

基于 0-1 规划模型的交巡警服务平台设置与调度研究

摘要

交巡警服务平台是在城市市区的交通要道和重要部位设置的,能够快速响应道路突发事故的交警站点,是警方保障公共交通安全的有力工具。然而交巡警服务平台的设置与调度问题涉及到路段案发率和到现场的距离等较多方面的因素,当前仍是一个有待研究的重点问题。本文基于 0-1 整数规划和 Floyd 算法对某市交巡警服务平台的管辖范围、新增平台的数量和位置以及对犯罪嫌疑人抓捕进行了较为深入的研究。

对于问题一,我们首先根据 Floyd 算法计算出 A 区任意两个路口之间的最短距离,其次构建了以出警时间最短和工作量分配最平均为目标的优化模型。由于出警时间最短是首要目标,我们首先保证节点能在 3 分钟内到达,将不能在 3 分钟内到达的 6 个节点就近分配平台;再以工作量最均衡为目标函数求解平台管辖范围。接着根据离封锁点最近的平台确定最大封锁时间为 8.015457 min,在此基础上建立以平均封锁时间最短为目标的优化模型,求解得到警力调度方案,平均封锁时间为 3.55 min。再次确定需要增加平台的具体数量和位置。首要任务是将出警时间尽量减少到三分钟内,故我们第一步确定增设 4 个平台效果最好,位置分别为 29、40、61 和 91 号路口;再确定每一个平台的管辖范围。

对于问题二,我们首先采用解问题一中的方法分别对于 A、B、C、D、E、F 区求出每个区平台管辖范围的最优解,再比较主要的指标,如最大出警时间、出警次数是否均衡等,其中各区最大出警时间相差相当大。为了给出一个合理的方案,我们考虑在 D、E 区分别增设 3 个平台,路口编号分别为 330、362、371、388、393 和 421。其次设接到报案时刻为计时起点,我们先求出 t 时刻时犯罪嫌疑人可能到达的路口集合,再求出该集合的边界,此边界即为需要在 t 时间内封锁住的路口。最后以平均围捕时间最短为目标函数建立优化模型,求解出围捕犯罪嫌疑人的最佳方案,计算得到最短封锁时间为报警后 6.05 min,交巡警抵达路口完成封锁的最长时间为 5.7 min。

关键词 交巡警服务平台 0-1 规划 Floyd 算法 犯罪嫌疑人追捕

一、问题重述

1.1 问题背景

新中国成立以来，特别是改革开放以来，中国交通运输的面貌发生了翻天覆地的变换，道路交通安全也越来越受到大家的重视。此时交通警察就显得格外重要，市区的交通要道和重要部位都应设置交巡警服务平台来维护交通秩序。但毕竟警务资源是有限的，所以就要根据每个地区的实际情况来合理地分配警力。但实际情况总是复杂的，王春梅^[1]等揭示了我国西部派出所设置的诸多不合理之处；吴江宏^[2]对杭州市公安局巡逻指挥系统的设计与实现做出了探讨。所以如何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。

1.2 需要解决的问题

针对市区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。

问题一：

根据附件所提供信息。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。

当发生重大突发事件时，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面封锁。实际中一个平台的警力最多封锁一个路口，试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以解决这些问题。

问题二：

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。如发现有明显不合理处，请给出合理的解决方案。

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逸。现调用全市警力来围堵该嫌疑人，试给出最合理的警力调度方案。

二、问题分析

2.1 问题一的分析

问题一首先要求为 A 区交巡警服务平台分配管辖范围。为了确定哪些平台管辖哪些路口，我们首先要求出 A 区各个路口之间的最短距离，其次需要构建以出警时间最短和工作量分配最平均为目标的优化模型。这是一个多目标优化问题，由于缩短出警时间是最首要的目标，并且题目要求尽量在 3 分钟内赶到现场，故我们首先找出不能在三分分钟内赶到现场的路口节点并将其单独处理，分配给最近的平台；对于其他路口我们再以工作量最平均为目标分配管辖路口。

问题一其次要求给出封锁 13 个交通要道的警力调度方案。我们首先根据离封锁点最近的平台确定最大封锁时间，在此基础上建立以平均封锁时间最短为目标的优化模型，求解得到警力调度方案。

问题一随后要求确定需要增加平台的具体数量和位置。首要任务是将出警时间尽量减少到三分分钟内，故我们第一步确定增设 2、3、4、5 个平台的最佳位置，使得最大出警时间最小；再确定每一个平台的管辖范围使得工作量尽量均衡。

2.2 问题二的分析

问题二首先要求分析该市现有交巡警服务平台的设置方案是否合理并给出合理方案。为评价该市交巡警平台设置的合理性，我们首先采用解问题一中的方法分别对于 A、B、C、D、E、F 区求出每个区平台管辖范围的最优解，再比较主要的指标，如最大出警时间、出警次数是否均衡等，观察是否合理。为了给出一个合理的方案，我们考虑在出警时间长、工作量大的区增设新的平台。

问题二其次要求给出搜捕嫌疑犯的最佳围捕方案，设接到报案时刻为计时起点，我们先求出 t 时刻时犯罪嫌疑人可能到达的路口集合，再求出该集合的边

界，此边界即为需要在 t 时间内封锁住的路口。最后以平均围捕时间最短为目标函数建立优化模型，求解出围捕犯罪嫌疑人的最佳方案。

三、模型假设

为简化分析，本文做出如下假设：

- (1) 不考虑各个平台警力流动的情况，即每个交巡警服务平台的警力大致相同；
- (2) 各个平台需要管辖自身所在路口，且一个路口至多由一个平台管辖；
- (3) 一个平台的警力最多封锁一个路口或出口；
- (4) 将最大出警时间最短作为首要目标，即保证最大出警时间最短后再考虑各平台工作量的均衡性；
- (5) 假设围捕时犯罪嫌疑人了解警方动态，即封锁时必须将犯罪嫌疑人行动范围完全封锁。

四、符号说明

表 4.1 为本文中用到的主要符号及说明。

表 4.1 本文中用到的主要符号及说明

符号	含义	单位
X, X^*	平台管辖路口所用决策矩阵	-
T, T^*	在决策矩阵下，各路口到其所属平台的最短出警时间矩阵	min
W	案发率向量	次
G	出警次数（工作量）向量	次
$Dist, Dist^*$	路口两两之间的最短距离矩阵和有效最短路径矩阵	m
r	放置新增平台的最优位置所用决策向量	-
$Contain$	犯罪嫌疑人路径的包围圈	-

五、问题求解

5.1 问题一的求解

本问要求为交巡警服务平台分配管辖范围，故先采用 Floyd 算法求出 A 区任意两个路口之间的最短路径，接下来针对出警时间最短和任务分配最均衡两个目标进行优化。

5.1.1 构造最短路径矩阵

我们采用 Floyd 算法计算 A 区任意两个路口节点之间的最短距离。Floyd 算法^[3]，又称 Floyd-Warshell 算法，是解决任意两点间最短路径的一种算法，其原理如图 5.1.1.

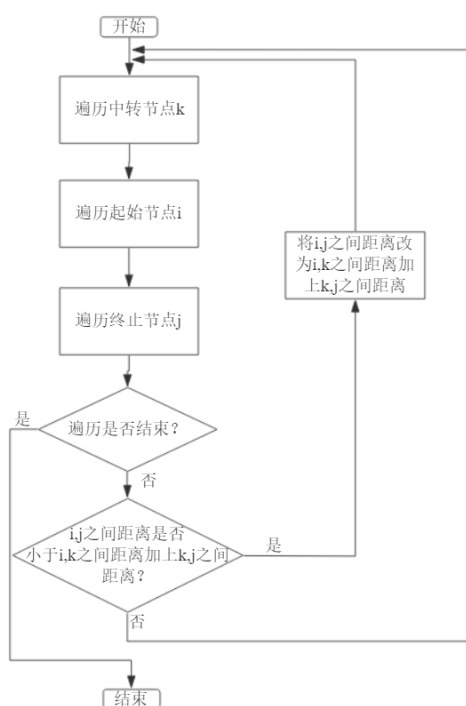


图 5.1.1 Floyd 算法流程图

设 A 是一个包含 A 区任意两个结点之间路径的矩阵，任意两个节点 i, j 之间的路径长度为 $A_{i,j}$ 。若两点之间没有直达路径，则其路径长度用 ∞ 表示。设中转

节点为 k , $k \neq i, j$, 遍历每一个中转节点 k , 若经过中转节点 k 后路径长度更小, 则用更小的路径长度代替原来的路径长度, 即

$$\begin{aligned} & \text{if } A_{i,j} > A_{i,k} + A_{k,j} \\ & \text{then } A_{i,j} = A_{i,k} + A_{k,j} \end{aligned}$$

根据附件中提供的数据, 可以使用 Matlab 软件编程得到最短路矩阵 $Adist$ 。具体内容参见支撑材料 $Adist.xlsx$ 。

5.1.2 构造平台管辖范围的优化模型

交巡警平台的设置有两方面的约束: 一是要使得服务平台管辖区域内出现突发事件时, 警车到达事发地的时间尽可能短; 二是要使各个交巡警服务平台的工作量尽可能均衡, 平衡平台之间的负荷。

根据第一个方面的限制, 我们首先构造决策矩阵 X 。矩阵 X 的每一行代表平台, 每一列代表路口。其中对第 i 行、第 j 列的元素 $X_{i,j}$, 有

$$\begin{aligned} X_{i,j} &= \begin{cases} 0, j \text{路口不属于} i \text{平台管辖} \\ 1, j \text{路口属于} i \text{平台管辖} \end{cases} \\ & i = 1, 2, \dots, 20 \\ & j = 1, 2, \dots, 92 \end{aligned} \quad (5.1.1)$$

构造矩阵 B , 矩阵 B 的每一行代表平台, 每一列代表路口, 其中第 i 行、第 j 列的元素 $B_{i,j}$ 代表从平台 i 到路口 j 平台 i 的最短路径长度, $i = 1, 2, \dots, 20$; $j = 1, 2, \dots, 92$ 。构造矩阵 T , 有

$$\begin{aligned} T_{i,j} &= X_{i,j} \cdot B_{i,j} \\ & i = 1, 2, \dots, 20 \\ & j = 1, 2, \dots, 92 \end{aligned} \quad (5.1.2)$$

T 的某一元素 $T_{i,j}$ 代表在决策矩阵下, A 区各路口到其所属平台的最短出警时间。故对某一个平台 i 而言, 到达现场最大时间为 $\max T_{i,j}, 1 \leq j \leq 92$ 。

根据第二个方面的限制, 我们构造发案向量 W , 其中向量的每一个分量 W_j 代表路口 j 的发案量。构造工作量矩阵 G ,

$$G = X \cdot W \quad (5.1.3)$$

若要各个平台之间的工作量尽可能均衡, 则要求 G 的标准差 $\sigma(G)$ 最小, 其中

$$\sigma G = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (G_i - G)^2} \quad (5.1.4)$$

由于第一个方面的限制，即出警时间最短是最首要的目标；且根据题意出警时间应尽量不大于三分钟，故我们首先找出出警时间不能满足在三分钟以内的节点，即 28、29、38、39、61 和 92 号节点，对其进行单独处理；再在剩余的节点中将出警时间不大于三分钟作为限制条件，优化模型如式 (5.1.5)。

$$\begin{aligned} & \min \sigma(G); \\ s.t. & \left\{ \begin{array}{l} \sum_{i=1}^{20} X_{i,j} = 1, j = 1, 2, \dots, 92 \\ \sum_{j=1}^{92} X_{i,j} \geq 1, i = 1, 2, \dots, 20 \\ \max T_{i,j} \leq 3, 1 \leq i \leq 20, 1 \leq j \leq 92 \\ X_{i,j} = 0 \text{ or } 1, \text{ 且 } i = j \text{ 时取 } 1, \\ i = 1, 2, \dots, 20, j = 1, 2, \dots, 92 \end{array} \right. \end{aligned} \quad (5.1.5)$$

编写 Lingo 程序求解，可得各个交巡警平台所管辖节点的范围、最大出警时间和工作量（出警次数）如表 5.1.1。

表 5.1.1 各个交巡警平台所管辖节点的范围、最大出警时间和工作量

平台编号	管辖节点						最大出警时间(min)	出警次数
1	69	71	74	77	79	80	1.758346	7.4
2	39	40	72	76			3.682186	7.1
3	44	54	55	64	67	70	2.971886	7.7
4	57	60	62	63	65	66	2.835638	7.3
5	49	53	56	59			2.083697	6.1
6	50	51	52	58			2.384147	6.1
7	30	48	61				4.190202	6.5
8	35	45	47				2.079663	6.8
9	32	34	36				1.769033	6.4
10							0	1.6
11	26	27					1.643303	4.6
12	25						1.788854	4.0
13	21	22	23	24			2.708314	8.5

14							0	2.5
15	28	29	31				5.700525	6.4
16	33	37	38	46			3.405877	6.5
17	41	42	43				1.791112	7.0
18	84	85	87	90	91		2.629628	7.0
19	68	73	75	78	81	82	2.787958	7.7
20	83	86	88	89	92		3.601269	7.3

从表中可以看到，多数平台管辖节点的数量为3~6个，其中10、14号平台管辖节点仅为自身，这是该平台处距离其他节点较远所致。各平台管辖区域如图5.1.2所示，区域内部由线段连接。由表5.1.1中数据可算得平均出警时间为2.49分钟，满足尽可能小的原则；平均出警次数为6.225次，各平台工作量较为均衡，模型效果好。

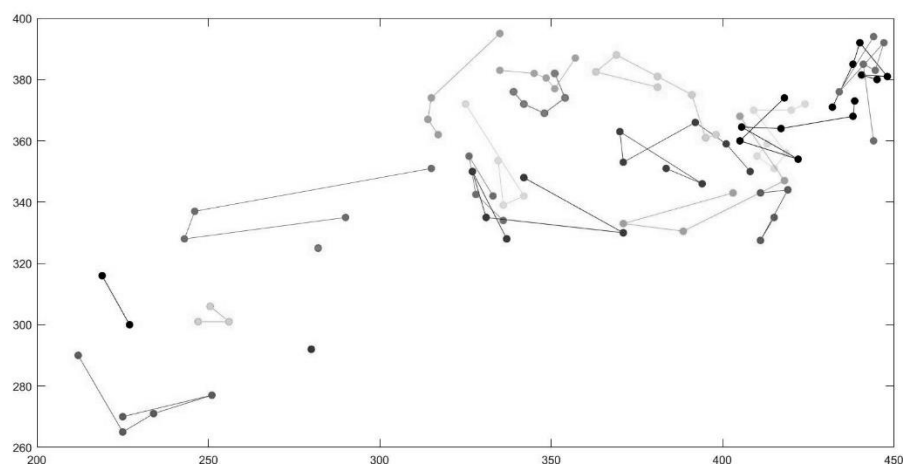


图 5.1.2 各平台管辖区域

5.1.3 构建 A 区封锁道路警力调度方案

为了快速封锁 13 条交通要道，必须选择 13 个最优平台使得赶赴交通要道的时间最短。为此建立决策矩阵 X^* ，矩阵的第 i 行代表平台 i ；第 j 列代表路口 j ，则

$$X^*_{i,j} = \begin{cases} 0, j \text{路口不由} i \text{平台封锁} \\ 1, j \text{路口由} i \text{平台封锁} \end{cases} \quad (5.1.6)$$

$$i = 1, 2, \dots, 20$$

$$j = 1, 2, \dots, 13$$

构造矩阵 B^* ，矩阵 B 由原矩阵 B 中 13 个路口的列组成。构造矩阵 T^* ，有

$$T^*_{i,j} = X^*_{i,j} \cdot B^*_{i,j} \quad (5.1.7)$$

$$i = 1, 2, \dots, 20$$

$$j = 1, 2, \dots, 13$$

T^* 的某一元素 $T^*_{i,j}$ 代表在决策矩阵 X^* 下，A 区各平台到其应封锁路口的最短封锁时间。故对某一个平台 i 而言，到达现场最大时间为 $\max T^*_{i,j} = 8.015457, 1 \leq j \leq 13$ 。

要建立最优的全封锁模型，首先需要使得最大封锁时间 $\max T^*_{i,j}, 1 \leq i \leq 20, 1 \leq j \leq 13$ 最小，记为 T^*_{max} 。在确定最大封锁时间 T^*_{max} 以后还应该使得平均封锁时间 T 最小，

$$T = \frac{1}{13} \sum_i \sum_j T_{i,j}, 1 \leq i \leq 20, 1 \leq j \leq 13$$

故建立优化模型如式 5.1.8.

$$\begin{aligned} & \min T; \\ s. t. & \left\{ \begin{aligned} & \sum_{i=1}^{20} X^*_{i,j} = 1, j = 1, 2, \dots, 13 \\ & \sum_{j=1}^{13} X^*_{i,j} \leq 1, i = 1, 2, \dots, 20 \\ & \max T^*_{i,j} \leq T^*_{max}, 1 \leq i \leq 20, 1 \leq j \leq 13 \\ & X^*_{i,j} = 0 \text{ or } 1, \text{ 且 } i = j \text{ 时取 } 1, \\ & i = 1, 2, \dots, 20, j = 1, 2, \dots, 13 \end{aligned} \right. \quad (5.1.8) \end{aligned}$$

编写 Lingo 程序求解，解得封锁各节点的平台编号和封锁用时如表 5.1.2.

表 5.1.2 封锁各节点的平台编号和封锁用时

平台编号	封锁节点编号	封锁节点路口	封锁用时(min)
2	11	38	3.982186
4	13	62	0.35

5	12	48	2.475826
7	9	29	8.015457
8	10	30	3.06082
9	3	16	1.53254
10	5	22	7.707918
11	7	24	3.805274
12	1	12	0
13	6	23	0.5
14	4	21	3.264966
15	8	28	4.751842
16	2	14	6.741662

从表 5.1.2 中可以看出最大封锁时间约为 8.02 分钟，计算得平均封锁时间约为 3.55 分钟，均较短，故模型效果较好。

5.1.4 确定增设交巡警平台的数量和位置

为了进一步减少出警时间、平衡平台之间的工作量，我们现考虑在 2~5 个路口处增设平台。^[4]由于首要原则是将出警时间尽量减少到三分钟内，故我们第一步确定增设 2、3、4、5 个平台的最佳位置，使得最大出警时间最小；再确定每一个平台的管辖范围使得工作量尽量均衡。

首先分别考虑增设 2、3、4、5 个平台时的放置新增平台的最优位置，使得最大出警时间最小。设决策向量 \mathbf{r} ， j 为路口，有式 5.1.9.

$$r_j = \begin{cases} 1, j \text{ 为平台} \\ 0, j \text{ 不为平台} \end{cases}, 1 \leq j \leq 92 \quad (5.1.9)$$

设 92 个路口两两之间最短距离组成的矩阵为 $Dist_{92 \times 92}$ ，有效最短路矩阵为 $Dist^*_{92 \times 92}$ ，其中有效最短路矩阵 $Dist^*_{92 \times 92}$ 代表从路口到平台的最短距离，即不是平台 ($r_j = 0$) 的列 j 的值均为 ∞ 。计算方法为式 5.1.10.

$$Dist^*_{i,j} = Dist_{i,j} \cdot r_j + [1 - r_j] \times inf \quad (5.1.10)$$

其中 inf 是一个相对于 $\frac{1}{1-r_j}$ 的低阶无穷大量, 相对于 $Dist_{i,j}$ 任何一个元素的高阶无穷大量, 在程序中定义为 100.

规划模型如式 5.1.11.其中目标函数意为先对每一行求最小值, 该最小值向量即为每一个路口到所有可能平台距离的最小值; 再对每一列求最大值, 意为所有可能的平台到所有路口距离的最大值, 最后再求最小值, 该值即为最大出警时间对应的出警距离。

$$\begin{aligned} \min \quad & \max_{1 \leq j \leq 92} \min_{1 \leq i \leq 92} Dist_{i,j}^* \\ s. t. \quad & \begin{cases} \sum_{j=1}^{92} r_j = P \\ r_j = 1, 1 \leq j \leq 20 \end{cases} \end{aligned} \quad (5.1.11)$$

使用 Lingo 软件编程得到增加 2、3、4、5 个平台时, 新增平台所在路口位置和最大出警时间 (见表 5.1.3) .

表 5.1.3 新增平台所在路口位置和最大出警时间

	新增 2 个平台	新增 3 个平台	新增 4 个平台	新增 5 个平台
	29	29	29	22
	48	36	40	28
路口位置编号		48	61	38
			91	61
				91
最大出警时间/min	3.682186	3.601269	2.708314	2.385372

从表 5.1.3 中可以看到, 新增 4 个平台时已经可以将最大出警时间压缩至 3 分钟以内, 故我们具体考察新增 4 或 5 个平台 的情况。为了进一步平衡工作量, 我们按照 5.1.2 的方法, 编写 Lingo 程序确定各个平台的管辖范围, 使得各个平台的工作量尽可能均衡。

新增 4 个平台时各平台的出警次数如表 5.1.4 所示，各平台的平均出警次数为 5.1875，目标函数（标准差，见式 5.1.5）值为 1.595909.

表 5.1.4 新增 4 个平台时各平台的出警次数

平台编号	出警次数	平台编号	出警次数
1	6.4	13	8.5
2	5.7	14	2.5
3	6	15	3.7
4	5.8	16	5.6
5	6.3	17	6.1
6	5.9	18	6.1
7	5.9	19	6.1
8	5.5	20	5.7
9	6	29	2.7
10	1.6	40	6
11	4.6	61	2
12	4	91	5.8

新增 5 个平台时各平台的出警次数如表 5.1.5 所示，各平台的平均出警次数为 4.98，目标函数（标准差，见式 5.1.5）值为 1.481351.

表 5.1.5 新增 5 个平台时各平台的出警次数

平台编号	出警次数	平台编号	出警次数
1	6.4	14	2.5
2	6.6	15	3.7
3	6	16	5.3
4	5.8	17	6.1
5	6.1	18	6.4
6	6.1	19	6
7	6.2	20	6.2
8	5.4	22	3.8

9	6.1	28	2.7
10	1.6	38	4.3
11	4.6	61	2
12	4	91	5.9
13	4.7		

由表 5.1.4、表 5.1.5 结果可见，新增 5 个平台比起新增 4 个平台对目标函数值，即出警次数的标准差的降低作用不大，考虑到修建平台会消耗较多成本，故我们认为新建 4 个平台较为合适，新增平台位置见表 5.1.3，使用 Lingo 软件按 5.1.2 节方法计算得到平台管辖区域如表 5.1.6.

表 5.1.6 新建四个平台时各个交巡警平台所管辖节点的范围、最大出警时间和出警次数

平台编号	管辖节点编号						最大出警时间(min)	出警次数
1	1	65	69	70	71	73	2.359909	6.4
2	2	66	68	74	75		2.84943	5.7
3	3	44	54	55	64		2.270886	6
4	4	57	60	62	63		1.868154	5.8
5	5	49	53	56	58		2.301889	6.3
6	6	50	51	52	59		2.324786	5.9
7	7	30	33				1.650077	5.9
8	8	32	47				2.079663	5.5
9	9	35	36	45			1.095084	6
10	10						0	1.6
11	11	26	27				1.643306	4.6
12	12	25					1.788854	4
13	13	21	22	23	24		2.708314	8.5
14	14						0	2.5
15	15	31					2.968164	3.7
16	16	34	37	46			2.379097	5.6
17	17	41	42	72			2.597337	6.1
18	18	80	82	83	89		1.823402	6.1

19	19	67	76	77	78	79	2.650733	6.1
20	20	81	84	86			2.677943	5.7
29	28	29					0.948683	2.7
40	38	39	40	43			2.714419	
61	48	61					2.9	
91	85	87	88	90	91	92	2.002498	

5.2 问题二的求解

5.2.1 分析现有交巡警服务平台设置的合理性

为了分析全市现有平台设置的合理性，我们首先仿照 5.1.1 节采用 Floyd 算法算出每个区内部道路的最短距离，再采用 5.1.2 节的方法算出每个区平台管辖范围的最优解，最后比较主要的指标，如最大出警时间、平均出警次数等，这里的最大出警时间指的是某个区内各平台最大出警时间的最大值。

表 5.2.1 全市六个区平台管辖的相关指标

区域	最大出警时间(min)	平均出警次数	出警次数标准差
A	5.70053	6.225000	1.711980
B	4.470312	8.3	1.603122
C	6.860544	11.01176	4.105360
D	16.06282	7.533333	2.034699
E	19.10506	7.96	3.370223
F	8.479848	7.28	4.431782

从表 5.2.1 可看出各区的出警次数标准差相差不大但最大出警时间差别非常大，D、E 区远大于其他区。可见各区的警力资源分配相当不均衡，现有平台设置明显不合理，并且主要体现在平台位置设置不合理，从而使得各区的最大出警时间相差较大。由于现有平台已经建成，所以只能采取增加平台的方法。为了尽量减小最大出警时间，我们考虑在 D、E 区增设平台。

5. 2. 2 对全市交巡警服务平台设置的改进

仿照 5.1.4 节的方法，我们采用 Lingo 软件编写程序，分别计算在 D、E 区增加 1、2、3、4 个平台的结果，由于出警次数的标准差差别不大，这里我们主要考虑对最大出警时间的影响。计算结果如表 5.2.2.

从表 5.2.2 中可以看出，无论是 D 区还是 E 区，在增加平台数量超过 3 后最大出警时间的下降都不大，故在 D、E 区增加 3 个平台最为合适。新增平台的位置如表 5.2.3.

表 5.2.2 在 D、E 区增加 1、2、3、4 个平台的结果

区域	增加平台数量	最大出警时间(min)	区域	增加平台数量	最大出警时间(min)
D	1	8.106913	E	1	8.573172
	2	7.808483		2	8.118817
	3	4.840709		3	6.580274
	4	4.724369		4	6.232651

表 5.2.3 D、E 区新增平台位置

区域	增加平台位置	区域	增加平台位置
D	330	E	388
	362		393
	371		421

5. 2. 3 建立追捕犯罪嫌疑人的最佳围堵方案

犯罪嫌疑人在案发到报警的三分钟内已经驾车逃逸。为方便分析，现取犯罪嫌疑人的车速 $v_{\text{犯}}$ 也为60km/h。设报警时刻为计时起点，在 t 时刻处犯罪嫌疑人可能逃逸的最大距离为 $d_{\text{逃逸}} = v_{\text{犯}} \cdot (t + 3)$ ，采用 5.1.1 节的 Floyd 算法计算出全市任意两个路口之间的最短距离，再找出犯罪嫌疑人可能到达的最远路口集合，将此集合记为 E 。以 E 集合中所有路口为起点，将所有终点的集合记为 C ，再从 C 集合中剔除 E 集合的路口，得到犯罪嫌疑人路径的包围圈（需封锁路口） $Contain$ ，图 5.2.1 展示了三者关系。

$$Contain = C \setminus E$$

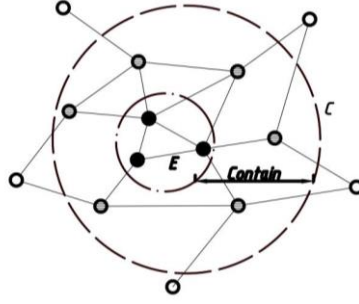


图 5.2.1 $C, E, Contain$ 关系

现考虑如何调动全市警力尽快赶往该包围圈，我们采取类似于 5.1.3 节的封锁方法探究不同的 t 值下的围捕方案。规划模型如式 5.2.1.图 5.2.2 为本方法的流程图。

$$\begin{aligned}
 & \min T; \\
 s.t. \left\{ \begin{aligned}
 & \sum_{i=1}^{80} X^*_{i,j} \leq 1, j = 1, 2, \dots, 582 \\
 & \sum_{j \in Contain} X^*_{i,j} = 1, i = 1, 2, \dots, 80 \\
 & \sum_{j \notin Contain} X^*_{i,j} = 0, i = 1, 2, \dots, 80 \\
 & \max T^*_{i,j} \leq t, 1 \leq i \leq 80, 1 \leq j \leq 582 \\
 & X^*_{i,j} = 0 \text{ or } 1 \\
 & i = 1, 2, \dots, 80, j = 1, 2, \dots, 582
 \end{aligned} \right. \quad (5.2.1)
 \end{aligned}$$

类似于 5.1.3 节，其中 X^* 为决策矩阵， T^* 的某一元素 $T^*_{i,j}$ 代表在决策矩阵 X^* 下，全市各平台到其应封锁路口的最短封锁时间。

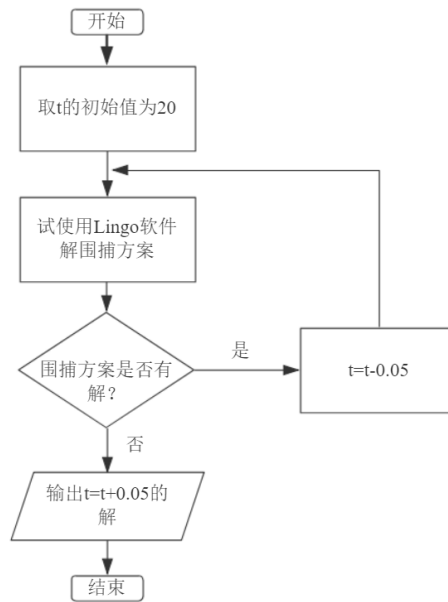


图 5.2.2 确定围捕方案的流程图

依据图 5.2.2 流程图进行计算，得到能够使围捕方案有解的 $\min t = 6.05$ ，即最短封锁时间为报警后6.05min，围捕方案如表 5.2.4 所示。

表 5.2.4 平台围捕方案

平台	位置	封锁路口	封锁时间(min)
A1	1	68	1.207107
A4	4	62	0.35
A14	14	14	0
A15	15	29	5.700525
A17	17	41	0.85
A19	19	76	1.432099
C3	168	168	0
C5	170	217	3.778924
C6	171	240	4.903991
C7	172	218	0.4716991
C9	174	215	4.860321
F1	475	558	2.052614
F6	480	562	1.941649
F7	481	549	1.835756
F8	482	482	0

从表 5.2.4 可以看出，为完成对犯罪嫌疑人在报警后的6.05min 的有效完全封锁，全市交巡警需封锁 15 个平台，示意图为图 5.2.3，其中*组成包围圈（需封锁路口），·表示案发地点。交巡警抵达路口完成封锁的最长时间为5.7min。

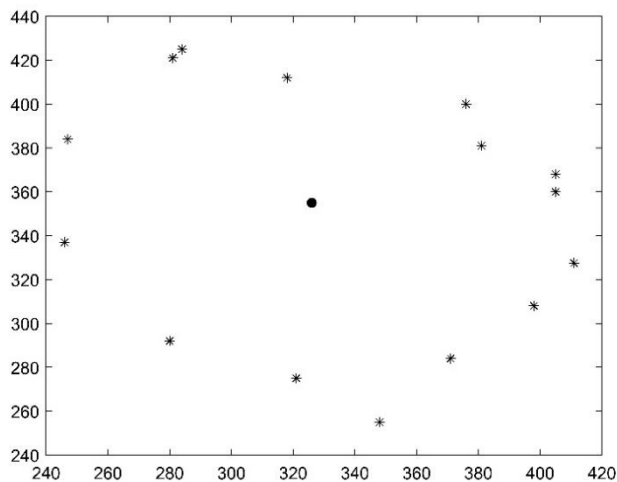


图 5.2.3 需封锁平台图示意

六、模型评价与推广

6.1 模型的优缺点

本文将交巡警平台的设置和调度问题转化为 0-1 整数线性规划问题，建立了以最小工作量标准差为目标函数的求解模型，大大简化了问题的求解难度。本模型分析全面深入，详细阐述了交巡警平台设置和调度的具体方案；主要为定量分析，结果准确可靠；模型适应性广，可移植性强，可以用于不同城市、不同类型的平台设置。

另一方面，本文做了较多假设，且没有考虑交警、犯罪嫌疑人的反应时间；认为速度恒定，且事故总在路口处发生，故与现实情况有一定差别。

6.2 模型的推广

本文所述模型不仅可以设置和调度交巡警平台，也可以用于消防、急救、物流配送等行业领域。另外，可以构造加权效用函数以平衡最大出警时间和工作量标准差之间的关系，根据不同需求得到最佳解。

参考文献

- [1]王春梅, 慕三英. 我国西部农村公安派出所规划布局问题研究 [J]. 江西公安专科学校学报, 2006(2) : 31-33
- [2]吴江宏. 杭州市公安局巡逻指挥系统的设计与实现 [D]. 成都: 电子科技大学, 2012: 36-37
- [3] 邱晓鹏, 王丽君. 基于 Floyd 算法的最短路径优化研究 [J]. 太原师范学院学报(自然科学版), 2019, 18(02): 53-67.
- [4]孙庆珍, 李明, 贾燕. 基于多目标决策的城市应急设施选址问题 研究 [J]. 科技和产业, 2014, 14(6) : 5-8.

附录

Floyd 算法求最短路矩阵

floyd.m

```
%%说明：本程序使用 floyd 算法计算 A 区之间两点间的最短距离
%%
%清空工作区
clear;
clc;
%读取数据
route_A = csvread("route.A.csv",1);
%第一列为起点，第二列为终点，第三列为距离
%建立邻接矩阵
Adist = zeros(92);
for row = 1 : 140
    src = route_A(row,1);
    dst = route_A(row,2);
    dist = route_A(row,3);
    Adist(src,dst) = dist;
    Adist(dst,src)=dist;
end
for row = 1 : 92
    for col = 1: 92
        if(Adist(row,col) == 0)
            Adist(row,col) = Inf;
        end
    end
    Adist(row,row) = 0;
end
%floyd 算法
[Aroute,Adist] = floydfunc(Adist);
```

floyd.m 配套函数

floydfunc.m

%%说明：本函数是 Floyd 算法的具体函数，与 floydtest.m 配合使用。

%%

```
function [Aroute,Adist] = floydfunc(Adist)
```

%建立路由矩阵

```
Aroute = zeros(92,92);
```

```
for row = 1 : 92
```

```
    for col = 1 : 92
```

```
        Aroute(row,col) = col;
```

```
    end
```

```
end
```

%对于起点 i, 终点 j, 遍历中转结点 k

```
for k = 1 : 92
```

```
    for i = 1 : 92
```

```
        for j = 1 : 92
```

```
            if(Adist(i,j)>Adist(i,k)+Adist(k,j))
```

```
                Adist(i,j)=Adist(i,k)+Adist(k,j);
```

```
                Aroute(i,j)=Aroute(i,k);
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

解 5.1.2 优化模型所用 Lingo 程序

lingo1

MODEL:

sets:

cop/1..20/:maxtime,load; !定义平台集合, 每个平台最大出警时间, 出警次数;

node/1..92/:crimerate;!定义路口集合, 每个路口的案发率;

```

link(cop,node):x,distance;!定义决策矩阵, 每个路口之间的最短路;
endsets
data:!导入案发率, 最短路数据;
crimerate=@ole('C:\Users\12563\Desktop\大二下\数学建模\train
1\crimerate.xlsx','crimerate');
distance=@ole('C:\Users\12563\Desktop\大二下\数学建模\train 1
\A_distance.xlsx','A_distance');
enddata
@for(cop(i):maxtime(i)=@max(node(j):x(i,j)*distance(i,j)););!计算
最大出警时间;
@for(cop(i):load(i)=@sum(node(j):x(i,j)*crimerate(j)););!计算出警
次数;
meanload=1/20*@sum(cop(i):load(i));!计算平均出警次数;
min=@sqrt(1/20*@sum(cop(i):(load(i)-meanload)*(load(i)-
meanload));));!目标函数;
@for(link(i,j)|j#ne#28#and#j#ne#29#and#j#ne#38#and#j#
ne#39#and#j#ne#61#and#j#ne#92:x(i,j)
*distance(i,j)<=30);!出警时间小于 3min;
@for(node(j):@sum(cop(i):x(i,j))=1);!一个路口由一个平台管理;
@for(cop(i):@sum(node(j):x(i,j))>=1);!一个平台至少管理一个路口;
@for(link(i,j):@BIN(x(i,j)));!0,1 规划;
@for(cop(i):x(i,i)=1);!各平台管理自身所在路口;
!处理不能在 3min 内到达的特殊路口;
x(15,28)=1;
x(15,29)=1;
x(16,38)=1;
x(2,39)=1;
x(7,61)=1;
x(20,92)=1;

```

解 5.1.3 优化模型所用 Lingo 程序

Lingo2

MODEL:

sets:

cop/1..20/:locktime;!定义平台集合, 封锁时间;

exit/1..13/;!定义出口集合;

link(cop,exit):y,exit_distance;!定义决策矩阵, 平台与出口最短路;

endsets

data:!导入最短路数据;

exit_distance=@ole('C:\Users\12563\Desktop\大二下\数学建模
\train 1

\A_distance.xlsx','A_exit_distance');

enddata

@for(cop(i):locktime(i)=@sum(exit(j):y(i,j)*exit_distance(i,j)););!

计算平台封锁时间;

min=1/13*@sum(cop(i):locktime(i));!目标函数: 最大封锁时间最短下
平均时间最短;

@for(exit(j):@sum(cop(i):y(i,j))=1);!一个出口由一个平台封锁;

@for(cop(i):@sum(exit(j):y(i,j))<=1);!一个平台最多封锁一个出口;

@for(link(i,j):@BIN(y(i,j)));!0,1 规划;

@max(cop(i):locktime(i))<80.2;!最大封锁时间;

寻找新增平台位置的 lingo 程序

Lingo3

MODEL:

sets:

cop/1..92/:r;!定义(潜在)平台集合, 决策变量;

node/1..92/:mintime;!定义路口集合, 最短被接管时间;

link(cop,node):distance,ufdistance;!定义最短路, 有效最短路;

endsets

data:!导入最短路数据;

distance=@ole('C:\Users\12563\Desktop\大二下\数学建模\train 1

```

\A_distance.xlsx','All_distance');
enddata
min=@max(node(j):mintime(j));!目标函数：最大最短被接管时间取得
最小值;
copnum=25;!平台总数=22,23,24,25;
@for(link(i,j):ufdistance(i,j)=r(i)*distance(i,j)+(1-r(i))*100;);!计算
有效最短路;
@for(node(j):mintime(j)=@min(cop(i):ufdistance(i,j)););!计算最短
被接管时间;
@sum(cop(i):r(i))=copnum;!平台总数限制;
@for(cop(i)|20#ge#i:r(i)=1);!前 20 个路口为平台;
@for(cop(i):@BIN(r(i));;!0,1 规划;

```