

# 巡警服务平台的设置与调度

## 【摘要】

本文主要对交巡警服务平台的设置与调度进行设计,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,使警务资源最大化利用,建立最优化模型,并设计算法,采用 *Lingo* 求解和 *Matlab* 编程求解。针对本题,需要运用 *Floyd* 算法,通过编程求出任意两点间的最短路径,以及路径的相关线路矩阵,具体矩阵(见附表)。

针对问题一,我们需要解决三个问题。首先,我们需要调度全区的 20 个巡警服务平台的警力资源,在发生重大突发事件时快速封锁 13 条进出该区的交通要道。为了实现这一目标,我们使用了最短路径算法来确定每个巡警平台所需封锁的最近路口,并提供了具体的调度方案。此外,我们也考虑到了当这 20 个服务平台封锁了 13 个交通要道后,仍有 7 个服务平台可以作为备用,我们采用混合型拓扑结构,使其在其他 13 个平台出现紧急情况时提供补位。我们使用了均值-方差模型来选择这 7 个巡警服务平台的补位点位置,以尽量减小补位所需时间。我们计算出 20 个平台封锁 13 个交通要道所需的最快时间为 8.015 分钟。有关具体的调度方案,请参考 5.1.3.4 模型 II 的详细结果。

针对问题一的另外两个问题,我们需要确保在 3 分钟内能够到达事发地点。从提供的 A 区数据中,我们可以观察到每个节点的犯罪率各不相同,这意味着一些巡警服务平台可能承担了过大的工作负荷,而一些偏远节点可能导致出警时间过长,无法在 3 分钟内到达目的地。当增加平台时,会对出警时间产生影响,因此我们的目标是在减少到案发地时间的同时,实现每个平台工作负荷的均衡分配。根据计算结果,我们需要增加 5 个巡警服务平台,并将它们分别设置在点 28、39、61、79、90。具体增加的位置和管辖范围请参考具体结果 5.1.4.4。

针对问题二,在地点 P(第 32 个节点)发生重大刑事案件后的 3 分钟内接到报警,犯罪嫌疑人已经驾车逃跑。为了迅速围堵嫌疑人,我们派遣了全市的巡警平台展开行动。首先,我们采用邻接矩阵分析嫌疑人可能到达的节点情况。然后,根据嫌疑人可能出现的点位,我们对路线进行围堵,确定巡警平台需要占据的点位。最后,结合第一问中求得的各点位最短路径,我们就能够计算出巡警平台围堵的具体路线,进而确定围堵所需的时间和方案。这样的分析方法能够帮助我们制定高效的围堵策略,以确保尽快抓捕嫌疑人。最后我们求得围堵需要的时间为: 8.56 分钟,具体围堵方案见 5.2.5.5 模型 IIII 具体结果。

**关键词:** *Floyd* 算法 均值-方差模型 混合型拓扑结构 邻接矩阵 多目标规划

## 一、问题重述

### 1.1 引言

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。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。合理的分配能够使警务资源充分利用，让人们有困难能及时找到警察。

在交巡警服务平台的设置与调度的优化策略中，需要考虑诸多因素，如何选择路径、在哪那些路口进行封锁、如何减少出警时间等。根据本题给出的交通网络图，不同地区、不同位置、不同路径、城市出入口等在巡警服务平台和警力调度方案均不同。

### 1.2 问题提出

请根据不同问题要求，结合附件中给出的相关数据，建立数学模型，研究以下问题：

1. 通过中心城区 A 的交通网络和交巡警服务平台的示意图和附件中相关的数据信息，试给出以下 3 个问题的最优方案。

(1) 试给出各交巡警服务平台分配具体管辖范围，使其在所管辖的范围内出现突发事件时，尽量能够在 3 分钟内有交巡警（警车的时速为 60km/h）到达事发地点。

(2) 对于发生突发情况，则需要对出入 A 区的 13 个路口进行封锁，当一个平台的警力只能封锁一个路口时，如何在最快的时间内对所有出入 A 区的路口进行封锁，给出具体的 A 区交巡警服务平台警力合理的调度方案及最快出警时间。

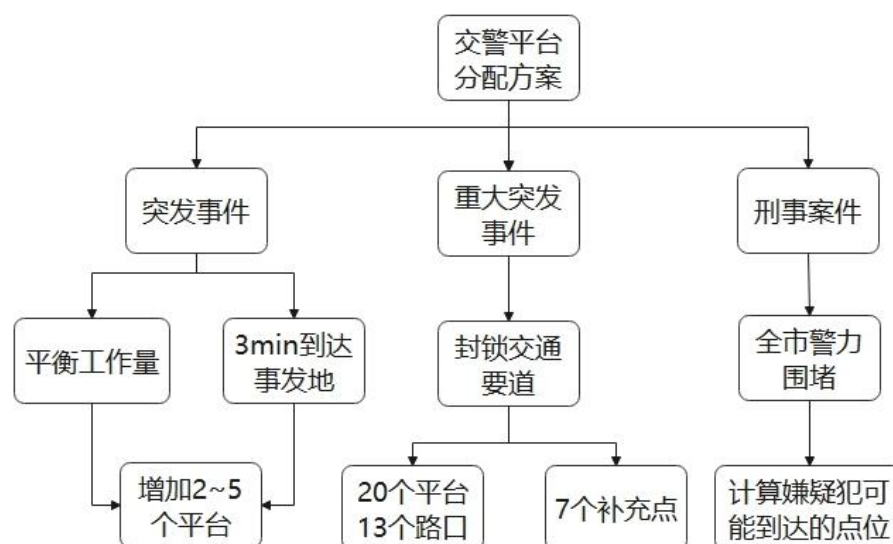
(3) 根据现有交巡警服务平台的工作量不均衡和有些地方出警时间过长的实际情况，准备在该区内再增加 2 至 5 个平台，从而平衡工作量和减少出警时间，请确定需要增加平台的具体个数和位置。

2. 根据在全市六区交通网络与平台设置和每个区的交巡警服务平台点位的示意图，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和围堵具体使用的时间。

## 二、问题分析

本题主要是对交巡警服务平台的设置与调度进行设计进行研究。针对问题 1 第一问，需给出各交巡警服务平台分配具体管辖范围，针对问题 1 的第二问，

对于重大突发事件给出具体的 A 区交巡警服务平台警力合理的调度方案及最快出警时间；针对问题 1 的第三问，为了平衡工作量以及减少出警时间，给出增加平台的具体个数和位置。针对问题 2，当发生刑事时，给出调度全市交巡警服务平台警力资源的最佳围堵方案，结合以上问题，分别做如下分析：



图一：交警平台流程图

## 2.1 问题 1 分析

在第一问需要研究三个小问，需要分别设计在发生突发情况和重大突发情况的条件下 巡警平台的设置与调度，设计最优求解算法，针对三个小问做如下具体分析：

①在城区 A 的交通网络和交巡警服务平台已知的情况下，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 分钟内有巡警到达事发地，其中还要考虑每个服务平台的工作量，让每个平台既能以最短时间到达事发地点的同时又能保证工作量不会太大。运用最短路算法，找出各个平台能到达点的最短路径，从而确定该平台所管辖的范围。

②针对发生重大突发情况，需要调派全区的 20 个巡警平台快速封锁出入 A 区的 13 个路口。考虑到每个平台的警力只能封锁一个路口，我们建立了"0-1"变量，以建立平台与需要封锁的路口之间的联系。通过这个联系，确定是否派遣该平台的警力去封锁对应的路口，并确保每个平台只封锁一个路口。

同时，剩下 7 个巡警平台需要进行补位。当其他 13 个平台遇到突发情况时，这 7 个平台可以迅速填充并封锁相应的位置。因此，我们需要确定这 7 个巡警平台的补位点，以确保能够在最短时间内进行有效的补位。

③为了平衡巡警服务平台的工作量和出警时间过长的问题，我们决定增加平

台数量。我们目前有 20 个交警平台，并且已经了解了 A 区各个节点的发案次数。我们考虑新增 2 至 5 个交警平台，以平衡各个交警平台的工作量为目标，使得每个巡警服务点的工作量差最小化。通过这样的平衡措施，我们可以有效减轻某些平台的工作负担，提高整体的工作效率，并确保各个巡警服务点承担相对均衡的工作量和出警时间。

### 2.2 问题 2 分析

在该市的地点 P（第 32 个节点）发生了一起重大刑事案件，在案发后的 3 分钟内接到报警，犯罪嫌疑人已经驾车逃跑。我们首先需要设置嫌疑人的逃逸速度，以确定嫌疑人在案发后的 3 分钟内可能出现在哪些节点。然后，我们对这些节点进行围堵，确定围堵的点位。

一旦确定了围堵点位，我们就可以派遣警力迅速到达这些点位并进行封堵。这里使用与第一问类似的派遣模型，计算出巡警平台到达这些点位的最短时间。这样能够确保警力在最短时间内到达并实施封堵。

### 三、符合说明

符号	描述说明
$F_{ij}$	第 $i$ 个交警平台是否选取第 $j$ 个节点
$L_{ij}$	第 $i$ 个巡警平台到第 $j$ 个路口最短路径
$x_{ij}$	第 $i$ 交警平台是否封锁第 $j$ 个节点
$A_j$	A 区第 $j$ 个节点的发案率
$f_i$	第 $i$ 个路口是否增加巡警服务平台
$y_j$	嫌疑人在第 $j$ 个节点活动
$a_i$	第 $i$ 个增设点的均值
$c_j$	第 $j$ 个节点是否有巡警平台封锁
$t_i$	全市交警平台到达第 $i$ 节点最短用时

### 四、模型假设

1. 假设交巡警到事发地的过程中不存在堵车等情况。
2. 假设交巡警和嫌疑犯驾驶车辆时车速不变。

3. 假设巡警平台所处点位由自己管辖。
4. 假设嫌疑犯的逃跑速度为 $1500m/min$

## 五、模型的建立与求解

针对本题为了应对各种不同情况,我们需要制定不同突发情况下最佳的巡警服务平台的警力调度方案。为了达到最优结果,我们将建立一个优化模型,并根据各种突发情况确定模型的约束条件。通过设计相应算法,我们将得出最佳的警力调度方案。

### 5.1问题1: 交警平台管辖范围安排

在这个问题中,我们需要解决三个具体问题。首先,我们需要确保在发生突发情况时能够在三分钟内到达现场,同时要考虑每个交警服务平台的工作负荷。我们希望能够实现每个平台快速响应,并且确保各个巡警服务平台的工作负荷相对均衡。其次,当面临重大突发事件时,我们需要制定一个调度方案,涉及到20个平台如何封锁13个要道。

这个方案旨在有效协调不同平台的行动,以确保相关要道的封锁工作能够高效有序地进行。最后,为了针对一些巡警平台工作负荷不均衡、出警时间过长的问題,我们计划增设一些新的平台,并根据实际情况确定这些平台的位置。这样的调整能够帮助平衡各个巡警服务点的工作负荷,提高工作效率。

#### 5.1.1 平台与各节点的最短路径

在突发情况发生时,我们需要迅速调配整个区域的20个巡警平台前往13个交通要道,以实施封锁措施。为了能够高效地进行计算和决策,我们需要了解20个巡警平台到达这13个交通要道的最短路径和时间。

为此,在区域数据预处理阶段,我们采用了Floyd算法来计算各个节点之间的最短路径,并得到了一个 $92 \times 92$ 的最短距离矩阵。这个矩阵将帮助我们进行后续的计算与求解工作。计算所得到的表格如下:

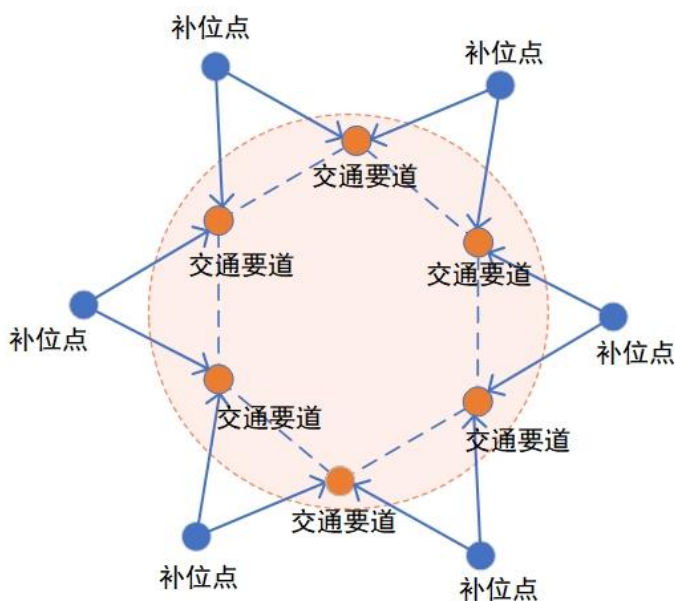
表一: 部分节点间的最短路径

节点	1	2	3	4	5	6	7	8	9	10
1	0	1899	3884	4535	9374	9538	11500	9023	9225	14650
2	1899	0	2112	5685	7834	9842	9728	7250	7453	12877
3	3884	2112	0	4043	5722	7730	7616	5139	5342	10766
4	4535	5685	4043	0	4920	5002	7657	8327	8987	14411
5	9374	7834	5722	4920	0	2943	2737	3536	4695	10042
6	9538	9842	7730	5002	2943	0	2767	3566	4726	10072
7	11500	9728	7616	7657	2737	2767	0	2478	2909	7328
8	9023	7250	5139	8327	3536	3566	2478	0	1160	6506

9	9225	7453	5342	8987	4695	4726	2909	1160	0	5424
10	14650	12877	10766	14411	10042	10072	7328	6506	5424	0

### 5.1.2.1 最优补位点选取

第二个问题中，当发生重大突发事件时，我们需要派遣全区的20个巡警平台来封锁13个交通要道。当全区已经将13个交警平台的警力调派到这些交通要道进行封锁时，为了防止警力在途中遇到其他突发事件需要支援，全区剩余的7个交警平台应前往指定的补位点，随时准备提供支援。这样可以确保警力能够迅速到达并有效地应对各种突发情况，具体如下：



图二：混合型拓扑结构示意图

### 5.1.2.1 确定目标

我们的目标是进行最优补位点的选取，通过最优选取点的位置应建立在两个交通要道之间，通过让选取的补位点到交通要道的均值方差差值应尽量小建立目标，具体目标如下：

$$\text{Min } z = \sum_{i=1}^7 \sum_{j=1}^{13} \frac{(F_{ij}L_{ij} - a_i)^2}{13}$$

### 5.1.2.2 模型I的约束条件

为了求出剩余7个交警平台应到达指定补位点，需要对模型做以下约束：

#### (1) 对巡警平台封锁交通要道的约束

在全区已经派出了13个交警平台的警力对要道进行封锁的情况下，剩下的7个交警平台需要进行补位工作。为了提高效率并平衡工作量，我们需要遵守一个约束条件：一个交警平台在进行补位时最多负责封锁两个交通要道。这意味着某个

节点可能会有多个交警平台同时进行封锁操作，以确保封锁工作的高效进行。具体的约束条件如下：

$$\sum_{j=1}^{13} F_{ij} \leq 2f_i \quad i = 1..92$$

### (2) 对出警时间的约束

在选择补位点时，我们需要确保警力能够在规定的时间内到达补位点。为了实现这一目标，我们需要确保各个补位交警平台的警力能够在小于3分钟的时间内到达指定的交通要道。考虑到警车的速度为每分钟1000米，我们需要满足以下具体约束条件：

$$F_{ij}L_{ij} \leq 3000 \quad i = 1..7, j = 1..13$$

### (3) 对剩余警力的约束

当全区的20个巡警服务平台中有13个平台被派遣到交通要道进行快速封锁封锁时，还剩下的7个巡警平台可以进行补位。由此可得，我们最多只能选择7个交通节点作为补位点，则可以得知第*i*个部位点*f<sub>i</sub>*所有点位求和等于7。具体约束如下所示：

$$\sum_{i=1}^{92} f_i = 7$$

### (4) 对补位点的约束

当剩余的7个交警平台对13个交通要道随时准备补位时，需要保证13个交通要道都能被补位到，则可以得到第*i*个巡警平台是否在第*i*个点位为补位点*F<sub>ij</sub>*之和等于1，具体约束如下：

$$\sum_{i=1}^{92} F_{ij} = 1 \quad j = 1..13$$

### (5) 距离均值的计算

*a<sub>i</sub>*为第*i*个增设点到达13个交通要道的距离的均值，其中第*i*个交警平台是否选取第*j*个节点作为增设点*F<sub>ij</sub>*，第*i*个交警平台到第*j*个节点的最短路径*L<sub>ij</sub>*，求出第*i*个增设点到达13个交通要道的距离的均值，具体计算式子如下：

$$a_i = \sum_{j=1}^{13} \frac{F_{ij}L_{ij}}{13} \quad i = 1..7$$

### (6) 对“0-1”变量的约束

考虑到交警平台是否选择该节点作为增设点位，则需要对第*i*个交警平台是否选取第*j*个节点作为增设点 $F_{ij}$ 进行“0-1”变量约束，具体约束如下：

$$F_{ij} \in \{0,1\} \quad i=1..92, j=1..13$$

由于我们需要确定该节点是否作为补位增设点，则需要对第*i*个节点是否作为补位增设点 $f_i$ 进行“0-1”变量约束，具体约束如下：

$$f_i \in \{0,1\} \quad i=1..92$$

#### 5.1.2.3 模型I建立和求解

综合上述目标模型和约束条件分析，为了求得剩下7个巡警服务平台的补充点位，以补充点位到交通要道的均值方差最小为目标，其中第*i*个交警平台是否选取第*j*个节点作为增设点 $F_{ij}$ 为变量， $L_{ij}$ 为第*i*个交警平台到第*j*个节点的最短路径， $a_i$ 为第*i*节点均值，建立最优化模型，具体模型如下

$$\begin{aligned} \text{Min } z &= \sum_{i=1}^{92} \sum_{j=1}^{13} \frac{(F_{ij}L_{ij} - a_i)^2}{13} \\ \text{s.t. } &\begin{cases} \sum_{i=1}^{92} f_i = 7 \\ \sum_{j=1}^{13} F_{ij} \leq 2f_i & i=1..92 \\ \sum_{i=1}^{92} F_{ij} = 1 & j=1..13 \\ F_{ij}S_{ij} \leq 3000 & i=1..92, j=1..13 \\ a_i = \sum_{j=1}^{13} \frac{F_{ij}L_{ij}}{13} & i=1..92 \\ f_i \in \{0,1\} & i=1..92 \\ F_{ij} \in \{0,1\} & i=1..92, j=1..13 \end{cases} \end{aligned}$$

#### 5.1.2.4 模型I的具体结果

综上所述，结合上述建立的模型，利用 *Lingo* 求解，一共有20个巡警平台分配给13个交通要道，为了防止警力在途中遇到突发事件，全区剩余7个交警平台应到达指定补位点，随时支援。求解得7个补充点位的位置如下：

表二：7个巡警平台的补充点位及管辖节点

巡警点位	13	14	16	25	29	48	62
管辖道路节点	22 23	14 21	16 38	12 24	28 29	30 48	62



根据上表，我们选择了节点13、14、16、25、29、48、62作为候补巡警平台。当其他13个巡警平台在发生突发情况时无法前往封锁地点时，这7个候补平台将快速行动并进行封锁操作。

### 5.1.3 重大突发事件平台调度

在发生重大突发事件时，我们将调度全区的20个巡警平台，以最快速度封锁13个交通要道。除了这20个主要平台，我们还根据模型I的计算结果选择了7个备选平台作为候补。这意味着这7个备选平台也需要前往指定的位置，以防止出现意外情况。因此，在解决这个问题时，我们考虑了让所有20个平台都前往规定的位置。A区节点和巡警平台示意图如下所示：

#### 5.1.3.1 确定模型II目标

在面对重点突发事件时，我们20个交警平台应快速做出响应，且交警的出警速度为1000m/min，到达预案中的点位（13条交通要道点位和7条增设的补充点位）。故我们的目标是以最晚到达预设点位的速度最快，于是可以确立的目标如下：

$$\text{Min } T = f(t)_{\max}$$

其中， $f(t)$ 是一个关于交警平台到达预案中的点位的时间的函数，它表示随着封锁节点的不同，各个交警平台到达预案中的点位的时间， $f(t)_{\max}$ 则表示各个交警平台到达预案中的点位中最晚的时间。

$$f(t) = \frac{\sum_{i=1}^{20} L_{it} x_{it}}{1000}$$

#### 5.1.3.2 约束条件

为了能在重大突发情况时快速封锁对应点位，则需要对模型做以下约束：

##### (1) 对“0-1”变量的约束

考虑到节点需要巡警平台进行封锁，则需要对第*i*交警平台是否封锁第*j*个节点  $x_{ij}$  进行“0-1”变量约束，具体约束如下：

$$x_{ij} \in \{0,1\} \quad i = 1 \dots 20, j = 1 \dots 92$$

##### (2) 对管辖节点的约束

根据题目要求，在面对重点突发事件时，我们已经根据模型I得出了要去往的交通节点的节点以及需要补位的节点记作  $c_j$ ，当  $c_j$  为1时表示是第*j*个节点必须得有一个巡警平台进行封锁，故我们约束节点能够派遣的平台数，具体约束如

下：

$$\sum_{i=1}^{20} x_{ij} = c_j \quad j = 1 \dots 92$$

### (3) 对巡警平台管辖数的约束

根据题目要求，在面对重点突发事件时，由于警力资源有限，一个交警平台节点的警力最多只能封锁一个节点，因此第*i*个巡警平台最多只能前往一个节点进行封锁，故我们约束巡警节点能够前往的平台数，具体约束如下：

$$\sum_{j=1}^{92} x_{ij} \leq 1 \quad i = 1 \dots 20$$

#### 5.1.3.3 模型II的建立和求解

我们面对突发事件时考虑安排13个交警平台完成对13个交通要道进行封锁，随后安排7个交警平台去7个补位点对13个交通要道进行补充支援，依据我们的目标分析以及约束条件的分析，建立以下模型：

$$\begin{aligned} & \text{Min } T = f(t)_{\max} \\ & s.t. \left\{ \begin{aligned} & f(t) = \frac{\sum_{i=1}^{20} L_{it} x_{it}}{1000} \\ & \sum_{i=1}^{20} x_{ij} = c_j \quad j = 1 \dots 92 \\ & \sum_{j=1}^{92} x_{ij} \leq 1 \quad i = 1 \dots 20 \\ & x_{ij} \in \{0,1\} \quad i = 1 \dots 20, j = 1 \dots 92 \end{aligned} \right. \end{aligned}$$

#### 5.1.3.4 模型II具体结果

综上所述，结合上述建立的模型，利用 *Lingo* 求解出当发生重大突发情况时，出动 20 个巡警平台快速封锁 13 个路口和 7 个候补点位的调度方案，最快的封锁时间为 8.015 分钟，调度方案如下：

表三：调度方案表

位置	时间	具体路径	位置	时间	具体路径
(2, 38)	3.982	2 → 40 → 39 → 38	(11, 24)	3.805	11 → 25 → 24
(3, 16)	6.026	3 → 45 → 35 → 36 → 16	(12, 23)	6.477	12 → 25 → 24 → 23
(6, 48)	2.506	6 → 47 → 48	(13, 22)	0.905	13 → 22
(7, 29)	8.015	7 → 30 → 237 → 238 → 29	(14, 21)	3.265	14 → 21
(9, 30)	3.492	9 → 34 → 33 → 32 → 30	(15, 28)	4.751	15 → 28

(10, 12)	7.587	10 → 26 → 27 → 12	(16, 14)	6.741	16 → 14
(19, 62)	5.033	19 → 77 → 76 → 64 → 63 → 4 → 62			

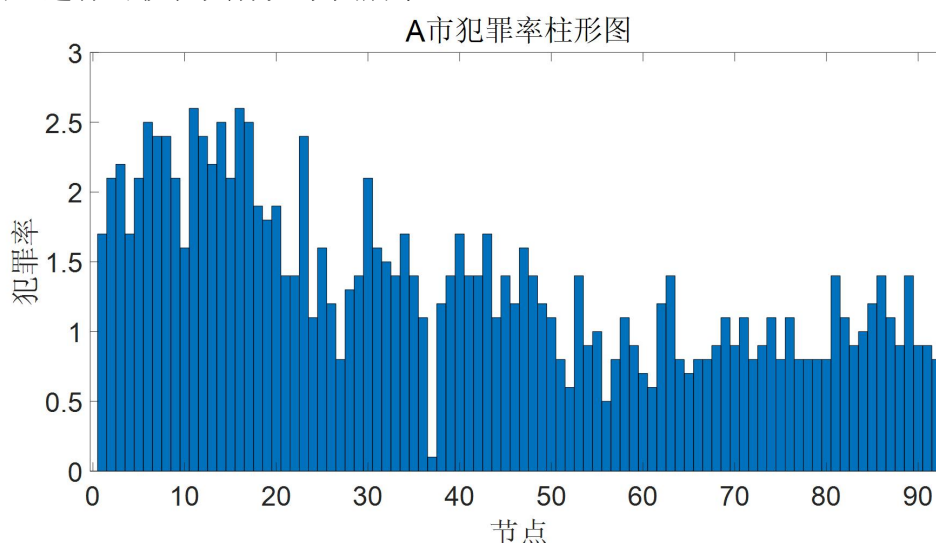
根据表格中的数据，我们可以看出从第 7 个巡警服务平台到第 29 个封锁口的时间最长。封锁是在最后一个点位封锁完成后才算封锁完毕，因此这个时间是指最快封锁所需的最短时间。根据数据计算得出最短封锁时间为 8.015457 分钟，这表明可以实现快速封锁。

在利用 20 个巡警平台封锁 13 个交通要道之后，剩下 7 个巡警平台。我们可以将它们调度到特定的点位，以备封锁中出现意外情况时能够快速补充人力。7 个巡警平台的调度点位如下：(1, 13)，表示第 1 个巡警平台到达第 14 个点位。完整的点位分布如下：(4, 29)，(5, 25)，(8, 13)，(17, 14)，(18, 62)，(20, 16)。

通过合理的调度和分配，我们能够充分利用巡警平台的资源，确保封锁工作的高效和可靠性，并在需要时能够迅速应对突发状况。

#### 5.1.4 最优平衡工作量

考虑到巡警服务平台存在工作量和出警时间过长的情况，通过增加平台来平衡工作量和出警时间，我们拥有交警平台 20 个，以及已知 A 区各个节点的发案次数，进行可视化具体如下图所示：



图三：A 区各个节点犯罪率柱形图

通过绘制区内各个节点的犯罪率折线图，我们可以清楚地看到前 20 个节点的犯罪率相对于其他节点偏高。这表明这些节点所需的工作量会比较大，犯罪率介于 1.6 至 2.5 次之间。相比之下，60 至 80 节点的犯罪率相对较低，仅为 0.8 至 1.3 次之间，因此可以推断这些区域的巡警平台负担相对较轻。

##### 5.1.4.1 确定模型 III 目标

由于只设置 20 个交警平台时，各个交警平台不能够及时的到达现场处理案

件并且工作量不均衡，在准备增加 2 到 5 个平台时，为了让各个交巡警能够在 3 分钟之内及时的到达现场并且以每个巡警服务点的最大的工作量差值最小为目标，可以建立的目标如下：

$$\text{Min } z = f(s)_{\max}$$

其中， $f(s)$  是一个关于交警平台服务点工作量差值最小的函数，随着  $s$  的不同，表示着各个交警平台服务点工作量的差值， $f(s)_{\max}$  表示着各个交警平台服务点工作量的最大差值，其中， $f(s)$  的函数关系式解释是 20 个交警平台各自管理范围为  $x_{ij}$ ，根据题目给出的 A 区第  $j$  个节点的发案次数  $A_j$ ，我们考虑新增 2 到 5 个交警平台平衡各交警平台的工作量，计算每个巡警服务点工作量差值

$$f(s) = \left| \sum_{j=1}^{92} A_j x_{sj} - \bar{A} \right|$$

其中， $x_{ij}$  为第  $i$  交警平台管理是否管理 A 区第  $j$  个节点， $A_j$  为第  $j$  个节点的发案次数， $\bar{A}$  为每个交警平台平均管理发案次数， $f_i$  为该节点是否为交警平台。 $\bar{A}$  计算公式如下：

$$\bar{A} = \sum_{j=1}^{92} A_j / \sum_{j=1}^{92} f_i$$

#### 5.1.4.2 约束条件

为了求得增设几个巡警服务平台来平衡每个巡警平台的工作量，需要做如下约束：

##### (1) “0-1” 变量的约束

考虑到是否在该节点增设巡警服务平台，则需要对第  $i$  个节点是否为巡警服务平台进行“0-1”变量约束，具体约束如下：

$$f_i \in \{0,1\} \quad i = 1 \dots 92$$

考虑到该节点需要巡警平台进行管辖，则需要对第  $i$  交警平台是否管辖第  $j$  个节点  $x_{ij}$  进行“0-1”变量约束，具体约束如下：

$$x_{ij} \in \{0,1\} \quad i = 1 \dots 20, j = 1 \dots 92$$

##### (2) 对巡警服务平台个数的约束

我们原有 20 个交警平台，现考虑增设 2 到 5 个交警平台，新增平台从 92 个交通节点中选取，从而我们可以得到巡警平台的总数不能低于 22 个，不能高于 25 个，具体约束如下：

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

因为有 20 个交警平台是已知的，所以我们对“0-1”变量  $f_i$  进行赋值，前 20 个的值都为 1，具体式子如下：

$$f_i = 1 \quad i = 1 \dots 20$$

### (3) 全区管控约束

根据题目要求，在面对重点突发事件时，第  $j$  个节点必须有一个巡警平台进行管辖，故我们约束每个节点能够派遣的巡警平台数，与“0-1”变量建立关系，具体约束如下：

$$\sum_{i=1}^{92} f_i x_{ij} = 1 \quad j = 1 \dots 92$$

### (4) 对出警速度约束

当出现突发事件时，交警应第一时间到达现场，在题目中规定交警的出警时间为 3 分钟，且警车的时速为 1000m/min，以及在数据预处理中我们可知 A 区第  $i$  个平台到第  $j$  个节点的最短路径  $L_{ij}$ ，通过与“0-1”变量建立关系，所以出警速度具体约束如下：

$$L_{ij} x_{ij} < 3000 f_i \quad i = 1 \dots 92, j = 1 \dots 92$$

#### 5.1.4.3 模型III的建立与求解

我们以平衡各交警平台的工作量为目标，考虑增设交警平台、交警平台全区管控、交警平台出进速度，建立工作量最优各交警平台管辖范围模型，建立的最优化模型如下：

$$\begin{aligned} & \text{Min} = f(s)_{\max} \\ & s.t. \begin{cases} f(s) = \left| \sum_{j=1}^{92} A_j x_{sj} - \bar{A} \right| \\ \bar{A} = \sum_{j=1}^{92} A_j / \sum_{j=1}^{92} f_i \\ 22 \leq \sum_{i=1}^{92} f_i \leq 25 \\ \sum_{i=1}^{92} f_i x_{ij} = 1 \quad j = 1 \dots 92 \end{cases} \end{aligned}$$

$$s.t. \begin{cases} f_i = 1 & i = 1 \dots 20 \\ L_{ij} x_{ij} < 3000 f_i & i = 1 \dots 92, j = 1 \dots 92 \\ x_{ij} \in \{0,1\} & i = 1 \dots 92, j = 1 \dots 92 \\ f_i \in \{0,1\} & i = 1 \dots 92 \end{cases}$$

#### 5.1.4.4 模型III具体结果

综上所述，结合上述建立的模型，利用 *Lingo* 求解出增加巡警服务平台的个数从而平衡各个巡警服务平台的工作量，与此同时增加的服务平台数与原来的20个服务平台一样都需要能在3分钟内到达事发地。我们计算得到需要增加5个巡警服务平台来平衡每个巡警平台工作量和出警时间，下列表格是各个巡警服务平台的管辖节点。

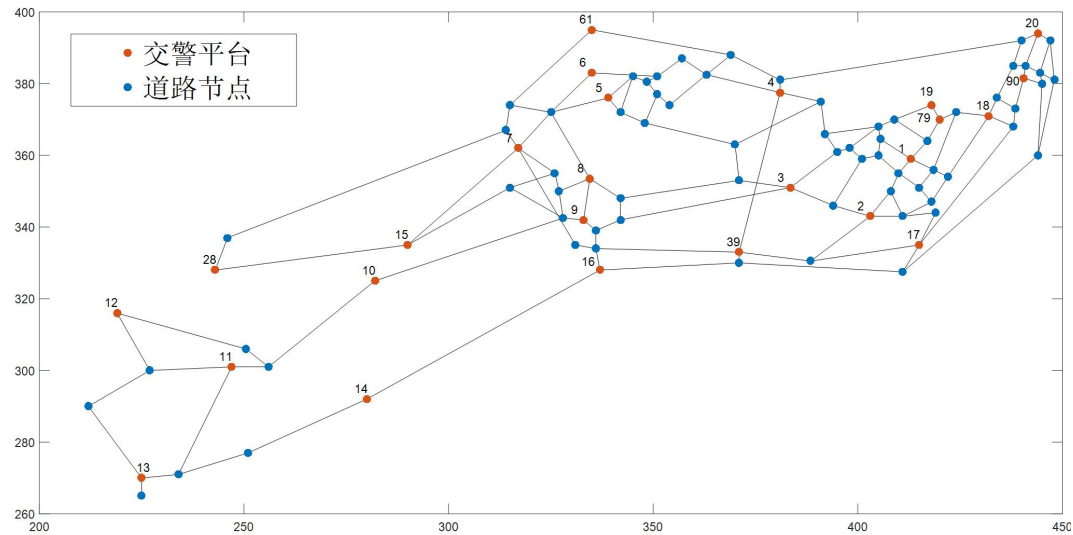
表四：各巡警点位管辖道路的节点表

巡警点位	管辖道路节点点位	节点总数	案发率（次）
1	1 43 71 76 78 79	6	7.2
2	2 44 68 73	4	5
3	3 54 55 65 67	5	5.6
4	4 57 64 62 66	5	5.3
5	5 47 51 53 58	5	7
6	6 50 52 56 89	5	5.6
7	7 31 32	3	5.5
8	8 34 37	3	5.4
9	9 33 45	3	4.9
10	10	1	1.6
11	11 25 26 27	4	6.2
12	12	1	2.4
13	13 21 22 23 24	5	8.5
14	14	1	2.5
15	15	1	2.1
16	16 35 36	3	5.1
17	17 41 42	3	5.3
18	18 72 74 84 87 90	6	6.8
19	19 69 75 77 80	5	5.3
20	20 83 86 91	4	5.1

29	28 29	2	2.7
40	38 39 40 70	4	5.2
48	30 48 49 61	4	5.3
63	60 63	2	2.1
89	81 82 85 88 89 92	6	6.8

我们增设了5个巡警服务平台，它们分别位于第29、40、48、63和89个节点上。通过上述表格可以看出，每个巡警平台所管理的节点数量都在3到6个之间，相比之前，巡警平台的工作负荷有所减轻。需要注意的是，个别平台所管理的节点数量较少，这是因为这些平台无法在3分钟内到达案发地点，因此它们只负责管理自身所在的节点。

这样的调整能够有效平衡各个巡警服务平台的工作负荷，并确保在限定时间内能够到达事发地点。通过增设新平台并合理分配节点，我们可以更好地应对突发事件，提高交巡警服务的效率。25个巡警平台点位示意图如下：

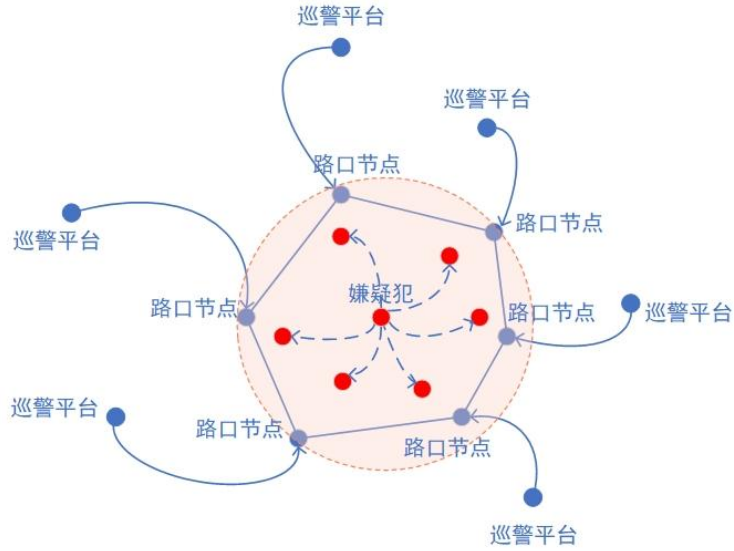


图四：A区节点和25个巡警平台示意图

### 5.2 问题2：围堵嫌疑人的最佳方案

在这个问题中，涉及到三类不同类型的突发事件需要由交警平台处理，包括一般突发事件、重大突发事件和重大刑事案件。在这个问题中，我们考虑的是当市区的某个地点（第32个节点）发生严重的刑事案件，警方在案发三分钟后接到报警，犯罪嫌疑人已经驾车逃跑。

为了迅速围堵嫌疑人，我们需要设计一个最佳的交巡警服务平台警力调度方案。首先，我们希望尽可能地限制嫌疑人的活动范围，然后调动全市的交警平台占领围堵嫌疑人可能逃往的节点。具体的调度如下图所示：



图五：围堵嫌疑犯示意图

### 5.2.1 节点扩散分析

现已知重大刑事案件发生节点，我们使用邻接矩阵描述该点可到达的节点，当两节点可达时设为1，否则设为0。依据全市交通路口的节点边数据，我们可以得到点与点间的邻接矩阵  $S_{m \times n}$ ，具体如下：

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mn} \end{bmatrix} \quad s_{ij} \in \{0,1\}$$

其中，邻接矩阵  $S$  为582行582列的矩阵。

当我们想知道一个节点往后走  $x$  步后可到达的节点，则我们可对邻接矩阵求幂计算到，设  $g(x)$  为各节点走  $x$  步后可到达的各节点情况的函数，具体如下：

$$g(x) = S^x$$

其中， $x$  的定义域为正整数， $g(x)$  函数表示各节点往后走  $x$  步后可到达的节点情况，我们记矩阵  $g(a) = S^a$  中的元素为  $s_{ij}^a$ ， $g(a) = S^a$  具体如下：

$$g(a) = \begin{bmatrix} s_{11}^a & \cdots & s_{1n}^a \\ \vdots & \ddots & \vdots \\ s_{m1}^a & \cdots & s_{mn}^a \end{bmatrix}$$

### 5.2.2 封锁节点分析

我们考虑交警平台在一定时间内封锁节点。由于在现实中一个交警平台最多封锁一个节点，我们出于简化模型的角度考虑，使交警平台封锁的节点不加限制，且设犯罪嫌疑人的逃脱速度为2000m/min，全市交警平台到达第  $i$  节点最短用时为  $t_i$ 。



对于 $i$ 节点当交警平台到达该节点的最短用时 $t_i$ 比犯罪嫌疑人到达该节点的最短用时小，则表示该点可被交警平台占领，具体如下：

$$\begin{cases} t_j = \left( \frac{A_{ij}}{1000} \right)_{\min} & i = 1 \dots 80 \\ s_{kj}^a = 0 & 2000(t_j + 3) \leq L_{kj}, j = 1 \dots 582, k = 32 \end{cases}$$

其中， $t_j$ 为全市80个交警平台到达第 $j$ 节点最短用时， $A_{ij}$ 为全市第 $i$ 个交警平台到第 $j$ 个节点的最短路径， $s_{kj}^a$ 表示第 $a$ 步邻接矩阵从 $k$ 节点到第 $j$ 节点的元素， $L_{kj}$ 表示案发节点 $k$ 距第 $j$ 节点的最短路径。

### 5.2.3 活动节点分析

在上述对交警平台封锁节点分析中，我们对节点扩散矩阵 $S^x$ 中的32行元素进行赋值为0处理，那么若 $S^x$ 中第32行第 $j$ 列存在不为0的元素，则表示嫌疑人从案发节点到第 $j$ 节点可达。设犯罪嫌疑人可活动的节点为 $y_i$ ，经我们封锁后嫌疑人可活动节点数应不再增长，具体如下：

$$y_j = \begin{cases} 1 & \exists s_{ij}^a > 0 \\ 0 & \forall s_{ij}^a = 0 \end{cases} \quad i = 32, a \in N^*$$

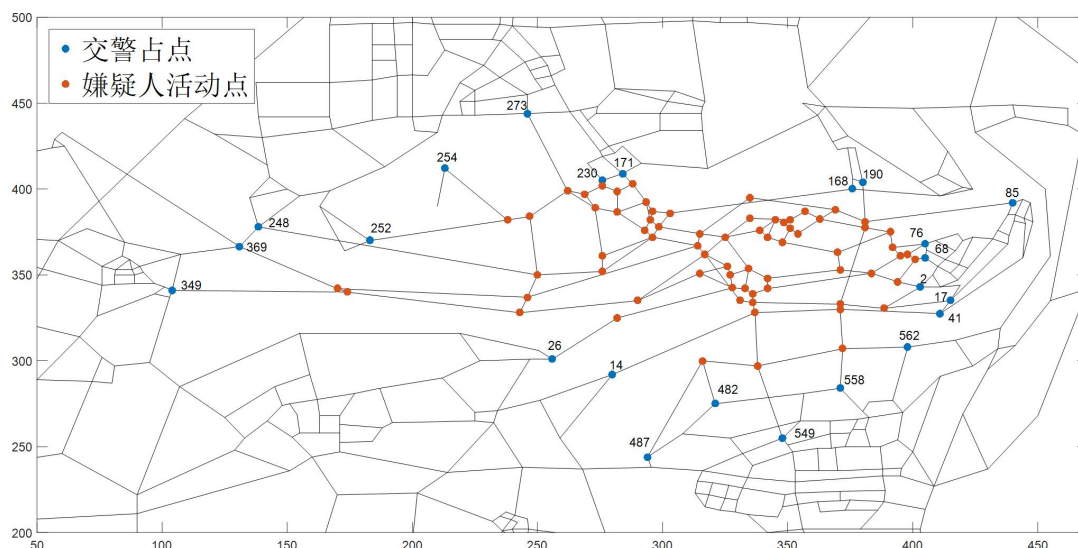
其中， $y_j$ 表示嫌疑人可活动节点。

### 5.2.4 模型求解

根据以上模型，我们编写了Matlab程序，通过设置嫌疑人的逃逸速度，并计算出嫌疑人可能逃跑到的点位。得到了嫌疑人的活动点位后，我们可以进行围堵操作。然后将嫌疑人的点位以及对应可进行围堵的点位在全市路线图上进行可视化展示。具体的结果如下所示：

表六：交警堵截点位表

交警占位点	85 41 2 190 26 14 17 76 14 17
	76 68 252 254 273 230 171 168
	369 248 349 558 549 562 487 482



图六：嫌疑人活动点和巡警封堵点

嫌疑人的密集活动点位为3~10、28~40、44~67、231~247。结合嫌疑人可能出现的位置表和封堵点示意图可以清晰看出嫌疑犯的活动轨迹和对嫌疑犯进行堵截的点位，我们将嫌疑人可能逃跑出的点位都成功进行围堵，此时我们确定了哪些点位可以进行堵截后，就需要派遣平台警力进行围堵，下文就是派遣巡警去围堵嫌疑人的具体方案。

### 5.2.5 最优全市交警平台占点调度

通过以上的数学表达式，我们可以计算出交警的占位点和嫌疑人的占位点。接下来，我们需要确定巡警服务平台到这些占位点的最优方案，也就是平台到占位点位置的最短时间，以实现对嫌疑犯的快速拦截。这个问题与第一问类似，我们需要确定最快的封锁时间。

因此，在求解这个问题时，我们可以沿用第一问的模型，只需要修改一些约束条件，就能找到最优的占位点调度方案。通过这样的优化求解，我们能够确定出最佳的巡警平台占位点调度方案。

#### 5.2.5.1 确定目标

我们从维护社会安全的角度出发，对于重大刑事案件，我们的交警平台应以最快的速度完成对占点的封锁，将嫌疑犯进行控制，故我们沿用第一问的平台最优响应目标，具体如下：

$$\text{Min } T = f(t)_{\max}$$

其中， $f(t)$ 是一个关于全市交警平台到达预案中的点位的时间的函数，它表示随着封锁节点的不同，各个交警平台到达预案中的点位的时间， $f(t)_{\max}$ 则表示各个交警平台到达预案中的点位中最晚的时间。

$$f(t) = \frac{\sum_{i=1}^{80} A_{it} x_{it}}{1000}$$

#### 5.2.5.2 修改模型II的约束条件

为了使得封锁的时间最快，需要对目标做如下约束：

##### (1) “0-1”变量的约束

考虑到该节点需要巡警平台进行管辖，则需要对第*i*交警平台是否管辖第*j*个节点  $x_{ij}$  进行“0-1”变量约束，具体约束如下：

$$x_{ij} \in \{0,1\} \quad i=1\ldots 20, j=1\ldots 92$$

##### (2) 全市交警平台占点约束

我们求得警察点位的位置  $g_j$ ，所以我们需要警察所求得的点位都要进行占领，则可以得知第*i*个交警平台封锁路口  $x_{ij}$  之和为1。

$$\sum_{i=1}^{80} x_{ij} = g_j \quad j=1\ldots 582$$

且题目中明确要求每个巡警服务平台只能封锁一个点位，则可得第*j*个节点必须有一个巡警平台进行封锁，第*i*巡警平台封锁节点  $x_{ij}$  之和等于1，故我们约束节点所能够派遣的巡警平台数，具体约束如下：

$$\sum_{j=1}^{582} x_{ij} \leq 1 \quad i=1\ldots 80$$

##### (3) 全市交警平台占点时间约束

我们交警平台去封锁占点的时间应比嫌疑人到达该占点的时间短，这样才能实现封锁，且交警平台的速度为1000m/min，嫌疑人的速度设为1500m/min，具体约束如下：

$$1.5(A_{ij} + 3000)x_{ij} < L_{kj} \quad i=1\ldots 80, j=1\ldots 582, k=32$$

其中， $A_{ij}$  为全市第*i*个交警平台到第*j*个节点的最短路径， $x_{ij}$  为第*i*交警平台是否封锁第*j*节点， $L_{kj}$  表示案发节点*k*到第*j*节点的最短路径。

#### 5.2.5.3 模型IIII的建立和求解

依据对节点扩散分析使用 Matlab 软件编程计算出的巡警平台封锁点位，以及讨论了全市平台占点调度的时间、占点约束，建立以整体最优时间目标的优化模型。具体如下：

$$\begin{aligned}
& \text{Min } T = f(t)_{\max} \\
& \left\{ \begin{aligned}
& f(t) = \frac{\sum_{i=1}^{80} A_{it} x_{it}}{1000} \\
& \sum_{i=1}^{80} x_{ij} = g_j \quad j = 1 \dots 582 \\
& \sum_{j=1}^{582} x_{ij} \leq 1 \quad i = 1 \dots 80 \\
& 1.5(A_{ij} + 3000)x_{ij} < L_{kj} \quad i = 1 \dots 80, j = 1 \dots 582, k = 32 \\
& x_{ij} \in \{0,1\} \quad i = 1 \dots 80, j = 1 \dots 582
\end{aligned} \right.
\end{aligned}$$

#### 5.2.5.4 设计最优占点调度求解算法的具体描述

根据以上建立的最优化模型IIII,设计相对应的求解算法,具体算法步骤如下:

**Step1:** 数据初始化,我们把全市节点的邻接矩阵 $S$ 、全市 582 节点到 582 节点的最短路径 $L_{ij}$ 、全市交巡警平台 $f_i$ 数据导入 **matlab**。设置节点扩散最大步数 $a = 20$ 、交警接到报警的时间 $t_1$ 、创建两 0-1 变量 $x_i$ 、 $y_i$ 记录交警平台占领点和嫌疑人活动点,初始化嫌疑人案发点 $y_{32} = 1$ 。创建一个记录嫌疑人活动点数的变量 $n$ ,该值初始值为 $n = 0$ 。创建 582 大小的单位矩阵 $g$ 用于计算节点扩散。

**Step2:** 编写一个从 1 到  $a$  循环,依据节点扩散函数编写 $g = g(x)$ 用于计算当前节点下一步可扩散的节点。遍历案发点的扩散节点 $j$ ,判断交警最小到该节点 $j$ 的时间是否小于案发点到达该点的时间,其中交警接时间 $t_1 = 3$ ,交警速度为 1000m/min,嫌疑人速度为 1500m/min,为给交警合理安排时间,故设置嫌疑人的速度为 2000m/min。若该扩散节点 $j$ 交警到达时间比嫌疑人到达时间短,且该点没被交警平台占领 $x_j = 0$ ,不是嫌疑人的活动点 $y_j = 0$ ,则该点被交警平台占领即设 $x_j = 1$ ,否则该点为嫌疑人的活动点即 $y_j = 1$ 。记录嫌疑人活动点数 $n$ 。

**Step3:** 判断程序结束的标志是案发节点进行扩散后,嫌疑人的活动点不再增加,即对 $y_i$ 求和与 $n$ 相等。

**Step4:** 额外的,当我们安排全市交警平台去占领 $x_i$ 点时,若无解,我们可以设置一些占领点位为不可占领,即在程序初始化阶段额外设置 $y_j = 1$ 。

#### 5.2.5.5 模型IIII的具体结果

综上所述,根据5.2.5.4的具体算法求解步骤利用 **Matlab** 编程实现,具体程序见附录,结合第一问中计算的最短路算法,计算出全市80个巡警服务平台到封锁点的最短路径,通过这些最短路径,我们可以得到各个巡警平台到达堵截点位的最快速度。经过计算,共有24个点位被堵截,这24个点位能够实现对嫌疑犯的最

快封锁。具体的堵截点位如下表所示：

表七：巡警占点位置表

占点位置	巡警时间	罪犯时间	占点位置	巡警时间	罪犯时间
(1, 85)	7.82	9.75	(17, 17)	3	7.10
(2, 41)	6.44	7.01	(18, 76)	6.13	6.16
(3, 2)	5.11	5.73	(19, 68)	5.23	6.11
(4, 190)	5.65	7.63	(167, 252)	6.81	11.14
(11, 26)	3.90	6.48	(169, 254)	5.22	9.99
(14, 14)	3	6.70	(170, 273)	4.70	8.54
(171, 230)	5.06	5.21	(475, 558)	5.05	7.41
(172, 171)	7.98	8.08	(476, 549)	6.03	7.15
(175, 168)	7.88	8.32	(175, 168)	4.94	7.61
(320, 369)	8.56	13.96	(480, 562)	7.57	9.76
(321, 248)	8.41	13.68	(481, 487)	7.56	7.45
(322, 349)	5.06	15.26	(482, 482)	5.05	7.41

根据上表，例如（1，85）表示节点1的巡警平台到达节点85并进行堵截所需的时间为7.82分钟，而嫌疑犯到达该点位的时间为9.75。在这些堵截点位中，最快的封锁时间为3分钟，包括点（14，14），（171，230），（481，487）。而最晚的封锁时间为8.56分钟。由于封锁操作必须在最后一个点位完成后才能生效，所以嫌疑犯被完全堵截使其无法逃脱的最短时间为8.56分钟。

## 六、模型评价与推广

### 6.1 模型优点

- 1) 本模型考虑题目已知数据利用数学公式建立模型。
- 2) 结合最短路径算法。
- 3) 采用混合型拓扑逻辑结构，对 13 个交通要道进行了警力补充。
- 4) 模型四计算交警占位时间复杂度为  $O(n^2)$ 。
- 5) 借鉴了马科维兹的均值-方差模型，综合考虑补充点位对交通要道的路径长度以及差值。

### 6.2 模型缺点

- 1) 在计算全市的最短路径时，出于时间复杂度的考虑，采用的是 Dijkstra 算法，并不是全局最优。

### 6.3 模型推广

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。该模式适用于合理地设置交巡警服务平台。

## 七、参考文献

[1] 最短路算法学习总结（一）

[https://blog.csdn.net/Reborn\\_Lee/article/details/82468058](https://blog.csdn.net/Reborn_Lee/article/details/82468058)

[2] *Floyd* 算法

<https://baike.baidu.com/item/%E7%A6%BB%E6%95%A3%E5%8C%96/10501557>

[3] 均值-方差模型

[https://blog.csdn.net/zk168\\_net/article/details/103628359](https://blog.csdn.net/zk168_net/article/details/103628359)

## 附录

### 一、支撑材料文件列表

1. problem\_tu3.m

2. problem3.m

3. problem4.m

### 二、支撑材料文件列表

1. 模型 *I lingo* 代码

sets:

aa/1..92/:A,x;

bb/1..13/;;

cc(aa,bb):F,s;

endsets

data:

s=@ole('C:\Users\1\Desktop\92, öE½l ¨ µ½' i13, ö½»Í ¨ Òª µÀµÃ¾äÄë. xlsx', '¾äÄë1');

```

enddata
min=@sum(aa(i):@sum(bb(j):(F(i,j)*s(i,j)-A(i))^2/13));
@for(aa(i):@sum(bb(j):F(i,j))<=2*x(i));
@sum(aa(i):x(i))=7;
@for(bb(j):@sum(aa(i):F(i,j))=1);
@for(cc(i,j):F(i,j)*s(i,j)<=4000);
@for(aa(i):A(i)=@sum(bb(j):F(i,j)*s(i,j)/13));
@for(cc(i,j):@bin(F(i,j)));
@for(aa(i):@bin(x(i)));
@for(cc(i,j):G(i,j)=F(i,j)*s(i,j));
end

```

## 2. 模型 II lingo 代码

```

sets:
aa/1..20/:;
bb/1..13/:T;
cc(aa,bb):L,x,P;
endsets
data:
L=@ole('C:\Users\1\Desktop\20 到 13 交通要道距离.xlsx','距离1');
enddata
min=@max(cc(i,j):L(i,j)*x(i,j)/100);
@for(bb(j):@sum(aa(i):x(i,j))=1);
@for(cc(i,j):@bin(x(i,j)));
@for(aa(i):@sum(bb(j):x(i,j))<=1);
@for(cc(i,j):L(i,j)*x(i,j)=P(i,j));
end

```

## 3. 模型 III lingo 代码

```

sets:
aa/1..92/:f;
bb/1..92/:A;
cc(aa,bb):x,L;
endsets
data:

```





```

@for(aa(i):@for(bb(j):1.5*(A(i,j)+3000)*x(i,j)<L(j)));
@for(aa(i):@sum(bb(j):x(i,j))<1);
@for(aa(i):@for(bb(j):(A(i,j)+3000)*x(i,j)/1000=T(i,j)));
!@for(aa(i):@sum(bb(j):x(i,j))>=G(i));
!@for(aa(i):@bin(G(i)));
end

```

## 第二问 matlab 代码

```

%%
%交警平台节点封锁算法
%输入节点的邻接矩阵 S、各点的最短路径 L、交警平台的编号 f
%设置运行步数 steps、案发节点 casenode
%输出警察封锁节点 blockade、嫌疑人活动节点 movement
function [blockade,movement]=nodeBlockade(S,L,f)
%初始化数据
movement=zeros(582,1);
blockade=zeros(582,1);
g=eye(size(L,1));
steps=20;%运行步数
casenode=32;%案发节点
policev=1000;%交警速度
suspectsv=2000;%嫌疑人速度
ptime=0;%交警接到报警时间
numb=0;%活动节点数
%限制占点
%movement(i)=1;
movement(casenode)=1;
%围堵算法
for i=1:steps
    g=g*S;
    for j=1:size(g,2)
        if blockade(j)==1
            g(casenode,j)=0;
        end
    end
end

```

```

    if g(casenode, j)>0&&blockade(j)==0&&movement(j)==0
        for k=1:size(f)
            if suspectsv(L(f(k), j)+ptime)<policev*L(c, j)
                g(c, j)=0;
                blockade(j)=1;
            end
        end
        if blockade(j)==0
            movement(j)=1;
        end
    end
end
if sum(movement)==numb
    break;
else
    numb=sum(movement);
end
end
end
end

```