2014 高教社杯全国大学生数学建模竞赛

承　诺　书

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从 A/B/C/D 中选择一项填写）： AB

我们的参赛报名号为（如果赛区设置报名号的话）：

所属学校（请填写完整的全名）： 湘潭大学

参赛队员 (打印并签名) ： 1. 孟晨

2. 李继龙

3. 尚艳艳

指导教师或指导教师组负责人 (打印并签名)： 李成福

日期： 2014 年 9 月 13 日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2014 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）：

评阅人										
评分										
备注										

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

摘要

本文主要解决交巡警服务平台的设置与调度问题。

对问题一第一小问，我们首先利用图论中的思想，将每个点到每个点的直达距离制成一个 92 阶方阵，再利用 Floyd 算法算出 A 区的最短距离矩阵。然后将该矩阵中含有服务平台的列选出，形成一个不同服务平台到各个节点的距离压力矩阵，将其和犯案压力矩阵做数组乘法，得到不同服务平台对于各个节点的工作压力矩阵。

然后以每个服务平台的管辖数量为约束条件（见式（5）），所有服务平台的总工作压力最小为目标函数（见式（4）），建立 0-1 规划模型（见式（7）），并利用 Lingo 求解，确立最优方案。最优区域分配方案结果见表 1：A 区服务平台管辖区域分配方案表。

对问题一第二小问，我们对 A 区最短距离矩阵进行处理，得到每个服务平台到每个路口的最短距离，将其写成一个 20×13 的矩阵。然后采用了数学分析中极限逼近的思想，根据如果存在一一对应关系，该矩阵必然行满秩这一性质。设定一个 k 值，然后将矩阵中所有大于 k 值的元素化为 0，然后不断减小 k 值，从而使其不满秩，进而找到一个过滤临界值，即为封锁全区的最短时间。然后再从剩下的对应关系中，找出最佳方案。

我们最终得到的最优封锁方案是（路口标号—服务平台标号）：12—11，14—16，16—9,21—14,22—10,23—12,24—13，28—15,29—7,30—8,38—1,48—4,62—20。所需时间为 8.0155 分钟。

对问题一第三小问，我们共计划增加 4 个服务平台，其中两个服务平台为了缓解管辖区域压力，建立增加后的 0-1 规划模型（见式（8）），求出使得目标函数的工作压力最小所对应增加的节点标号。另外两个服务平台为了缓解重大事故封锁方案压力，计算增加两列之后的过滤临界值，然后找到增加后的矩阵的最小的过滤临界值，再找出该临界值对应增加的节点标号。

我们得到的最优增加方案为：23 号节点，29 号节点，71 号节点，88 号节点。

对问题二第一小问，我们从常规报案压力，重大事故封锁路口压力和人口密度服务平台分布这三个方面考虑其合理性。第一步，先利用 Floyd 算法求得全市的最短距离矩阵，然后将该矩阵中含有交巡警服务平台的列数保留，再将其中大于 2.5 分钟，3 分钟，4 分钟，5 分钟，6 分钟的元素化为 0，再分别计算其秩，即为该时间内无法到达事故地点的服务平台数。第二步，用和第一题第二小问一样的方法计算出封锁路口所用的理论最短时间。第三步，对人口密度和服务平台数做回归分析，并用 F 检验，检验其显著性。

我们最终的结论是：该设置方案基本合理。

对于问题二第二小问，先计算 3 分钟内犯罪嫌疑人所能到达的所有节点标号，然后根据交巡警服务平台的分布，建立动态规划模型进行求解。

关键词： Floyd 算法　矩阵秩逼近法　过滤临界值　0-1 规划模型　回归分析

一、问题重述

“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、服务群众四大职责。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。

二、模型假设

1. 假设所有交巡警服务平台的警察接到报案后均能立即出警。
2. 假设每位警察均选择最短路线到达案发现场。
3. 假设出警过程中不会出现堵车，红绿灯等意外情况。
4. 假设每个交巡警服务平台在常规报案下警员充足。
5. 假设出警过程中，警车均能保持以最高时速 60km/h 行驶。

6. 假设所有的线路均为双行线。
7. 假设犯罪嫌疑人的逃跑速度和警车速度相同，均为 60km/h。

三、符号假设

符号	说明	量纲
$D=\left(d_{ij}\right)_{92\times 92}$	A 区 92 个节点的直达距离矩阵，d_{ij} 表示第 i 个节点到第 j 个节点的直达距离	千米
$S=\left(s_{ij}\right)_{92\times 92}$	A 区 92 个节点的最短距离矩阵，s_{ij} 表示第 i 个节点到第 j 个节点的最短距离	千米
A_i	A 区第 i 个节点 $\left(i=1,2,\cdots,92\right)$	
N_i	全市第 i 个节点 $\left(i=1,2,\cdots,582\right)$	
$C=\left(c_{ij}\right)_{92\times 20}$	A 区的平台工作压力矩阵，c_{ij} 表示第 j 个平台对第 i 个节点的工作压力	
$X=\left(x_{ij}\right)_{92\times 20}$	A 区管辖范围分配方案矩阵，x_{ij} 表示第 j 个平台是否管辖第 i 个节点	
k	过滤临界值，即封锁方案所需最短时间	分钟

四、问题分析

4.1 问题一分析

4.1.1 直达矩阵 D 的确立

1） 根据附件 2 中的全市交通路口节点数据，将 A 区 92 个节点的坐标 $\left(x_i,y_i\right)$ 利用 MATLAB 中的 xlsread 命令录入。

2） 建立一个 92 阶的零矩阵 D，令其主对角线元素 d_{ii} 为 0，其余元素均为 ∞。

3） 再根据 A 区交通路口线路图，若第 i 个节点到第 j 个节点连通，则利用公式 $\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$ 计算出其直达距离，写在直达矩阵 D 中 d_{ij} 和 d_{ji} 的位置。

4） 根据单位换算公式 1km=1000000mm，和地图的比例尺为 1:100000，将上述的直达矩阵 D 中每个元素除以 10，得到直达矩阵 D，其每个元素单位为 km。

4.1.2 将管辖区域划分为节点

因为每一条道路上均有至少两个节点相连，而且此图为无向赋权图，我们可以将节点分配当做区域分配。

4.1.3 时间与距离的转换

已知警车的时速为 60km/h，那么按照 3 分钟内到达现场的标准，交巡警服务平台和案发地点最短距离不应超过 3km。

4.2 问题二分析

4.2.1 直达矩阵 D' 的确立

1） 根据附件 2 中的全市交通路口节点数据，将全市 582 个节点的坐标 (x_i, y_i) 利用 xlsread 命令，录入 MATLAB。

2） 建立一个 582 阶的零矩阵 D'，令其主对角线元素 d_{ii} 为 0，其余元素均为 ∞。

3） 再根据全市交通路口线路图，若第 i 个节点到第 j 个节点连通，则利用公式 $\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$ 计算出其直达距离，写在直达矩阵 D' 中 d'_{ij} 和 d'_{ji} 的位置。

4） 根据单位换算公式 1km=1000000mm，和地图的比例尺为 1:100000，将上述的 D' 中每个元素除以 10，得到直达矩阵 D'，其每个元素单位为 km。

4.2.2 设置方案合理性分析

根据问题一中的考虑因素，我们认为判断一个城市交巡警服务平台设置是否合理的指标有三个：

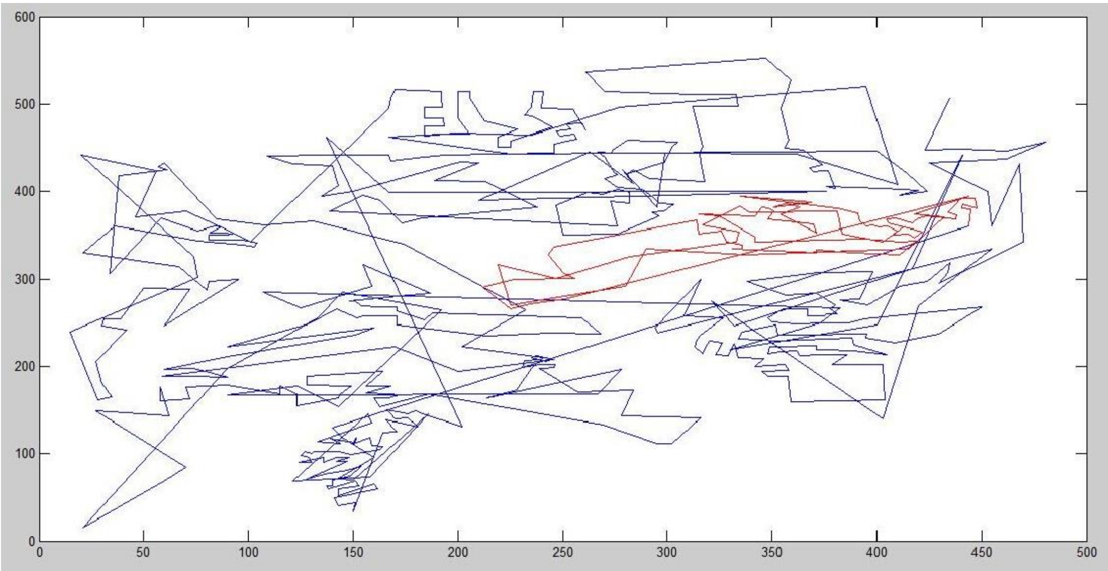
1） 常规报案各个服务平台的工作压力。

2） 出现重大事故，需要封锁全市路口所用最短时间。

3） 单位面积人口密度和交巡警服务平台数的正相关性

4.2.3 异常数据处理

附件 2 中的六个城区基本面积这一数据，和附件 1 中所给的全市六区交通网络与平台设置的示意图有明显差异。为了检验该示意图的合理性，我们又利用 MATLAB 将全市 582 个点的坐标绘制了一张示意图，该图如下：



经过对比，我们发现该图和附件 1 中所给的示意图差别不大，故我们认为附件 2 中所给的城区面积这一数据有误。

五、问题一模型的建立及求解

5.1 最短距离矩阵 S 的确立

我们利用 Floyd 算法，利用了直达矩阵 D 算出了 A 区 92 个站点间的最短距离矩阵 S。该矩阵第 i 行第 j 列的元素的意义是第 i 个节点到第 j 个节点的最短距离。

5.2 管辖区域分配

5.2.1 工作压力矩阵的确立

我们将最短距离矩阵中除了交巡警服务平台所在列数外其余的列数剔除，仅保留交巡警服务平台所在列数（1 列到 20 列）剩下一个 92×20 的矩阵$\tilde{S}$，其列意义为每个交巡警服务平台到某一节点所用最短距离。

将该矩阵中所有大于 3 的元素的值化为 0，然后利用 MATLAB 中的 sparse 命令，将其转化为稀疏矩阵，得到的三元组结点共有 1820 组，我们发现远远大于可以筛选的数量，于是我们引入新的参考因素——案发率矩阵B。

B是一个 92×20 的矩阵，其每一列的元素均为附件 1 中每个节点的案发率次数，b_{ij}为B的元素。

为了排除大于 3 分钟的元素，我们将$\tilde{S}$中所有大于 3 的元素改为原值的 10000 倍。

根据数组乘法，计算出每个节点的工作压力矩阵C，其中$c_{ij}=\tilde{s}_{ij} g b_{ij}$。则交巡警服务平台的工作压力矩阵为

$$C=\begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,20} \\ c_{1,1} & c_{1,1} & \cdots & c_{2,20} \\ \vdots & \vdots & \ddots & \vdots \\ c_{92,1} & c_{92,2} & \cdots & c_{92,20} \end{bmatrix}=\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{20} \end{bmatrix}^T \qquad (1)$$

其中向量C_j表示第j个交巡警服务平台对每个节点的工作压力。

5.2.2 0-1 规划模型

我们考虑每个交巡警服务平台的出警压力和全区各地到达现场的时间最短这两个因素，确立如下规划模型；

设$x_{ij}=\begin{cases}1, & \text{表示第}i\text{个节点是第}j\text{个服务平台的管辖范围} \\ 0, & \text{表示第}i\text{个节点不是第}j\text{个服务平台的管辖范围}\end{cases}$，则对节点的管辖范围分配矩阵如下为：

$$X=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,20} \\ x_{1,1} & x_{1,1} & \cdots & x_{2,20} \\ \vdots & \vdots & \ddots & \vdots \\ x_{92,1} & x_{92,2} & \cdots & x_{92,20} \end{bmatrix}=\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{20} \end{bmatrix}^T \qquad (2)$$

其中X_j为一维列向量，表示对第j个交巡警服务平台的节点分配情况。

因此，第 j 个交巡警服务平台对分配方案 X 的工作压力为：

$$X_j C_j = \sum_{i=1}^{92} x_{ij} \cdot c_{ij} \qquad (3)$$

为了使交巡警服务平台总体的工作压力最小，得到目标函数：

$$\min \sum_{j=1}^{20} \sum_{i=1}^{92} x_{ij} \cdot c_{ij} \qquad (4)$$

在分配的过程中，每个交巡警服务平台管辖的节点数不能相差太多，由 92/20=4.6，我们规定其为 3 个到 6 个，即：

$$3 \le \sum_{i=1}^{92} x_{ij} \le 6, j = 1, 2, \cdots, 20 \qquad (5)$$

在分配的过程中，每个节点仅需要一个交巡警服务平台即可，即：

$$\sum_{j=1}^{20} x_{ij} = 1, i = 1, 2, \cdots, 92 \qquad (6)$$

综上得到该问题的 0-1 规划数学模型为：

$$\min \sum_{j=1}^{20} \sum_{i=1}^{92} x_{ij} \cdot c_{ij}$$

$$s.t. \begin{cases} 3 \le \sum_{i=1}^{92} x_{ij} \le 6, j = 1, 2, \cdots, 20 \\ \sum_{j=1}^{20} x_{ij} = 1, i = 1, 2, \cdots, 92 \\ x_{ij} 取0或1，\; i = 1, 2, \cdots, 92，\; j = 1, 2, \cdots, 20 \end{cases} \qquad (7)$$

5.2.3 管辖区域方案

我们利用 Lingo 编写了程序计算上述 0-1 规划模型，求出的结果如下表：

表 1：A 区服务平台管辖区域分配方案表

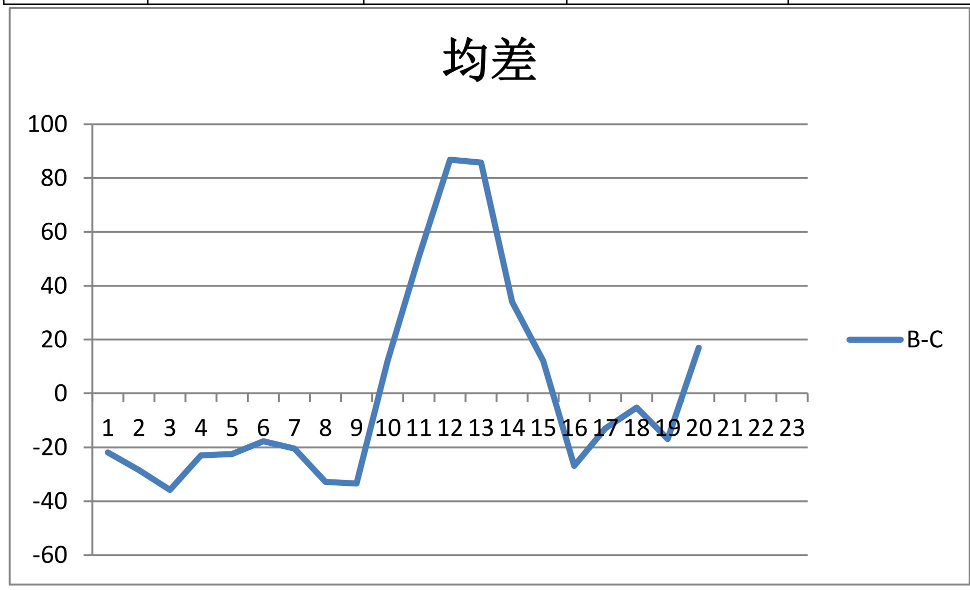
服务台标号	管辖区域节点标号	服务台标号	管辖区域节点标号
1	1 69 71 73 74 75	11	11 26 27
2	2 40 43 44 68 70	12	12 24 25
3	3 54 55 65 66 67	13	13 22 23
4	4 57 60 62 63 64	14	14 21 37
5	5 49 50 51 52 53	15	15 28 29
6	6 58 59	16	16 36 38 39
7	7 30 32 47 48	17	17 41 42 72 91 92
8	8 33 46	18	18 81 82 83 84 90

9	9 31 34 35 45	19	19 76 77 78 79 80
10	10 56 61	20	20 85 86 87 88 89

5.2.4 结果合理性分析

我们将分别每一个交巡警服务平台的工作压力计算出来得到下表：

站台	距离和	均值	均差	均差方
1	106.956208	128.8996646	-21.9434566	481.5152876
2	100.414236	128.8996646	-28.4854286	811.4196425
3	93.137352	128.8996646	-35.7623126	1278.943003
4	105.909713	128.8996646	-22.9899516	528.5378746
5	106.367519	128.8996646	-22.5321456	507.6975853
6	111.195046	128.8996646	-17.7046186	313.4535198
7	108.427784	128.8996646	-20.4718806	419.0978953
8	96.051039	128.8996646	-32.8486256	1079.032204
9	95.447703	128.8996646	-33.4519616	1119.033735
10	141.040223	128.8996646	12.1405584	147.3931583
11	179.41341	128.8996646	50.5137454	2551.638474
12	215.71113	128.8996646	86.8114654	7536.230525
13	214.61424	128.8996646	85.7145754	7346.988436
14	162.930168	128.8996646	34.0305034	1158.075162
15	141.01978	128.8996646	12.1201154	146.8971973
16	101.960422	128.8996646	-26.9392426	725.7227919
17	115.87358	128.8996646	-13.0260846	169.67888
18	123.682069	128.8996646	-5.2175956	27.22330385
19	111.946525	128.8996646	-16.9531396	287.4089423
20	145.895144	128.8996646	16.9954794	288.84632



经过计算，20 个交巡警服务平台的方差为 1346.241697，标准差为 36.691167。

该数据的意义表示任意两个服务平台间工作压力平均偏差 28%。

5.3 封锁路口调动方案

5.3.1 最短距离矩阵的确立

我们将最短距离矩阵S中不含有交巡警服务平台的节点的行剔除，将不含有路口节点的列剔除，生成一个 20×13 的矩阵S^*，其第 i 行第 j 列的元素意义为第 i 个交巡警服务平台到第 j 个路口的最短距离。

5.3.2 利用矩阵行秩求解最优方案

本问的目的在于求解一个每个路口都有一个对应的交巡警服务平台的最优方案。根据常识，我们知道完全封锁路口是由最后一名警察决定的，故该方案使得最后一个到达路口的警察所用时间最短。

我们知道，如果一个矩阵列满秩，则证明其列向量的极大线性无关组的个数等于其维度，即该列向量组的每一个维度均至少有一个列向量与之对应。

综上所述，我们先将其零元素化成 0.001，然后将矩阵S^*中大于 k 的值化为 0，得到矩阵S^{**}然后计算其行秩的方法，若该矩阵仍然行满秩，则我们认为最优方案存在，然后通过不断减小 k 的值，找到能够使S^{**}行满秩的最小的 k 值，该值即为最短封锁时间。

最后得到结果，最长封锁距离为 8.0155km，即最短封锁时间为 8.0155 分钟。

我们将S^{**}化为稀疏矩阵，并对数据进行处理，制成如下表格：

表 2：封锁路口方案可行表

原始标号	路口标号	对应的小于 k 值的交巡警服务平台标号
1	12	10 11 12 13
2	14	13 14 16
3	16	2 3 5 6 7 8 9 10 14 15 16
4	21	11 13 14
5	22	10 11 12 13 14
6	23	11 12 13 14
7	24	11 12 13
8	28	15
9	29	7
10	30	5 6 7 8 9 10 15 16
11	38	1 2 3 4 7 8 9 16 17 29

12	48	4 5 6 7 8 9 15 16
13	62	1 2 3 4 5 6 7 17 18 19 20

由该表格，我们可得到封锁方案如下表：

表 3：封锁路口方案表

原始标号	路口标号	封锁该路口的服务平台
1	12	11
2	14	16
3	16	9
4	21	14
5	22	10
6	23	12
7	24	13
8	28	15
9	29	7
10	30	8
11	38	1
12	48	4
13	62	20

封锁所用最短时间为 8.0155 分钟。

5.4 交警平台增加方案

根据题意，我们假设增加 4 个交巡警服务平台，其中 2 个为了缓解管辖区域的压力，另外 2 个为了缓解封锁路口压力。

5.4.1 缓解管辖区域压力

我们随机增加两个交巡警服务平台，然后生成一个 92×22 的矩阵，利用 5.2 中的方法，重新建立工作压力矩阵和规划方程。使得工作压力的最小值最小，然后求得最小值对应的列数，即为增加的服务平台数。

我们利用 MATLAB 编写了一个程序，其算法为：将 5.2.1 中的矩阵$\tilde{S}$增加 2 行，这两行来自最短距离矩阵S 中第 21 行到第 92 行（即没有设置交巡警服务平台的节点），然后根据以下规划方程：

$$\min \sum_{j=1}^{22}\sum_{i=1}^{92} x_{ij}\cdot c_{ij}$$

$$s.t.\begin{cases} 3 \le \sum_{i=1}^{92} x_{ij} \le 6,\ j=1,2,\cdots,22 \\ \sum_{j=1}^{22} x_{ij}=1, i=1,2,\cdots,92 \\ x_{ij}\text{取}0\text{或}1\text{，}\ i=1,2,\cdots,92\text{，}\ j=1,2,\cdots,22 \end{cases} \tag{8}$$

算出一个最小值 $\sum_{j=1}^{22}\sum_{i=1}^{92} x_{ij}\cdot c_{ij}$，然后每改变增加的行数，该最小值也发生变化，选出其中的最小值，找出对应增加的行数。

最终该程序运行出的结果为：

% x =68

% y =51

即该增加交巡警服务平台的节点标号为 88 和 71。

5.4.2 缓解封锁路口压力

我们再随机增加两个交巡警服务平台，然后生成一个 22×13 的矩阵，利用 5.3 中的方法，重新计算 k 值。

通过改变增加的列数，求得很多组不同的 k 值，选择其中最小的 k 值，然后求得该最小的 k 值对应的列数，即为增加的服务平台数。

我们用 MATLAB 用以上算法编写一个程序，其运行结果如下：

最终确定最小的 k 值为 6.750 分钟。

增加的方案为：增加第一个的交巡警服务平台可选方案为 21,22,23,24,25,26,27，增加第二个的交巡警服务平台可选方案为 28,29。

5.4.3 增加方案总结

根据 5.4.1 和 5.4.2 中的结论，我们再将 5.4.2 中的方案代入 5.4.1 中去，最终发现使得工作压力最小的方案为 71 号节点，88 号节点，23 号节点，29 号节点。

六、问题二模型的建立及求解

6.1 最短距离矩阵 $S^{'}$ 的确立

我们利用 Floyd 算法，利用了直达矩阵 $D^{'}$ 算出了全市 582 个站点间的最短距离矩阵 $S^{'}$。该矩阵第 i 行第 j 列的元素的意义是第 i 个节点到第 j 个节点的最短距离。

6.2 合理性判断

6.2.1 通过常规报案判断

我们将全市含有交巡警服务平台的列数保留，其他列剔除，最终形成一个 582×80 的矩阵。

然后先后进行五次筛选，每次将其大于 2.5，3，4，5，6 的元素化为 0，然后利用 MATLAB 中的 rank（A）命令计算该矩阵五次筛选的秩为 73，76，80，80，80。

我们发现，该矩阵的秩在五次筛选的过程中，距离其满秩的值差距不大，在 3 分钟内，有 4 个交巡警服务平台无法到达任何节点数，该四个站点为 10，14,177,378。

相比较所有的服务平台个数而言，此数据只是其中很少的一部分。

或者根据 5.2 中的 0-1 规划模型，计算总工作压力。

$$\min\sum_{j=1}^{80}\sum_{i=1}^{582}x_{ij}\cdot c_{ij}$$

$$s.t.\begin{cases}5\le\sum_{i=1}^{582}x_{ij}\le 10, j=1,2,\cdots,80\\ \sum_{j=1}^{80}x_{ij}=1, i=1,2,\cdots,582\\ x_{ij}\text{取}0\text{或}1，i=1,2,\cdots,582，j=1,2,\cdots,80\end{cases}\tag{9}$$

我们发现其总工作压力值和 5.2 中的数据差距不大。

故我们认为在常规报案情况方面，该设置方案基本合理。

6.2.2 通过封锁方案判断

我们将全市含有交巡警服务平台的列数保留，其他列剔除，将全市含有路口的行数保留，其他行剔除，最终形成一个 17×80 的矩阵，然后利用 5.3 中计算 k 的临界值的方法计算其最短时间为 12.78 分钟，考虑全市的面积和封锁难度，

再和 5.3 中的 8.0155 分钟相比，我们认为该数据的增加量不大，故我们认为在重大事件封锁方案方面，该设置方案基本合理。

6.2.3 城区人口判断

根据常识，一个城区的人口密度越大，则需要的交巡警服务平台越多。其满足正相关性。

我们根据附件 2 中六城区基本数据中的每个城市的面积和城区人口数，计算出各个城区的人口密度。该数据为 2.73,0.20,0.22,0.19,0.18,0.19。

然后根据附件 2 中交巡警服务平台中的数据，得到每个城区的交巡警服务平台数为 20,8,17,9,15,11。

然后将该两组数据进行做回归分析，再进行 F 检验，其公式如下：

$$F = \frac{SSR / m}{SSE / (\mathrm{n-m-1})}$$

我们调用 MATLAB 中的 reglm 函数，计算 F 值为 29.3884。

在本题为一元回归模型，样本观测值为 6，故 m=1，n-m-1=4。

我们查表得到在置信度 α 为 0.05 时候的 F 值为 7.71，29.3884＞7.71，我们接受这个假设，认为其合理。

6.2.4 结论

根据常规报案情况，重大事件封锁路口情况和人口比例工作压力情况三个方面考虑，我们认为全市的交巡警服务平台设置基本合理。

6.3 围堵方案的求解

我们假设犯罪嫌疑人的速度与警车相同，然后计算三分钟内，犯罪嫌疑人有可能到达的节点数为 7,8,9,30,31,32,33,34,35,36,45,46,47,48。

再根据交巡警服务平台的设置，建立动态规划模型。

七、模型的评价及改进

7.1 模型的评价

7.1.1 模型的优点

1、在求解管辖区域分配问题时，我们利用了矩阵的 0-1 规划模型，并用 Lingo 求解，清除直接地计算出最优方案。

2、在求解封锁道路方案的时候，我们巧妙地利用了矩阵行满秩的思想，利用如果存在对应关系必定行满秩这一性质，将其中时间过长的元素剔除，而且剔除后行满秩的临界值必然大量减少其方案。
3、在计算增加方案时，我们充分考虑了任意将两个节点变为服务平台的情况，并考虑其最大收益，能够充分取到全局最优解。

7.1.2 模型的缺点

1、在考虑交巡警服务平台的增加方案的时候，我们是分别考虑两个因素的影响，而未将其综合考虑。

7.2 模型的改进方向

1、在求解交巡警服务平台的增加方案的时候，我们可以对程序进行优化。考虑增加三列的情况。

2、在判断全市常规报案是否合理的时候，我们还可以根据每个城区的人均占地面积，设定警察到达事发现场所用的合理时间，然后将全市各节点最短距离矩阵分块，将管理外区的情况非别化为零矩阵，分别进行过滤，然后再计算该矩阵的秩。

参考文献

[1]汪晓银，周宝平，数学建模与数学实验[M]，北京：科学出版社，2012.8。

[2]司守奎，孙玺菁，数学建模方法与应用[M]，北京：国防工业出版社，2014.2。

[3]肖华勇，实用数学建模与软件应用[M]，西安：西北工业大学出版社，2008.11。

[4]冯予 陈萍，概率论与数理统计[M]，国防工业出版社，2012.3。

八、附录

8.1计算直达距离

```
load demo02
juzhen=zeros(92);
j=1;
for i=1:140
    juzhen(xzuo(i),yzuo(j))=distance(xzuo(i),yzuo(j));
    juzhen(yzuo(j),xzuo(i))=distance(xzuo(i),yzuo(j));
    j=j+1;
end
%size(sparse(juzhen))
for i=1:92
for j=1:92
if i==j
            juzhen(i,j)=0;
end
if i~=j & juzhen(i,j)==0
            juzhen(i,j)=Inf;
end
end
end
save D:\matlabcode\2011B\zhidajuzhen
zuiduanjuli=floydsuanfa(juzhen)/10;
save D:\matlabcode\2011B\zuiduanjulizuiduanjuli
```

8.2计算两点间距离

```
function distance=jisuan(a,b)
load demo01!三点坐标
ah=find(zdata==a);
bh=find(zdata==b);
x1=data(ah,1);
x2=data(bh,1);
y1=data(ah,2);
y2=data(bh,2);
distance=sqrt((x1-x2).^2+(y1-y2).^2);
```

8.3Floyd算法

```
function [D,R]=floyd(A)
%用floyd算法实现求任意两点之间的最短路程。可以有负权
%参数D为连通图的权矩阵
D=A;n=length(D);
for i=1:n
    for j=1:n
```

```
        R(i,j)=i;%赋路径初值
    end
end
for k=1:n
    for i=1:n
        for j=1:n
            if D(i,k)+D(k,j)<D(i,j)
                D(i,j)=D(i,k)+D(k,j);%更新 D(i,j),说明通过k的路程更短
                R(i,j)=R(k,j);%更新R(i,j), 需要通过k
            end
        end
    end
    hl=0;
    for i=1:n
        if D(i,i)<0
           hl=1;
           break;%跳出内层的for循环
        end
    end
    if(hl==1)
        fprintf('有负回路')
        break;%跳出最外层循环
    end
end
```

8.4数据筛选

```
load zuiduanjuli   %最短距离矩阵
for i=1:20
    min_dis2(:,i)=zuiduanjuli(:,i);
end
min_dis2=min_dis2/10;
%zuiduanjuli2  92行20列
save D:\matlabcode\2011B\zuiduanjuli2 min_dis2
for i=1:92
    for j=1:20
        if min_dis2(i,j)>3 %过滤zuiduanjuli2
            min_dis2(i,j)=10000;
        end
    end
end
min_dis3=min_dis2;
save D:\matlabcode\2011B\xianzhijuzhen min_dis3
```

8.5 0-1规划循环

```
%load guanxiatest  %min_dis3
%load realfananshuju2  %realfananshuju2
%load zuiduanjuli
%tmp=min_dis3;
tic


%------计算最距离矩阵  92*22
% result=zeros(72);
% for i=1:72
%     for j=1:72
%         if  i>j
%             hang1=zuiduanjuli(:,i);
%             hang2=zuiduanjuli(:,j);
%             tmp1=[tmp';hang1';hang2']';  %最短距离矩阵
%             gongzuoyalijuzhen=tmp1.*realfananshuju2;
%             result(i,j)=o1jisuan(gongzuoyalijuzhen);
%         else
%             continue;
%         end
%     end
% end

% hang1=zuiduanjuli(:,4);
% hang2=zuiduanjuli(:,7);
% tmp1=[tmp';hang1';hang2']';  %最短距离矩阵 92*22
% gongzuoyalijuzhen=tmp1.*realfananshuju2;
%
% o1jisuan(gongzuoyalijuzhen)
load result
% for i=1:72
%     for j=1:72
%         if i<=j
%             result(i,j)=10000;
%         end
%     end
% end
zuixiao=min(min(result));
[x,y]=find(result==zuixiao)
%xlswrite('a.xlsx',zhaodao);
% x =68  zuiduanjuli里面的88列
% y =51  zuiduanjuli里面的71列
Toc
```

01jisuan

```
function  value=o1jisuan(gongzuo)
%------为计算准备初始化值
a=zeros(92);
for i=1:92
    for j=1:92
        if i==j
            a(i,j)=1;
        end
    end
end
Aeq=[a a a a a a a a a a a a a a a a a a a a a a];
Beq=linspace(1,1,92)';
%size(Beq)
%size(Aeq)

for i=1:22
    for j=1:92
        b(i,j)=0;
    end
end
c=[b b b b b b b b b b b b b b b b b b b b b b];
%size(c)
for i=1:22
    for m=1+(i-1)*92:92*i
        c(i,m)=1;
    end
end
A=[c;-c];
%size(A)
tmp11=linspace(6,6,22)';
tmp22=linspace(-3,-3,22)';
B=[tmp11;tmp22];
%size(B)
%------初始值计算结束

%计算f
a1=[];
for i=1:22
    a1=[gongzuo(:,i);a1];
end
%size(a1)
```

```
options=optimset('display','off');
%x0=linspace(0,0,2024)';
[x fval]=bintprog(a1,A,B,Aeq,Beq);
value=min(fval);
```

8.6 筛选最小值对应的列数

```
function zuixiao(a)
%(1)B=sort(A) 对一维或二维数组进行升序排序,并返回排序后的数组,当A为二维时,对数组每一列进行排序.
%eg: A=[1,5,3],则sort(A)=[1,3,5]
%A=[1,5,3;2,4,1],则sort(A)=[1,4,1;2,5,3]
%(2)B=sort(A,dim),对数组按指定方向进行升序排序,
%dim =1,表示对每一列进行排序,,dim=2表示对每一行进行排序.
%(3)B=sort(A,dim,mode),mode为指定排序模式,mode为"ascend"时,进行升序排序,为"descend "时,进行降序排序.
%(4)[B,I]=sort(A,.....),I为返回的排序后元素在原数组中的行位置或列位置.
[B,I]=sort(a);
zuixiao=B(2:4,:);
zuisuo=I(2:4,:);
%zui=[zuixiao;zuisuo];

%save D:\matlabcode\2011B\zuiduanfuwu zuixiao zuisuo
%在保存正数到txt文件时用save result.txt p -ascii
%dlm('filename',M,'D',R,C)
%将矩阵M的R行和C列用分割符D写入到filename中
```

8.7 秩逼近法

```
load zuiduanjuli
% 1-20 line
% 12 14 16 21 22 23 24 28 29 30 38 48 62 colume
%min_dis_matr
% min_dis_matr(1:20,12);
a=[12 14 16 21 22 23 24 28 29 30 38 48 62];
b=1;
for i=a
    tmp(:,b)=zuiduanjuli(1:20,i);
    b=b+1;
end
%tmp为封锁方案矩阵20*13
%save D:\matlabcode\2011B\fengsuofangan tmp
%把所有的inf换成一个很大的数把 0 换成一个很小的数
```

```
for i=1:20
    for j=1:13
        if tmp(i,j)==0
            tmp(i,j)=0.0001;
        end
    end
end
%计算K值
tmp1=tmp;
a=0;
for k=30:-0.001:1
    for i=1:20
        for j=1:13
            if tmp1(i,j)>k
                tmp1(i,j)=0;
            end
        end
    end
    if rank(tmp1)~=13
        a=k+0.001;  %注释
        break;
    end
end
%最优封锁方案
 for i=1:20
        for j=1:13
            if tmp(i,j)>a
                tmp(i,j)=0;
            end
        end
 end
 zuiyou=tmp;
 %rank(zuiyou)
 %save D:\matlabcode\2011B\zuiyoufangan zuiyou
 %sparse(zuiyou)
```

8.8 循环秩逼近法

```
tic
load zuiduanjuli
% 1-20 line
% 12 14 16 21 22 23 24 28 29 30 38 48 62 colume
%min_dis_matr
% min_dis_matr(1:20,12);
a=[12 14 16 21 22 23 24 28 29 30 38 48 62];
```

```
b=1;
%提出前面20行
for i=a
    tmp(:,b)=zuiduanjuli(1:20,i);
    b=b+1;
end

%提出后面72行
c=1;
for i=a
    tmp2(:,c)=zuiduanjuli(21:92,i);
    c=c+1;
end
save tmp tmp tmp2
%size(tmp2)
%tmp为封锁方案矩阵20*13
%tmp2为72*13
%
% kset=zeros(72);
% %test=zeros(72);
% for i=1:72
%     for j=1:72
%         if i~=j
%             hang1=tmp2(i,:);
%             hang2=tmp2(j,:);
%             tmp1=[tmp;hang1;hang2];
%             %test(i,j)=i*j;
%             kset(i,j)=funck(tmp1);
%         else
%             continue;
%         end
%     end
% end
% load kset
% for i=1:72
%     kset(i,i)=10000;
% end
% zuixiao=min(min(kset));
% find(kset==zuixiao)
toc
```