其中， $f(u, v)$  表示从节点  $u$  到节点  $v$  的流量， $V$  表示节点集合。

#### 2. 边容量约束

每条边的流量不能超过其容量限制：

$$0 \leq f(u, v) \leq c(u, v)$$

其中， $c(u, v)$  表示边  $(u, v)$  的容量。

### 3.最大流量

最大流问题的目标是最大化从源点  $s$  到汇点  $t$  的总流量  $|f|$ ：

$$\max |f| = \sum_{v \in V} f(s, v)$$

### 4.二分查找法

二分查找法是一种在有序数组中查找目标值的高效算法。其基本思想是通过逐步缩小查找范围，将问题规模减半，直到找到目标值或确定目标值不存在。

### 5.二分查找算法步骤

(1) 初始化左右边界  $T_{\min}$  和  $T_{\max}$ 。

(2) 计算中点  $T_{\text{mid}}$ ：

$$T_{\text{mid}} = \frac{T_{\min} + T_{\max}}{2}$$

(3) 判断  $T_{\text{mid}}$  是否满足条件。如果满足，则更新右边界：

$$T_{\max} = T_{\text{mid}}$$

否则，更新左边界：

$$T_{\min} = T_{\text{mid}}$$

(4) 重复步骤2和3，直到左右边界足够接近。

### 6.最大流二分查找法结合

我们通过二分查找法找到一个最小的时间  $T$ ，使得在此时间内警察能够封锁所有关键交通要道。对于每个给定的  $T$ ，通过最大流算法判断是否能满足封锁要求。

### 7.具体数学模型

目标函数：

$$\min T$$

约束条件：

$$\text{s.t.} \quad \begin{cases} \sum_{i \in P} x_{ij} \geq 1, & \forall j \in K \\ \sum_{j \in K} x_{ij} \leq 1, & \forall i \in P \\ x_{ij} \cdot d_{ij} \leq T, & \forall i \in P, \forall j \in K \\ \text{flow}(T) \geq |K| \end{cases}$$

其中， $P$  为警察局集合， $K$  为关键交通要道集合， $d_{ij}$  为警察局  $i$  到路口  $j$  的时间， $x_{ij}$  为二元决策变量，表示是否由警察局  $i$  封锁路口  $j$ 。



### 5.2.3 模型的求解

$$\begin{aligned}
 & \min T \\
 & \text{s.t.} \\
 & \begin{cases} \sum_{i \in P} x_{ij} \geq 1, & \forall j \in K \\ \sum_{j \in K} x_{ij} \leq 1, & \forall i \in P \\ x_{ij} \cdot d_{ij} \leq T, & \forall i \in P, \forall j \in K \\ \text{flow}(T) \geq |K| \end{cases}
 \end{aligned} \tag{1}$$

### 5.2.4 模型求解的结果

匹配对: [(2, 16), (4, 48), (7, 29), (8, 30), (10, 12), (11, 24), (12, 22), (13, 23), (14, 21), (15, 28), (16, 14), (19, 38), (20, 62)]

A2	A4	A7	A8	A10	A11	A12
16	48	C	D	E	F	G
A13	A14	A15	A16	A19	A20	
1	2	3	4	5	6	

### 5.2.5 结果的检验

### 5.2.6 问题的结论

### 5.2.7 模型检验与分析

### 5.2.8 误差分析

### 5.2.9 灵敏度分析

### 5.2.10 稳定性分析

### 5.2.11 小结

## 5.3 问题三模型的建立与求解

### 5.3.1 模型建立

基于问题一已建立的扩散搜索模型，引入工作量判定函数，综合考虑A区每个交巡警平台范围内的路径长度与节点发案率，为使平台工作量均衡并且尽量保证出警时间在三分分钟内，最终确定增加平台的位置与数目，以达到最优设置方案。

### 5.3.2 模型建立的数学思想

### 5.3.3 模型建立的准备

### 5.3.4 模型的求解

### 5.3.5 工作量函数的建立

工作量由工作时间决定，即出警时间与办案时间，其中，出警时间是（路程/速度）\*时间，出警时间由所调查资料可知，平均时长为30分钟。

要计算警察完成附近路口任务的总工作量，我们可以定义如下的工作量函数  $W$ ，综合考虑多个因素：

$$W = \sum_{i=1}^N \left( R_i \times T_{\text{case}} + 2 \times \frac{D_i}{V} \right) \quad (2)$$

其中：

- $N$  是路口数量
- $R_i$  是第  $i$  个路口的发案率
- $T_{\text{case}}$  是处理每个案件的固定时间，设为 30 分钟（0.5 小时）
- $V$  是警察行进的速度（单位：公里/小时）
- $D_i$  是从起点到第  $i$  个路口的距离（单位：公里）

### 5.3.6 基于优化函数的扩散算法

1. 定位每一个交巡警服务平台，对所有平台，以相同速率进行扩散搜索，直到搜索到第一个路口节点。
2. 搜索到第一个路口节点后，进行工作量函数的计算，根据计算结果，调整该交巡警平台的扩散搜索速率。
3. 继续重复此循环，直到3km内所有路口节点被搜索完毕，程序终止。

优点在于无需去除重叠区域且工作量分配更加均匀。

#### 1. 定位每一个交巡警服务平台：

设交巡警服务平台集合为  $P = \{p_1, p_2, \dots, p_m\}$ ，路口节点集合为  $N = \{n_1, n_2, \dots, n_n\}$ 。

#### 2. 以相同速率进行扩散搜索，直到搜索到第一个路口节点：

对每一个交巡警服务平台  $p_i \in P$ ，设初始搜索速率为  $v_0$ ，在时间  $t$  内的搜索半径为  $r_i(t) = v_0 \cdot t$ 。搜索半径  $r_i$  满足：

$$r_i(t) = \min\{r \mid \exists n_j \in N, \text{dist}(p_i, n_j) \leq r\}$$

3. 搜索到第一个路口节点后，进行工作量函数的计算，根据计算结果调整该交巡警平台的扩散搜索速率：

设工作量函数为  $W(p_i)$ ，则新的搜索速率  $v_i$  为：

$$v_i = f(W(p_i))$$

其中，函数  $f$  根据具体工作量计算结果调整搜索速率。

4. 继续重复此循环，直到3km内所有路口节点被搜索完毕，程序终止：

设搜索终止距离为  $R = 3 \text{ km}$ ，程序在满足以下条件时终止：

$$\forall n_j \in N, \text{dist}(p_i, n_j) \leq R \text{ for some } p_i \in P$$

#### 5.3.7 模型求解的数学原理

#### 5.3.8 模型求解的准备

#### 5.3.9 模型求解的过程

#### 5.3.10 模型求解的结果

#### 5.3.11 结果的检验

#### 5.3.12 问题的结论

#### 5.3.13 模型检验与分析

#### 5.3.14 误差分析

#### 5.3.15 灵敏度分析

#### 5.3.16 稳定性分析

#### 5.3.17 小结

### 5.4 问题四模型的建立与求解

#### 5.4.1 数据预处理

#### 5.4.2 模型建立

#### 5.4.3 工作量函数的建立

工作量由工作时间决定，即出警时间与办案时间，其中，出警时间是（路程/速度）\*时间，出警时间由所调查资料可知，平均时长为30分钟。

#### 5.4.4 模型建立的数学思想

#### 5.4.5 模型建立的准备

#### 5.4.6 模型的求解

#### 5.4.7 模型求解的数学原理

#### 5.4.8 模型求解的准备

#### 5.4.9 模型求解的过程

#### 5.4.10 模型求解的结果

#### 5.4.11 结果的检验

#### 5.4.12 问题的结论

#### 5.4.13 模型检验与分析

#### 5.4.14 误差分析

#### 5.4.15 灵敏度分析

#### 5.4.16 稳定性分析

#### 5.4.17 小结

### 5.5 问题五模型的建立与求解

#### 5.5.1 模型建立

#### 5.5.2 最大流算法的模型建立

最大流问题是在给定一个源点（source）和汇点（sink）的流网络中，找到从源点到汇点的最大流量。网络中每条边都有一个容量限制，表示该边能通过的最大流量。最大流算法的目标是找到一种流量分配方式，使得从源点到汇点的总流量最大。

##### 1.流量守恒约束

在网络的每个节点  $v$ （除了源点和汇点）处，流量守恒原则要求：

$$\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$$

其中， $f(u, v)$  表示从节点  $u$  到节点  $v$  的流量， $V$  表示节点集合。

##### 2.边容量约束

每条边的流量不能超过其容量限制：

$$0 \leq f(u, v) \leq c(u, v)$$

其中， $c(u, v)$  表示边  $(u, v)$  的容量。

##### 3.最大流量

最大流问题的目标是最大化从源点  $s$  到汇点  $t$  的总流量  $|f|$ :

$$\max |f| = \sum_{v \in V} f(s, v)$$

#### 4.二分查找法

二分查找法是一种在有序数组中查找目标值的高效算法。其基本思想是通过逐步缩小查找范围，将问题规模减半，直到找到目标值或确定目标值不存在。

#### 5.二分查找算法步骤

(1) 初始化左右边界  $T_{\min}$  和  $T_{\max}$ 。

(2) 计算中点  $T_{\text{mid}}$ :

$$T_{\text{mid}} = \frac{T_{\min} + T_{\max}}{2}$$

(3) 判断  $T_{\text{mid}}$  是否满足条件。如果满足，则更新右边界:

$$T_{\max} = T_{\text{mid}}$$

否则，更新左边界:

$$T_{\min} = T_{\text{mid}}$$

(4) 重复步骤2和3，直到左右边界足够接近。

#### 6.最大流二分查找法结合

我们通过二分查找法找到一个最小的时间  $T$ ，使得在此时间内警察能够封锁所有关键交通要道。对于每个给定的  $T$ ，通过最大流算法判断是否能满足封锁要求。

#### 7.具体数学模型

目标函数:

$$\min T$$

约束条件:

$$\text{s.t.} \quad \begin{cases} \sum_{i \in P} x_{ij} \geq 1, & \forall j \in K \\ \sum_{j \in K} x_{ij} \leq 1, & \forall i \in P \\ x_{ij} \cdot d_{ij} \leq T, & \forall i \in P, \forall j \in K \\ \text{flow}(T) \geq |K| \end{cases}$$

其中,  $P$  为警察局集合,  $K$  为关键交通要道集合,  $d_{ij}$  为警察局  $i$  到路口  $j$  的时间,  $x_{ij}$  为二元决策变量, 表示是否由警察局  $i$  封锁路口  $j$ 。

5.5.3 模型的求解

$$\begin{aligned} &\min T \\ &\text{s.t.} \\ &\begin{cases} \sum_{i \in P} x_{ij} \geq 1, & \forall j \in K \\ \sum_{j \in K} x_{ij} \leq 1, & \forall i \in P \\ x_{ij} \cdot d_{ij} \leq T, & \forall i \in P, \forall j \in K \\ \text{flow}(T) \geq |K| \end{cases} \end{aligned}$$

(3)

5.5.4 模型求解的结果

匹配对: [(2, 16), (4, 48), (7, 29), (8, 30), (10, 12), (11, 24), (12, 22), (13, 23), (14, 21), (15, 28), (16, 14), (19, 38), (20, 62)]

A2	A4	A7	A8	A10	A11	A12
16	48	C	D	E	F	G
A13	A14	A15	A16	A19	A20	
1	2	3	4	5	6	

5.5.5 模型建立的准备

5.5.6 模型的求解

参考文献

[1] 何晓群.多元统计分析.北京：中国人民大学出版社，2012.

[2] 徐维超. 相关系数研究综述[J]. 广东工业大学学报,2012,29(3):12-17.

## 6 附录