基于 0-1 整数规划模型的交巡警服务平台位置优化问题

摘要

交巡警是交通警察巡警合一的警务模式，是将刑事执法、治安管理、交通管理、服务群众四大职能有机融合的新型防控体系。本文通过建立**整数规划模型**，分析了日常巡逻时的平台位置设置问题和封锁道路时任务分配问题。该问题的研究能为交巡警平台规划提供指导性意见。

针对问题一，本文先运用 **Flody 算法**求出各交巡警服务平台（后称“平台”）与各路口节点（后称路口），设计距离及任务多目标**混合整数规划**模型，对 3 分钟无法到达的六个点按就近分配处理，给出较优的平台管辖范围。

针对问题二，本文以最大封锁时间最短为目标，将问题转化为极小化极大问题，建立 0-1 整数规划模型，得到最短的最大封锁时间为 **8.015min**。以最短的最大封锁时间为约束，再建立以各个平台完成各自封锁任务的平均时间最短，即封锁时间总和最短为目标的指派模型，最终得到 A 区交巡警服务平台警力合理的调度方案。

针对问题三，在原 20 个平台不变的情况下，本文先以实现所有路口均可在 3 分钟内到达为约束条件，建立增设平台的 **0-1 整数规划模型**。再附加问题一的距离及任务 0-1 整数规划模型为约束。最终得到任务均衡要增加平台最少 4 个，分别在 **29,38,48,92**。

针对问题四，本文先进行平均每个平台负责的节点数，3 分钟达到的节点数，各区平均发案率，各区人口，平均每个平台负责的区域面积等信息统计，计算相关指标均值和方差，发现原设置有 138 个路口无法在三分钟内到达，还有各区出警时间、任务不平衡的问题。再通过只增减不调平台的方法改变任务不均衡和减少 3 分钟内无法到达的路口节点。

针对问题五，逃犯若想逃出主城六区，就必然经过 17 个全市出入口。在逃犯走最短路径到达 17 个全市出入口的前提下，以就近原则和封锁时间小于逃逸时间（减 3 分钟）为约束条件，建立对 17 各出入口的 **0-1 整数规划**指派模型，得到局部最优的平台封锁出入口分配方案。此外，我们还分析了逃犯不走最短路线情况下，以 32 号路口为中心的 1-10 级（超过 10 级逃犯出市）的封锁圈。

关键字：交巡警服务平台　0-1 整数规划　混合整数规划　Floyd 算法

一、 问题重述

1.1 问题一重述

合理分配中心城区 A 内各交巡警服务平台的管辖范围, 使其在所管辖的范围内出现突发事件时, 能在 3 分钟内有交巡警到达事发地, 其中警车的速度为 60 km/h 。

1.2 问题二重述

对于重大突发事件, 如何调度 A 区内 20 个平台的警力资源, 快速全封锁该区的 13 个出入口。

1.3 问题三重述

在 A 区内增加 2~5 个平台, 以解决服务平台的工作量不均以及部分地方出警时间过长的实际问题。

1.4 问题四重述

针对全市 6 区的情况, 分析研究现有交巡警服务平台设置方案的合理性, 并对明显不合理的平台设置给出改进方案。

1.5 问题五重述

如果地点 P 处发生重大刑事案件, 在案发 3 分钟后接到报警, 犯罪嫌疑人已驾车逃跑。试设计调度全市服务平台警力资源的最佳围堵方案。

二、 模型假设

(1) 出警过程中, 警车行驶的总是最短路径;
(2) 所有道路均为双行道;
(3) 在较短的时间内, 服务平台管辖范围里不会出现两个及两个以上的突发事件;
(4) 警车抵达路口后, 封锁路口需要的时间忽略不计;
(5) 交巡警服务平台接警后, 准备时间忽略不计, 视为立刻出发;
(6) 犯罪嫌疑人驾车逃跑时, 行驶速度为 60km/h 。

三、 符号说明

表 1　主要符号说明

符号	意义	单位
$d_{ij}^{(k)}$	最短路径	m
D	距离矩阵	
t_p	围堵时间	min

四、 问题分析

4.1 问题一的分析

由题目附件 2 中所给信息可得 A 区交巡警平台与路口节点的路线图（无向图），通过计算图中各边的权值获得 A 区的路线图（赋权无向图），运用 **Floyd 算法**计算各平台到各路口节点的最短距离。我们先以三分钟到达作约束条件剔除 3 分钟未能到达的路口节点，对 3 分钟未能到达的路口节点按就近分配与最近平台绑定；我们综合考虑任务量和出警路程（单程），对 3 分钟内能到达的路口作**混合整数规划**确定任务和路程均衡的各平台管辖范围。

4.2 问题二的分析

对于重大突发事件, 调度 A 区 20 个交巡警服务平台的警力资源, 对该区 13 个出入口实现快速全封锁。所谓全封锁, 即为 A 区 13 个出入口全部有警力到达。因此, 要实现最快速的全封锁, 就应该使得警车到达最后封锁的路口所用时间最短, 即各服务平台与其要封锁的路口间的最长距离要最短。在此基础上对平台封锁任务进行再优化，实现封锁总时间最短。

4.3 问题三的分析

根据问题一已分配的服务平台管辖范围, 结合题目所给图形可以发现现有交巡警服务平台的分布主要存在两个问题: 首先是存在有任何平台在 3 分钟内都无法到达的路口, 我们首先要为这些点根据实际情况增设部分交巡警服务平台。其次是任务量问题, 我们可以根据所给数据计算出每个交巡警服务平台的任务量（所管辖范围的发案率与路程的乘积）, 根据任务量的大小再调整部分交巡警服务平台。

4.4 问题四的分析

问题四要求按照设置交巡警服务平台的原则和任务, 分析研究全市 6 区现有交巡警服务平台设置方案的合理性。而交巡警服务平台应该优先保证 3 分钟内到达所有路口。因此, 先找出全市的所有三分钟无法到达的路口；再分区优化，添加平台来保证 3 分钟内到达所有路口。

4.5 问题五的分析

在犯罪嫌疑人逃跑 3 分钟后接到报案, 开始对其进行围捕,3 分钟后犯罪嫌疑人已经离开案发现场较远, 在不能准确判断逃离方向的情况下, 选择逃犯最有可能出逃的路线——出城路径最短的路线。确定路线后, 调度距离 17 个出入口最近的平台封锁出入口，比较其封锁时间与逃逸时间，验证其封锁性。我们还根据犯罪嫌疑人的逃跑位置，得到 1~10 级封锁圈。

五、 问题建立与求解

5.1 问题一：平台最优管辖范围模型的建立与求解

5.1.1 任务及路程均衡的平台最优管辖范围模型的建立

最短路径模型

1. 模型定义

   在交巡警服务平台管辖范围分配问题中，我们需要计算 A 区 92 个交通路口节点之间的最短路径距离。该问题可抽象为**加权无向图**的最短路径问题，具体定义如下：

$$\text{图} G = (V, E, W)$$

$$V = \{v_1, v_2, \ldots, v_{92}\} \quad (\text{节点集：92 个路口})$$

$$E = \{(v_i, v_j) \mid \text{节点} i \text{ 与} j \text{ 有道路直接连接}\}$$

$$W = \{w_{ij} \mid w_{ij} = \text{节点} i \text{ 到} j \text{ 的实际距离}\}$$

2. 欧氏距离

   对于任意两个节点 $v_i(x_i, y_i)$ 和 $v_j(x_j, y_j)$:

$$d_{ij} = \frac{1}{10}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (\text{单位：公里})$$

3. 邻接矩阵

$$w_{ij}=\begin{cases} d_{ij}, & (v_i,v_j)\in E, \\ \infty, & v_i \text{ 与} v_j \text{ 之间无边}, \end{cases} \qquad i,j=1,2,\ldots,92.$$

$$w_{ii}=0, \quad i=1,2,\ldots,92.$$

4. 最短路径算法模型

采用 Floyd 算法计算最短路径，其动态规划递推式为：

$$d_{ij}^{(k)}=\min\left(d_{ij}^{(k-1)},\, d_{ik}^{(k-1)}+d_{kj}^{(k-1)}\right)$$

5.1.2 特殊节点处理

定义：特殊节点指所有平台到该节点的距离均 >3km 的节点（即不存在平台能在 3 分钟内到达）

$$b_{ij}=\begin{cases} 1, & \text{警台}i \text{ 到节点}j \text{ 距离不超过 3 分钟（公里）} \\ 0, & \text{警台}i \text{ 到节点}j \text{ 距离超过 3 分钟（公里）} \end{cases}$$

特殊节点集合：$S=\{j \mid \sum_{i\in[1,20]} b_{ij}, \text{其中} j\in[1,92]\}$

结果得到：S={28, 29, 38, 39, 61, 92}

根据就近分配原则得到：

$$\begin{cases} \text{平台 15 管辖节点 28,29} \\ \text{平台 16 管辖节点 38} \\ \text{平台 2 管辖节点 39} \\ \text{平台 7 管辖节点 61} \\ \text{平台 20 管辖节点 92} \end{cases}$$

5.1.3 平台最优管辖范围模型建立结果

决策变量

$$X_{ij}=\begin{cases} 1, & \text{警台}i \text{ 管辖节点}j \\ 0, & \text{警台}i \text{ 不管辖节点}j \end{cases}$$

目标函数

$$\min Z=\sum_{i\in[1,20]}\sum_{j\in[1,92]} d_{ij}X_{ij}$$

约束条件

$$
\begin{cases}
\textbf{非特殊节点}j\notin \text{S}\ \textbf{必须被 3 分钟内的平台覆盖:} \\
\sum_{i\in I} X_{ij}\cdot b_{ij}=1, \quad \forall j\notin S \\
\textbf{特殊节点强制就近分配:}\ X(15,28)=1, X(15,29)=1, X(16,38)=1, \\
X(2,39)=1, X(7,61)=1, X(20,92)=1 \\
\textbf{平台自身管辖:} \\
X(i,i)=1 \\
\textbf{每个节点被唯一平台管辖:} \\
\sum_{i=1}^{20} X_{ij}=1, \quad \forall j=1,\ldots,92
\end{cases}
\tag{1}
$$

5.1.4 模型求解

算法：贪婪算法

对于每个非特殊节点 j, 找到使其距离 $d(i,j)$ 最短的交警平台 i:

$$
i^* = \arg\min_{i:b_{ij}=1} d_{ij}
$$

5.1.5 模型优化

考虑到任务量分配的合理性，增加任务均衡性为约束条件，建立优化模型。

$$
\sigma_T = \sqrt{\frac{1}{20}\sum_{i=1}^{20}(T_i-\bar{T}_A)^2} \le 2
$$

其中，$\bar{T}_A$ 为 A 区的各平台平均任务量，$\bar{T}_A=6.225$；T_i 为第 i 个平台的任务量。

完整数学模型为

$$
\min Z = \sum_{i\in[1,20]}\sum_{j\in[1,92]} d_{ij}X_{ij}
$$

$$
s.t.\begin{cases}
\sum_{i\in I} X_{ij}\cdot b_{ij}=1, \quad \forall j\notin S \\
X(i,i)=1 \\
X(15,28)=1, X(15,29)=1, X(16,38)=1, X(2,39)=1, X(7,61)=1, X(20,92)=1 \\
b_{15,28}=b_{15,29}=b_{16,38}=b_{2,39}=b_{7,61}=b_{20,92}=1 \\
\sum_{i=1}^{20} X_{ij}=1, \quad \forall j=1,\ldots,92 \\
\sigma_T=\sqrt{\frac{1}{20}\sum_{i=1}^{20}(T_i-\bar{T}_A)^2}\le 2
\end{cases}
$$

5.1.6 结果分析与展示

表 2　A 区各交警平台管辖节点

平台	管辖节点	平台	管辖节点
1	1, 18, 44, 66, 72, 80	11	11, 25, 26, 27
2	17, 39, 40, 78	12	12
3	2, 3, 54, 55, 68, 76	13	13, 21, 22, 23, 24
4	4, 57, 60, 62, 63, 64, 65	14	14
5	6, 49, 50, 51, 52, 53, 59	15	15, 28, 29, 31
6	7, 48, 56, 58	16	35, 38
7	5, 30, 34, 47, 61	17	41, 42, 43
8	9, 16, 32, 36, 37	18	19, 85, 87, 88, 89, 90, 91
9	8, 33, 45, 46	19	67, 69, 70, 71, 73, 74, 75, 77, 79
10	10	20	20, 81, 82, 83, 84, 86, 92

表 3　交警平台管辖节点数与负荷

平台	节点数	负荷	平台	节点数	负荷	平台	节点数	负荷	平台	节点数	负荷
1	6	7.1	6	4	5.4	11	4	6.2	16	2	2.6
2	4	6.4	7	5	8.1	12	1	2.4	17	3	4.5
3	6	8.2	8	5	7.4	13	5	8.5	18	7	8.2
4	7	7.3	9	4	6.4	14	1	2.5	19	9	8.3
5	7	8.5	10	1	1.6	15	4	6.4	20	7	8.5

总负荷(案发率 * 路程):124.500000m;负荷标准差为:2.308309;总路程为:167.427128m;最大负荷为（案发率 * 路程）:8.500000m。
此模型求解结果为任务与路程双目标的局部最优解 (未达到负荷标准差最小)。

5.2 问题二：封锁出入口时间优化模型

5.2.1 欧式距离

交警平台 i 与出入口 e_j 的距离:

$$d(i,e_j)=\sqrt{(x_{e_j}-x_i)^2+(y_{e_j}-y_i)^2}$$

距离矩阵:

$$\mathbf{D}_{20\times 13}=\begin{bmatrix} d(p_1,e_1) & \cdots & d(p_1,e_{13}) \\ \vdots & \ddots & \vdots \\ d(p_{20},e_1) & \cdots & d(p_{20},e_{13}) \end{bmatrix}$$

5.2.2 扩展距离矩阵

$$\mathbf{D}_{ext}=[\mathbf{D}_{20\times 13}\mid \mathbf{0.0001}_{20\times 7}]_{20\times 20}$$

5.2.3 最小化最大封锁时间模型

决策变量:

$$X_{ij}=\begin{cases}1, & \text{警台}i\text{ 管辖节点}j\\ 0, & \text{警台}i\text{ 不管辖节点}j\end{cases}$$

t_j为封锁节点j的时间

目标函数:

$$\min\left(\max_j t_j\right)\quad \text{其中}\quad t_j=\sum_i X_{ij}d_{ij}\quad \text{(时间单位:分钟)}$$

约束条件:

- 每个节点必须封锁:

$$\sum_{i=1}^{20} X_{ij}=1,\quad \forall j\in\{\text{13 个出入口}\}$$

- 管辖唯一性:

$$\sum_{j\in\{\text{13 个出入口}\}} X_{ij}\le 1,\quad \forall i=1,\ldots,20$$

5.2.4 结果展示

最大封锁时间为 8.0155min。封锁时间总和为 49.1231min.

表 4　交警平台序号与围堵节点的对应关系

交警平台序号	10	16	9	14	11	13	12	15	7	8	2	5	4
围堵的节点	12	14	16	21	22	23	24	28	29	30	38	48	62

5.3 问题三：增设平台优化模型

5.3.1 阶段 1：最小化新增平台数模型

决策变量

$$\text{x(i)} = \begin{cases} 1, & \text{在节点} i \text{ 新增平台} \\ 0, & \text{否则} \end{cases} \quad \text{其中 } i \in [21, 92]$$

$X_{ij} \in \{0, 1\}$

目标函数

$$\min y = \sum_{i \in [21,92]} x(i) \quad \text{其中 } y \in \{2, 3, 4, 5\}$$

约束条件

- 非平台不管辖:

$$X_{ij} \leq x(i)$$

- 全覆盖约束:

$$\sum_{i=1}^{92} b_{ij} X_{ij} \geq 1, \quad \forall j$$

- 管辖唯一性:

$$\sum_{i=1}^{92} X_{ij} = 1, \quad \forall j$$

- 新增数量限制:

$$2 \leq \sum_{i \in [21,92]} x(i) \leq 5$$

5.3.2 阶段 2：最小化最大平台任务量模型

我们定义每个平台 i 的任务量为该平台所管辖的所有节点的发案率之和。

设 f_j 为节点 j 的发案率，则平台 i 的任务量：$F_i = \sum_{j=1}^{92} f_j X_{ij}$

$$\min \max_i DF_i$$

$$s.t. \quad \begin{cases} \text{阶段 1 所有约束} \\ \sum_{i=21}^{92} x_i = y_{\min} \quad \text{(固定新增数量)} \\ DF_i = \sum_{j=1}^{92} f_j X_{ij} d_{ij} \quad \forall i \\ X(i,i) = X(15,28) = X(15,29) = X(16,38) = X(2,39) = X(7,61) = X(20,92) = 1 \\ b_{15,28} = b_{15,29} = b_{16,38} = b_{2,39} = b_{7,61} = b_{20,92} = 1 \end{cases}$$

5.3.3 模型求解

步骤一：求解最小 y 能覆盖所有节点

- 对所有 y=2 到 5 穷举新增位置组合
- 若新增位置满足 $\sum_j b_{ij} = 1$ 对所有 j 都成立，则可行

步骤二：优化新增平台位置

- 根据最小化最大平台任务量模型计算任务均衡性，得到最理想组合

5.3.4 结果分析与展示

增加最少 4 个平台，分别在 29,38,48,92 路口节点。

表 5　各交警平台管辖节点

平台	管辖节点	平台	管辖节点
1	2, 43, 68, 74, 75, 76	13	13, 21, 22, 23, 24
2	3, 17, 44, 71, 73	14	14
3	54, 55, 64, 66, 67	15	15, 31
4	4, 57, 58, 60, 62, 63, 65	16	8, 35
5	7, 49, 50, 51, 52, 53, 56	17	41, 42, 70, 72
6	5, 6, 59	18	1, 77, 78, 82, 88
7	9, 30	19	19, 69, 79, 80, 83
8	16, 34, 37, 45, 47	20	81, 84, 85, 86, 89, 91
9	36, 46	21	28, 29
10	10	22	32, 33, 48, 61
11	11, 25, 26, 27	23	18, 20, 87, 90, 92
12	12	24	38, 39, 40

总负荷:124.500000m；负荷标准差为 (3456 种情况下最小的):2.085209；总路程为:136.395175m。

表 6　各平台节点数和负荷

平台	节点数	负荷	平台	节点数	负荷	平台	节点数	负荷	平台	节点数	负荷
1	6	7.7	7	2	4.2	13	5	8.5	19	5	5.4
2	5	7.8	8	5	7.4	14	1	2.5	20	6	7.3
3	5	4.3	9	2	2.3	15	2	3.7	21	2	2.7
4	7	7.6	10	1	1.6	16	2	3.8	22	4	4.9
5	7	8.0	11	4	6.2	17	4	4.5	23	5	6.6
6	3	5.5	12	1	2.4	18	5	5.3	24	3	4.3

5.4 问题四: 平台增设优化模型

5.4.1 模型建立

决策变量

$$x_i = \begin{cases} 1 & \text{在节点} i \text{ 新增平台} \\ 0 & \text{否则} \end{cases}, \quad \forall i \in \mathcal{U}$$

其中 $\mathcal{U}$ 为未覆盖交通节点集合

目标函数

$$\min \sum_{i \in \mathcal{U}} x_i$$

约束条件

- **全覆盖约束**:

$$\sum_{i \in S_j} x_i \geq 1, \quad \forall j \in U$$

其中 $S_j = \{i \in \mathcal{U} \mid d_{ij} \leq 3\,\text{km}\}$ 表示能覆盖节点 j 的候选平台集合。

- **二元约束**:

$$x_i \in \{0, 1\}, \quad \forall i \in U$$

5.4.2 模型求解

贪心算法求解

步骤一: 识别未覆盖交通节点集合 $U = \left\{ j \in [1, 2, \cdots, 582] \mid \min_{i \in [1,2,\cdots,80]} d_{ij} > 3\,\text{km} \right\}$

步骤二: 初始化新增平台位置集合 $S = \emptyset$

步骤三: 每次选择能覆盖最多未覆盖节点的位置新增平台，更新 S 和 U

步骤四: 迭代直到所有节点均被覆盖，输出 S

5.5 问题五: 围捕方案优化模型

5.5.1 模型建立

时间

交巡警到达围堵点的时间为 t_p

$$t_p \leq t_{escape}$$

其中，t_{escape} 是嫌疑犯到达该点的时间（3 分钟 + 最短路径时间）

圈层

一级围堵圈 Q_1: 与 32 节点直接相邻的节点

二级围堵圈 Q_2: 与一级围堵圈相邻的节点, 不包括一级节点 …

交警平台调度优化

目标函数：$min\ max\ t_p$

约束条件：每个平台 p 只负责一个围堵点 q，优先调度距离最近的平台

5.5.2 数学模型

目标函数

最小化所有围堵点的最大到达时间:

$$min\ max_{k \in K\ m \in M}\ t_{(P_m, S_k)}$$

其中，K 是围堵层级集合，S_k 是第 k 级围堵圈节点集合，P 是可用交警平台集合，P_m 是第 m 个交警平台的节点

约束条件

1. 每个 q 被至少一个 p 覆盖：$\sum_{m \in M} \geq 1 \, \forall \, k \in K$ 2. 每个 p 至多负责一个 q：$\sum_{k \in K} \leq 1 \, \forall \, m \in M$ 3.$t_{(P_m, S_k)} \leq 3 + t_{(32, S_k)}$

5.5.3 模型求解

最短路径

使用 Dijkstra 算法计算所有节点到 32 节点的最短距离

围堵圈生成的终止

$$3 + t_{(32,S_k)} \leq mint_{(P_m,S_k)}$$

5.5.4 结果展示

表 7　17 个平台与出入口对应关系

平台	路口	平台	路口	平台	路口
96	151	177	202	324	325
99	153	178	203	327	328
175	177	166	264	386	332
181	317	323	362	100	387
379	418	478	483	484	541
485	572	479	578		

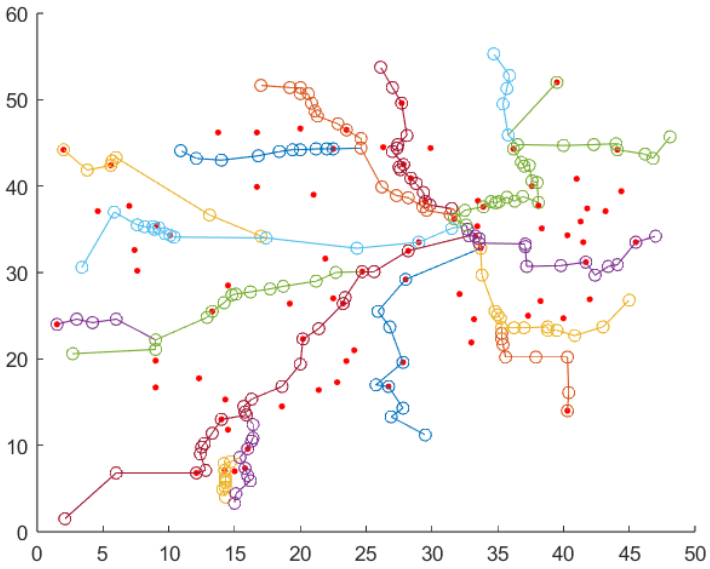

图 1　逃跑最短路线图

六、 模型评价

6.1 模型优点

模型求解采用合适的算法，例如贪心算法，节约算力和时间，同时保证准确性。

6.2 模型缺点

问题五没有考虑搜查范围尽可能小，仅封锁出入口，会使搜查范围过大。

附录 A　代码 $problem_1$

```
clc,clear,close all

load("weight.mat");
load("distances.mat");
fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E93");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];

scatter(T.X./10, T.Y./10, 10, 'b', 'filled'); % 蓝色实心点，大小100[1,5](@ref)
hold on
scatter(T.X(1:20)./10, T.Y(1:20)./10, 10, 'r', 'filled'); % 蓝色实心点，大小100[1,5](@ref)
scatter(T.X(14)./10, T.Y(14)./10, 100, 'g', 'filled'); % 蓝色实心点，大小100[1,5](@ref)


% % 创建digraph对象
% G = digraph(weight);
% [pathNodes, dist] = shortestpath(G, 1, 13)
% % 计算所有节点对的最短距离（距离矩阵）
% distances = distances(G);
% save("distances.mat","distances")


distances1 = distances(1:20,1:92);
result2 = find(sum(distances1 < 3) == 0);
[i,j] = min(distances1);
j(result2);
panduan_matrix = distances1 < 3;
panduan_matrix(15,28) = 1;
panduan_matrix(15,29) = 1;
panduan_matrix(16,38) = 1;
panduan_matrix(2,39) = 1;
panduan_matrix(7,61) = 1;
panduan_matrix(20,92) = 1;
anfashu = T.anfashu';
fuhe = T.anfashu'.*panduan_matrix;
fuhe(find(fuhe == 0)) = 10e6;

for num = 9:-0.1:8
    try
        [x, fval, assignment] = t1fun(fuhe,num);
    catch
        zuidarenwushu = num + 0.1
        break
```

```
    end
end

result = cell(20,1);
for i = 1:20
    result{i} = find(assignment(i,:) == 1);
end

res = zeros(20,2);
for k = 1:20
    res(k,1) = length(result{k});
    fprintf("第%d个交警平台管辖的节点:",k)
    fprintf(" %d",result{k})
    fprintf("\n")
    for kk = result{k}
        res(k,2) = res(k,2) + T.anfashu(kk);
    end
end

T = table([1:20]',res(:,1),res(:,2),'VariableNames',{'交警平台序号','管辖节点数','负荷'})
fprintf("3分钟到不的点:")
fprintf(" %d",result2)
fprintf("\n")
fprintf("\n总案发率（案发效*路程）:%6f\n",fval) % 122.8000
fprintf("\n标准差为:%6f\n",std(res(:,2))) % 2.002998
fprintf("\n总路程为:%6f\n",sum(sum(assignment.*distances1))) % 137.4178
fprintf("\n最案发数为（案发效*路程）:%6f\n",zuidarenwushu) % 8.5
% 6.6+6.9+7.2+8.1+7.3+6.2+2.4+8.5+2.5+4.8+6.1+7.3+7.9+8.4+5.5+5.7+7.6+5.2+7+1.6


% C = fuhe;
% nPerson = size(fuhe,1);
% nTask = size(fuhe,2);
% fun = @(x) fuhe.*x(1:92)';
% Aeq = kron(eye(nTask), ones(1, nPerson)); % 任务约束：每列和为1
% beq = ones(nTask, 1);               % 17×1全1向量
% lb = zeros(nPerson*nTask,1);        % 下限为0
% ub = ones(nPerson*nTask,1);         % 上限为1
% options = optimoptions('fminimax', 'Display', 'iter-detailed', 'UseParallel', true);
% [x, fval, exitflag] = fminimax(fun, zeros(1840,1), [], [], Aeq, beq, lb, ub, [], options);
% assignment = reshape(x, [nPerson, nTask])
```

附录 B　代码 $problem_2$

```
clc,clear,close all
```

```
load("weight.mat");
load("distances.mat");
fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E93");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];
entrance_out = readtable(fullPath, 'Sheet', '全市区出入口的位置',"Range","C2:C14");

entrance_out = table2array(entrance_out);
C_orig = distances(1:20,entrance_out);
C1 = [C_orig,0.0001.*ones(20,7)];
while true
    mat_result = C1;
    [M, linearIdx] = max(C1(:));
    [row, col] = ind2sub(size(C1), linearIdx); % 精确定位坐标
    maxx = C1(row, col);
    C1(row, col) = 0;
    if det(C1) == 0
        break
    end
end
C1(row, col) = maxx;
result = C1(:,1:13);
fprintf("到达的最长时间为%6f分钟\n",maxx)

C = result;
C(find(C == 0)) = 10e6;
nPerson = 20;
nTask = 13;

% 目标函数向量化
f = C(:);                  % 340×1向量

% 约束条件构建
Aeq = kron(eye(nTask), ones(1, nPerson)); % 任务约束：每列和为1
beq = ones(nTask, 1);                % 17×1全1向量

A = repmat(eye(nPerson), 1, nTask); % 人员约束：每行和 1
b = ones(nPerson, 1);                % 20×1全1向量

% 变量范围及整数约束
intcon = 1:length(f);      % 所有340个变量为整数
lb = zeros(size(f));       % 下限为0
ub = ones(size(f));        % 上限为1

% 求解混合整数规划
```

```
options = optimoptions('intlinprog', 'Display', 'final');
[x, fval] = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, options);

% 重构分配矩阵（20×17）
assignment = reshape(x, [nPerson, nTask]);

% 输出结果
% disp('最优分配矩阵（行为交警平台，列为封锁点）:');
% disp(assignment);          % 每列有且仅有一个1，每行至多一个1
disp(['总时间: ', num2str(fval)]);

[i,j] = find(assignment == 1);
T = table(i,entrance_out(j),'VariableNames',{'交警平台序号','围堵的节点'})
maxx
```

附录 C　代码 $problem_3$

```
clc,clear,close all

load("weight.mat");
load("distances.mat");

fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E93");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];

t1_3 = find(sum(distances(1:20,1:92) < 3) == 0)
[i,j] = find((distances(t1_3,:) < 3) == 1);
t1_3_3 = unique(j')

tot = cell(3456,1);
num = 0;
for i = t1_3_3
    for j = t1_3_3
        for m = t1_3_3
            for n = t1_3_3
                res = distances([i,j,m,n],t1_3) < 3;
                rr = sum(res);
                flag = any(rr(:) == 0);
                if flag == 0
                    num = num +1 ;
                    tot{num} = [i,j,m,n];
                end
            end
```

```
        end
    end
end

% ss = [];
% sss = [];
% for i = 1:length(tot)
%     dis = distances([1:20,tot{i}],1:92) < 3;
%     fuhe = dis.*T.anfashu';
%     fuhe(find(fuhe == 0)) = 10e6;
%     s = 0;
%     for num = 9:-0.1:0 % 8.5
%         try
%             [x, fval, assignment] = t1fun(fuhe,num);
%             result = cell(24,1);
%             for j = 1:24
%                 result{j} = find(assignment(j,:) == 1);
%             end
%             res = zeros(24,1);
%             for k = 1:24
%                 for kk = result{k}
%                     res(k,1) = res(k,1) + T.anfashu(kk);
%                 end
%             end
%             s = num;
%         catch
%             break
%         end
%     end
%     ss = [ss,s]; % 全是 8.5
%     sss = [sss,std(res)]; % 2.0852 91 [i,j] = min(sss) 28 48 87 38
% end

dis = distances([1:20,[28 48 87 38]],1:92) < 3;
fuhe = dis.*T.anfashu';
fuhe(find(fuhe == 0)) = 10e6;
[x, fval, assignment] = t1fun(fuhe,8.5);
result = cell(24,1);
for i = 1:24
    result{i} = find(assignment(i,:) == 1);
end
res = zeros(24,2);
for k = 1:24
    res(k,1) = length(result{k});
    fprintf("第%d个交警平台管辖的节点:",k)
    fprintf(" %d",result{k})
    fprintf("\n")
```

```
    for kk = result{k}
        res(k,2) = res(k,2) + T.anfashu(kk);
    end
end
T = table([1:24]',res(:,1),res(:,2),'VariableNames',{'交警平台序号','管辖节点数','负荷'})
fprintf("\n总案发率:%6f\n",fval)
fprintf("\n标准差为(3456种情况下最小的):%6f\n",std(res(:,2)))
fprintf("\n总路程为:%6f\n",sum(sum(assignment.*distances([1:20,[28 48 87 38]],1:92))))
```

附录 D　代码 $problem_4$

```
clc,clear,close all

load("weight.mat");
load("distances.mat");
fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E583");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];
jjpt = readtable(fullPath, 'Sheet', '全市交巡警平台',"Range","B2:B81");

dis1 = distances(table2array(jjpt),:);
[i,j] = find(min(dis1)<3 == 0);

t_4 = j

[i,j] = find((distances(t_4,:) < 3) == 1);

t_4_3 = unique(j')

dis = (distances(t_4_3,t_4) < 3).*distances(t_4_3,t_4);
scatter(T.X(t_4)./10, T.Y(t_4)./10, 10, 'b', 'filled');
anfashu = T.anfashu;
anfashu = anfashu(t_4)';
fuhe = dis.*anfashu;
% fuhe(find(fuhe == 0)) = 10e6;

% [x, fval, assignment] = t4fun(fuhe,20);
% result = cell(294,1);
% for i = 1:294
%     result{i} = find(assignment(i,:) == 1);
% end
% for i = 1:294
%     if length(cell2mat(result(i))) > 0
%         fprintf(" %d",cell2mat(result(i)))
```

```
%        fprintf("\n")
%    end
% end
% fval
```

附录 E　代码 $problem_5$

```
clc,clear,close all

load("weight.mat");
load("distances.mat");
load("dis.mat");
fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E583");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];
jjpt = readtable(fullPath, 'Sheet', '全市交巡警平台',"Range","B2:B81");
luko = readtable(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");
churuluko = readtable(fullPath, 'Sheet', '全市区出入口的位置',"Range","B2:B18");
ttt = readmatrix(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");

% mat = inf(582,582);
% targetPoint = [T.X./10, T.Y./10];
% points = [T.X./10, T.Y./10];
% distances = pdist2(targetPoint, points, 'euclidean');
% for i = 1:size(ttt,1)
%     mat(ttt(i,1),ttt(i,2)) = 1;
%     mat(ttt(i,2),ttt(i,1)) = 1;
% end
% weight = distances.*mat;
% weight(isnan(weight)) = 0;
% weight;
% save('weight.mat', 'weight');

result = distances(table2array(jjpt),table2array(churuluko));
matt = result < repmat(dis_list',80,1) + 3;
result1 = result.*matt; %   行号是交警平台按excel里按循序排好的，列号是出入口（按循序）
det(result1(1:17,:)) %  证明有效 前17个交警平台可以在罪犯没到达出入口之前到出入口

C = result1;
C(C == 0) = 1000;
nPerson = 80;
nTask = 17;

% 目标函数向量化
```

```
f = C(:);                  % 340×1向量

% 约束条件构建
Aeq = kron(eye(nTask), ones(1, nPerson)); % 任务约束：每列和为1
beq = ones(nTask, 1);                % 17×1全1向量

A = repmat(eye(nPerson), 1, nTask); % 人员约束：每行和 1
b = ones(nPerson, 1);                % 20×1全1向量

% 变量范围及整数约束
intcon = 1:length(f);      % 所有340个变量为整数
lb = zeros(size(f));       % 下限为0
ub = ones(size(f));        % 上限为1

% 求解混合整数规划
options = optimoptions('intlinprog', 'Display', 'final');
[x, fval] = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, options);

% 重构分配矩阵（20×17）
assignment = reshape(x, [nPerson, nTask]);

disp(['总时间：', num2str(fval)]);

[i,j] = find(assignment == 1);
aa = table2array(jjpt);
bb = table2array(churuluko);
aa = aa(i);
bb = bb(j);
cc = [aa,bb] % 第一列为交警平台 ，第二列为出入口
assignment .* C
max(max(assignment .* C))
```

附录 F　代码 $problem_5 jiedianfenji$

```
clc,clear,close all

fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E583");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];
jjpt = readtable(fullPath, 'Sheet', '全市交巡警平台',"Range","B2:B81");
luko = readtable(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");
churuluko = readtable(fullPath, 'Sheet', '全市区出入口的位置',"Range","B2:B18");
ttt = readmatrix(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");
```

```
scatter(T.X(jjpt.Var1)./10, T.Y(jjpt.Var1)./10, 10, 'b', 'filled')
hold on
scatter(T.X(churuluko.Var1)./10, T.Y(churuluko.Var1)./10, 10, 'k', 'filled')

for i = 1:928
    plot([T.X(luko.Var1(i))./10, T.X(luko.Var2(i))./10],[T.Y(luko.Var1(i))./10,
        T.Y(luko.Var2(i))./10],'b')
end

tot = {};
smallArray = [32];
while true
    bigArray = fun(smallArray);
    mask = ismember(bigArray, smallArray);
    smallArray = bigArray;
    result = bigArray(~mask);
    tot{end+1} = result';
    if any(ismember(bigArray, churuluko.Var1))
        break
    end
end

scatter(T.X(32)./10, T.Y(32)./10, 30, 'k', 'filled')

for i = 4 %   选择展示第1-10级拦截路口
    jibieluko = cell2mat(tot(i))'
    shuliang = length(jibieluko)
    scatter(T.X(tot{i})./10, T.Y(tot{i})./10, 10, 'r', 'filled') %
end
```

附录 G　代码 $problem_5shortestpath$

```
clc,clear,close all

load("weight.mat");
load("distances.mat");
fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E583");
T = removevars(T, {'Var4'});
T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];
jjpt = readtable(fullPath, 'Sheet', '全市交巡警平台',"Range","B2:B81");
luko = readtable(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");
churuluko = readtable(fullPath, 'Sheet', '全市区出入口的位置',"Range","B2:B18");
ttt = readmatrix(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");
```

```
hold on
scatter(T.X(jjpt.Var1)./10, T.Y(jjpt.Var1)./10, 10, 'r', 'filled')
scatter(T.X(jjpt.Var1)./10, T.Y(jjpt.Var1)./10, 10, 'r', 'filled')

dis_list = zeros(17,1);
G = digraph(weight);
num = 1;
for i = churuluko.Var1'
    [path, dis] = shortestpath(G, 32, i);
    dis_list(num) = dis;
    num = num +1;
    fprintf("罪犯从32到%d耗时%.2f分钟,路径为",i,dis)
    fprintf('%d ', path)
    fprintf('\n')
    plot(T.X(path)./10, T.Y(path)./10, 'o-');
end
save("dis.mat",'dis_list')
```

附录 H　代码 fun

```
function [smallArray] = fun(smallArray)

    fullPath = fullfile(pwd, '2011B附件2_全市六区交通网路和平台设置的数据表.xls');
    T = readtable(fullPath, 'Sheet', '全市交通路口节点数据',"Range","A2:E583");
    T = removevars(T, {'Var4'});
    T.Properties.VariableNames = ["xuhao", "X", "Y", "anfashu"];
    ttt = readmatrix(fullPath, 'Sheet', '全市交通路口的路线',"Range","A2:B929");
    [r, c] = find(ismember(ttt, smallArray));
    for i = 1:length(r)
        if c(i) == 1
            c(i) = 2;
        elseif c(i) == 2
            c(i) = 1;
        end
        smallArray(end + 1) = ttt(r(i),c(i));
    end

end
```

附录 I　代码 $t1fun$

```
function [x, fval,assignment] = t1fun(fuhe,num)
```

```
C = fuhe;
nPerson = size(fuhe,1);
nTask = size(fuhe,2);
f = C(:);
Aeq = kron(eye(nTask), ones(1, nPerson)); % 任务约束：每列和为1
beq = ones(nTask, 1);                % n×1全1向量
A = zeros(nPerson,nPerson*nTask);
for i = 1:nTask
    A(:,[(nPerson*(i-1)+1) : nPerson*i]) = diag(C(:,i));
end
b = num*ones((nPerson), 1);
intcon = 1:length(f);      % 所有变量为整数
lb = zeros(size(f));       % 下限为0
ub = ones(size(f)); % 上限为1
options = optimoptions('intlinprog', 'Display', 'off');
[x, fval,exitflag] = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, options);
assignment = reshape(x, [nPerson, nTask]);
end
```