

# 承诺书

赛区评阅编号（由赛区组委会评阅前进行编号）：

## 2011 高教社杯全国大学生数学建模竞赛

### 编 号 专 用 页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：


全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

# 交巡警服务平台的设置与调度模型

## 摘要

本文讨论了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、调度警务资源等问题。

问题一，对于 A 区交巡警服务平台管辖范围分配，即要考虑路口节点的归属，首先将其转化为图论问题，用改进的 Dijkstra 算法求任意两个节点之间最短路径，确定在 3 分钟内各交巡警服务平台能够到达的路口节点，作为各交巡警服务平台的管辖范围初步划分。对于同时属于多个各交巡警服务平台的管辖范围的路口节点，按照时间最短原则确定其归属。对于局部工作量和时间不均衡的情况，建立 0-1 整数规划模型进一步优化服务平台管辖范围，得到最终的管辖范围分配方案。

运用所得管辖方案解决发生重大事件时调用全区 20 个服务平台对 13 条交通要道的封锁问题。用类似于查表法查找 13 个交通要道所属的交巡警服务平台，对某些一个平台管辖多个交通要道的情况按路口节点到达平台时间的长短来进一步调整，从而得到该区交巡警服务平台警力合理的调度方案。

针对 1、2 号平台管辖路口节点数远远大于其它服务平台的不合理性，建立工作量和时间均衡的多目标线性规划模型，具体求解时将其转化为出警时间和最小的单目标线性规划，对 1、2 号服务平台进行局部优化，得出需在 69 号新设 1 个服务平台；同时，针对 28、29、92 号路口节点 3 分钟内不能到达的问题，根据实际情况考虑在 28 或 29 号位置设置一个服务平台单独管理这两个路口节点。综上，仅需要增加 2 个服务平台很大程度上缓解现有交巡警服务平台的工作量不均衡和某些地方出警时间过长的情况。

问题二，针对全市 5 个区用问题一的最短路模型，考虑工作量和时间均衡原则，求解出该市现有服务平台的管辖方案。分析管辖方案里明显不合理的服务平台将其删除，再利用增设服务平台模型减轻工作量过重的服务平台。

为了解决在 P（第 32 个节点）犯案的犯罪嫌疑人围堵问题，借用图论中“割点”概念，调度全市警力尽快占据“割点”来在最短时间内达到将犯罪嫌疑人围堵在一个固定的区域，最后完成合围所花费的时间为 7.36 分钟。

**关键字：** Dijkstra 算法 管辖范围 0-1 整数规划 多目标线性规划

## 目录

1 问题的提出 .....	3
2 基本假设 .....	3
3 符号说明 .....	3
4 问题分析 .....	4
5 问题一的模型建立与求解 .....	4
5.1 中心城区 A 交巡警平台管辖范围模型 .....	4
5.1.1 管辖范围分配策略 .....	4
5.1.2 基于 Dijkstra 的各平台管辖范围初步分配 .....	5
5.1.3 争议路口节点归属划分 .....	6
5.1.4 基于各服务平台辖区工作量和时间尽量均衡的局部优化 .....	7
5.2 交通要道全封锁的警力调度模型 .....	10
5.3 增设交警服务平台模型 .....	11
6 问题二的模型建立与求解 .....	13
6.1 现有交巡警服务平台设置的合理性评价模型 .....	13
6.2 最佳围堵方案警力调度模型 .....	15
7 结果分析与检验 .....	16
8 模型评价 .....	16
8.1 模型优缺点 .....	16
8.2 模型改进 .....	17
9 参考文献 .....	17
附录 .....	18

## 1 问题的提出

“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（参见附件）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 基本假设

- (1) 每个交巡警服务平台的职能和警力配备基本相同；
- (2) 所有的案件都发生在路口节点处；
- (3) 每一个路口节点只归属一个交巡警服务平台管辖；
- (4) 交巡警服务平台接到报警后立即赶往出事路口节点，即不考虑反应时间；
- (5) 重大刑事案件犯罪嫌疑人驾车逃跑方向随机

## 3 符号说明

符号	符号说明
$v_i$	代表第 $i$ 个交通路口节点 ( $i=1,2, \dots, 582$ , 下同)
$e_{v_i v_j}$	代表第 $v_i$ 点到第 $v_j$ 点的路线
$W_{v_i v_j}$	代表节点 $v_i$ 到节点 $v_j$ 的距离，即边权
$a$	代表警车的速率（单位 km/h）
$S_i$	所有交通路口节点的集合

$T_i$	第 $i$ 个服务平台的出警时间
$Q_i$	第 $i$ 个平台的案发率

## 4 问题分析

问题要求在某市各个区域道路网中划分交巡警平台管辖范围。将道路网中，路口节点看作图中对应的节点  $v_i$  (编号  $i=1,2,3, \dots, 582$ )，其中 A 区域的交巡警平台所在路口节点  $v_i$  ( $i=1,2,3, \dots, 20$ )，B 区域  $v_i$  ( $i=93,94,95, \dots, 100$ )，C 区域  $v_i$  ( $i=166,167,168, \dots, 182$ )，D 区域  $v_i$  ( $i=320,321,322, \dots, 382$ )，E 区域  $v_i$  ( $i=372,373,374, \dots, 182$ )，F 区域  $v_i$  ( $i=475,476,477, \dots, 485$ )；出入市区的路口 ( $i=151,153,177,202,203,264,317,325,328,332,362,387,418,483,541,572,578$ )；

在警务资源有限的条件下，根据城市的辖区地域特征、人口分布、交通状况、治安状况以及出警要求合理地用设置交巡警服务平台管辖范围。其要考虑的相关因素有：

辖区地域特征：城市现有交巡警平台设置及其周边路口节点；

交通状况：城市道路连接状况；

治安状况：城市路口节点案发率；

出警要求：遇到突发事件时尽量能在 3 分钟内赶到。

## 5 问题一的模型建立与求解

由题意，问题一需解决下面三个小问：

(1) 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；

(2) 给出重大突发事件发生时该区交巡警服务平台警力合理的调度方案；

(3) 确定需要增加服务平台的具体个数和位置。

下面分别就此建立模型求解。

### 5.1 中心城区 A 交巡警平台管辖范围模型

本小节主要研究区域 A 的交通网络和该区的 20 个交巡警服务平台，分配各平台的管辖范围（不考虑 A 区服务平台对其它区域的管辖）。

由题目要求，确定各交巡警服务平台分配管辖范围原则如下：

- 各服务平台能够快速出警，突发事件发生尽量在 3 分钟内赶到；
- 各服务平台辖区工作量尽量均衡，即尽量保持各平台接案数量均衡；
- 各服务平台出警时间尽量均衡，且尽量在工作时间以内

#### 5.1.1 管辖范围分配策略

对于 A 区的现有交通网络和交巡警服务平台，进行分配管辖范围拟按如下步骤进行：

- (1) 对数据进行处理，求出所有点任意两点间的距离矩阵；
- (2) 在此基础上以 3 分钟为分界点，人工把大于 3 分钟的点选出来；
- (3) 以各服务平台为中心，以巡警 3 分钟内可达距离为半径，计算各服务平台所能管辖的路口节点；
- (4) 将所有的路口节点自然分成几类：只归属某一个服务平台的节点、可归属两个或两个以上平台的节点、不归属任何一个平台的。
- (5) 将不同类别节点按照采用不同的方式确定其归属。

### 5.1.2 基于 Dijkstra 的各平台管辖范围初步分配

对于图  $G(V, E)$ ，在 A 区  $V = \{v_1, v_2, \dots, v_i\} \quad i \in (1, 92)$

$E = \{e_{v_1 v_1}, e_{v_1 v_2}, \dots, e_{v_j v_j}\} \quad j \in (1, 92)$ ，利用欧式距离  $d_{ij}$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

其中  $(x_i, y_i), (x_j, y_j) \in S \quad i, j \in (1, 92)$

则图  $G = (V, E)$  对应的节点间距离的邻接矩阵  $A(G)$  为：

$$A(G) = \begin{bmatrix} & v_1 & \cdots & v_i \\ \begin{bmatrix} 0 & \cdots & w_{e_i e_1} \\ \vdots & 0 & \vdots \\ w_{e_1 e_i} & \cdots & 0 \end{bmatrix} & v_1 \\ & \vdots \\ & v_i \end{bmatrix}$$

因为各服务平台管辖范围内出现突发事件时，尽量能在 3 分钟内有交巡警到达事发地。警车时速  $60km/h$ ，因此交巡警从服务平台能够达到的最远距离  $l_{\max}$  为：

$$l_{\max} = 60km/h \times \frac{3}{60}h = 3km$$

以每一个服务平台为起点搜索路径长度小于  $3km$  的节点，使其属于该服务平台管辖范围。具体计算，本文运用 **Dijkstra** 算法，计算从服务平台到任意路口节点的最短距离。然后将最短距离大于  $3km$  的节点舍去，生成新的邻接矩阵。经过这样的数据筛选处理，大大地减少了搜索最短路的运算量，提高了程序运行效率。

**Dijkstra** 算法求解服务平台到各节点的最短路

步骤 1：令  $l(u_0) = 0$ ，对于  $v \neq u_0$ ，令  $l(v) = \infty, S_0 = \{u_0\}, i = 0$ 。

步骤 2：对每个  $v \in \overline{S_i} (\overline{S_i} = V \setminus S_i)$ ，用  $\min_{u \in S_i} \{l(v), l(u) + w(uv)\}$  代替  $l(v)$ ，当  $u, v$  不相邻时， $w(uv) = \infty$ 。计算  $\min_{u \in S_i} \{l(v)\}$ ，把达到这个最小值的一个顶点记为  $u_{i+1}$ ，令

$$S_{i+1} = S_i \cup \{u_{i+1}\}$$

步骤 3：若  $i = |V| - 1$ ，则停止；若  $i < |V| - 1$ ，则用  $i+1$  代替  $i$ ，转至步骤 2。

算法结束时，从  $u_0$  到各顶点  $v$  的距离由的最后一次的标号  $l(v)$  给出。在  $v$  进入  $S_i$  之前的标号  $l(v)$  叫  $T$  标号， $v$  进入  $S_i$  时的标号  $l(v)$  叫  $P$ 。算法就是不断修改各个点的标号，直至获得  $P$  标号。若在算法运行过程中，将每一顶点获得  $P$  标号所由来的边在图上标明，

则当算法结束时， $u_0$ 至各个点的最短路也在图上标示出来了。

用 MATLAB 编程求解（程序见附件），计算结果如下见表 5-1。

表 5-1 各服务平台在 3 分钟内能达到的节点

交巡警 平台位 置标号	各服务平台在 3 分钟内能达到的节点																							
1	42	43	44	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	78	79	80	
2	40	42	43	44	66	67	68	69	70	71	72	73	74	75	76	78	76	78						
3	43	44	54	55	64	65	66	67	68	70	76													
4	57	58	60	62	63	64	65	66																
5	47	48	49	50	51	52	53	56	58	59	58	59												
6	47	48	50	51	52	56	58	59	59															
7	30	31	32	33	34	47	48	47	48															
8	31	32	33	34	35	36	37	45	46	47	46	47												
9	31	32	33	34	35	36	37	45	46	46														
10	10																							
11	25	26	27																					
12	25																							
13	21	22	23	24																				
14	14																							
15	31																							
16	33	34	35	36	37	45	46																	
17	40	41	42	43	70	72																		
18	71	72	73	74	77	78	79	80	81	82	83	84	85	87	88	89	90	91	90	91				
19	64	65	66	67	68	69	70	71	73	74	75	76	77	78	79	80	81	82	83					
20	81	82	83	84	85	86	87	88	89	90	91													

5.1.3 争议路口节点归属划分

由表格发现，所有节点可分为三类点：①只归属某一个平台；②可归属两个或两个以上的平台；③不归属任何一个平台。其中的第一类点已经不需要再修正，这里我们仅对二三类点修正，具体步骤如下：

（1）对于可归属两个或两个以上平台的节点，这类点的特点是在一次寻优中可同时归属于几个平台。由于这类点的数目不大，我们采用人工查找距离邻接矩阵的方法，将这些点逐一地归属到距离它最近的平台管辖。

（2）对于不归属任何一个平台（6 个），这类点的特点是在一次寻优中不归属任何一个平台。由于这类点不能实现突发事件发生时 3 分钟内赶到，但也不能因此无人管辖。所以我们强制将这些点进行归属。用同步骤 1 相同的方法，把这些点强制归属到离它最近的平台管辖。

进行修正后点的归属结果见表 5-2。

表 5-2 修正后所有服务平台管辖的交通路口节点

交巡警 平台位 置标号	交巡警服务平台所管辖的路口节点标号	案发率	时间 min
-------------------	-------------------	-----	--------



1	67	68	69	71	73	74	75	76	78	10.3	8.976919
2	40	43	44	70	72	38	39			10.9	13.79393
3	54	55	65	66						5.6	6.900963
4	57	60	62	63	64					6.6	6.92449
5	49	50	51	52	53	56	58	59		9.7	11.31455
6	6									2.5	0
7	30	32	47	48	61					9.6	8.484299
8	33	46								5	1.757701
9	31	34	35	45						8.2	4.077559
10	10									1.6	0
11	26	27								4.6	2.543303
12	25									4	1.788854
13	21	22	23	24						8.5	6.499225
14	14									2.5	0
15	28	29								4.8	10.45237
16	36	37								3.8	1.726454
17	41	42								5.3	1.834886
18	80	81	82	83						6.1	3.094912
19	77	79								3.4	1.432099
20	84	85	86	87	88	89	90	91	92	11.5	12.19368

分析修正后的点归属表格，编号为 6 的交巡警服务平台除了自己以外没有任何一个路口节点归属于它，在保证求解结果的情况下，我们判定 6 号平台属于市区平台设施建设中设置不合理的服务平台。

#### 5.1.4 基于各服务平台辖区工作量和时间尽量均衡的局部优化

在现有服务平台分布不变的情况下，我们对 6 号平台及其周围的服务平台，按出警时间和案发率均衡进行局部更优化。

重新将 6 号平台和其周围的 5、7 号平台重新进行优化分配。我们采用多目标规划来建立模型。

出警时间：  $t_i = \frac{l_i}{a}$   $i = 5, 6, 7$  建立出警时间之和最短的目标函数：

$$\text{Min } S = (\sum_{i=5}^7 T_i) \quad (1)$$

案发率：  $Q_i$  建立各服务平台案发率均等的目标函数：

$$\text{Min } Z = \{|Q_5 - Q_6| + |Q_6 - Q_7| + |Q_7 - Q_5|\} \quad (2)$$

基于多目标线性规划的求解，我们将多目标转化成单目标，即将其中任意一个目标函数转化成约束函数。

现在我们求出平均案发率  $\bar{Q} = \frac{\sum_{i=5}^7 Q_i}{3} = 7.26$ ，(2) 可以转化成三个平台每天处理案发

事件大事相等，但是要达到各个交巡警服务平台工作量完全均等的理想状态是不符合实际情况的。所以我们引入一个松弛量  $\Delta$  来表征案发率在一定区间范围内认为它们的工作量近似均等，由于地域不是很大，我们将一个路口节点处的最大案发率下取整即  $\Delta = 2$

来构造约束条件：

$$5.26 < Q_i < 9.26 \quad i = 5, 6, 7$$

(1) 式目标函数不变，转化为单目标函数求解：

$$\text{Min} S = \left( \sum_{i=5}^7 T_i \right)$$

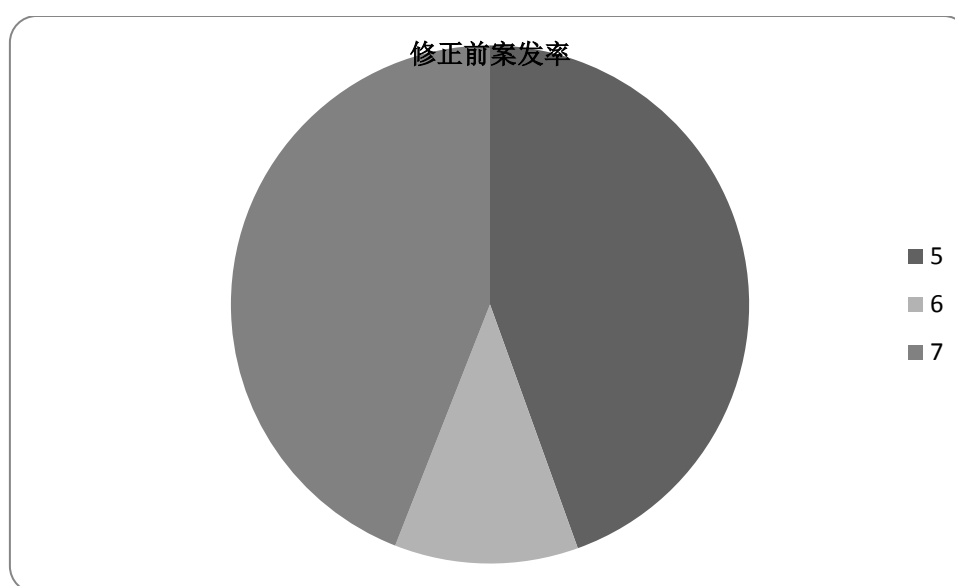
$$\text{st. } 5.26 < Q_i < 9.26 \quad i = 5, 6, 7$$

我们采用 lingdo 软件经过多次迭代求得 5、6、7 号平台的最优节点分配。

**表 5-3 5、6、7 号平台局部最优分配**

交巡警 平台位 置标号	交巡警平台所管辖的路口节点标号					案 发 率	时间 min
5	50	52	53	56	59	6.6	7.283343
6	47	51	58	49		7.2	9.084912
7	30	32	48	61		8	7.203675

表 5-3 我们完成对 5、6、7 号交巡警服务平台管辖范围局部划分，下面我们为 5、6、7 改进前后的总体数据画出树状图



**图 5.1 未对 5、6、7 号平台局部优化前 A 区各服务平台案发率**

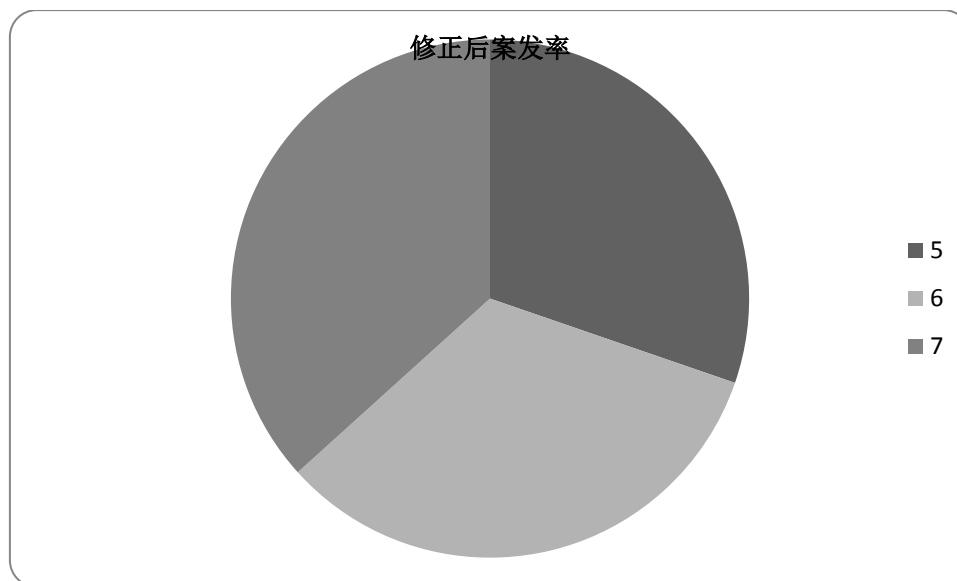


图 5.2 对 5、6、7 号平台局部优化后 A 区各服务平台案发率

对比案发率在 5、6、7 局部优化前后可知，5、6、7 三个服务平台的案发率差异大大缩小了，满足了各服务平台案发率均等的原则，因此优化效果较好。

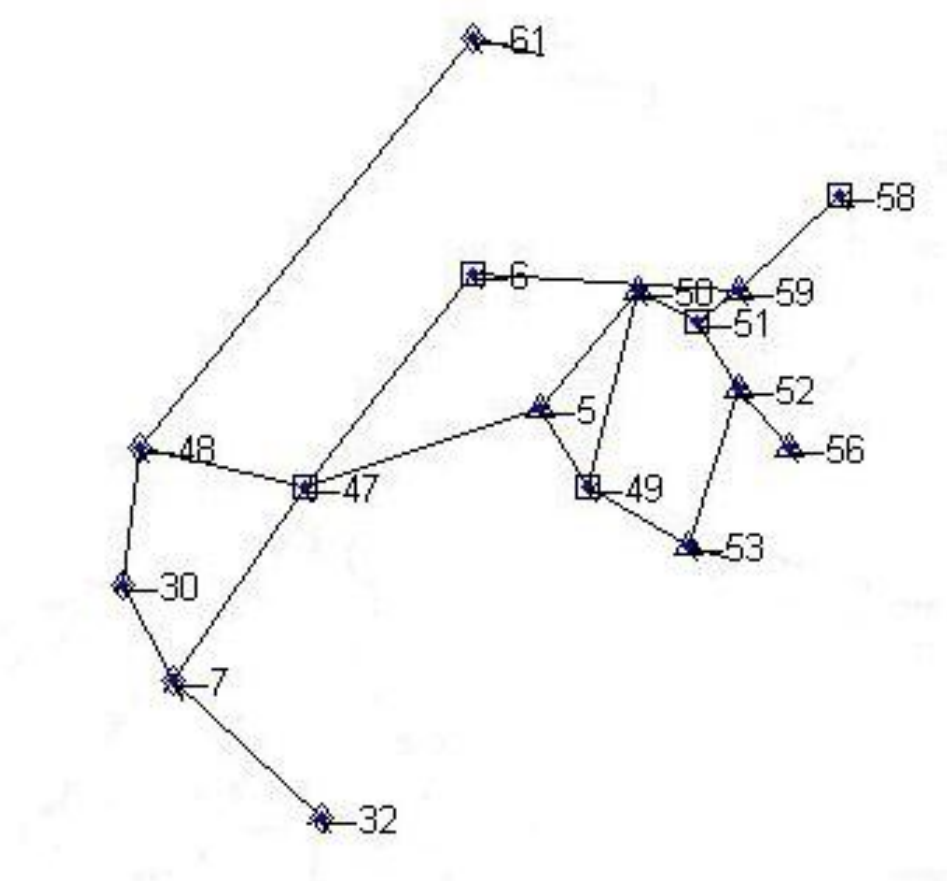


图 5.3 对 5、6、7 号平台局部优化后各自管辖范围示意图

注：△标示归属 5 号服务平台管辖的路口节点  
□标示归属 6 号服务平台管辖的路口节点  
◇标示归属 7 号服务平台管辖的路口节点

表 5-4 A 区 20 个平台的总管辖范围

交巡警 平台位 置标号	交巡警平台所管辖的路口节点标号										案发 率	时间 (min)
1	67	68	69	71	73	74	75	76	78		10.3	8.976919
2	40	43	44	70	72	38	39				10.9	13.79393
3	54	55	65	66							5.6	6.900963
4	57	60	62	63	64						6.6	6.92449
5	50	52	53	56	59						6.6	7.283343
6	47	51	58	49							7.2	9.084912
7	30	32	48	61							8	7.203675
8	33	46									5	1.757701
9	31	34	35	45							8.2	4.077559
10											1.6	0
11	26	27									4.6	2.543303
12	25										4	1.788854
13	21	22	23	24							8.5	6.499225
14											2.5	0
15	28	29									4.8	10.45237
16	36	37									3.8	1.726454
17	41	42									5.3	1.834886
18	80	81	82	83							6.1	3.094912
19	77	79									3.4	1.432099
20	84	85	86	87	88	89	90	91	92		11.5	12.19368

## 5.2 交通要道全封锁的警力调度模型

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，对进入该区的 13 条交通要道实现快速全封锁。而实际中一个平台的警力最多封锁一个路口，我们要用已完成的各平台管辖范围划分来制定出交警服务平台警力合理的调度方案。制定方式如下步骤：

(1) 将 13 个交通要道按其距离最短为约束求出所属平台，得出管辖结果在表 5-5 一二列。

(2) 由于一个服务平台的警力最多封锁一个路口的限制，所以对于某些一个服务平台对应几个交通要道的平台进行调整。调整规则为找出属于这个平台最优的路口，其他路口按时间排列仅次于所属平台的第二个服务平台甚至第三个直到找出不管辖其他路口节点某个服务平台，以 13 号平台为例，在初步计算中要封锁 21、22、23、24 四个路口节点，对此必须做出调整，以 24 好路口节点为例，在其通往所有服务平台的时间里找出除 13 号以外的时间最优服务平台，即 11，判断出 11 没有封锁其他平台的任务，则将 24 归属于由 11 号服务平台所封锁，其他点一次计算，最后得出第一种调度方案。

(3) 上面的调度过于生硬，为此我们将 13 个交通要道及其调配的 13 个平台全部打乱，找出 13 个服务平台到其所调度的交通要道时间最小而得到第二种调度方案。

此种方案为用 0-1 线性规划的思想来解决，即求出 13 个路口节点到 13 个服务平台的两两之间的相互到达的时间得出矩阵

$$N = \begin{matrix} & v_i & \cdots & v_j \\ \begin{bmatrix} t_{im} & \cdots & t_{jm} \\ \vdots & \vdots & \vdots \\ t_{in} & \cdots & t_{jn} \end{bmatrix} & v_m \\ & \vdots \\ & v_n \end{matrix}$$

$v_i$  这一横列代表所要封锁的 13 个路口节点；

$v_m$  这一竖列代表要封锁路口节点的 13 个服务平台。

在上面的矩阵中找出每行没列都只取一个时间的 13 个时间的最小，得出此种模型下的调度方案。

优化前后调度方案分配结果对比如表 5-5。

表 5-5 优化前后调度方案对比

出口标号	所属平台	初次调度	初次调度时间	优化后调度	优化后调度时间
12	12	12	0	12	0
14	14	14	0	14	0
16	16	16	0	9	1.532540322
21	13	9	11.53916738	16	10.00662706
22	13	10	7.707917747	10	7.707917747
23	13	13	0.5	13	0.5
24	13	11	3.805274077	11	3.805274077
28	15	15	4.751841748	15	4.751841748
29	15	7	8.01545686	7	8.01545686
30	7	8	3.060819834	8	3.060819834
38	2	2	3.982185925	2	3.982185925
48	7	5	2.475825881	5	2.475825881
62	4	4	0.35	4	0.35
总时间 (T)		46.18848945		46.18848945	

(4) 比较两种方案，发现两种方案的  $T_1 = T_2$ ，但是由于初次调度方案里的  $Max(T_{1i}) > Max(T_{2j})$   $i, j \in (1, 13)$ ，认为优化后调度方案优于初次调度方案。

所以快速全封锁 A 区的 13 条交通要道的调度方案为优化后调度方案，即表 4-5 中的优化后调度方案。

### 5.3 增设交警服务平台模型

通过前面模型分析可知，现有的交巡警服务平台设置存在不合理性，具体表现在以下两方面：

①1、2 号交巡警服务平台工作量远大于其他的平台。

②28、29、92 号路口节点按照现有的平服务台分布不能实现突发事件时 3 分钟到达。

1) 针对①种情况，

设平台  $i$  管辖的路口节点集  $S_i = \{v_{i1}, v_{i2}, \dots, v_{ij}\} \quad j \in (1, n)$ ;

出警时间目标函数:  $MinS = T_m + T_1 + T_2$ ;

出警时间约束条件:  $m \in (A_1 \cup A_2)$

我们建立单目标规划模型:

$$\begin{aligned} MinS &= (T_m + T_1 + T_2) \quad S_i = \{v_{i1}, v_{i2}, \dots, v_{ij}\} \quad j \in (1, n) \\ st. \quad m &\in (A_1 \cup A_2) \end{aligned}$$

根据此思想得出在 1、2 号服务平台不变的前提下, 以  $(A_1 \cup A_2)$  内的所有点 (16 个) 为新设置的服务平台所得出的各个最小时间的点的归属方案, 其中有 4 个点若在其位置设置服务平台, 它们仅仅包含自己的一个路口点, 显然需要舍去, 剩下 12 个待选服务平台的相应数值, 见表 5-6。

表 5-6 12 个待选服务平台信息情况

新平台设置点标号	38	39	43	44	67	68	69	70	71	72	73	75
总时间	19.3	22.6	21.9	22.6	21.8	21.8	21.5	22.7	21.7	22.5	22.5	21.8
新平台所管辖点数目	1	1	2	1	3	2	3	1	3	1	1	1

分析所得数据, 在时间最优的前提下考虑每个服务平台的工作量近似均等, 则选取新分配方案中包含三个路口节点的服务平台, 即 67、69、71 (如见表 5-7)。

表 5-7 3 个待选服务平台的最长出警时间

点号	最长时间
67	13.79
69	11.32
71	12.19

由表 5-7 比较这三个平台最长出警时间, 找出最长出警时间中最短的一个。所以选择以 69 号路口节点位置为新设置的平台位置, 从而得出 1、2、69 三个服务平台的综合信息如表 5-8。

表 5-8 1、2、69 号服务平台综合信息

平台点	管辖路口							时间累加	案件累加率
1	67	68	73	74	75	76	78	7.336607	8.1
2	38	39	40	43	44			11.32747	9.2
69	70	71	72					2.319141	3.9

分析表里的已经基本达到所有方案中的最优条件, 所以确定此位置点。

2) 针对②种问题中的 28、29 号点我们直接在两点中任意一点位置设置一个新的服务平台, 让新设置的这个服务平台仅管辖这两个点。而 92 号点只有这一个点, 而且已经有可以管辖 92 号点的服务平台, 不必专门为了某一个点而增设一个服务平台, 这样会有浪费警力资源的弊端。

综上, 新增服务平台数为 2 个, 位置为在 28 (或者 29)、69 号路口节点处。

## 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的合理性评价模型

设置交巡警平台服务的原则及任务：

①警情主导警务原则：根据管区道路交通流量、拥堵状况、治安复杂情况、发案量高低，科学确定服务平台管控区域；

②快速处警原则：城区接警后确保快速到达现场；

③任务：提高附近居民的安全性。

服务平台设置合理性指标：服务平台管辖的路口节点数过少或过多。

下面具体给出判断服务平台设置合理性的步骤：

(1) 利用交巡警平台管辖范围模型 I，求出全市（主城六区 A,B,C,D,E,F）各交巡警服务平台的管辖范围。

(2) 分析表格中可知存在明显不合理的服务平台点，如存在只管辖自己的服务平台点和除自己外仅管辖一个路口节点的平台，我们用 EXCEL 表格处理删除这两类平台点，这些点分别为：6、10、12、14、169、322、325、372、376、377、378。列出除此之外的服务平台信息见表 6-1。

**表 6-1 删除不合理服务平台后的平台信息**

交巡警平台编号	交巡警平台位置标号	交巡警平台编号	交巡警平台位置标号	交巡警平台编号	交巡警平台位置标号
A1	1	B8	100	D9	328
A2	2	C1	166	E2	373
A3	3	C2	167	E3	374
A4	4	C3	168	E4	375
A5	5	C5	170	E8	379
A7	7	C6	171	E9	380
A8	8	C7	172	E10	381
A9	9	C8	173	E11	382
A11	11	C9	174	E12	383
A13	13	C10	175	E13	384
A15	15	C11	176	E14	385
A16	16	C12	177	E15	386
A17	17	C13	178	F1	475
A18	18	C14	179	F2	476
A19	19	C15	180	F3	477
A20	20	C16	181	F4	478
B1	93	C17	182	F5	479
B2	94	D1	320	F6	480
B3	95	D2	321	F7	481
B4	96	D4	323	F8	482
B5	97	D5	324	F9	483
B6	98	D7	326	F10	484
B7	99	D8	327	F11	485

(3) 转至 (1)，得到除去不合理服务平台后全市交巡警服务平台管辖范围结果由

于此表格非常庞大有 85 行数据，在此不在正文里列出表格。

(4) 对上一步骤求出的表格中的数据进行分析及处理，在此由于范围的扩大我们将最大案发率进行上取整得松弛量  $\Delta = 3$ ：

$$\bar{Q} = \frac{\sum_{i=1}^n Q_i}{n}$$

$$Q_{\max} = \bar{Q} + \Delta = 12.7$$

$$Q_{\max} \leq Q_i < 2Q_{\max} \quad (5-1)$$

$$2Q_{\max} \leq Q_i < 3Q_{\max}$$

$$MinT = \sum_{i=1}^n T_i$$

得出在此服务平台情况下得出在理想状态中的每个服务平台都应为  $\bar{Q} = 9.7$ ，以松弛量为指标得出工作量的最大限度，即  $Q_{\max}$ ，当在  $Q_{\max} < Q_i < 2Q_{\max}$  情况下，平台  $i$  内部增设一个新的服务平台，当在  $2Q_{\max} \leq Q_i < 3Q_{\max}$  情况下，服务平台  $i$  内部设置两个新的服务平台，依此类推，在确定新增服务平台个数的前提下以  $MinT = \sum_{i=1}^n T_i$  为目标函数找出最佳新平台的位置，即确定新服务平台所在的路口节点编号。

表 6-2 新确定平台位置

新设服务平台编号	平台新增路口节点编号	平台所包含的路口节点	时间	案发率
13 号平台	13	14 21 23 24	11.567	9.6
	22	372 459	4.8596	4.9
167 号平台	167	248 250 251 252 255	14.855	8
	249	258 259 260 261	6.8294	5.5
171 号平台	216	230 231 240 243 244	13.773	11
	246	253 254	12.206	5.3
179 号平台	179	275 278 292 295 296	10.272	10.1
	279	281 282 285 288 289 290 291	15.617	8.3
180 号平台	180	268 269 270 297 298 305	16.934	10.6
	299	300 301 302 303 304 305 316	17.889	9.6
	307	309 310 311 312	8.2891	5.9
321 号平台	299	300 301 302 303 304 313 314 315 316	13.939	11.3
	354	355 356 357 360	3.2147	4.9
379 号平台	379	417 421 422 423	9.5508	9
	418	419 420	4.6202	4.1
382 号平台	382	401 402 403 404 407 408	15.215	8.1
	409	410 411 412 413 414 415 416	15.628	7.3
383 号平台	383	452 460 461 470	7.4677	7.4



	453	454	455	462	463	464	469						7.3818	6.2
475 号平台	475	550	551	555	556	565							6.8908	8.8
	557	558	559	561	563	564							10.814	7.1
476 号平台	476	532	533	534	535	543	544	545	546	547			14.011	9.9
	552	553	554										1.8267	3.7
477 号平台	477	492	493	494	495	496	497	498	499	500	501		23.681	9.9
			502	503	504	520	530							
	505	506	507	508	509	516	517	518	519	521	522	523	27.08	8.4
478 号平台	478	512	528	536	537	538	539	542					14.391	10.3
	513	514	515	524	525	526	527						15.069	5.3

## 6.2 最佳围堵方案警力调度模型

针对 P（第 32 个节点）发生特大刑警事件，对事件进行背景分析：

- (1)在三分钟后才接到报案；
- (2)嫌疑犯犯案位置固定
- (3)嫌疑犯逃跑路线各个路线概率均等
- (4)嫌疑犯逃跑时是驾车逃跑，假设其速率与警车速率相等
- (5)搜索逃犯进行封锁应该尽量在最快时间内

基于上面的背景下我们建立如下模型，在此引入一个割点的概念，即在一幅图中除去割点位置则得到的图为一个不连通图，在此虽然不能求出全市的割点，但是我们可以定义一个局部割点，即除去此局部割点在局部范围内为非连通点。除此之外我们根据图像的特点，即在都写割点附近，假设我们控制其中两个（或者更多，只要能控制）点就可以阻断这整条线路，我们将此种情况近似考虑成伪割点。

有割点的定义即其特性我们很容易想到要达到搜索嫌疑犯的目的必须先控制其逃跑的范围，即进行封锁嫌疑犯逃出某些范围的必须经过的点，然后内部巡警进行搜索抓住嫌疑犯。我们要控制的点即图中的割点。

经过 Matlab 的 Dijkstra 算法求出 P 点到图中所有点的最短时间。

根据最短时间，我们将要在尽量最短的时间内抓到嫌疑犯，转化成在最短时间内控制一个圈的所有割点，让嫌疑犯不能逃出一定区域，我们将这个在一定时间内的时间设在十分钟左右，我们找出在此范围内的割点（除去嫌疑犯 3 分钟逃出区域内的割点），因为数据两太大，在此我们选出部分割点进行说明，现举出部分可能有用的割点：

表 6-3 部分割点标号

割点标号	28	29	38	40	41	214	215	241	248	370	371	561
------	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----

对此我们也要求出很容易看得出的伪割点，即 50、53 两点可以代替 47 号割点，点 62、4 可以当做一个伪割点，点 5、6 可以看成一个伪割点，在此不再赘述。

将上面信息进行整理得出 P 点到达各个割点的时间  $t_p$  和由前面程序求出的表格中找出最近平台到达割点的时间  $t_i$ ，当  $(t_p - 3) > t_i$  时说明控制割点成功。最后得出围堵方案如下：

1 → 4、4 → 62、2 → 40、17 → 41、480 → 561、5 → 53、6 → 50、15 → 29、320 → 371、167 → 248、174 → 214、171 → 241、169 → 253

说明：箭头左边为交巡警服务平台所在的路口标号，右边是该平台需要围堵的路口标号。例如 320 → 371 说明 320 号平台要围堵 371 号路口。

从而得出，完成合围所花费的时间为 7.36 分钟。

## 7 结果分析与检验

在模型中我们为进行优化而进行了大量的数据处理，在处理中我们应与处理前的数据进行比较。在这个比较的过程中为了量化我们处理后的方案比处理前的方案更加优，优多少的问题我们引入标准差  $S$  的概念，即：

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

其中  $x$  代表一个样本  
 $x_i$  代表样本  $x$  中的个体  
 $\bar{x}$  代表样本  $x$  中的平均值  
 $n$  代表样本的容量

(1)对于第一问交警车尽量在三分钟内到达用 Dijkstra 算法求出三分钟内能够归属的点的表并对重复出现路口节点及强制归属后得到的表 4-2，对其内的时间和案发率进行求标准差（用  $S_t$  代表时间标准差，用  $S_q$  代表案发率标准差），对其求出时间的标准查为  $S_t = 4.33$ ，案发率的标准差为  $S_q = 2.95$ 。在对异常点 5、6、7 三服务平台点调整后的时间标准差为  $S_t = 4.02$ ，案发率标准查问  $S_q = 2.66$  知改进在此处表现不明显。

对于 5、6、7 三点，在处理前这三个服务平台点的时间标准查为  $S_t = 4.81$ ，案发率标准差为  $S_q = 3.37$ ，在处理后得到三个服务平台点的时间标准差为  $S_t = 0.87$ ，案发率标准差为  $S_q = 0.57$  知改进在此处的效果很好。

(2)对已第一问中第三个小问，得出改进前时间的标准差为  $S_t = 3.52$ ，改进后耳朵时间标准差为  $S_t = 3.17$ ，在此处的改进效果还是可以接受的。

(3)在第二问中第一小问中分析全市的交巡警服务平台的设置问题，在原基础上得出的表格的时间标准差为  $S_t = 15.58$ ，案发效率标准差为  $S_q = 4.38$ 。在进行调整后的出的结果的时间标准差为  $S_t = 9.48$ ，案发率标准差为  $S_q = 2.69$ 。从这个问题上易知效果非常好。

综上所有标准差得出咋大多数情况下我们所进行的优化计算都是有效果的，只是在第一问中第一小问不明显，其他都还是有教明显的效果。

## 8 模型评价

### 8.1 模型优缺点

优点：

(1)我们先将节点距离的邻接矩阵作适当的数据处理，删除距离大于  $3km$ ，因此大大得减少了用 Dijkstra 算法的量，并且采用了改进后的 Dijkstra 算法（代码见附录）进一步提高了程序运行的效率。

(2)模型 I 充分考虑到实际情况，利用出警时间，工作量分配以及辖区案发率均衡等原则对一次模型结果的某些点做进一步修正，以得出更加人性化的管辖范围。

(3)经过局部修正后的模型解为局部最优解，但经检验同时也为全局最优解。

(4)模型建立时综合考虑了多方面的因素如案发率、出警时间等。

**缺点：**

(1)题目中还给出了各区域面积和该区人口，我们建立的模型中没有考虑各区平均每个自然人所享受的警力分布对优化辖区范围的影响。

(2)在第一题的第二小问的模型中进行的第一次调整是手工进行查表所得到的结果，虽然第二次调整对调度方案有所改进，但是第二次所做的调整是在第一次调整的基础上进行的优化，不一定是全局最优，但肯定的较优的。

## 8.2 模型改进

针对缺点(2)，我们在解决类似于一个平台封锁多个路口节点的问题应该在全局的角度上进行，在编程上应该在求 13 个服务平台封锁的 13 个交通要道距离最优的前提下增设一个限制条件，即模型如下：

$$\text{Min}Z = \sum_{i \in m} L_{v_i q_j}$$

s.t.

$$q_j \in A \text{ 且不重复}$$

符号说明：

$v_i$  代表 13 个需要封锁的路口节点

$q_i$  代表 13 个封锁  $v_i$  的服务平台

$L_{v_i q_j}$  代表  $v_i$  距离  $q_i$  的最短距离

$m$  代表 13 个进出 A 区的交通要道标号集合

在此模型下得出的即为全局最优的调度方案。

## 9 参考文献

[1]孙惠泉，图论及其应用，北京：科学出版社，2004 年 9 月第一版

[2]韩中庚，数学建模方法及其应用，北京：高等教育出版社，2005 年 6 月第一版

[3]王海英、黄强、李传涛、褚宝增，图论算法及其 MATLAB 实现，北京：北京航空航天大学出版社，2010 年 2 月第一版

[4]董辰辉、彭雪峰，MATLAB2008 全程指南，北京：电子工业出版社，2009 年 3 月第一版

[5]王沫然，MATLAB 与科学计算，北京：电子工业出版社，2005 年 1 月第一版

[6]郭耀煌，运筹学原理与方法，成都：西南交通大学出版社，1994 年 9 月第一版

[7]全国大学生数学建模竞赛组委会编，全国大学生数学建模竞赛优秀论文汇编（1992-2000），北京：中国物价出版社，2002 年 3 月第一次版

[8]梁福军，科技论文规范写作与编辑，北京：清华大学出版社，2010 年 6 月第一版

## 附录

### 0.1 DijkstraI 算法函数

```
dijkf2.m
function a=dijkf2(a)
%求任意两顶点之间的最短距离 dijkstraI 算法
%a（输入量）表示图的邻接矩阵
%a（输出量）表示所求最短路的距离矩阵
n=length(a);
for i=2:n
    for j=1:i-1
        a(i,j)=a(j,i);
    end
end
for k=1:n-1
    b=[1:k-1,k+1:n];
    kk=length(b);
    a_id=k;
    b1=[k+1:n];
    kk1=length(b1);
    while kk>0
        for j=1:kk1
            te=a(k,a_id)+a(a_id,b1(j));
            if te<a(k,b1(j))
                a(k,b1(j))=te;
            end
        end
        miid=1;
        for j=2:kk
            if a(k,b(j))<a(k,b(miid))
                miid=j;
            end
        end
        a_id=b(miid);
        b=[b(1:miid-1),b(miid+1:kk)];
        kk=length(b);
        if a_id>k
            miid1=find(b1==a_id);
            b1=[b1(1:miid1-1) b1(miid1+1:kk1)];
            kk1=length(b1);
        end
    end
    for j=k+1:n
        a(j,k)=a(k,j);
    end
end
end
```

```
a;
```

### 1.1 得出 A 区网络图邻接矩阵代码段

A\_linjie.m

```
B=xlsread('A 区交通路口坐标.xlsx');
C=xlsread('A 区交通路口的路线.xlsx');
[m n]=size(B);
A=zeros(m,m)+inf;
for i=1:92
    A(i,i)=0;
end
[m n]=size(C);
for i=1:m
    a=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2);
    if a>30
        A(C(i,1),C(i,2))=inf;
    else
        A(C(i,1),C(i,2))=a;
    end
    A(C(i,2),C(i,1))=A(C(i,1),C(i,2));
end
xlswrite('A 区网络图邻接矩阵.xlsx',A)
```

### 1.2 绘制 A 区交通路口分布图程序段

A\_gplot.m

```
B=xlsread('A 区交通路口坐标.xlsx');
C=xlsread('A 区交通路口的路线.xlsx');
c=xlsread('出入 A 区的路口标号.xlsx');
[m n]=size(B);
A1=zeros(m,m)+inf;
for i=1:92
    A1(i,i)=0;
end
[m n]=size(C);
for i=1:m
    A1(C(i,1),C(i,2))=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2);
    A1(C(i,2),C(i,1))=A1(C(i,1),C(i,2));
end
D=B(:,2:3);
gplot(A1,D,'k')
hold on
plot(B((1:20),2),B((1:20),3),'ro');
plot(B(c,2),B(c,3),'r*');
plot(B(:,2),B(:,3),'.');
hold off
hold on
```

```

for k=1:m
    text(B(k,2),B(k,3),['\leftarrow',num2str(k)]);
end
hold off

```

### 1.3 A 区各路口归属分配方案程序段

```

A_fenpei.m
a=dijkf2(A)
%对 a 中数据进行筛选
D=triu(a);
D=D((1:20),:);
[m n]=size(D);
for i=1:m
    for j=1:n
        if D(i,j)>30|j<=20
            D(i,j)=inf;
        end
    end
end
%对各路口进行归属划分
F=zeros(20,9);
for j=21:n
    k=1;
    if min(D(:,j))~=inf
        [r,c]=find(D==min(D(:,j)));
        r=r(find(c==j));
        c=j;
    end
    while 1
        if F(r,k)~=0;
            k=k+1;
        else
            break;
        end
    end
    F(r,k)=c;
end
xlswrite('A 区各路口归属分配方案.xlsx',F);

```

### 1.4 A 区各服务平台管辖各路口的发案率总和程序段

```

A_faanylvsum.m
%求发案率
b=xlsread('A 区各路口发案率.xlsx');
b=b(:,2);
[m n]=size(F);
c=zeros(1,20)';

```

```

for i=1:m
    for j=1:n
        if F(i,j)~=0
            c(i)=c(i)+b(F(i,j));
        end
    end
end
end
xlswrite('A 区各服务平台管辖各路口的发案率总和.xlsx',c);

```

### 1.5 A 区各服务平台到各管辖路口的时间总和程序段

```

A_timesum.m
a1=a/10;
[m n]=size(F);
d=zeros(1,20)';
for i=1:m
    for j=2:n
        if F(i,j)~=0
            d(i)=d(i)+a1(F(i,j));
        end
    end
end
end
xlswrite('A 区各服务平台到各管辖路口的时间总和.xlsx',d);

```

### 1.6 对 5、6、7 号交巡警服务平台管辖路口调整的 LINDO 程序段

```

A_tiaozheng.ltx
MIN
13.0107149x11+7.586585214x12+ 3.791352821x13+ 0x14+
5.977002108x15+ 11.9502818x16+ 14.54325522x71+
8.274201838x21+ 12.77565893x22+ 8.337297726x23+
11.9502818x24+ 5.973279695x25+ 0x26+ 6.741661516x72+
1.532540322x31+ 6.95667001x32+ 11.39503121x33+ 14.54325522x34+
12.71494121x35+ 6.741661516x36+ 0x73+
11.53916738x41+ 9.510693385x42+ 5.072332183x43+
8.68531626x44+ 2.708314152x45+ 3.264965543x46+ 10.00662706x74+
13.13204744x51+ 7.707917747x52+ 3.269556545x53+
6.882540622x54+ 0.905538514x55+ 5.067741181x56+
11.8094027x75+
14.53758595x61+ 9.113456261x62+ 4.675095059x63+
6.477002108x64+ 0.5x65+ 6.473279695x66+ 13.21494121x76+
13.66776497x71+ 8.24363528x72+ 3.805274077x73+ 3.59163002x74+
2.385372088x75+ 8.358651783x76+ 15.1003133x77
SUBJECT TO
x11+x12+x13+x14+x15+x16+x17=1
x21+x22+x23+x24+x25+x26+x27=1
x31+x32+x33+x34+x35+x36+x37=1
x41+x42+x43+x44+x45+x46+x47=1

```

```

x51+x52+x53+x54+x55+x56+x57=1
x61+x62+x63+x64+x65+x66+x67=1
x71+x72+x73+x74+x75+x76+x77=1
x11+x21+x31+x41+x51+x61+x71=1
x12+x22+x32+x42+x52+x62+x72=1
x13+x23+x33+x43+x53+x63+x73=1
x14+x24+x34+x44+x54+x64+x74=1
x15+x25+x35+x45+x55+x65+x75=1
x16+x26+x36+x46+x56+x66+x76=1
x17+x27+x37+x47+x57+x67+x77=1
END
INT 49

```

运行结果:

LP OPTIMUM FOUND AT STEP 25  
OBJECTIVE VALUE = 13.5457325

FIX ALL VARS.( 36) WITH RC > 0.121333

25 NEW INTEGER SOLUTION OF 13.5457325 AT BRANCH 0 PIVOT

BOUND ON OPTIMUM: 13.54573  
ENUMERATION COMPLETE. BRANCHES= 0 PIVOTS= 25

LAST INTEGER SOLUTION IS THE BEST FOUND  
RE-INSTALLING BEST SOLUTION...

OBJECTIVE FUNCTION VALUE

1) 13.54573

VARIABLE	VALUE	REDUCED COST
X11	0.000000	13.010715
X12	0.000000	7.586585
X13	0.000000	3.791353
X14	1.000000	0.000000
X15	0.000000	5.977002
X16	0.000000	11.950282
X71	0.000000	28.211020
X21	0.000000	8.274201
X22	0.000000	12.775659
X23	0.000000	8.337297
X24	0.000000	11.950282
X25	0.000000	5.973279
X26	1.000000	0.000000
X72	0.000000	14.985296
X31	1.000000	1.532540



X32	0.000000	6.956670
X33	0.000000	11.395031
X34	0.000000	14.543255
X35	0.000000	12.714941
X36	0.000000	6.741662
X73	1.000000	3.805274
X41	0.000000	11.539167
X42	0.000000	9.510694
X43	0.000000	5.072332
X44	0.000000	8.685316
X45	0.000000	2.708314
X46	0.000000	3.264966
X74	0.000000	13.598257
X51	0.000000	13.132048
X52	1.000000	7.707918
X53	0.000000	3.269557
X54	0.000000	6.882541
X55	0.000000	0.905538
X56	0.000000	5.067741
X75	0.000000	14.194775
X61	0.000000	14.537586
X62	0.000000	9.113457
X63	0.000000	4.675095
X64	0.000000	6.477002
X65	1.000000	0.500000
X66	0.000000	6.473279
X76	0.000000	21.573593
X77	0.000000	15.100313
X17	0.000000	0.000000
X27	0.000000	0.000000
X37	0.000000	0.000000
X47	1.000000	0.000000
X57	0.000000	0.000000
X67	0.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	0.000000
8)	0.000000	0.000000
9)	0.000000	0.000000
10)	0.000000	0.000000
11)	0.000000	0.000000

12)	0.000000	0.000000
13)	0.000000	0.000000
14)	0.000000	0.000000
15)	0.000000	0.000000
NO. ITERATIONS= 25		
BRANCHES=	0 DETERM.=	1.000E 0

## 1.7 交通要道封锁中最优化求解的 LINDO 程序段

```

MIN
3.182932662x530+3.87682225x532+ 1.456021978x547+2.475825881x548+
0.5x549+0.848528137x550+1.229316793x551+1.659433056x552+1.170820393x553+2.
083697125x556+2.301889355x558+1.520864387x559+5.375825881x561+
3.213517559x630+3.907407147x632+1.486606875x647+2.506410777x648+3.3194888
55x649+2.275458204x650+1.894669549x651+2.324785812x652+3.179186187x653+2.7490
49881x656+2.384146922x658+1.603121954x659+5.406410777x661+
0.583095189x730+1.140175425x732+1.280624847x747+1.290201971x748+3.2366468
25x749+3.585174963x750+3.965963618x751+4.396079881x752+3.907467219x753+4.8203
4395x756+ 5.03853618x758+ 4.257511213x759+4.190201971x761
SUBJECT TO
x530+x532+x547+x548+x549+x550+x551+x552+x553+x556+x558+x559+x561>=1
x630+x632+x647+x648+x649+x650+x651+x652+x653+x656+x658+x659+x661>=1
x730+x732+x747+x748+x749+x750+x751+x752+x753+x756+x758+x759+x761>=1
x530+x630+x730=1
x532+x632+x732=1
x547+x647+x747=1
x548+x648+x748=1
x549+x649+x749=1
x550+x650+x750=1
x551+x651+x751=1
x552+x652+x752=1
x553+x653+x753=1
x556+x656+x756=1
x558+x658+x758=1
x559+x659+x759=1
x561+x661+x761=1
2.1x530+1.5x532+1.6x547+1.4x548+1.2x549+1.1x550+0.8x551+0.6x552+1.4x553+0.5x5
56+1.1x558+0.9x559+0.6x561>5.266666667
2.1x630+1.5x632+1.6x647+1.4x648+1.2x649+1.1x650+0.8x651+0.6x652+1.4x653+0.5x6
56+1.1x658+0.9x659+0.6x661>5.266666667
2.1x730+1.5x732+1.6x747+1.4x748+1.2x749+1.1x750+0.8x751+0.6x752+1.4x753+0.5x7
56+1.1x758+0.9x759+0.6x761>5.266666667
2.1x530+1.5x532+1.6x547+1.4x548+1.2x549+1.1x550+0.8x551+0.6x552+1.4x553+0.5x5
56+1.1x558+0.9x559+0.6x561<9.266666667
2.1x630+1.5x632+1.6x647+1.4x648+1.2x649+1.1x650+0.8x651+0.6x652+1.4x653+0.5x6
56+1.1x658+0.9x659+0.6x661<9.266666667

```

2.1x730+1.5x732+1.6x747+1.4x748+1.2x749+1.1x750+0.8x751+0.6x752+1.4x753+0.5x756+1.1x758+0.9x759+0.6x761<9.266666667  
 END  
 INT 39

运行结果:

OBJECTIVE FUNCTION VALUE

1) 29.60739

VARIABLE	VALUE	REDUCED COST
X530	0.000000	0.000000
X532	0.000000	0.000000
X547	0.833334	0.000000
X548	0.000000	0.000000
X549	0.000000	0.000000
X550	1.000000	0.000000
X551	0.000000	0.000000
X552	1.000000	0.000000
X553	1.000000	0.000000
X556	1.000000	0.000000
X558	0.000000	0.000000
X559	0.370370	0.000000
X561	0.000000	0.000000
X630	1.000000	0.000000
X632	0.000000	0.000000
X647	0.166666	0.000000
X648	0.714286	0.000000
X649	0.000000	0.000000
X650	0.000000	0.000000
X651	1.000000	0.000000
X652	0.000000	0.000000
X653	0.000000	0.000000
X656	0.000000	0.000000
X658	1.000000	0.000000
X659	0.000000	0.000000
X661	0.000000	0.000000
X730	0.000000	0.000000
X732	1.000000	0.000000
X747	0.000000	0.000000
X748	1.000000	0.000000
X749	1.000000	0.000000
X750	0.000000	0.000000
X751	0.000000	0.000000
X752	0.000000	0.000000
X753	0.000000	0.000000
X756	0.000000	0.000000
X758	0.000000	0.000000

X759	0.629630	0.000000
X761	1.000000	0.000000
ROW	SLACK OR SURPLUS	DUAL PRICES
2)	4.203704	0.000000
3)	2.880953	0.000000
4)	3.629630	0.000000
5)	0.000000	1.500000
6)	0.000000	1.071429
7)	0.000000	1.142857
8)	-0.714286	1.000000
9)	0.000000	0.857143
10)	0.000000	0.785714
11)	0.000000	0.571429
12)	0.000000	0.428571
13)	0.000000	1.000000
14)	0.000000	0.357143
15)	0.000000	0.785714
16)	0.000000	0.642857
17)	0.000000	0.428571
18)	0.000000	-0.714286
19)	0.000000	-0.714286
20)	0.000000	-0.714286
21)	4.000000	0.000000
22)	4.000000	0.000000
23)	4.000000	0.000000
NO. ITERATIONS=	24	
BRANCHES=	0 DETERM.=	1.000E 0

### 1.8 在 1、2 号服务平台管辖区域内增设平台程序段

A_addfuwu.m
<pre> x3=[1 2 38 39 40 43 44 67 68 69 70 71 72 73 74 75 76 78]; B=xlsread('A 区交通路口坐标.xlsx'); C=xlsread('A 区交通路口的路线.xlsx'); %计算 x3 中各点之间的邻接矩阵 A1=zeros(92,92)+inf; for i=1:92     A1(i,i)=0; end [m n]=size(C); for i=1:m     if find(C(i,1)==x3)&amp;find(C(i,2)==x3)         A1(C(i,1),C(i,2))=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2); </pre>

```

        A1(C(i,2),C(i,1))=A1(C(i,1),C(i,2));
    end
end

a3=dijkf2(A1)/10;
%增设服务平台的计算
s=size(x3,2);
for tt=3:s
    t=x3(tt);
    [m n]=size(a3);
    D3=triu(a3);
    for i=1:m
        for j=1:n
            if i>=j
                D3(i,j)=inf;
            end
        end
    end
    for i=1:m
        for j=1:n
            if i~=1&i~=2&i~=t
                D3(i,j)=inf;
            end
        end
    end
    F3=zeros(m,n);
    for j=21:n
        k=1;
        if min(D3(:,j))~=inf
            [r,c]=find(D3==min(D3(:,j)));
            r=r(find(c==j));
            c=j;
            while 1
                if F3(r,k)~=0;
                    k=k+1;
                else
                    break;
                end
            end
            F3(r,k)=c;
        end
    end
    xlswrite('A1 增设服务平台.xlsx',F3,[num2str(t)]);
    [m n]=size(F3);
    d3=zeros(1,t)';
    for i=1:m
        for j=1:n

```

```

        if F3(i,j)~=0
            d3(i)=d3(i)+a3(i,F3(i,j));
        end
    end
end
    xlsxwrite('A1 增设服务平台.xlsx',d3,[num2str(t)],'L1');
end

```

## 2.1 绘制全市交通路口分布图程序段

B2\_fenbutu.m

```

B=xlsread('全市交通路口坐标.xlsx');
C=xlsread('全市交通路口的路线.xlsx');
c1=xlsread('全市平台位置.xlsx');
c2=xlsread('出入市区的路口标号.xlsx');
[m n]=size(C);
A=zeros(m,m);
for i=1:m
    A(C(i,1),C(i,2))=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2);
    A(C(i,2),C(i,1))=A(C(i,1),C(i,2));
end
[m n]=size(B);
D=B(:,2:3);
gplot(A,D,'k')
hold on
plot(B(c1,2),B(c1,3),'ro');
plot(B(c2,2),B(c2,3),'r*');
plot(B(:,2),B(:,3),'.');
hold off
hold on
for k=1:m
    text(B(k,2),B(k,3),['\leftarrow',num2str(k)]);
end
hold off

```

## 2.2 全市服务平台管辖信息求解程序段

B2\_guanxiaxinxi.m

```

B=xlsread('全市交通路口坐标.xlsx');
C=xlsread('全市交通路口的路线.xlsx');
c1=xlsread('全市平台位置 1.xlsx');
c2=xlsread('出入市区的路口标号.xlsx');
%求路口点之间的邻接矩阵
m=size(B,1);
A=zeros(m,m)+inf;
for i=1:m
    A(i,i)=0;
end

```

```

[m n]=size(C);
for i=1:m
    A(C(i,1),C(i,2))=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2);
    A(C(i,2),C(i,1))=A(C(i,1),C(i,2));
end
%求任意两点间最短路径，进而求解出相对应的时间
a=dijkf2(A)/10;

[m n]=size(a);
%取上三角矩阵
D=triu(a);
%将对角线及以下数据令为 inf
for i=1:m
    for j=1:n
        if i>=j
            D(i,j)=inf;
        end
    end
end
%将平台位置标号所不在的行令为 inf
for i=1:m
    if find(i==c1)
        continue;
    else
        D(i,:)=inf;
    end
end
%对交巡警平台进行管辖范围的分配
mm=size(c1,1);
F=zeros(m,32);
for j=1:n
    if find(j==c1)
        continue;
    else
        k=1;
        if min(D(:,j))~=inf
            [r,c]=find(D==min(D(:,j)));
            r=r(find(c==j));
            c=j;
            while 1
                if F(r,k)~=0;
                    k=k+1;
                else
                    break;
                end
            end
        end
    end
end

```

```

        F(r,k)=c;
    end
end
end
xlswrite('全市服务平台管辖信息.xlsx',F);
%求出各个交巡警平台的出警时间
[m n]=size(F);
t=zeros(1,m)';
for i=1:m
    for j=1:n
        if F(i,j)~=0
            t(i)=t(i)+a(i,F(i,j));
        end
    end
end
xlswrite('全市服务平台管辖信息.xlsx',t,'Sheet1','AE1');
%求出各个交巡警平台所管辖路口的发案率总和
e=zeros(1,m)';
for i=1:m
    if find(i==c1)
        e(i)=e(i)+B(i,5);
    end
    for j=1:n
        if F(i,j)~=0
            e(i)=e(i)+B(F(i,j),5);
        end
    end
end
xlswrite('全市服务平台管辖信息.xlsx',e,'Sheet1','AF1');

```

### 2.3 针对第 2 问解决交巡警服务平台设置方案中的调整函数

B2\_fenpeitiao Zheng

```

function f=B2_fenpeitiao Zheng(x)
%针对第 2 题解决交巡警服务平台设置方案中的调整

B=xlsread('全市交通路口坐标.xlsx');
C=xlsread('全市交通路口的路线.xlsx');
c1=xlsread('全市平台位置.xlsx');
c2=xlsread('出入市区的路口标号.xlsx');
%求路口点之间的邻接矩阵
m=size(B,1);
A1=zeros(m,m)+inf;
for i=1:m
    A1(i,i)=0;
end
[m n]=size(C);

```



```

for i=1:m
    if find(C(i,1)==x)&find(C(i,2)==x)
        A1(C(i,1),C(i,2))=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2);
        A1(C(i,2),C(i,1))=A1(C(i,1),C(i,2));
    end
end
%求任意两点间最短路径，进而求解出相对应的时间
a1=dijkf2(A1)/10;

s=size(x,2);
for tt=2:s
    t=x(tt);
    %数据筛选
    [m n]=size(a1);
    D1=triu(a1);
    for i=1:m
        for j=1:n
            if i>=j
                D1(i,j)=inf;
            end
        end
    end
    for i=1:m
        if find(i==x(1))
            continue;
        elseif find(i==x(tt))
            continue;
        else
            D1(i,:)=inf;
        end
    end
    %求出分配方案
    F1=zeros(m,n);
    for j=1:n
        if find(j==x(1))
            continue;
        elseif find(j==x(tt))
            continue;
        else
            k=1;
            if min(D1(:,j))~=inf
                [r,c]=find(D1==min(D1(:,j)));
                r=r(find(c==j));
                c=j;
                while 1
                    if F1(r,k)~=0;
                        k=k+1;

```

```

else
    break;
end
end
F1(r,k)=c;
end
end
end
xlswrite('A22 调整分配方案.xlsx',F1,[num2str(t)]);
%求出相对应的时间总和
[m n]=size(F1);
d1=zeros(1,m);
for i=1:m
    for j=1:n
        if F1(i,j)~=0
            d1(i)=d1(i)+a1(i,F1(i,j));
        end
    end
end
xlswrite('A22 调整分配方案.xlsx',d1,[num2str(t)],'AD1');
%求出相对应的发案率总和
e1=zeros(1,m);
for i=1:m
    if find(i==x(1))
        e1(i)=e1(i)+B(i,5);
    end
    if find(i==x(tt))
        e1(i)=e1(i)+B(i,5);
    end
    for j=1:n
        if F1(i,j)~=0
            e1(i)=e1(i)+B(F1(i,j),5);
        end
    end
end
xlswrite('A22 调整分配方案.xlsx',e1,[num2str(t)],'AE1');
end

```

说明：在解决第 2 问中的，交巡警服务平台设置方案中的调整问题时，将要增设平台的区域内的路口标号，写在一个向量 x 中，然后再调用 B2\_fenpeitiaozheng 函数，分析得出结果。

#### 2.4 P 点到各点的最短时间程序段

B2_ptime.m
<pre> B=xlsread('全市交通路口坐标.xlsx'); C=xlsread('全市交通路口的路线.xlsx'); %求路口点之间的邻接矩阵 </pre>

```

m=size(B,1);
A=zeros(m,m)+inf;
for i=1:m
    A(i,i)=0;
end
[m n]=size(C);
for i=1:m
    A(C(i,1),C(i,2))=sqrt((B(C(i,1),2)-B(C(i,2),2))^2+(B(C(i,1),3)-B(C(i,2),3))^2);
    A(C(i,2),C(i,1))=A(C(i,1),C(i,2));
end
a=dijkf2(A)/10;
time=a(32,:);

```