

交巡警服务平台的设置与调度

摘要

本题针对某市设置交巡警服务平台的相关情况来建立数学模型研究交巡警服务平台警力合理的调度方案。

对于问题一的第一小问：我们建立了最优化模型。首先计算出 A 城区每一个交巡警平台最远可以管辖的交通路口的范围，运用 MATLAB 编程进行求解出范围并画出图形。其次我们运用“Floyd 算法”分成两个步骤来求解每个平台最优具体所管辖的路口：先求得 A 城区每个有路径的相邻路口节点之间的距离，并做出对应的带权邻接矩阵。然后建立状态转移方程：每个交巡警平台到范围内交通路口节点的最短距离。我们根据“Floyd 算法”运用 MATLAB 进行编程，得到 A 城区每个各交巡警服务平台所管辖的最优交通路口节点。使得 A 城区交巡警驾驶时速为 60km/h 的警车尽量在 3 分钟内可以最快到达管辖的交叉路口。

对于问题一的第二个小问：我们建立了 0-1 整数规划模型来求解调度 20 个交巡警平台封锁 13 个路口的最佳方案。在 0-1 整数规划模型中，将平台的最长出警时间要尽可能缩短作为目标函数。可知 13 条交通要道需要全部被封锁，且每个交通要道只需一个交巡警平台去封锁、一个平台的警力最多封锁一个出口的约束条件。根据以上目标函数和约束条件，建立了 0-1 整数规划模型并运用 LINGO 进行求解，得到 A 区交巡警服务平台警力合理的调度方案。

对于问题一的第三个小问：我们建立了双目标的 0-1 整数规划模型来求解需要增加平台的具体个数和位置。首先，每个交通巡警平台的工作量要尽可能的均衡，即每个平台管辖的路口数目要尽可能相同，也就是方差最小；另外，每个巡警平台出警的时间也要尽可能缩短，即到达每个平台所管辖的距离最远的路口的时间要尽可能最短。此时，可根据如上两个目标建立如下 0-1 整数规划模型。约束条件是：设置的总平台数应当在 22-25 之间、第 i 个路口是否被选为巡警平台，若选为平台，则 $y_i=1$ ，否则为 0、第 j 个路口是否被第 i 个平台管辖，若被管辖，

则 x_{ij} 为 1，否则为 0、编号为前 20 的路口已经被选为平台。可建立双目标 0-1 规划模型，运用 LINGO 求解得到应在 28、38、61、87 路口处增设交巡警平台。

对于问题二的第一个小问：我们对该市现有交巡警服务平台设置方案进行了分析研究，发现存在明显不合理。考虑到附件二“全市交通路口节点数据”中给出的每个节点的案发率，我们对问题一建立的“最优化模型”进行了模型的优化。不仅要考虑在 3 分钟内到达路口，而且要根据每个节点案发率的不同，最优的给平台分配路口。将 A 区案发率归一化，近似的作为节点的距离权重。对每一个节点，优先选取上述值中最小值代表的平台，表示受其管辖。并应用 MATLAB 对建立的“改进后的最优化模型”求解出各城区各交巡警服务平台管辖的交叉路口。

对于问题二的第二个小问：根据广度优先原则，从 P 点开始依次走遍 A 区的所有路口节点进行排查。与此同时根据距离就近原则抽调 17 个平台去该城市的所有出入口进行封锁。B~F 区未分配封锁任务的平台，照常值班处理日常事务。

关键词：最优化模型 0-1 整数规划模型 Floyd 算法 双目标的 0-1 整数规划模型

一、 问题重述

建立数学模型分析求解下面的问题：

(1) 请为市中心城区 A 的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。

二、 问题分析

对于第一问：

针对第一小问，我们建立了最优化模型为各交巡警服务平台分配最优的管辖范围。首先计算出 A 城区每个各交巡警服务平台所能管辖的交通路口节点最大范围，其次我们运用“Floyd 算法”求解每个平台最优具体所管辖的路口，使得每个路口发生事故都可以得到最短时间的处理。

针对第二小问，该小题问要研究调度 20 个交巡警平台封锁 13 个路口的最佳方案。其中，如何使每个平台的出警时间最短是最为重要的。因而，在模型建立过程中，将其作为目标函数，而每一个平台最多只可以去封锁一个路口，并且每一个路口都必须被封锁。根据以上目标函数和约束条件，建立了 0-1 整数规划模型。

针对第三小问，要求在 A 区增设 2-5 个交通巡警平台，以解决 A 区交巡警服务平台的工作量不均衡和有些平台出警时间过长的情况。首先，每个交通巡警平台的工作量要尽可能的均衡，即每个平台管辖的路口数目要尽可能相同，也就是方差最小；另外，每个巡警平台出警的时间也要尽可能缩短，即到达每个平台所管辖的距离最远的路口的时间要尽可能最短。此时，可根据如上两个目标建立如下双目标的 0-1 整数规划模型。

对于第二问：

针对第一个小问：我们对该市现有交巡警服务平台设置方案进行了分析研究，发现存在明显不合理。考虑到附件二“全市交通路口节点数据”中给出的每个节点的案发率，我们对问题一建立的“最优化模型”进行了模型的优化。不仅要考虑在 3 分钟内到达路口，而且要根据每个节点案发率的不同，最优的给平台分配路口。将 A 区案发率归一化，近似的作为节点的距离权重。对每一个节点，优先选取上述值中最小值代表的平台，表示受其管辖。并应用 MATLAB 对建立的“改进后的最优化模型”求解出各城区各交巡警服务平台管辖的交叉路口。

针对第二个小问：根据广度优先原则，从 P 点开始依次走遍 A 区的所有路口节点进行排查。与此同时根据距离就近原则抽调 17 个平台去该城市的所有出入口进行封锁。B~F 区未分配封锁任务的平台，照常值班处理日常事务。

三、 模型假设与符号说明

3.1 模型假设

- 1、假设题目所给的数据真实可靠；
- 2、假设在 3 分钟内，交巡警驾驶的警车可以以时速为 60km/h 保持匀速行驶。
- 3、同一管辖范围内，一定时间只会出现一件突发事件。

3.2 符号说明

x_{ij} : 平台 i 是否被节点 j 管辖

d_{ij} : 平台 i 与节点 j 间的最短距离

Z : 出警时间

y_i : 节点 i 是否是平台

$map[i, j]$: 平台 i 到节点 j 的最短距离

四、 模型的建立与求解

5.1 问题一中第一小问的思路分析

我们建立了最优化模型为各交巡警服务平台分配最优的管辖范围。首先计算出 A 城区每个各交巡警服务平台所能管辖的交通路口节点最大范围，其次我们运用“Floyd 算法”求解每个平台最优具体所管辖的路口，使得每个路口发生事故都可以得到最短时间的处理。“Floyd 算法”是一种利用动态规划的思想寻找给定的加权图中多源点之间最短路径的算法

5.1.1 最优化模型的建立

首先得到 A 城区每个各交巡警服务平台所能管辖的交通路口节点最大范围。

交巡警驾驶时速为 60km/h 的警车尽量在 3 分钟内到达事发地，即最多行驶 3Km 的距离。已知地图距离和实际距离的比例是 1:100000，所以将 3Km 换算成地图距离 30mm。在图中以每一个属于 A 城区的交巡警平台做圆，半径取 30mm，可以得到 20 个圆，即每一个 A 城区交巡警平台最远可以管辖的交通路口的范围。

然后我们运用“Floyd 算法”求解每个平台最优具体所管辖的路口，具体分为两个步骤：

1.带权邻接矩阵：

我们首先将附件 2“全市交通路口节点数据”中属于 A 城区的路口节点筛选出来，得到对应地图上的横纵坐标。且根据附件 2“全市交巡警平台”中属于 A 城区的 20 个交巡警平台，也可以得到对应地图上的横纵坐标。

再根据附件 2“全市交通路口的路线”，计算出表格中有路径的两路口节点之间的欧氏距离。可以得到 A 城区每个有路径的相邻路口节点之间的距离，并做出相应的带权邻接矩阵。

欧氏距离求解公式如下（ x, y 分别表示点的横纵坐标）：

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2.状态转移方程：
其状态转移方程如下：

$$map[i,j]=\min\{map[i,k]+map[k,j],map[i,j]\}$$

$map[i,j]$ 表示交巡警平台*i*到交通路口节点*j*的最短距离，*k*是穷举*i,j*的断点。

5.1.2 最优化模型的求解

我们运用 MATLAB 编程求解 A 城区每个各交巡警服务平台所能管辖的交通路口节点最大范围并画出图形。（图中实圆点“●”表示交叉路口的节点，没有实圆点的交叉线为道路立体相交；星号“*”表示现有交巡警服务平台的设置点）见图 1：

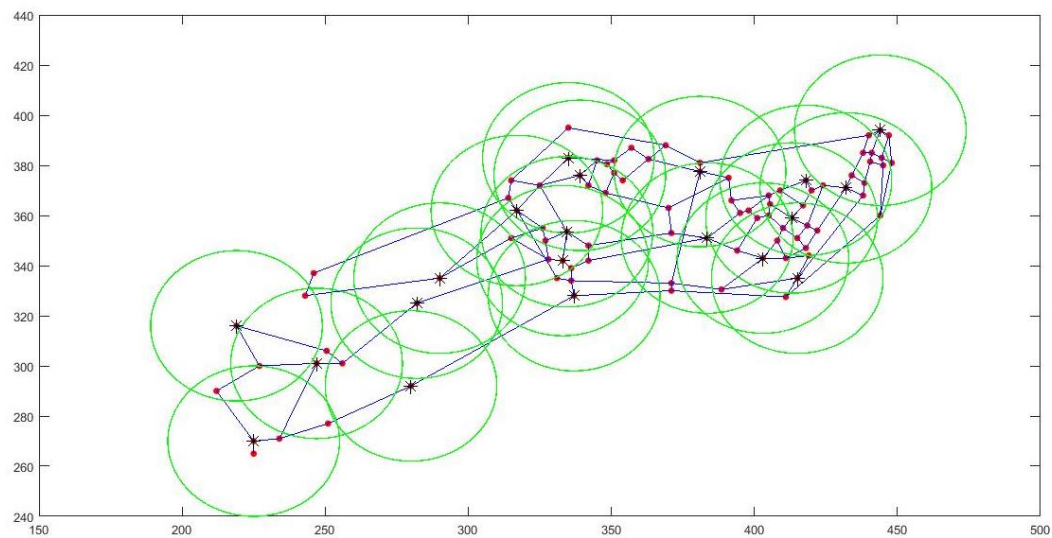


图 1 A 城区每个各交巡警服务平台所能管辖的交通路口节点最大范围

我们根据“Floyd 算法”运用 MATLB 进行编程，得到 A 城区每个各交巡警服务平台所管辖的最优交通路口节点。使得 A 城区交巡警驾驶时速为 60km/h 的警车尽量在 3 分钟内可以最快到达管辖的交叉路口。见表 1、表 2：

表 1 A 城区各交巡警服务平台管辖的交叉路口

该市 A 城区交巡警平台	所管辖的交叉路口
A1	68 71 72 73 74 79
A2	44 70 76 78 39 0
A3	43 54 55 65 67
A4	57 60 62 63 64 66
A5	49 50 53 56
A6	51 52 58 59
A7	30 48 61
A8	35 46 47
A9	32 34 45
A11	26 27

A12	25
A13	21 22 23 24
A15	31 28 29
A16	16 33 36 37 38
A17	40 41 42
A18	77 80 84 85 87 88
A19	69 75 81 82 83
A20	86 89 90 91 92

表 2: A 城区各交巡警服务平台管辖的最远交叉路口

该市 A 城区交巡警平台	最远路口编号	最远出警距离(单位: 百米)
A1	72	16.4031
A2	39	36.8219
A3	43	29.1165
A4	66	28.3564
A5	56	20.8370
A6	58	23.8415
A7	61	41.9020
A8	47	20.7966
A9	32	17.6903
A11	27	16.4330
A12	25	17.8885
A13	21	27.0831
A15	29	57.0053
A16	38	34.0588
A17	40	26.8794
A18	77	26.8554
A19	81	23.7147
A20	92	36.0127

5.2 问题一中第二小问的思路分析

该小问要研究调度 20 个交巡警平台封锁 13 个路口的最佳方案。其中, 如何使每个平台的出警时间最短是最为重要的。因而, 在模型建立过程中, 将其作为目标函数, 而每一个平台最多只可以去封锁一个路口, 并且每一个路口都必须被封锁。根据以上目标函数和约束条件, 建立了 0-1 整数规划模型。

5.2.1 0-1 整数规划模型的建立

0-1 整数规划模型:

(1) 目标函数:

平台 i 的最长出警时间要尽可能缩短 (v 表示警车的时速为 $60km/h$):

$$\min Z = \max_{\substack{1 \leq i \leq 20 \\ 1 \leq j \leq 13}} \frac{x_{ij} \times d_{ij}}{v}$$

(2) 约束条件一：

将出入 A 区的路口标号（交通要道）都对应上序号 1,2,...,13，记为 j，方便下面计算：

$$x_{ij} = \begin{cases} 1 & \text{交巡警服务平台 } i \text{ 封锁出入 A 区的路口 } j \\ 0 & \text{交巡警服务平台 } i \text{ 不封锁出入 A 区的路口 } j \end{cases}$$

表 3 出入 A 区的路口标号所定义的序号

序号 j	出入 A 区的路口标号
1	12
2	14
3	16
4	21
5	22
6	23
7	24
8	28
9	29
10	30
11	38
12	48
13	62

(3) 约束条件二：

13 条交通要道需要全部被封锁，且每个交通要道只需要一个交巡警平台去封锁：

$$\sum_{i=1}^{20} x_{ij} = 1 \quad (j = 1, 2, \dots, 13)$$

(4) 约束条件三：

一个平台的警力最多封锁一个出口：

$$\sum_{j=1}^{13} x_{ij} \leq 1 \quad (i = 1, 2, \dots, 20)$$

综上所述，建立的模型如下：

$$\min Z = \max_{\substack{1 \leq i \leq 20 \\ 1 \leq j \leq 13}} \frac{x_{ij} \times d_{ij}}{v}$$

$$s.t \begin{cases} \sum_{i=1}^{20} x_{ij} = 1 (j = 1, 2, \dots, 13) \\ \sum_{j=1}^{13} x_{ij} \leq 1 (i = 1, 2, \dots, 20) \\ x_{ij} = 1 | 0 \end{cases}$$

5.1.2 0-1 整数规划模型的求解

我们运用 LINGO 进行求解如上 0-1 整数规划模型，得到了结果如表：

表 4 A 城区交巡警平台所封锁的路口及最长距离和时间

该市 A 城区交巡警平台	封锁的交叉路口	距离(单位：百米)	时间 (min)
A3	16	60.2557	6.02557
A7	29	80.1546	8.01546
A8	48	30.9946	3.09946
A9	30	34.923	3.4923
A10	22	77.0792	7.70792
A11	12	37.9135	3.79135
A12	24	35.9163	3.59163
A13	23	5	0.5
A14	21	32.6497	3.26497
A15	28	47.5184	4.75184
A16	14	67.4166	6.74166
A17	62	78.2053	7.82053
A19	38	76.3928	7.63928

5.3 问题一第三小问的思路分析

问题一第三小问中，要求在 A 区增设 2-5 个交通巡警平台，以解决 A 区交巡警服务平台的工作量不均衡和有些平台出警时间过长的情况。首先，每个交通巡警平台的工作量要尽可能的均衡，即每个平台管辖的路口数目要尽可能相同，也就是方差最小；另外，每个巡警平台出警的时间也要尽可能缩短，即到达每个平台所管辖的距离最远的路口的时间要尽可能最短。此时，可根据如上两个目标建立如下双目标的 0-1 整数规划模型。

5.3.1 双目标的 0-1 整数规划模型的建立

(1) 目标函数：

求解时，每个平台所管辖的路口数目要均衡，即平台所管辖路口的数目的方差要达到最小： $\sum_{j=1}^{92} x_{ij}$ 表示第 i 个平台管辖的路口数目； $\frac{92}{\sum_{i=1}^{92} y_i}$ 表示 A 区每个平

台管辖路口数目的平均值； $\sum_{i=1}^{92} y_i$ 表示 A 区设置的平台的总数目）：

$$z = \min \left(\frac{\sum_{i=1}^{92} \left(\sum_{j=1}^{92} x_{ij} - \frac{92}{\sum_{i=1}^{92} y_i} \right)^2 \times y_i}{\sum_{i=1}^{92} y_i} \right)$$

每个平台的巡警出警时，要尽可能使每个平台到达最远距离的路口的时间最短：

$$\min = \max_{\substack{1 \leq i \leq 92 \\ 1 \leq j \leq 92}} \left(\frac{x_{ij} y_i d_{ij}}{v} \right)$$

（2）约束条件一：

设置的总平台数应当在 22-25 之间：

$$22 \leq \sum_{i=1}^{92} y_i \leq 25$$

（3）约束条件二：

第 i 个路口是否被选为巡警平台，若选为平台，则 $y_i=1$ ，否则为 0：

$$y_i = 0 | 1$$

（4）约束条件三：

第 j 个路口是否被第 i 个平台管辖，若被管辖，则 x_{ij} 为 1，否则为 0：

$$x_{ij} = 0 | 1$$

（5）约束条件四：

编号为前 20 的路口已经被选为平台：

$$\sum_{i=1}^{20} y_i = 20$$

综上所述，建立的双目标 0-1 整数规划模型为：

$$\min = \max_{\substack{1 \leq i \leq 92 \\ 1 \leq j \leq 92}} \left(\frac{x_{ij} y_i d_{ij}}{v} \right)$$

$$z = \min \left(\frac{\sum_{i=1}^{92} \left(\sum_{j=1}^{92} x_{ij} - \frac{92}{\sum_{i=1}^{92} y_i} \right)^2 \times y_i}{\sum_{i=1}^{92} y_i} \right)$$

$$s.t. \begin{cases} 22 \leq \sum_{i=1}^{92} y_i \leq 25 \\ x_{ij} = 0 | 1 \\ y_i = 0 | 1 \\ \sum_{i=1}^{20} y_i = 20 \\ i = 1, \dots, 92 \\ j = 1, \dots, 92 \end{cases}$$

5.3.2 双目标的 0-1 整数规划模型的求解

在求解该模型时，由于双目标规划求解困难，因而我们将双目标转化为单目标，分步求解。

第一步：使用 LINGO 软件，求解约束条件下目标函数 A 的最优解。此时，求得平台管辖的路口数目的方差最小时，需要增设的平台的编号及其所管辖的路口编号，见表：

表 5 增设的平台的编号及其所管辖的路口编号

增设的平台编号	管辖的路口编号
28	28 29
38	38 39
61	61
87	87 92

第二步：在求解目标函数 A 时，未考虑每个平台的出警时间。因而，应在目标函数 A 的最优解下，考虑出警时间，对每个平台的管辖范围进一步优化。得到如下结果，见表：

表 6 优化后平台的编号及其所管辖的路口编号

平台编号	管辖的路口编号
1	68 71 72 73 74 79
2	44 70 76 78
3	43 54 55 65 67

4	57 60 62 63 64 66
5	49 50 53 56
6	51 52 58 59
7	30 48 61
8	35 46 47
9	32 34 45
11	26 27
12	25
13	21 22 23 24
15	31
16	33 36 37
17	40 41 42
18	77 80 85
19	69 75 81 82 83
20	86
28	29
38	39
87	87 88 89 90 91 92

5.4 问题二第一小问的分析

我们对该市现有交巡警服务平台设置方案进行了分析研究,发现存在明显不合理。考虑到附件二“全市交通路口节点数据”中给出的每个节点的案发率,我们对问题一建立的“最优化模型”进行了模型的优化。不仅要考虑在 3 分钟内到达路口,而且要根据每个节点案发率的不同,最优的给平台分配路口。将 A 区案发率归一化,近似的作为节点的距离权重。对每一个节点,优先选取上述值中最小值代表的平台,表示受其管辖。

5.4.1 改进后最优化模型的求解

我们运用 MATLAB 编程求解出如下每个城区各交巡警服务平台管辖的交叉路口和最远管辖路口,结果见如下表 7 到表 16:

B 城区:

表 7 B 城区各交巡警服务平台管辖的交叉路口

该市 B 城区交巡警平台	所管辖的交叉路口
B1	101 102 103
B2	104 105 106 107 108 109 110 111 112 117 118 119 120 121 122 123
B3	113 114 115 116 126 128 129 131 136 154
B4	124 127 130 133 134 138 139 140 141 142 145 146 147 150 151
B5	135 137 143 144
B6	155 156 157 158 159 160 161 162 163 164 165
B7	148 149 152 153

B8	125 132
----	---------

表 8 B 城区各交巡警服务平台管辖的最远交叉路口

该市 B 城区交巡警平台	最远路口编号	最远出警距离(百米)
B1	101	29.8634
B2	123	33.6526
B3	154	28.6269
B4	124	32.0984
B5	135	11.4142
B6	155	28.804
B7	153	44.7031
B8	125	27.0323

C 城区：

表 9 C 城区各交巡警服务平台管辖的交叉路口

该 C 城区交巡警平台	所管辖的交叉路口
C1	262 263 264 265
C2	248 249 250 251 252 255 258 259 260 261
C3	189 190 191 192
C4	254
C5	222 223 224 225 226 273 276 277 283
C6	215 216 230 231 240 241 242 243 244 246 253
C7	217 218 227 228 229
C8	232 233 234 235 236 237 238 239 245 247
C9	211 212 213 214 219 220 221
C10	183 193 194 195 196 197 198 199
C11	184 185 186 187 188
C12	200 201 202
C13	203 204 205 206 207 208 209 210 284 286 287
C14	274 275 278 279 280 281 282 285 288 289 290 291 292 295 296
C15	268 269 270 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316
C16	266 267 317 318 319
C17	256 257 271 272 293 294

表 10 C 城区各交巡警服务平台管辖的最远交叉路口

C 城区交巡警平台	最远路口编号	最远出警距离(百米)
C1	264	66.2233

C2	252	38.1331
C3	191	27.1108
C4	254	22.2036
C5	283	29.1224
C6	253	58.2595
C7	229	14.9832
C8	239	65.6856
C9	211	23.6934
C10	199	55.0909
C11	186	19.6354
C12	202	64.609
C13	207	68.6054
C14	288	38.6761
C15	316	54.0704
C16	317	54.7516
C17	257	32.2534

D 城区：

表 11 D 城区各交巡警服务平台管辖的交叉路口

该 D 城区交巡警平台	所管辖的交叉路口
D1	347 348 349 350 370 371
D2	351 352 353 354 355 356 357 358 359 360 368 369
D3	367
D4	344
D5	345 361 362
D6	363 364 365 366
D7	343 346
D8	337 338 340 341 342
D9	329 330 331 332 333 334 335 336 339

表 12 D 城区各交巡警服务平台管辖的最远交叉路口

该 D 城区交巡警平台	最远出警距离（百米）	最远路口编号
D1	370	78.0848
D2	369	41.8504
D3	367	17
D4	344	48.4071
D5	362	81.0691
D6	363	18.8215
D7	346	18.1108
D8	337	36.5829
D9	332	160.6282

E 城区：

表 13 E 城区各交巡警服务平台管辖的交叉路口

该市 B 城区交巡警平台	所管辖的交叉路口
E2	437 438 456
E3	427 432 433 434 435 436 457
E4	424 425 426 428 429 430 431
E6	416
E7	458 459
E8	417 418 419 420 421 422 423
E9	387 388 389 390 391 392 393 394 395 396
E10	397 398 399 400 405 406
E11	401 402 403 404 407 408 409 410 411 412 413 414 415
E12	452 453 454 455 460 461 462 463 464 469 470
E13	465 466 467 468 471 472
E14	445 446 448 449 450 451 473 474
E15	439 440 441 442 443 444 447

表 14 E 城区各交巡警服务平台管辖的最远交叉路口

该市 E 城区交巡警平台	最远出警距离（百米）	最远路口编号
E2	32.0156	438
E3	28.1603	457
E4	11.1301	426
E6	28.2843	416
E7	62.3265	459
E8	81.1882	419
E9	191.0506	387
E10	17.5	397
E11	48.0903	407
E12	35.1172	452
E13	39.9835	471
E14	41.2557	446
E15	37.2132	439

F 城区：

表 15 F 城区各交巡警服务平台管辖的交叉路口

该市 F 城区交巡警平台	所管辖的交叉路口
F1	550 551 555 556 557 558 559 561 563 564 565
F2	532 533 534 535 543 544 545 546 547 552 553 554
F3	492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 516 517 518 519 520 521 522 523 529 530
F4	512 513 514 515 524 525 526 527 528 536 537 538 539 542
F5	573 575 576 577 578 579 580 581 582
F6	562 566 567 568 569 574
F7	486 490 491 531 548 549
F8	487 488 489 560
F9	510 511
F10	540 541 570
F11	571 572

表 16 F 城区各交巡警服务平台管辖的最远交叉路口

该市 F 城区交巡警平台	最远出警距离（百米）	最远路口编号
F1	43. 5479	561
F2	32. 3557	533
F3	52. 3952	523
F4	55. 7349	513
F5	84. 7985	582
F6	65. 832	574
F7	39. 4653	486
F8	47. 6987	560
F9	66. 0682	510
F10	70. 4178	541

5.5 问题二第二小问的分析

根据题可知 P 点处于 A 区。按照第一题中所计算得的分配方案，A 区的 20 平台中有封锁任务的 13 个平台，去该区的 13 个出入口，执行封锁任务，剩余 7 个没有封锁任务的平台，从 P 点开始，根据广度优先原则，依次走遍 A 区的所有路口节点，进行排查。与此同时，城市除 A 区外的其余 5 个区的所有平台，根据距离就近原则抽调 17 个平台去该城市的所有出入口进行封锁。B~F 区未分配封锁任务的平台，照常值班处理日常事务。

五、 模型的评价与优化

8.1 模型的优缺点分析

8.1.1 模型的优点

1、Floyd 算法是一种动态规划算法。此算法简单有效，由于三重循环结构紧凑，对于闭包稠密图，效率要高于 Dijkstra 算法。此算法容易理解，可以算出任意两个节点之间的最短距离，代码编写简单。

2、0-1 型整数线性规划是整数线性规划中的特殊情形，它的变量 x 仅取值 0 或 1。它和一般整数线性规划的约束条件形式是一致的。在实际问题中，如果引入 0-1 变量，就可以把有各种情况需要分别讨论的线性规划问题统一在一个问题中讨论了。求解得到的数据也很直截了当，十分简便。

8.1.2 模型的缺点

1、当需要处理的数据很多时，使用 Floyd 算法编程求解会十分繁琐，其算法的时间复杂度很高，会导致运行结果很慢。

2、在求解 0-1 规划时，尤其是此题中的双目标 0-1 规划，使用 LINGO 求解，代码的效率很低，相当于软件遍历所有可能的求解组合之后，进行对比，得到最优解，这会造成模型的求解过程非常慢。

8.2 模型的优化

1、问题一中，在求任意两点间的最短距离时，可以使用遗传算法对 Floyd 算法进行改进，以空间换取时间，通过提高其空间复杂度，以降低时间复杂度。

2、在求解双目标 0-1 规划模型时，寻找更多的约束条件，使用 MATLAB 编程求解，降低代码的运行时间。

六、 模型的推广

近年来，随着西安市经济的发展，现金的需求量进一步增加，现金大进大出的特征进一步呈现。如何合理有效的对我市现金进行调拨，实现调拨的最优化问题是近年来我市货币发行部门一直探讨和研究的热点问题。该问题可通过数学建模的方法，建立相关数学模型，并利用相关软件求得最优决策方案，以便实现决策的科学化和定量化。在现有的调拨模式下提出优化决策方案，再结合我市的地理位置、调拨工作量等数据对我市发行基金调拨工作提出优化方案是货币发行工作的创新和进步。这会为发行基金调拨决策提供科学的依据，为探索建立区域发行基金调拨中心提供理论指导，实现经验管理走向科学管理的转变。

根据各个盟市近几年的货币调运量及销毁量的整体态势，可建立相关货币中心设置模型及其各货币管理中心的管辖范围及其相关最优调度方案。

另外，近几年来随着网购热潮的兴起，快递公司的任务日趋增加，部分优秀的快递公司面临扩建的重大任务。这使得快递公司需要增设新的分公司。一个优秀的分公司增设方式不仅需要考虑到现有的快递公司业务，同时也要考虑到今后的扩建或缩小规模的方式。而这一问题仍然是管理中心的确定问题。与此同时，快递公司为了实现送快件成本的最小化，需要为各个快递员设置最优送快件的方式。这些问题可能会涉及到诸如中国邮递员问题或推销员问题。

参考文献

- [1] 冯晓东, 梁红硕. 路由算法的设计与优化(M). 石家庄: 石家庄职业技术学院学报, 2015。
- [2] 黄沙日娜, 赵国亮, 朱捷. 双目标瓶颈指派问题的遗传算法(A). 黑龙江科技大学, 2014。
- [3] 谢金星, 薛毅. 优化建模与 LINGO/LINGO 软件(M). 北京: 清华大学出版社, 2005。
- [4] 张瑞丰. 精通 MATLAB 6.5 (M). 北京: 中国水利水电出版社, 2004。
- [5] 全国大学生数学建模竞赛组委会. 大学数学建模的理论与实践(M). 长沙: 湖南教育出版社, 2004。

附件

运行环境 Matlab 2015b,Lingo 11.

1. 数据录入

利用 Matlab 的数据导入功能，将问题中的 Excel 中五张表的数据，通过复制、粘贴建立数据文件 (*.txt)，分别命名 xianlu,lukoujieidan,pingtai,churukou,renkou, 并保存至 Matlab 的工作文件夹中，等待程序读取数据。

2. 建立.m 文件，并命名为 myfloyd，计算两点间最短路径及距离

```
function [dist,mypath]=myfloyd(a,sb,db)
```

```
% a—邻接矩阵(aij)
```

```
% sb 起点的标号; db 终点的标号
```

```
% dist 最短路的距离; mypath 最短路的路径
```

```
n=size(a,1); path=zeros(n);
```

```
for i=1:n
```

```
    for j=1:n
```

```
        if a(i,j)~=inf
```

```
            path(i,j)=j; %j 是 i 的后续点
```

```
        end
```

```
    end
```

```
end
```

```
for k=1:n
```

```
    for i=1:n
```

```
        for j=1:n
```

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

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

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

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
dist=a(sb,db);
```

```
mypath=sb; t=sb;
```

```
while t~=db
```

```
    temp=path(t,db);
```

```
    mypath=[mypath,temp];
```

```
    t=temp;
```

```
end
```

```
return
```

3. 建立.m 文件，并命名为 yiwen，计算 A 区各平台管辖范围

```
load xianlu.txt,load lukoujiedian.txt,load pingtai.txt,load fal.txt;
```

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

```
    A(i,:)=lukoujiedian(i,:);
```

```

end
num=0;
A0=zeros(140,3);
for i=1:1:582
    if xianlu(i,1)<=92 && xianlu(i,2)<=92
        num=num+1;
        A0(num,1)=xianlu(i,1);A0(num,2)=xianlu(i,2);
    end
end
for i=1:1:140

A0(i,3)=sqrt((A(A0(i,1),2)-A(A0(i,2),2))^2+(A(A0(i,1),3)-A(A0(i,2),3)
)^2);
end
A1=zeros(92);
for i=1:1:140
    A1(A0(i,1),A0(i,2))=A0(i,3);
end
for i=1:1:92
    for j=1:1:92
        if A1(i,j)==0
            A1(i,j)=inf;
        end
        if i==j
            A1(i,j)=0;
        end
    end
end
for i=1:1:92
    for j=1:1:92
        if A1(i,j)~=inf
            A1(j,i)=A1(i,j);
        end
    end
end
A2=zeros(20,92);
for i=1:1:20
    for j=1:1:92
        A2(i,j)=myfloyd(A1,i,j);
    end
end
A3=zeros(92,2);
for i=1:1:92
    A3(i,1)=min(A2(:,i));
end

```

```

    for j=1:1:20
        if A3(i,1)==A2(j,i)
            A3(i,2)=j;
        end
    end
end
end
A5=zeros(1,20);
for i=1:1:92
    A5(A3(i,2))=A5(A3(i,2))+fal(i,1);
end

```

4. 建立.lg4 文件，并命名为 **erwen1**，计算 A 区封锁交通要道时的调度方案

```

sets:
pt/1..20/;
lk/1..13/;
link(pt,lk):k,d;
endsets
data:
d=A2;
A2 为 yiwen.m 文件运行后得到的 20*92 的矩阵
enddata
min=@max(link:k*d);
@for(pt(i):@sum(lk(j):k(i,j))<=1);
@for(lk(j):@sum(pt(i):k(i,j))=1);
@for(link:@bin(k));
end

```

5. 建立.lg4 文件，并命名为 **erwen2**，计算增加平台时的最优数量及位置

```

model:
sets:
set1/1..92/:y;
set2/1..92/;;
set3(set1,set2):x,d,v;
endsets
data:
d=A1;
A2 为 yiwen.m 文件运行后得到的 92*92 的矩阵
enddata
min=@sum(set1(i)|i#ge#21:y(i));
@for(set2(j):@sum(set1(i):x(i,j))=1);
@for(set3(i,j):x(i,j)*d(i,j)<=30);
@for(set1(i)|i#ge#21:@sum(set2(j):x(i,j))<=100*y(i));
@for(set3(i,j):@bin(x(i,j)));
@sum(set1(i)|i#ge#21:y(i))<=5;

```

```
@sum(set1(i)|i#ge#21:y(i))>=2;
@for(set1(i)|i#ge#21:@bin(y(i)));
end
```

6. 建立.m 文件，并命名为 **erwen**，计算 A 区增加平台后的各平台管辖范围

```
load xianlu.txt,load lukoujiedian.txt,load pingtai.txt;
for i=1:1:92
    T(i,:)=lukoujiedian(i,:);
end
num=0;
T0=zeros(140,3);
for i=1:1:582
    if xianlu(i,1)<=92 && xianlu(i,2)<=92
        num=num+1;
        T0(num,1)=xianlu(i,1);T0(num,2)=xianlu(i,2);
    end
end
for i=1:1:140

T0(i,3)=sqrt((T(T0(i,1),2)-T(T0(i,2),2))^2+(T(T0(i,1),3)-T(T0(i,2),3)
)^2);
end
T1=zeros(92);
for i=1:1:140
    T1(T0(i,1),T0(i,2))=T0(i,3);
end
for i=1:1:92
    for j=1:1:92
        if T1(i,j)==0
            T1(i,j)=inf;
        end
        if i==j
            T1(i,j)=0;
        end
    end
end
for i=1:1:92
    for j=1:1:92
        if T1(i,j)~=inf
            T1(j,i)=T1(i,j);
        end
    end
end
T2=zeros(24,92);
```

```

zs=[28 38 61 87];
pt=zeros(1,24);
for i=1:1:24
    pt(i)=i;
    if i>20
        pt(i)=zs(i-20);
    end
end
for i=1:1:24
    for j=1:1:92
        T2(i,j)=myfloyd(T1,pt(i),j);
    end
end
T3=zeros(92,2);
for i=1:1:92
    T3(i,1)=min(T2(:,i));
    for j=1:1:24
        if T3(i,1)==T2(j,i)
            T3(i,2)=pt(j);
        end
    end
end
end

```

7. 建立.m 文件，并命名为 **sanwen**，计算全城按区管理时各平台管辖范围

```

load xianlu.txt,load lukoujiedian.txt,load pingtai.txt;
R=lukoujiedian;
R0=zeros(928,3);
for i=1:1:928
    R0(i,1)=xianlu(i,1);
    R0(i,2)=xianlu(i,2);
    R0(i,3)=sqrt((R(R0(i,1),2)-R(R0(i,2),2))^2+((R(R0(i,1),3)-R(R0(i,2),3))^2));
end
R1=zeros(582);
for i=1:1:928
    R1(R0(i,1),R0(i,2))=R0(i,3);
end
for i=1:1:582
    for j=1:1:582
        if R1(i,j)==0
            R1(i,j)=inf;
        end
        if i==j
            R1(i,j)=0;
        end
    end
end

```

```

        end
    end
end
for i=1:1:582
    for j=1:1:582
        if R1(i,j)~=inf
            R1(j,i)=R1(i,j);
        end
    end
end
end
R2=zeros(80,582);
for i=1:1:80
    for j=1:1:582
        R2(i,j)=myfloyd(R1,pingtai(i,1),j);
    end
end
end
R3=zeros(582,2);
for i=1:1:582
    R3(i,1)=min(R2(:,i));
    for j=1:1:80
        if R3(i,1)==R2(j,i)
            R3(i,3)=pingtai(j);
        end
    end
end
end
end

```