

交巡警服务平台的设置与调度

摘要

在市区的交通要道及重要部位设有交巡警服务平台。平台间的职能与警力的配备大致相同。由于警务资源有限，应根据实际情况及需求对平台的位置、管辖区域、资源调度进行合理设置。

针对问题一，我们利用 MATLAB 软件对 A 区的 20 个交巡警服务平台在三分分钟内可以赶到的区域进行圈定，面对平台间的重复区域，我们以各个点到各平台的距离最短为原则，对各平台的管辖区域进行**遍历分析**，从而为每个平台分配管辖范围，结果如图 2 所示。

我们对 A 区节点图与各平台的管辖范围图进行分析，结合 13 条要道的节点位置可知，平台与节点具有一对一和一对多的关系；我们利用**图与网络模型**求解这些节点到各平台的最短距离，并以距离最短为封锁原则，具体方案见表 1。

我们采用**0-1 非线性规划模型**。以平台工作量的**变异系数**最小为目标函数，以一个平台可管辖多个节点，而每个节点只能由一个平台管辖，且该平台必须管理自己所在的节点为约束条件。利用遍历分析，依次将非平台的节点作为新平台，求解其变异系数，在每次遍历的结果中找最小值，该值对应的节点就是新添的平台，得到应增加 5 个平台，具体位置在节点 29、30、50、66、84 的结论。

针对问题二，首先我们利用问题一中的方法对所有平台的管辖区域进行划分，发现部分节点不在范围内，且各平台的工作量非常不均衡，因此现有的方案明显不合理；对不在范围内的节点，利用问题一中的方法对其进行处理，然后采用相同方法处理所有未设立平台的节点，最后得出结论：应在 30 个节点：69、84、109、141、163、220、235、246、271、274、279、311、316、358、449、469、530、536、553、557 处设立新平台。

我们采用**0-1 非线性规划模型**，以嫌疑犯逃跑的最大用时最小为目标函数，以交巡警的追捕时间小于嫌疑犯的逃跑时间，一个平台可围堵一个节点，且交巡警应先围堵全市的出入口节点为约束条件，按照距离最短原则，利用**图与网络模型**解决平台与节点间一对多的情况。最佳围堵方案见表 3。

关键词：遍历分析 图与网络模型 0-1 非线性规划模型 变异系数

一、问题重述

警察肩负着许多职能，为了有效实施这些职能，需在市区的部分交通要道及重要部位设立起交巡警服务平台，每一个平台的警力配备及职能大致相同。根据城市的实际发展情况及需要，在有限的警务资源下，合理地设置平台、分配它们的管辖区域、调度好警务资源是相关部门需要面临的实际课题。根据附件建立数学模型解决以下问题：

（1）根据附件信息，为各个交巡警服务平台设定管辖范围，以确保在某一管辖范围内，若出现突发事件，尽量能在三分钟之内有交巡警到达事发处（警车时速：60km/h）。面对重大突发事故，需调动全区 20 个平台的警力，快速全封锁进出的 13 条要道。现实条件下，一个平台的警力最多仅可封锁一个路口，设计合理的警力调度方案。拟再增加 2~5 个平台以解决部分地区出警时间过长及现有平台的工作量不均衡的情况，求解需要增加的平台个数及位置。

（2）针对全市的情况，依照设置新平台的原则和任务，分析该市现有的交巡警服务平台方案的合理性，若有明显不合理的地方，需给出解决方案。若该市 P 处发生重大案件，在案发三分钟后接到了报警，此时犯罪嫌疑人已驾车潜逃。请给出一个最佳围堵方案，调度全市平台警力资源以快速追捕嫌犯。

二、模型假设

1.假设嫌犯逃跑时的速度及警车追捕的速度一致。

三、符号说明

A_i	第 i 个平台的工作量
a_j	第 j 个节点的案发率
Z	平台工作量的变异系数
V	嫌疑犯逃跑和警察追捕速度
T	交巡警封锁的时间
T'	嫌犯逃跑的时间

S_{ij}	交巡警从第 i 个平台出发直到封锁 第 j 个节点所用的路程
L_{32j}	嫌疑犯从第 32 个节点逃跑到 第 j 个节点被抓捕的路程
M	平台数量的最大值
N	节点数量的最大值

四、问题分析

4.1 问题一的分析

对于问题一，根据题目要求交巡警需在三分钟内到达事发地，利用该时间及警车的时速，我们可绘制出每个平台能及时到达的范围；由于该范围会存在重合现象，可根据重合地区到平台的距离最短为原则对它们进行进一步划分，从而确定各个平台的管辖范围。

根据题目及附件资料可知，封锁该区至少需派出 13 个平台的警力；我们以交通要道到平台的距离最小为条件，求解出相应平台，当若干条要道对应的平台为同一个时，仍以距离最短为原则确定平台封锁的道路。

为了解决各平台工作量不均衡及部分地区出警时间过长的问題，我们可利用附件信息并建立合理指标，通过比较在不同地区设立新平台时的指标数值，确定需要增加的平台个数及具体位置。

4.2 问题二的分析

对于问题二，我们从人为划分的平台管辖区域是否包含多数节点及各平台的工作量是否均衡对现有的方案进行分析，若现有的方案具有明显不合理的地方，我们可利用附件信息建立合理指标，先比较不在管辖范围内的节点设立新平台时的指标数值，确定新平台的设立数量及具体位置，再比较各个非平台的节点处的指标数值，进一步添加新平台，从而解决现有方案的不合理之处。

为了快速搜捕嫌疑犯，各平台应先对全市的出入口节点进行围堵，在仍有余力的条件下围堵其余节点。由于一个平台可管理多个节点，而一个平台只能围堵一个节点，因此在挑选节点进行围堵时，应以节点与平台间的距离最短为原则。在满足以上条件的前提下，可求解出一个最佳围堵方案

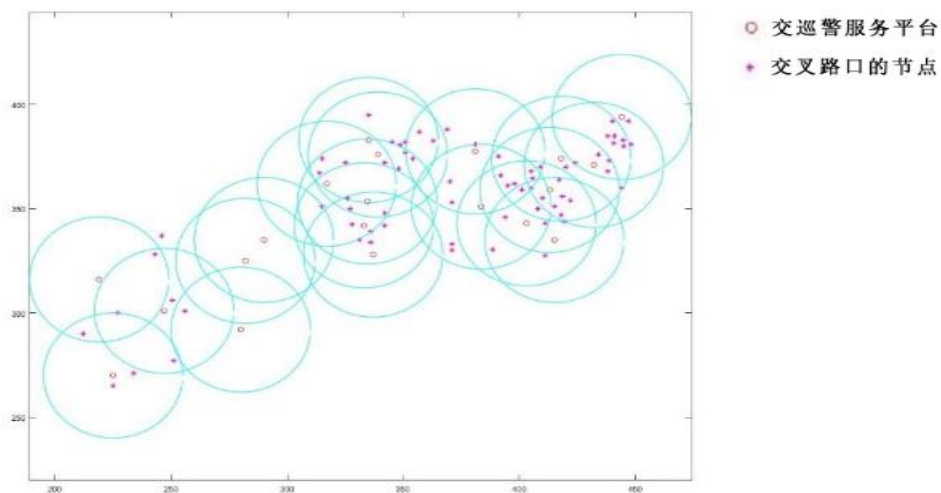
五、模型的建立及求解

5.1 利用 0-1 非线性规划及图与网络模型进行求解

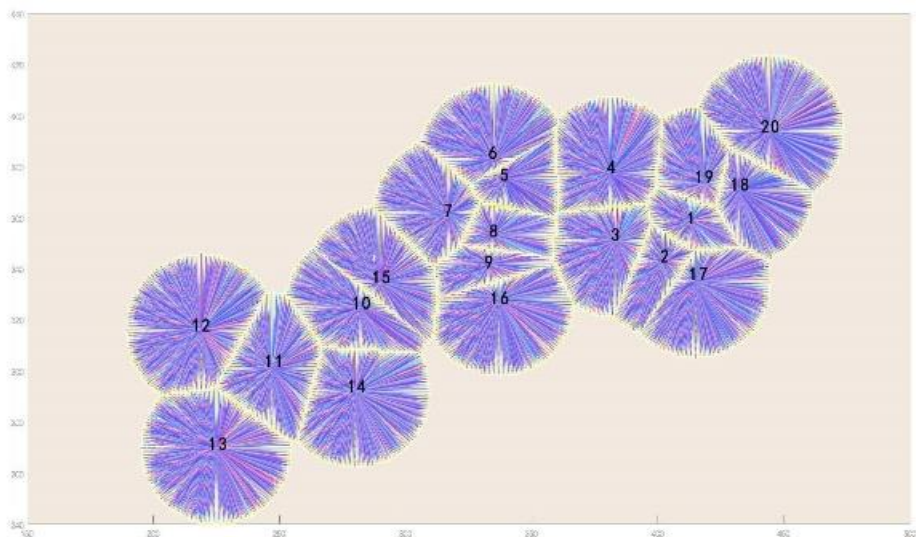
5.1.1 为交巡警服务平台划分管辖范围

根据附件一中的附图 1 可知，A 区共设置了 20 个交巡警服务平台。根据题目要求，当各平台所管辖的范围内发生突发事件时，交巡警尽量能在三分钟内赶到事发地，警车的时速可达到 60 千米每小时，即某平台的管辖范围应在以该平台为圆心，3km 为半径的圆内。

我们利用 MATLAB 软件，对附图一中的 20 个平台点进行处理，绘制出以 20 个平台为圆心，3km 为半径的圆，结果如下图（a）所示；由于部分平台的管辖范围会有所重合，因此我们以管辖范围内各个点到交巡警服务平台距离最短为原则，利用 MATLAB 软件对各个平台的管辖区域进行遍历分析，最后得到各个平台的管辖范围，结果如图 1（b）所示。



(a)



(b)

图 1：20 个平台点管辖区域划分图

图 1（b）中，数字代表平台编号，紫色区域表示该平台的管辖范围，之间的白色线段为两区域间的分界线。

由于各区域之间均各自设置了交巡警服务平台，A 区的交巡警仅需负责处理 A 区内的突发事件。因此需结合 A 区地图对上图进一步处理，结果如下所示：

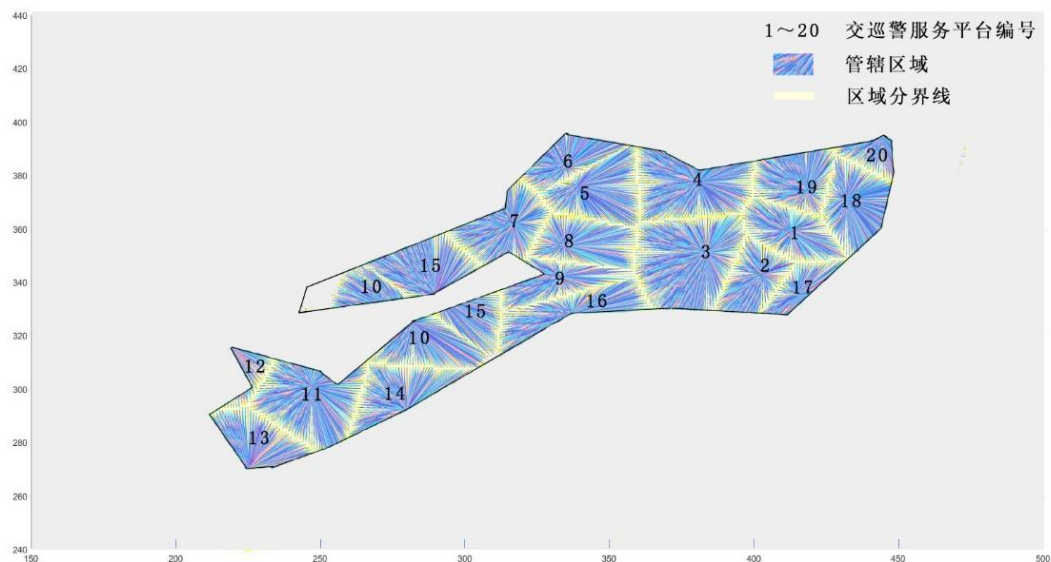


图 2：A 区交巡警服务平台管辖范围

上图中，数字表示交巡警服务平台的编号，蓝色色块表示某平台的管辖区域，两管辖区域间以白线相隔。其中，平台 10 左边的区域空白，代表没有平台可以在 3 分钟内赶到该区域，以该区域与各平台的距离最短为原则，我们将它

划入平台 12 的管辖区域内。

5.1.2 交巡警服务平台警力的调度方案

根据附件二提供的 A 区各个节点的坐标，结合 A 区的各个平台的管辖范围作图得：

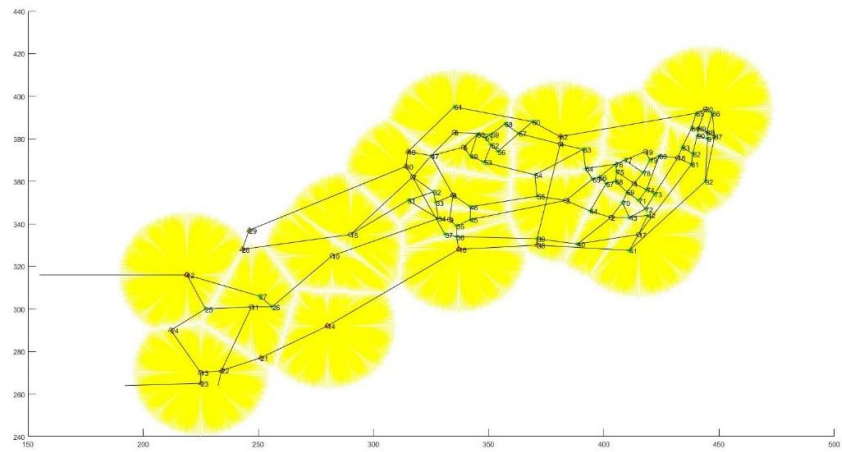


图 3：A 区各节点分布图

结合图 2、图 3 及该区的 13 个路口节点标号可得：平台 3 管辖范围包含 38 号交通要道；平台 4 管辖范围包含 62 号交通要道；平台 7 管辖范围包含 30、48 号交通要道；平台 11 管辖范围包含 21 号交通要道；平台 12 管辖范围包含 12、28、29 号交通要道；平台 13 管辖范围包含 22、23、24 号交通要道；平台 14 管辖范围包含 14 号交通要道；平台 16 管辖范围包含 16 号交通要道。

由于一个平台的交巡警警力最多仅可封锁一个路口，且出警速度应尽可能快，因此若某平台仅包含一个交通要道，该要道可由该平台进行封锁。但若平台管辖区域包含的要道大于其封锁能力，则需对其余交通要道进行再分配。

利用图与网络模型，将 A 区的地图网格化，以剩余交通要道到剩余平台的距离为指标，求解它们之间的最短距离矩阵，其中， y_i 表示第 i 个平台， x_j 表示第 j 个节点（由于数量较多，下列矩阵仅列出了未被分配的节点与其距离最短的平台）：

	y_5	y_7	y_8	y_9	y_{10}	y_{12}	y_{13}	y_{15}
x_{12}	176.3	149.1	140.9	130.1	75.9	0	59.8	170.3
x_{22}	177.5	150.4	142.1	131.3	77.1	68.8	9.1	171.5
x_{23}	191.6	164.4	156.2	145.4	91.1	64.8	5	185.6
x_{24}	182.9	155.7	147.5	136.7	82.4	35.9	23.9	83.6
x_{28}	113.1	85.7	102.3	97.8	141.9	217.8	228.1	47.5
x_{29}	106.2	80.2	104.9	107.2	151.4	227.3	237.6	57.0
x_{30}	31.8	5.8	30.6	34.9	79.1	155.0	165.2	44.0
x_{48}	24.8	12.9	31.0	42.0	86.2	162.1	172.3	51.1

由上述矩阵可知，12 号节点与平台 12 距离最近，22、23、24 号节点与平台 13 距离最近，28、29 号节点与平台 15 距离最近，30、48 号节点与平台 7 距离最近。由于一个平台仅可封锁一个节点，且需实现快速全封锁的目标，因此当平台与节点产生一对多关系时，我们选取距离该平台最近的节点进行封锁；对剩余节点，我们取其与所有平台的距离次小值，若仍存在一对多关系时，以同样的方法对其进行处理。

最后的警力调度方案如下表所示：

表 1：警力调度方案

平台	封锁节点	平台	封锁节点
3	38	11	21
4	62	12	12
5	48	13	23
7	30	14	14
8	2915	15	28
9	24	16	16
10	22		

5.1.3 平台的增加个数及位置

为了有效解决交巡警服务平台工作量不均及对某些地方的出警时间过长问题，我们采用 0-1 规划模型。

①模型建立

$$x_{ij} = \begin{cases} 0 & \text{第} i \text{个平台不管辖第} j \text{个节点} \\ 1 & \text{第} i \text{个平台管辖第} j \text{个节点} \end{cases}$$

由附件可知，上式中， $i=1、2、\cdots 20、\cdots m$ ， $j=1、2、\cdots 92$ 。

令 A_i 表示第 i 个平台的工作量， a_j 表示第 j 个节点的案发率。因此：

$$A_i = \sum_{j=1}^{92} a_j x_{ij}$$

我们将平台工作量的变异系数 Z 作为评判指标，变异系数越小，表示该平台的工作量越均衡：

$$\min Z = \frac{\sqrt{\frac{1}{m-1} \sum_{i=1}^m (A_i - \bar{A})^2}}{\bar{A}}$$

因为一个平台可管辖多个节点，而每一个节点只能由一个平台对其进行管辖，且该平台必须管理自己所在的节点，因此我们对上式进行约束：

$$s. t. \begin{cases} \sum_{i=1}^m x_{ij} = 1, & i \neq j \\ x_{ij} = 1, & i = j \end{cases}$$

②过程求解

结合图 2 和图 3 可知，没有平台可在 3 分钟内赶到节点 29 附近区域，因此我们首先在节点 29 处增加一个新平台，此时 $m=21$ 。利用 5.1.1 中的方法对 21 个平台重新进行区域划分，按照新划分的 21 个平台的管辖范围，确定每一个平台的管理节点。利用附件二中各节点的案发率对变异系数 Z 进行求解：

$$Z = 0.4718, \quad m = 21$$

继续添加新平台时，我们采用遍历的方法寻找可使变异系数达到最小值的平台点：依次在未安置平台的节点处添加新平台，并利用上述方法对变异系数进行求解，在求解得到的所有变异系数中寻找最小值，该值所对应的节点处就是新添加的平台位置。

依据题目要求并利用上述方法求得的增加新平台后变异系数的变化如下表所示：

表 2：增加新平台后变异系数变化表

新增平台位置	变异系数
节点 29	0.4718
节点 66	0.4340
节点 84	0.3969
节点 50	0.3654
节点 30	0.3372

利用 MATLAB 软件，将求得的变异系数作图表示如下：

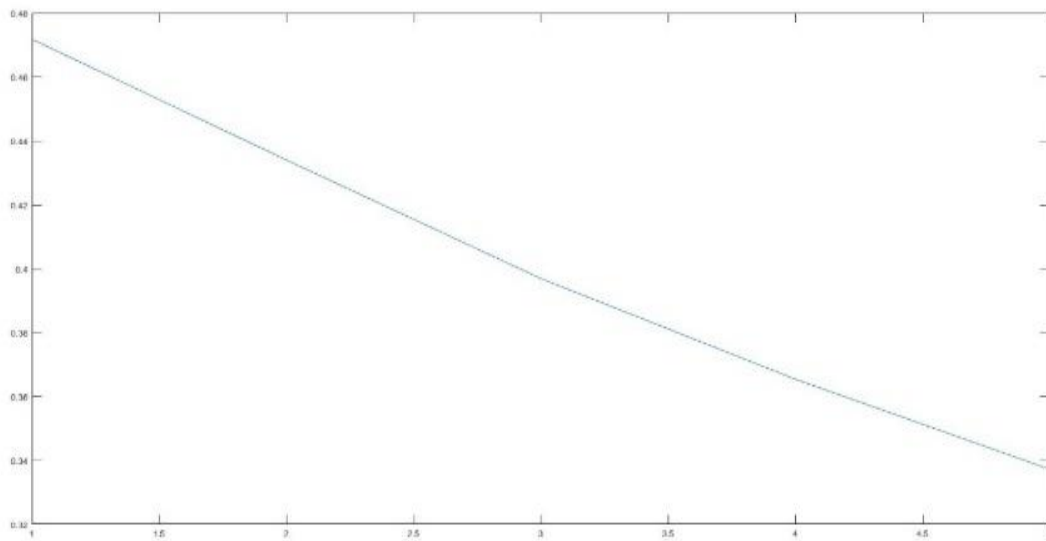


图 4：变异系数的下降趋势

由于变异系数的下降趋势呈线性，期间无明显转折节点。根据题目新增 2~5 个平台的要求，我们增加了 5 个平台，以缓解各平台工作量不均衡及某些地区出警时间过长的的问题。

5.1.4 结果分析

各个交巡警服务平台的管辖范围如图 2 所示，其中第 29 号节点附近的区域未在各平台的范围之内，根据该节点与各平台的距离最短原则，我们将它划入平台 12 的管辖区域内。

A 区的交巡警服务平台的警力调配方案见表 1。

在 A 区应增加 5 个平台，具体位置在节点 29、30、50、66、84。

5.2 利用 0-1 非线性规划模型进行求解

5.2.1 分析平台设置方案合理性

①方案合理性分析

根据附件资料，利用 5.1.1 中的方法对该市的所有平台管辖范围进行划分，结果如下图所示：

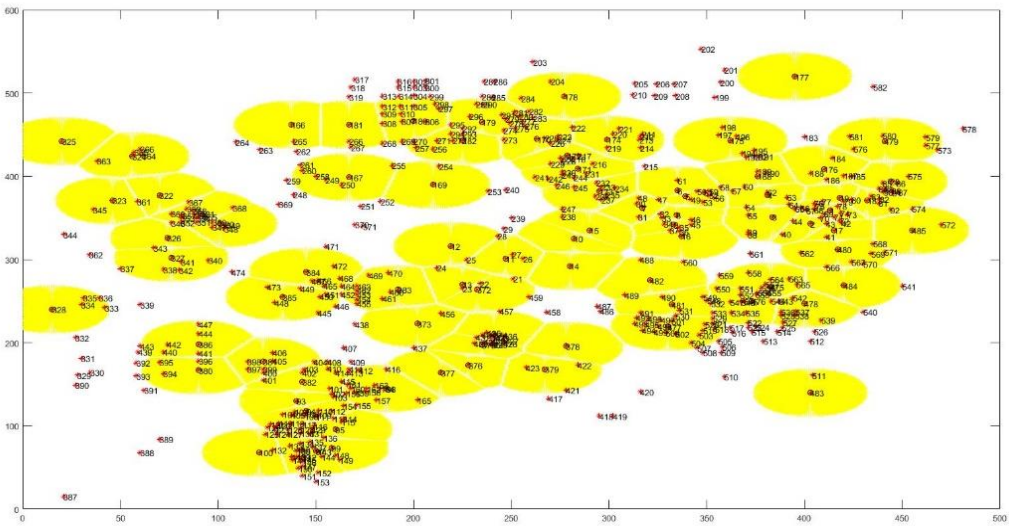


图 5：该市所有平台管辖区域划分图

上图中，黄色区域代表平台的管辖范围，其余数字为全市的节点编号，由图可知，仍有部分节点不在平台的管辖范围内；部分平台的管辖区域内节点排列紧凑，而部分平台的管辖区域节点稀疏，各平台的工作量不均衡。因此，该市目前的交巡警服务平台设置得不合理。

②解决方案

对目前为止未被包含进管辖区域的节点进行筛选，筛选结果如下图所示：

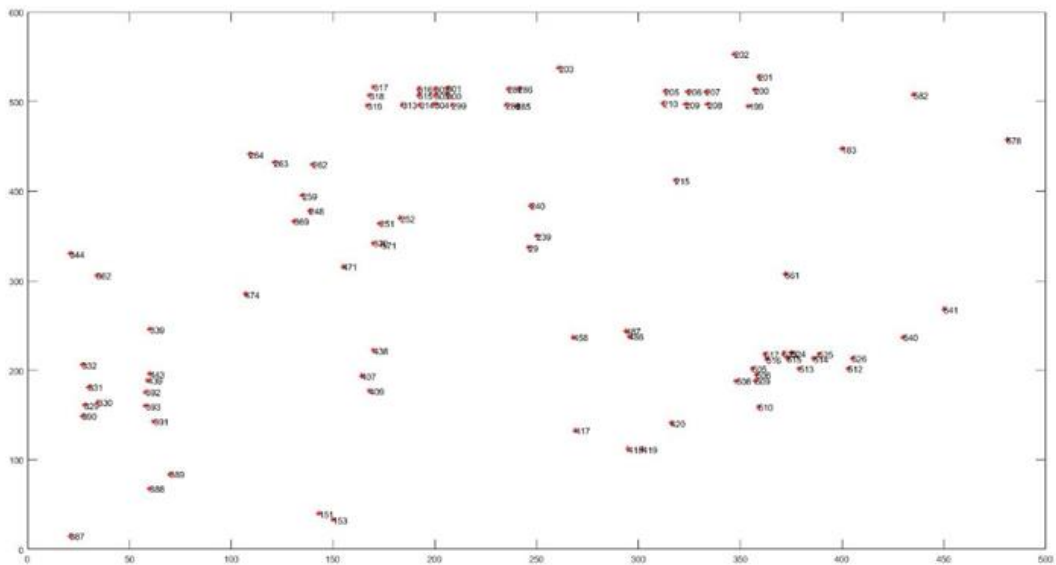


图 6：不在平台管辖范围内的节点分布图

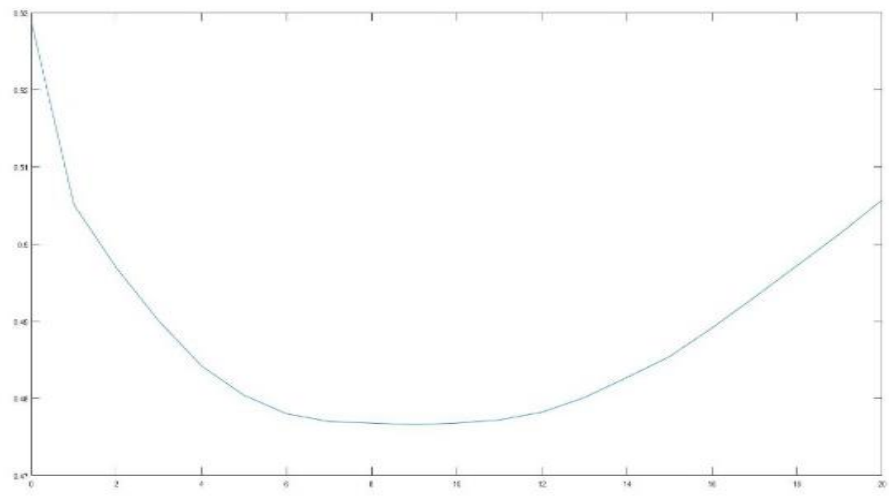
对于上述 93 个节点，我们仍以变异系数为指标，利用遍历的方法对所有未被划分进管辖区域的节点进行处理。求解出在任意一个节点设立平台时的变异系数，每一次遍历都挑选出变异系数最小值所对应的节点，该节点位置即为设置新平台的具体位置。20 次遍历结果如下表所示：

表 3：遍历结果表

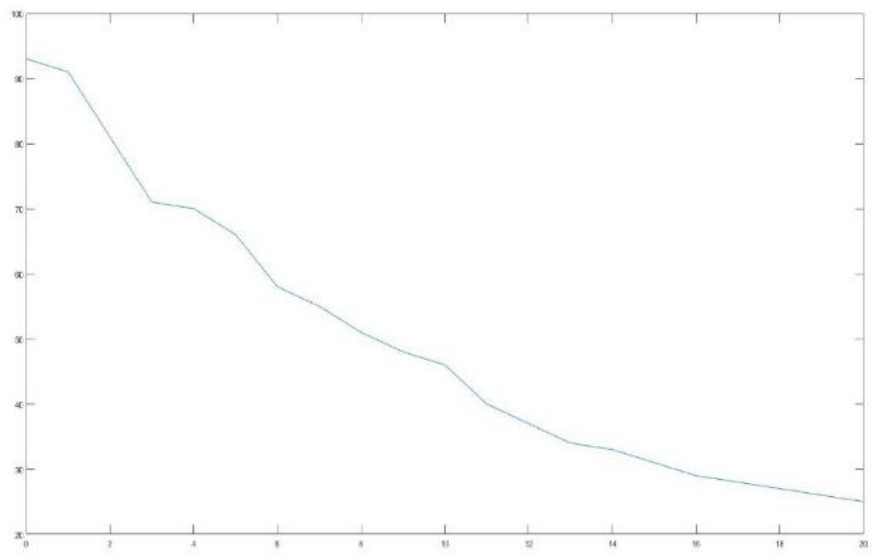
遍历次数	设立节点	变异系数	剩余节点数/个	遍历次数	设立节点	变异系数	剩余节点数/个
1	409	0.51	91	11	330	0.4772	40
2	524	0.50	81	12	319	0.4782	37
3	304	0.4900	71	13	458	0.4801	34
4	438	0.4842	70	14	240	0.4827	33
5	288	0.4804	66	15	526	0.4854	31
6	208	0.4780	58	16	151	0.4891	29
7	509	0.4770	55	17	474	0.4931	28
8	251	0.4768	51	18	417	0.4972	27
9	259	0.4766	48	19	478	0.5013	26
10	239	0.4768	46	20	339	0.5057	25

根据该表的变异系数，利用 MATLAB 软件绘制出变异系数变化图及不在

管辖范围内的节点个数变化图：



(a)



(b)

图 7：变异系数及无管辖平台节点数变化图

由图可知，在第 10 次遍历后，变异系数的值开始逐步增大，而不在平台管辖范围内的点减少速率降低。因此，在保证变异系数最少且被管辖到的节点数最多的条件下，我们新添了前十个交巡警服务平台。

新添加的十个平台可有效缓解多数节点不在平台管辖区域内的问题，但仍存在各平台工作量不均衡的问题。在已添加上述十个平台的基础之上，我们对工作量不均衡问题进行完善。

我们采用遍历法对所有不是交巡警服务平台的节点进行处理，依次在这些节

点处添加新平台，并求解整个市工作量的变异系数，在每一次遍历后得到的变异系数中寻找最小值，该值所对应的节点就是新添加的平台位置。我们一共进行了 20 次遍历，结合之前的 10 次遍历，30 个变异系数的变化如下图所示：

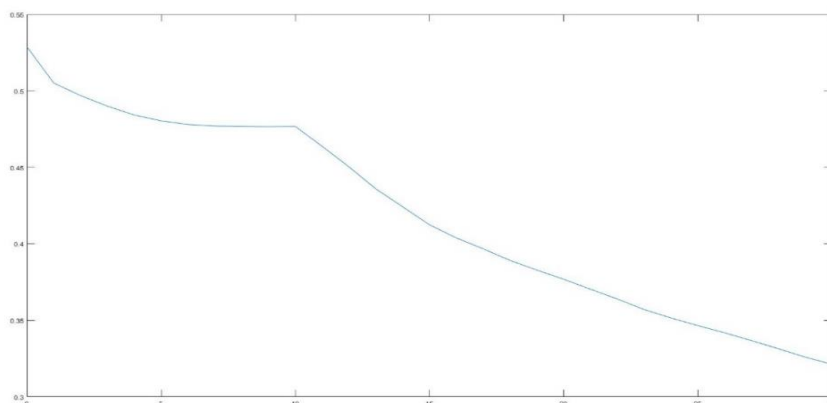


图 8：变异系数的变化

由上图可看出，到第 30 个变异系数时，变异系数的减小速度已明显放缓，之后若再增加平台，变异系数不会产生明显变化，但对新平台的投入并不会因此减少，且在问题一中，题目要求增加 2~5 个新平台，其上限为一个区 5 个，所以我们认为添加这 30 个新平台是合理的。

这 30 个新平台位置分别为：节点 109、553、274、311、358、246、141、469、271、530、163、84、69、536、279、557、235、220、316、449 处。

5.2.2 设计最佳围堵方案

我们采用 0-1 变量模型对本题进行求解。

①模型建立

$$x_{ij} = \begin{cases} 0 & \text{第 } i \text{ 个平台不封锁第 } j \text{ 个节点} \\ 1 & \text{第 } i \text{ 个平台封锁第 } j \text{ 个节点} \end{cases}$$

由附件资料可知，全市共有 80 个服务平台，582 个节点。

根据题目要求，全市的交巡警服务平台需要对嫌疑犯进行快速围堵，而在交巡警接到报案时，嫌疑已驾车逃跑三分钟，假设嫌犯驾车逃跑的速度与警车的追捕速度相同，均为 $V = 60\text{km/h}$ ；我们用 T 表示交巡警封锁的时间，用 T' 表示嫌犯逃跑的时间，用 S_{ij} 表示交巡警从第 i 个平台出发直到封锁第 j 个节点所用的路程，用 L_{32j} 表示嫌疑犯从第 32 个节点逃跑到第 j 个节点被捕的

路程（由于嫌疑犯在该市的第 32 个节点开始逃跑，故 $j = 32$ ）。所以，目标函数为：

$$\min f = \max T$$

该目标函数的意义为嫌疑犯逃跑所用的最大时间应取到最小值。它的约束条件为：

$$s. t. \begin{cases} T \leq T' - 3 \\ \sum_{i=1}^M x_{ij} \leq 1 \\ \sum_{j=1}^N x_{ij} \leq 1 \\ T = \frac{S_{ij}}{V} \\ T' = \frac{L_{i32}}{V} \end{cases}$$

上式中， M 为平台数量的最大值， N 为节点数量的最大值。

②过程求解

利用 MATLAB 软件对上式进行求解，得到了 576 组解，其中，存在着一个平台对应多个节点的情况，但是根据实际情况可知，一个平台的警力资源最多只能围堵一个节点，结合附件 2 中全市的出入口所在位置，我们增加约束条件：先对全市的出入口节点进行围堵，若仍出现一个平台对应多个节点的情况，可以利用图与网络模型，将全市的区域划分成方格，以方便求解节点与平台间的最短距离，并按照距离最短原则进行处理。满足该条件之后，再对剩余的节点依照相同原则进行围堵。根据以上约束条件，求解出的结果如下表所示：

表 4：平台围堵节点对应表

平台	节点	平台	节点	平台	节点	平台	节点	平台	节点
1	69	17	41	170	225	323	361	382	402
2	43	18	83	171	231	324	--	383	460
3	55	19	79	172	227	325	325	384	467
4	62	20	86	173	233	326	343	385	448
5	49	93	103	174	220	327	362	386	332

6	61	94	108	175	202	328	328	475	557
7	30	95	115	176	188	372	22	476	545
8	33	96	138	177	177	373	456	477	498
9	35	97	151	178	203	374	432	478	542
10	239	98	164	179	418	375	429	479	578
11	26	99	153	180	--	376	--	480	567
12	25	100	387	181	317	377	--	481	531
13	23	166	264	182	272	378	--	482	489
14	--	167	250	320	349	379	423	483	483
15	238	168	189	321	356	380	396	484	541
16	36	169	254	322	--	381	405	485	572

③结果分析

该市目前现有的交巡警服务平台的设置方案有明显不合理之处。我们对该方案的解决方法为：增加 30 个新平台，这些平台对应的具体位置为：节点 69、84、109、141、163、220、235、246、271、274、279、311、316、358、449、469、530、536、553、557 的所在处。

为了尽快搜捕到嫌疑犯，由于一个交巡警服务平台仅可围堵一个节点，且该市范围内的出入口节点都应先被围堵，因此，我们给出的最佳围堵方案如表 3 所示。

六、模型评价

6.1 模型的优点

- ①我们采用 0-1 规划模型分析道路管辖和围堵逃犯问题，通过数字 0 和 1 的分布得出结论，操作简单，结果直观。
- ②我们使用图与网络模型刻画了该城市的交通路网，利用图与网络算法求解路网中的最短路等特征，简化了求解过程且更有逻辑性。
- ③第二题设计最佳围堵方案时，我们考虑到了在现实情况下就近的平台应该优先封锁市区出入口所在的节点，因此更为贴合实际情况。

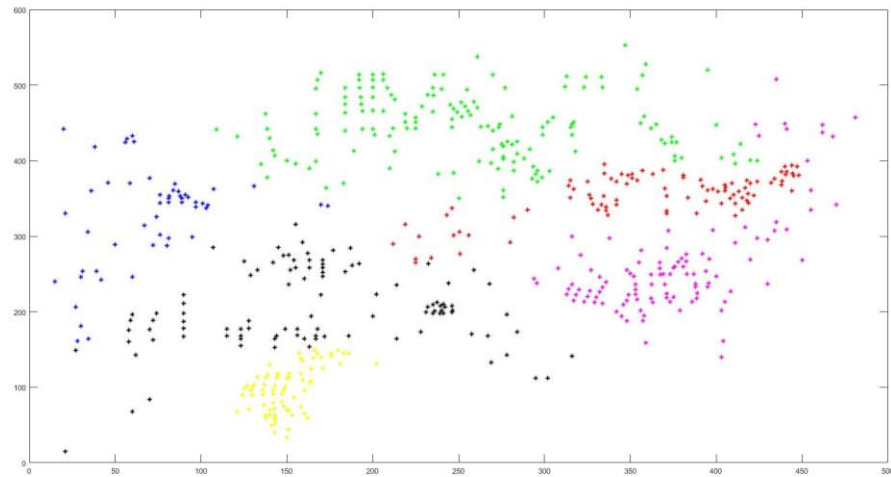
6.2 模型的缺点

- ①在第二问的围堵逃犯问题中，我们为了计算方便，从而假设逃犯逃跑速度和巡警封锁速度相同，但这样的假设有失实际性。
- ②本文大部分采用 0-1 规划模型解决问题，模型较单一。

七、参考文献

- [1]卓金武，MATLAB 在数学建模中的应用，北京：北京航空航天大学出版社，2014.9[M]
- [2]姜启源，数学模型，北京：高等教育出版社，2011.1[M]
- [3]司守奎，数学建模算法与应用，北京：国防工业出版社，2011.8[M]

附录

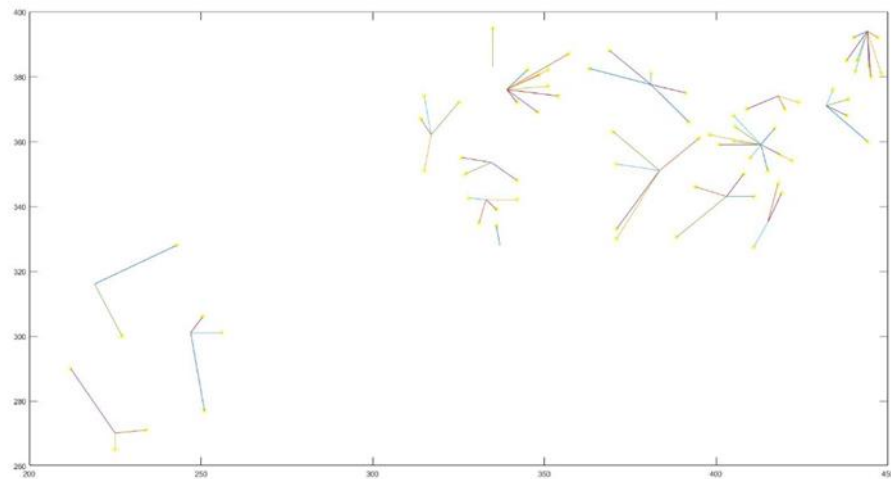


1.

```
A_area = xlsread('data.xls', '全市交通路口节点数据', 'B2:C93');
B_area = xlsread('data.xls', '全市交通路口节点数据', 'B94:C166');
C_area = xlsread('data.xls', '全市交通路口节点数据', 'B167:C320');
D_area = xlsread('data.xls', '全市交通路口节点数据', 'B321:C372');
E_area = xlsread('data.xls', '全市交通路口节点数据', 'B373:C475');
F_area = xlsread('data.xls', '全市交通路口节点数据', 'B476:C583');

areas = [A_area; B_area; C_area; D_area; E_area; F_area];

figure('Name', '交通物理网络图')
plot(A_area(:,1), A_area(:,2), 'r*')
hold on
plot(B_area(:,1), B_area(:,2), 'y*')
plot(C_area(:,1), C_area(:,2), 'g*')
```

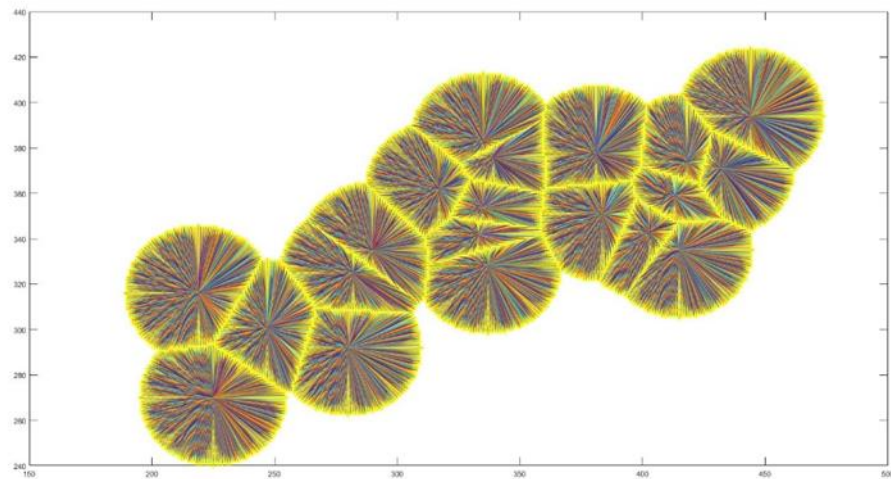



3.

```
A_area = xlsread('data.xls', '全市交通路口节点数据', 'B2:C93');

police_points = A_area(1:20,:);
mapA_x = A_area(21:size(A_area,1),1);
mapA_y = A_area(21:size(A_area,1),2);

count = 1;
for m = 1:numel(mapA_x)
    for k = 1:20
        distance_list(k) = sqrt((mapA_x(m)-
police_points(k,1))^2+(mapA_y(m)-police_points(k,2))^2);
    end
    min_dist = min(distance_list);
    min_place = find(distance_list==min_dist);
    if min_dist <= 30
        plot_x(count) = mapA_x(m);
        plot_y(count) = mapA_y(m);
        min_police(count) = min_place(1);
        count = count+1;
    end
end
plot(plot_x, plot_y, 'y*')
hold on
for k = 1:size(min_police,2)
    plot([plot_x(k) police_points(min_police(k),1)], [plot_y(k)
police_points(min_police(k),2)])
end
hold off
```



4.

```

A_x_limit = [170 530];
A_y_limit = [230 430];

A_area = xlsread('data.xls', '全市交通路口节点数据', 'B2:C93');

police_points = A_area(1:20,:);
mapA_x = A_x_limit(1):1:A_x_limit(2);
mapA_y = A_y_limit(1):1:A_y_limit(2);

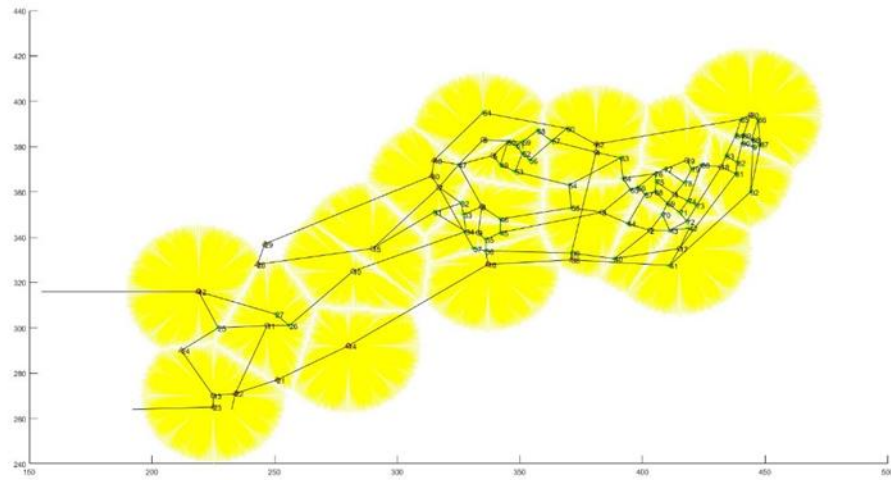
count = 1;
for m = 1:numel(mapA_x)
    for n = 1:numel(mapA_y)
        for k = 1:20
            distance_list(k) = sqrt((mapA_x(m)-
police_points(k,1))^2+(mapA_y(n)-police_points(k,2))^2);
        end
        min_dist = min(distance_list);
        min_place = find(distance_list==min_dist);
        if min_dist <= 30
            plot_x(count) = mapA_x(m);
            plot_y(count) = mapA_y(n);
            min_police(count) = min_place(1);
            count = count+1;
        end
    end
end
plot(plot_x, plot_y, 'y*')
hold on
for k = 1:size(min_police,2)
    plot([plot_x(k) police_points(min_police(k),1)], [plot_y(k)

```

```

police_points(min_police(k),2)]
    end
hold off

```



5.

```

A_x_limit = [170 530];
A_y_limit = [230 430];

A_mainRoad = xlsread('data.xls', '全市区出入口的位置', 'C2:C14');
A_area = xlsread('data.xls', '全市交通路口节点数据', 'B2:C93');
A_road = xlsread('data.xls', '全市交通路口的路线', 'A2:B144');
areas = xlsread('data.xls', '全市交通路口节点数据', 'B2:C583');

police_points = A_area(1:20,:);
mapA_x = A_x_limit(1):1:A_x_limit(2);
mapA_y = A_y_limit(1):1:A_y_limit(2);

count = 1;
for m = 1: numel(mapA_x)
    for n = 1: numel(mapA_y)
        for k = 1:20
            distance_list(k) = sqrt((mapA_x(m)-
police_points(k,1))^2+(mapA_y(n)-police_points(k,2))^2);
        end
        min_dist = min(distance_list);
        min_place = find(distance_list==min_dist);
        if min_dist <= 30
            plot_x(count) = mapA_x(m);
            plot_y(count) = mapA_y(n);
            min_police(count) = min_place(1);
            count = count+1;
        end
    end
end

```

```

        end
    end
    hold on
    for k = 1:size(min_police,2)
        plot([plot_x(k)  police_points(min_police(k),1)], [plot_y(k)
police_points(min_police(k),2)], 'y')
    end
    plot(A_area(1:20,1), A_area(1:20,2), 'bo')
    plot(A_area(21:size(A_area,1),1), A_area(21:size(A_area,1),2),
'g*')
    for k = 1:size(A_area,1)
        text(A_area(k,1), A_area(k,2), sprintf('%d', k))
    end
    plot(A_area(A_mainRoad,1), A_area(A_mainRoad,2), 'r^')
    for k = 1:size(A_road,1)
        plot([areas(A_road(k,1),1) areas(A_road(k,2),1)],
[areas(A_road(k,1),2) areas(A_road(k,2),2)], 'k-')
    end
    hold off

```

6. 求最短路：

```

A_mainRoad = xlsread('data2.xls', '全市区出入口的位置', 'C2:C14');
A_area = xlsread('data2.xls', '全市交通路口节点数据', 'B2:C93');
A_road = xlsread('data2.xls', '全市交通路口的路线', 'A2:B141');

graph_mat = zeros(size(A_area,1));
for k = 1:size(A_road,1)
    graph_mat(A_road(k,1),A_road(k,2)) =
sqrt((A_area(A_road(k,1),1)-
A_area(A_road(k,2),1))^2+(A_area(A_road(k,1),2)-
A_area(A_road(k,2),2))^2);
    graph_mat(A_road(k,2),A_road(k,1)) =
sqrt((A_area(A_road(k,1),1)-
A_area(A_road(k,2),1))^2+(A_area(A_road(k,1),2)-
A_area(A_road(k,2),2))^2);
end

road_net = graph(graph_mat);
shortest_road_mat = distances(road_net);
xlswrite('output1_2.xlsx', shortest_road_mat, 'shortest_road_mat',
'A1');

```

7. 第一题第三问

```

A_area = xlsread('data1_3.xlsx', 1, 'C2:D93');
A_case = xlsread('data1_3.xlsx', 1, 'F2:F93');
police_num = 22;

```

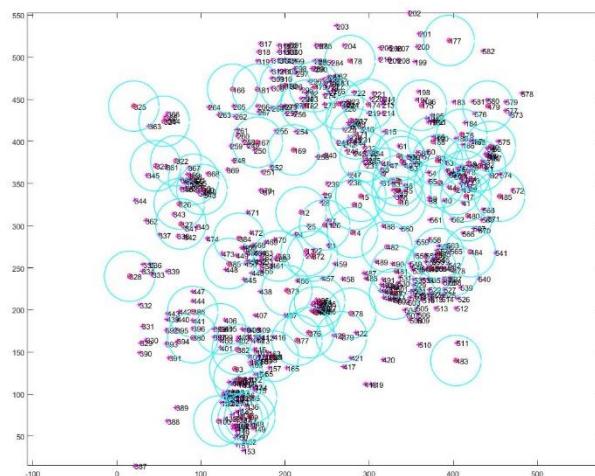
```

for n = police_num:92
    police_points = A_area([1:21 n],:);
    mapA_x = A_area(:,1);
    mapA_y = A_area(:,2);

    count = 1;
    for m = 1:numel(mapA_x)
        for k = 1:police_num
            distance_list(k) = sqrt((mapA_x(m)-
police_points(k,1))^2+(mapA_y(m)-police_points(k,2))^2);
        end
        min_dist = min(distance_list);
        min_place = find(distance_list==min_dist);
        if min_dist <= 30
            plot_x(count) = mapA_x(m);
            plot_y(count) = mapA_y(m);
            min_police(count) = min_place(1);
            count = count+1;
        end
    end
    for k = 1:police_num
        case_sumNum(k) = sum(A_case(find(min_police==k)));
    end
    vary_rate(n) = std(case_sumNum)/mean(case_sumNum);
end

xlswrite('output1_3.xlsx', vary_rate, sprintf('%d',police_num),
'A2');

```



8.

```
clear;
```

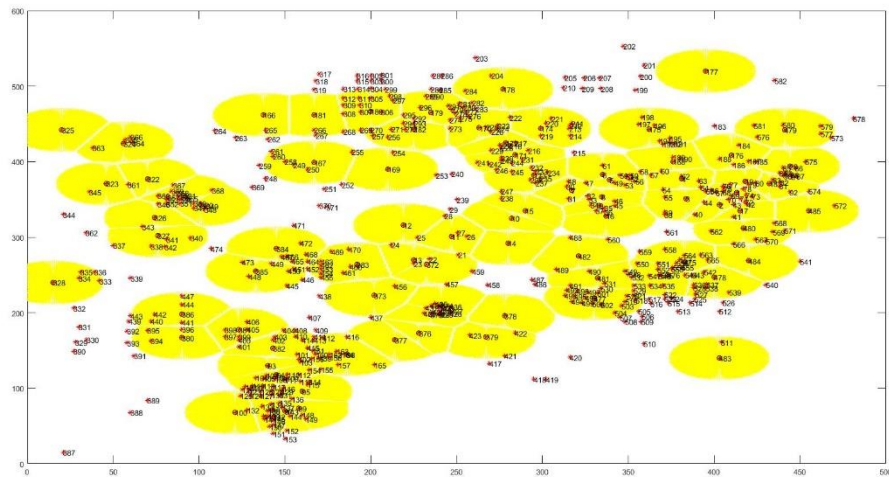
```

areas = xlsread('data.xls', '全市交通路口节点数据', 'B2:C583');
polices = xlsread('data.xls', '全市交巡警平台', 'B2:B81');

figure('Name', '全市交巡警平台的最大管辖范围')
plot(areas(:,1), areas(:,2), 'm*')
hold on
for k = 1:size(areas,1)
    text(areas(k,1),areas(k,2),sprintf('%d',k))
end
for k = polices'
    plot_circle(areas(k,1),areas(k,2),30)
end
plot(areas(polices,1), areas(polices,2), 'ro')
hold off

function [] = plot_circle(x,y,r)
    theta=0:0.1:2*pi;
    Circle1=x+r*cos(theta);
    Circle2=y+r*sin(theta);
    c=[123,14,52];
    plot(Circle1,Circle2,'c','linewidth',1);
    axis equal
end

```



9.

```

clear

x_limit = [0 500];
y_limit = [0 600];

areas = xlsread('data.xls', '全市交通路口节点数据', 'B2:C583');

```



```

polices = xlsread('data.xls', '全市交巡警平台', 'B2:B81');

police_points = areas(polices,:);
map_x = x_limit(1):x_limit(2);
map_y = y_limit(1):y_limit(2);

count = 1;
for m = 1: numel(map_x)
    for n = 1: numel(map_y)
        for k = 1: size(police_points,1)
            distance_list(k) = sqrt((map_x(m)-
police_points(k,1))^2+(map_y(n)-police_points(k,2))^2);
        end
        min_dist = min(distance_list);
        min_place = find(distance_list==min_dist);
        if min_dist<=30
            plot_x(count) = map_x(m);
            plot_y(count) = map_y(n);
            plot_police(count) = min_place(1);
            count = count+1;
        end
    end
end
for k = 1: count-1
    plot([plot_x(k) police_points(plot_police(k),1)], [plot_y(k)
police_points(plot_police(k),2)], 'y')
    hold on
end
plot(areas(:,1),areas(:,2),'r*')
for k = 1: size(polices,1)
    plot(areas(polices(k),1),areas(polices(k),2),'ko')
end
for k = 1: size(areas,1)
    text(areas(k,1),areas(k,2),sprintf('%d',k))
end
hold off

```

10. 第二题第一问的新增平台的代码

```

clear;
areas = xlsread('data.xls', '全市交通路口节点数据', 'B2:C583');
polices = xlsread('data.xls', '全市交巡警平台', 'B2:B81');
cases = xlsread('data.xls', '全市交通路口节点数据', 'E2:E583');
police_num = 81;

polices_new = [polices'];

```

```

    police_points = areas(polices_new,:);
    map_x = areas(:,1);
    map_y = areas(:,2);
    count = 0;
    count_map = 0;
    for m = 1:numel(map_x)
        count_map = count_map+1;
        for k = 1:size(police_points,1)
            distance_list(k) = sqrt((map_x(m)-
police_points(k,1))^2+(map_y(m)-police_points(k,2))^2);
        end
        min_dist = min(distance_list);
        min_place = find(distance_list==min_dist);
        if min_dist > 30
            count = count+1;
            count_maps(count) = count_map;
        end
    end
    clearvars distance_list min_dist min_place;
    for n = count_maps
        police_points = areas([polices_new n],:);
        map_x = areas(:,1);
        map_y = areas(:,2);
        count = 1;
        for m = 1:numel(map_x)
            for k = 1:size(police_points,1)
                distance_list(k) = sqrt((map_x(m)-
police_points(k,1))^2+(map_y(m)-police_points(k,2))^2);
            end
            min_dist = min(distance_list);
            min_place = find(distance_list==min_dist);
            if min_dist<=30
                min_police(count) = min_place(1);
                count = count+1;
            else
                min_police(count) = 0;
                count = count+1;
            end
        end
        for k = 1:police_num
            case_sumNum(k) = sum(cases(find(min_police==k)));
        end
        vary_rate(n) = std(case_sumNum)/mean(case_sumNum);
    end
end

```

```

xlswrite('output2_1.xlsx',      [[1:size(areas,1)]'      vary_rate'],
sprintf('%d', police_num), 'A1');

```

11. 求全市的最短路

```

A_mainRoad = xlsread('data2.xls', '全市区出入口的位置', 'C2:B929');
A_area = xlsread('data2.xls', '全市交通路口节点数据', 'B2:C583');
A_road = xlsread('data2.xls', '全市交通路口的路线', 'A2:B929');

graph_mat = zeros(size(A_area,1));
for k = 1:size(A_road,1)
    graph_mat(A_road(k,1),A_road(k,2)) =
sqrt((A_area(A_road(k,1),1)-
A_area(A_road(k,2),1))^2+(A_area(A_road(k,1),2)-
A_area(A_road(k,2),2))^2);
    graph_mat(A_road(k,2),A_road(k,1)) =
sqrt((A_area(A_road(k,1),1)-
A_area(A_road(k,2),1))^2+(A_area(A_road(k,1),2)-
A_area(A_road(k,2),2))^2);
end

road_net = graph(graph_mat);
plot(road_net)
shortest_road_mat = distances(road_net);
xlswrite('output2_2.xlsx',  shortest_road_mat,  'shortest_road_mat',
'A1');

```

12. 最后的围堵

```

clear;
areas = xlsread('data.xls', '全市交通路口节点数据', 'B2:C583');
polices = xlsread('data.xls', '全市交巡警平台', 'B2:B81');
road = xlsread('data2.xls', '全市交通路口的路线', 'A2:B929');

police_points = areas(polices,:);
map_x = areas(:,1);
map_y = areas(:,2);
count = 1;
for m = 1: numel(map_x)
    for k = 1:size(polices,1)
        distance_list(k) = sqrt((map_x(m)-
police_points(k,1))^2+(map_y(m)-police_points(k,2))^2);
    end
    min_dist = min(distance_list);
    min_place = find(distance_list==min_dist);
    plot_x(count) = map_x(m);
    plot_y(count) = map_y(m);
    min_police(count) = min_place(1);
end

```

```

        count = count+1;
    end
    graph_mat = zeros(size(areas,1));
    for k = 1:size(road,1)
        graph_mat(road(k,1),road(k,2)) = sqrt((areas(road(k,1),1)-
areas(road(k,2),1))^2+(areas(road(k,1),2)-areas(road(k,2),2))^2);
        graph_mat(road(k,2),road(k,1)) =
graph_mat(road(k,1),road(k,2));
    end
    road_net = graph(graph_mat);
    shorestPath_mat = distances(road_net);
    count = 0;
    for k = 1:numel(polices)
        policesId_manage = find(min_police==k);
        for m = 1:numel(policesId_manage)
            shorestPath(polices(k),policesId_manage(m)) =
shorestPath_mat(polices(k),policesId_manage(m));
            if shorestPath(polices(k),policesId_manage(m)) <
shorestPath_mat(30,policesId_manage(m))+30
                count = count+1;
                block_points(count) = policesId_manage(m);
                block_police(count) = polices(k);
            end
        end
    end
    for k = 1:size(polices,1)
        shorestPath(polices(k), shorestPath(polices(k),:)==0) = 1000;
        if
numel(find(shorestPath(polices(k),:)==min(shorestPath(polices(k),:))
)) == 1;
            police_block(k) =
find(shorestPath(polices(k),:)==min(shorestPath(polices(k),:)));
        else
            police_block(k) = 0;
        end
    end
    % xlswrite('output2_2.xlsx', shorestPath, 'shorestPath', 'B2');
    % xlswrite('output2_2.xlsx', [block_police block_points],
'block_points', 'A1');
% xlswrite('output2_2.xlsx', [polices police_block], 'police_block',
'A1');

```