2011 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。

我们参赛选择的题号是（从 A/B/C/D 中选择一项填写）: B

我们的参赛报名号为（如果赛区设置报名号的话）: 20022012

所属学校（请填写完整的全名）: 长沙学院

参赛队员 (打印并签名) ：1. 陆石杠

2. 曹外游

3. 胡　豆

指导教师或指导教师组负责人　(打印并签名): 张逵

日期: 2011 年 9 月 12 日

赛区评阅编号（由赛区组委会评阅前进行编号）:

2011 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

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。然而现有的警务资源相当有限，本文针对现有的警务资源分配与调度问题做了较为深层次的研究。

针对第一问，本文建立了三个模型来分别对管辖范围分配、警务调度和工作量不均衡等问题进行了求解。为解决警员最快到达事发地，建立基于聚类和图论分析的最短路径模型。对于 A 市的管辖范围分配，由于出警速度要求是尽可能在 3 分钟内赶到事发路口，根据路程最短原则，判断该路口的归属，通过 MATLAB 计算分析得各个服务平台的管辖范围(见表 1、2)，其中有 6 个点的出警时间超过了三分钟。针对 A 区交通要道完成快速封锁，即要求所有警力分配方案中，最晚到达封锁路口的警力，费时最短，建立极大极小和总体路程最小两类非方阵指派模型。运用行优先选取算法与匈牙利算法求解出最快封锁调配方案的最优解，其所用最短时间为 8.155 分钟，所有满足最慢到达时间为 8.0155 分钟的解都为可行解（见表 3）。在为解决出警时间过长的条件下，运用最短路径模型，确定增加警务平台的最少个数为 4 个，分别为 29,40,48,90，工作总量方差总和为 168.82；为使各个平台工作量均衡，建立 0-1 规划模型，确定 4 个警务平台的具体位置（见表 4）。

针对第二问，首先，本文从设置交巡警服务平台的原则和任务出发，综合评价现有设置的合理性，得出不合理点为处警时间过长节点数、平台工作量、人口密度与平台数；为解决这些不合理性，建立 0-1 规划模型，使各区的交巡警平台在处警时间和平台工作量上达到最优（见表 5）。然后，为解决题中的具体围堵案例，建立元素树权传递算法；在假定逃犯逃离速度与警车速度相同的情况下，求解出具体的最佳围堵方案为：动用 20 个交巡警服务平台，用时 8.79 分钟实现围堵（见表 6）。

关键词：最短路径模型、非方阵指派问题、0-1 规划模型、元素树权传递算法、匈牙利算法

1 问题重述

1.1 背景资料

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。由于现有警务资源是有限的，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重要问题。

1.2 待解决问题

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）对于城区 A 根据相关的数据信息。请为各交巡警服务平台分配管辖范围，使其尽量能在 3 分钟内有交巡警（警车的时速为 60km/h）到达其管辖范围的事发地。对于突发事件，怎样调度各平台警务资源，使其最快速度封锁进出该区的 13 个交通要道且一个平台警力最多封锁一个路口。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 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。

2 问题假设

1、假设交巡警的管辖范围只考虑路口交点
2、假设每个交巡警服务平台的职能和警力配备基本相同
3、假设警车速度恒定
4、忽略出巡时其他因素的影响
5、忽略道路的方向性
6、相邻两节点间的路程为两点坐标的距离。
7、不考虑其他因素对车速的影响。

3 符号说明

符号	说明
V	交通路口顶点
P_{ij}	第 i 个节点由第 j 个平台管辖则为 1，否则为 0
Q_i	第 i 个节点日均案发数
C_{ij}	第 i 个节点由第 j 个平台最短距离
S	城区所有节点的日均案发次数
a	增加的交巡警服务平台个数

注：部分符号见文中说明

4 问题分析

4.1 问题重要性分析

如今城市的迅速发展，给交通和管理带来了巨大压力。现在警察都肩负着多重的职责任务重大，但现有的警力资源相当有限。对于现有的交通要道和路口，存在着很高的事故率和巨大的交通拥堵压力，所需要的人员较多。而且全城的服务覆盖率也变得越来越重要，能够在短时间内给与人民最及时的帮助，让现有的警务资源的分配与调度问题变得越来越重视。

4.2 问题解决思路

在第一问中，只针对 A 城区进行分析，其中存在三个问题需要解决：

首先，考虑 A 城区的管辖范围分配问题，由于要求尽可能在 3 分钟内到达事发地，知道警车速度为 60km/h，可以将问题转化求解路口到警务平台的最短路问题，路程最短说明所用时间一定是最少的，通过图论的最短路求解方法求解出结果。

然后，分析 A 城区的警务调度问题，题目要求怎样调度能使 20 个警务平台封锁 13 个要道路口的速度最快，即可以看作完成全面封锁的时间最少问题，也就是所有分配方案中，最慢达到要道口的警务人员所用时间最短的解（可能存在多解），问题可转化成了求极大极小值和总体最小 2 类非方阵指派问题。

最后，研究怎么增加平台和调整，能使交巡警服务平台的工作量均衡，并且出警时间不会过长（由题，定义出警时间超过三分钟才为出警时间过长）。通过尽量减少超出三分钟路口点的数量来达到解决问题的目的，工作量的均衡性考虑工作量的方差最小，从而建立工作量最小的优化模型。

在第二问中，针对全市区进行分析，其中存在两个问题需要解决：

首先，按照设置交巡警服务平台的原则和任务，即：

1、根据管区道路交通流量、拥堵状况、治安复杂情况、发案量高低，科学确定平台管控区域；

2、快速处警原则：城区接警后确保快速到达现场；

3、方便与安全原则：按照醒目、规范，方便群众和确保安全的原则，科学设置平台。

平台设置在遵循上述三大原则的基础上，应当结合辖区地域特征、人口分布、交通状况、治安状况和未来城市发展规划等实际情况，在充分考虑现有警力和财力并确保安全的条件下，科学确定平台的数量和具体位置。

再根据题意，从快速出警、人口分布、交通状况、现有警力和财力，这五方面对全市交巡警服务平台设置方案的合理性进行评价。首先对该市各区内现有设置方案合理性评价，即分别对 A、B、C、D、E、F 这六个区进行评价，然后再评价各区相互之间的分配合理性。由不合理性，建立 0-1 规划优化模型，求解出各区的最佳交巡警服务平台的设置。

然后，题目要求在发生重大事故三分钟后，全市出动警务力量进行围堵封锁的最佳方案。对此可以理解为包围圈最小且调动人员速度最快的围堵是最佳的封锁方案。由于逃犯的逃走路线不定，且已经逃离案发现场三分钟后，警察再进行围堵，即存在一定范围的围堵，只考虑歹徒想越逃越远，这会出现动态的变化范围求解问题，那么警察要想围住必须在逃犯在一定时间内到达某一可能节点前，警察都已经赶到即可，建立元素树权传递算法求解。

5 模型的建立与求解

5.1 基于最短路问题的交巡警服务平台管辖范围分配

5.1.1 模型的建立

为交巡警服务平台合理的分配管辖范围，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。据题意服务平台的管辖范围即其所管辖的节点集合，因此问题也可理解为在非服务平台路口有突发事件时，尽快有交巡警到达该路口，服务平台是固定不动的，转化为距离即该路口应当归属于距离其最近的服务平台管辖，问题转化为图论中的最短路问题。

构造加权无向图 G(V,E)，顶点集$V=\{v_1,...,v_i\}\cup\{u_1,...,u_j\}$，$i$ 为服务平台数，j 为普通节点数，每个顶点代表城市交通路口，v_i 代表第 i 个服务平台，u_j 代表第 j 个普通节点，边集 E 代表交通路口的路线集。

问题转化为，对图 G 中的每一个顶点 u_j，均需找到对其到路程最近的顶点 v_i，即在顶点 v_i 中寻找到顶点 u_j 路程最短的点。只须 u_j 到所有的顶点 v 的最短路程，通过比较即可得到距离其最短路程的顶点。

Floyd 算法：求任意两点间的最短路.

D(i,j)：*i* 到 *j* 的距离.

R(i,j)：*i* 到 *j* 之间的插入点.

输入: 带权邻接矩阵 *w(i,j)*

(1)赋初值：

对所有 i,j, $d(i,j)\leftarrow w(i,j)$, $r(i,j)\leftarrow j$, $k\leftarrow 1$

(2) 更新 $d(i,j)$, $r(i,j)$

对所有 i,j，若 $d(i,k)+d(k,j)<d(i,j)$，则 $d(i,j)\leftarrow d(i,k)+d(k,j)$, $r(i,j)\leftarrow k$

(3) 若 $k=v$，停止．否则 $k\leftarrow k+1$，转（2）.

对顶点 u_j 与 v_i 中的所有顶点使用 Floyd 算法求解其最短路程为 l_{ij}，若 $l_{kj}=\min_i\{l_{ij}\},k\in(1,i)$，则将顶点 u_j 归属于顶点 v_k。

对所有 u_j 顶点按上述算法进行计算，得到顶点 v_i 的集合 $vu_i=\{u_{r,...,}u_s\}$，此即为服务平台 v_i 的管辖范围。

5.1.2 模型的求解：

城区 A 总节点数为 92，服务平台数为 20，顶点集$V=\{v_1,...,v_{20}\}\cup\{u_1,...,u_{72}\}$，各交通路口即构成网络图 G ,如图 1 根据上述模型及 Floyd 算法，使用 matlab 编程（程序见附件 1,2)，求解即可得到每个服务平台的管辖范围如下：

表 1 城区 A 服务平台管辖范围

服务平台	管辖范围
1	1，67，68，69，71，73，74，75，76，78
2	2，39，40，43，44，70，72
3	3，54，55，65，66
4	4，57，60，62，63，64
5	5，49，50，51，52，53，56，58，59
6	6
7	7，30，32，47，48，61
8	8，33，46
9	9，31，34，35，45

10	10
11	11，26，27
12	12，25
13	13，21，22，23，24
14	14
15	15，28，29
16	16，36，37，38
17	17，41，42
18	18，80，81，82，83
19	19，77，79
20	2084，85，86，87，88，89，90，91，92

由表发现 6,10,14 三个服务平台只管辖本身，分析可知这主要是由于，这里是依据事故发生时巡警到达时间最短进行管辖范围分配的，而这些服务平台距离其他顶点均较远。

要求在有突发事件时，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，转化为路程为在 3 公里内至少有一个服务平台。计算中发现存在有 6 个普通节点，距离最近的服务平台也超过了 3 公里，具体情况如下：

表 2 三分钟内无法到达节点

节点	最近服务台路程
28	47.518
29	57.005
38	34.059
39	36.822
61	41.902
92	36.013

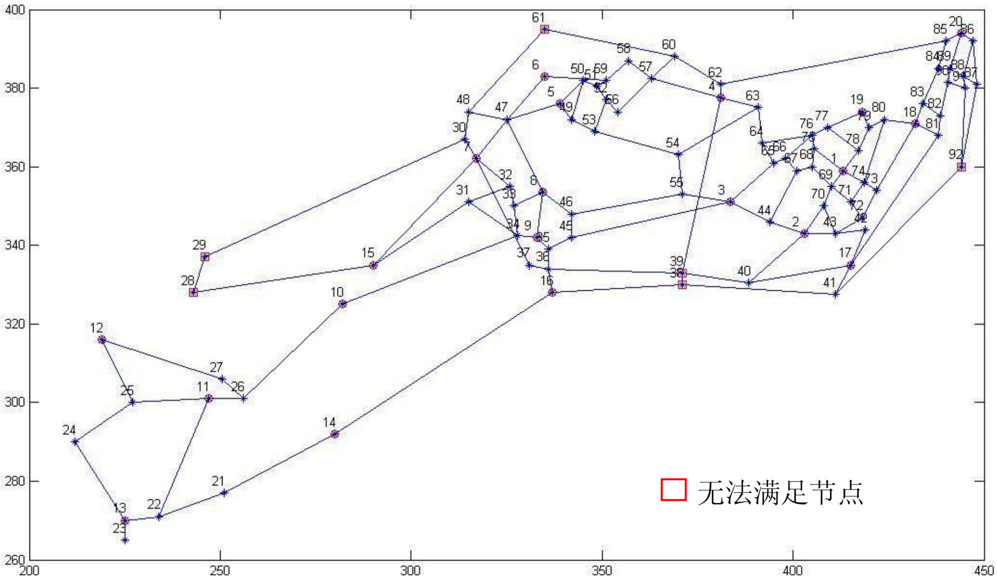


图 1 城区 A 交通网络图

从图 1 中也可看出这些点距离其他节点均较远，而服务平台是固定不动的，因此无

论怎样也无法在三分钟之内到达。但整个模型算法均将路程最短作为管辖范围分配的唯一依据，也即满足了在事件发生时，巡交警集尽量在 3 分钟之内能够到达事发地。。

5.2 基于非方阵指派问题的调度模型

5.2.1 模型建立

本文将 nn 方阵极大极小与总体极小 2 类指派问题的矩阵作业法推广到 mn 非方阵情形（假定 $m \geq n$）。mn 非方阵极大极小指派问题在数学上可以用 mn 个决策变量描述。令 x_{ij} 为 0-1 决策变量，即 $x_{ij}=1$ 表示第 j 个警务平台出警到第 i 个节点，$x_{ij}=0$ 表示第 j 个警务平台不出警到第 i 个节点。c_{ij} 表示第 i 个节点到第 j 个警务平台的路程，则 c_{ij} 构成了以知的 $m \times n$ 路程矩阵。从而建立极大极小指派问题的数学模型：

$$\min(z)=\max\{c_{ij}x_{ij}\}$$

$$s.t\quad \begin{cases} \sum_{j=1}^{n} x_{ij}=1 \\ \sum_{j=1}^{m} x_{ij}\leq 1 \\ x_{ij}=0\text{或}1 \qquad i\in\{1,2,\ldots\ldots,n\},\ j\in\{1,2,\ldots.m\}。 \\ \sum_{j=1}^{m}\sum_{i=1}^{n} x_{ij}=n \end{cases} \tag{1}$$

这是一个非线性整数规划问题，其中目标函数是非线性的，引进 1 个实变量 y_{n+1}，就可以将上述极大极小指派问题改进成混合整数线性规划问题，等价的混合整数线性规划为

$$\min z=y_{n+1}$$

$$s.t\quad \begin{cases} \sum_{j=1}^{n} x_{ij}=1 \\ \sum_{j=1}^{m} x_{ij}\leq 1 \\ x_{ij}=0\text{或}1 \qquad i\in\{1,2,\ldots\ldots,n\},\ j\in\{1,2,\ldots.m\}。 \\ \sum_{j=1}^{m}\sum_{i=1}^{n} x_{ij}=n \\ y_{n+1}-c_{ij}x_{ij}\geq 0 \end{cases} \tag{2}$$

其中只有 1 个实变量 y_{n+1}，但增加了 mn 个约束不等式。

$m \times n$ 非方阵总体极小指派问题的数学模型由如下线性整数规划表述：

$$\min z=\sum_{j=1}^{m}\sum_{i=1}^{n} x_{ij}c_{ij}$$

$$s.t\quad \sum_{j=1}^{n} x_{ij}=1$$

$$\sum_{j=1}^{m} x_{ij}\leq 1$$

$$x_{ij}=0或1 \qquad i\in\{1,2,.....,n\},\ j\in\{1,2,....m\}。 \tag{3}$$

$$\sum_{j=1}^{m}\sum_{i=1}^{n}x_{ij}=n$$

5.2.2 极大极小指派问题的行优先选取算法

不失一般性， 假定极大极小指派问题中 m > n；否则，考虑路程矩阵的转置．将 n×n 方阵极大极小指派问题的矩阵作业解法推广到 m×n 非方阵情形的基本思想是：在 m×n 路程矩阵中适当取 n 行，使得 n×n 代价矩阵极大极小指派问题的解就是原 m×n 路程矩阵极大极小指派问题的解．而 n×n 路程方阵极大极小指派问题的解可以用矩阵作业法有效地求得．把 mn 个路程 c_{ij} 按由小到大的顺序排成一行，由于其中可能有相等者， 因此这 mn 个路程由小到大为$h\in[1;mn]$个不同级别的实数 $R_1;R_2;....;R_h$（$R_1<R_2<...<R_h$），等于R_k（$1\le k\le h$）的路程c_{ij}有 s_k个．显然，$s_k\ge 1$ 且 $\sum_{k=1}^{h}s_k=mn$．又记 R 为一个不小于R_h 的数， 它也可以看作∞符号．如果 h = 1， 即所有$c_{ij}=R_h$，则式 (2) 显然有 $C_m^n n!$ 个解，每个解的目标函数都是R_h，因此不妨设 h > 1．

令 $R_i^c=\min\limits_{1\le j\le n}\{c_{ij}\}$， $R_{\max}^c=\max\limits_{i}\{R_i^c\}$，$i$ =1,2. . . ,n; $R_j^r=\min\limits_{1\le i\le n}\{c_{ij}\}$；$j$ =1,2, . . . ,m, 而 $R_{\max}^r=\max\{R_{j1}^r,R_{j2}^r,.....R_{js}^r\mid n\le s\le m,R_{jk}^r>R_{\max}^r,j^k\ne j^1,...,j^s\}$， $R_{\max}^o=\max\{R_{\max}^r,R_{\max}^c\}$．

按题意（m > n)，每个节点由且只由 1 个警务平台管理．最后完成任务的代价不小于 $R_{\max}^c$．又要从 m 个警务平台中挑出 n 个，使得每个警务平台围堵且只围堵一个节点；按题意，完成任务路程最大的不应该被指派，$R_{\max}^r$ 的定义满足这些要求，故最后到达指定节点的路程也不小于 $R_{\max}^r$．因而最优目标函数值$z^*\ge R_{\max}^o$．

记 $R_{\max}^o$在 h 个代价级别中的排序是 k，即 $R_{\max}^o=R_k$． 在路程矩阵$C_{mn}=(c_{ij})$中将所有大于R_k（$R_k=R_{\max}^o$）的元素用 R 覆盖，叫作 C_{mn} 的基本覆盖．由$R_{\max}^r$ 的定义，C_{mn}基本覆盖后存在 s 行 n 列的子矩阵C_{sn}，使得$R_{\max}^o$ 为C_{sn} 的最大非覆盖元素且C_{sn}中每行每列至少有 1 个非覆盖元素．显然，所有非覆盖元素中的任何一种可行分配 G（如 果存在的话）都是极大极小模型的 1 个最优解．

由于$n\le s\le m$，所以C_{sn} 中有 C_n^s个 n×n 基本覆盖方阵可以应用矩阵作业法求解． 如果这C_n^s 个 n×n 基本覆盖方阵都没有可行解，则最优解只能在更高一级的覆盖中寻找．

5.2.3 基于非方阵指派问题的调度模型求解

针对 A 城区的的具体情况，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，即需要将封锁$m=13$个路口的任务，指派给$n=20$个服务平台。问题可视为非方阵指派问题，因此可建立上述数学模型，并可采用行优先选取算法对模型进行求解。本文采用 matlab 编程（程序见附件 3）进行模型的求解，得到 1 种该区服务平台警力合理的调度方案如下：

表 3 封锁调度方案

出口节点	**12**	**14**	**16**	**21**	**22**	**23**	**24**	**28**	**29**	**30**	**38**	**48**	**62**
指派平台	10	16	2	11	12	14	13	15	7	5	1	4	3

该调度方案下，最后背封锁的出口为 29，整个城区被完全封锁所需时间为8.0155 分钟。

5.3 确定增加交巡警服务平台的个数和位置模型建立与求解

要求依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，拟在该区内再增加若干平台。在前述确定的分配方案下，求得各个服务平台的工作总量如下：

表 4 各服务平台工作量

平台	1	2	3	4	5	6	7	8	9	10
总发案率	10.3	9.7	5.6	6.6	9.7	2.5	9.6	5	8.2	1.6
平台	11	12	13	14	15	16	17	18	19	20
总发案率	4.6	4	8.5	2.5	4.8	5	5.3	6.1	3.4	11.5

可以方差描述各平台工作量的均衡性，求得方差为 160.198。

又由表 6 可知存在 6 个节点无法在三分钟之内到达，出警时间过长。在不变动现有平台位置的情况下，若想在缩短过长出警时间，只能增加服务平台的数目。

而好的分配方案应当在顾及过长出警时间的情况下，把握好工作量的均衡性，但考虑到交巡警的职责和服务性，增加平台吸收工作量，因此在满足尽量快速出警的下，再考虑工作量的均衡性。

因此为确定增加节点数及位置确立如下算法：

Setp1： 对出警时间过长节点，搜寻所有满足 3 分钟可到达节点。

Setp2：计算得到所有节点突发事件均能 3 分钟出警至少需要增加节点数。

Setp3：结合增加数目得到可能分配方案，使用管辖范围分配模型确定管辖范围。

Setp4：求得新方案中工作量方差最小者即为所求方案。

上述算法简单易于计算，但其是基于原有分配，而且并不能够得到完全均衡的分配方案，在增加平台个数确定的情况下，要使得交巡警服务平台的工作量均衡，可建立以工作量的方差最小的优化模型如下：

1、目标函数

工作量方差总和最小，即：

$$\min z = \sum_{j=1}^{20+a}\left(\sum_{i=1}^{92} Q_i P_{ij} - S\right)$$

2、约束条件

$$s.t\quad \begin{cases} \sum_{j=1}^{20+a} P_{ij} = 1 \\ \sum_{i=1}^{92} P_{ij} \ge 1 \\ P_{ij}C_{ij} \le C_{ik}\text{，（其中 } k \ne j\text{）} \\ S = \sum_{i=1}^{92} Q_i /(20+a); \\ a = [2,5] \end{cases}$$

注： P_{ij} =1 表示第 i 个节点由第 j 个交巡警服务平台管辖。P_{ij} =0 表示第 i 个节点不由第 j 个交巡警服务平台管辖。
Q_i 表示第 i 个节点日案发次数。S 表示所对应区所有节点的平均日案发次数。
a 表示增加的交巡警服务平台个数。
C_{ij} 表示第 i 个节点由第 j 个交巡警服务平台的路程。

同样依据本题具体情况，依据上述算法使用 matlab 编程求解得到增加平台数目为 4 个，分别为 29,40,48,90，工作总量方差总和为 168.82

5.4 全市现有交巡警服务平台设置方案合理性评价

根据题意，从快速处警、人口分布、交通状况、现有警力和财力，这五方面对全市交巡警服务平台设置方案的合理性进行评价。

首先，对该市各区区内现有设置方案合理性评价，即分别对Ａ、Ｂ、Ｃ、Ｄ、Ｅ、Ｆ这六个区进行评价，然后再评价各区相互之间的分配合理性。

5.4.1 对Ａ区交巡警服务平台设置方案合理评价

5.4.1.1 对快速处警进行评价

本文设为若任一节点处发生交通事故，交巡警能在三分钟内到达现场则认为该节点能快速处警；反之则不能。

由以上定义，即用区域内未能快速处警的发案量与总发案量的比值（下文称之为未处警率），来评价区域内快速处警的能力大小。

运用 matlab 对附件二中的数据进行处理，计算得到Ａ区中未能快速处警的节点数为 6。再根据每个节点的日案发次数，从而得出日未处警量为 6.7．而总日案件量为 124.5。从而未处警率为 5.38%。根据小概率事件的评价标准（即小于 0.03%为小概率事件）。显然 5.38%>>0.03%。即Ａ区在快速处警设置上明显不合理。

5.4.1.2 对Ａ区案发量区域特性与平台区域设置评价

若把任何一个节点与到该节点的路程小于等于３ｋｍ的其他节点连接在一起的树杈型网络看作是这一个节点的一个域（下文中称这一个域为为节点域，这一个点位中心节点）。则在这一个域里的日发案量为各节点日案发量的总和（下文称为节点域量）。各个节点域内包含的交巡警服务平台个数称为平台量，平台量与节点域量得比值称为平案比。对附件二中，Ａ区得数据进行处理，得到各个节点域量、平台量和平案比部分值如下所示：

中心节点	平台量	节点域量	平案比
70	5	31.2	0.160256
68	4	28.4	0.140845
66	5	27	0.185185
76	4	27	0.148148
62	1	8.4	0.119048
54	1	8.1	0.123457
22	1	7.4	0.135135
29	0	2.7	0
14	1	2.5	0.4
61	0	2	0
10	1	1.6	0.625

A

中心节点	平台量	节点域量	平案比
10	1	1.6	0.625
14	1	2.5	0.4
15	1	3.7	0.27027
12	1	4	0.25
89	2	17.6	0.113636
90	2	17.6	0.113636
4	1	9.2	0.108696
38	0	4.3	0
39	0	4.3	0
61	0	2	0
92	0	6	0

B

注：A 表是案节点域量降序排列，B 表是按平案比降序排列。

由以上 A 表可以看出，节点域量最大的节点是 70，其次为 68、66、76。而最小的是节点 10 其次为 61、14、29。其中节点域量越大表示该域日案发量越多。又由题知只有节点 1~20 设有交巡警服务平台。因此 A 区的交巡警服务平台未考虑中心节点的设置。

由 B 表可以看出，平案比最大的节点是 10，其次为 14、15、12；而最小的节点依次为 38、39、61、92。其中平案比越大表示交巡警服务平台日处理的案件量少。因此，有 B 表分析得知，日处理案件量少的节点设置了较多的交巡警服务平台。

综上可得，交巡警区域设置上明显不合理。

5.4.2 对Ｂ、Ｃ、Ｄ、Ｅ和Ｆ区交巡警服务平台设置方案合理评价

5.4.2.1 对快速处警进行评价

同Ａ区的评价方法，得出Ｂ区的未覆盖率为 5.27%,C 区的未覆盖率为 23.45%，Ｄ区的未覆盖率为 13.27%，Ｅ区的未覆盖率为 28.31%，Ｆ区的未覆盖率为 25.18%.从而对于Ｂ、Ｃ、Ｄ、Ｅ、Ｆ区的未覆盖率都远远大于 0.03%，即它们在快速处警情况下，设置都明显不合理。

5.4.2.2 对Ｂ、Ｃ、Ｄ、Ｅ和Ｆ区案发量区域特性与平台区域设置评价

同Ａ区的评价方法，得Ｂ区节点域量最大的节点是 134，其次为 135、99、129。而最小的是节点 153 其次为 152、151、124。其中节点域量越大表示该域日案发量越多。又由题知只有节点 93~100 设有交巡警服务平台。因此 B 区的交巡警服务平台未充分考虑中心节点的设置。

平案比最大的节点是 129，其次为 130、128、117；而最小的节点依次为 165、153、100、93。其中平案比越大表示交巡警服务平台日处理的案件量少。

综上可得，B 区交巡警区域设置上明显不合理。

C 区节点域量最大的节点是 223，其次为 172、228、227。而最小的是节点 319 其次为 318、317、316。其中节点域量越大表示该域日案发量越多。又由题知只有节点 166~182 设有交巡警服务平台。因此 C 区的交巡警服务平台未充分考虑中心节点的设置。

平案比最大的节点是 231，其次为 179、171、274；而最小的节点依次为 177、264、253、239。其中平案比越大表示交巡警服务平台日处理的案件量少。

综上可得，C 区交巡警区域设置上明显不合理。

Ｄ区节点域量最大的节点是 358，其次为 353、346、347。而最小的是节点 330 其次为 329、337、339。其中节点域量越大表示该域日案发量越多。又由题知只有节点 320~328 设有交巡警服务平台。因此 D 区的交巡警服务平台未充分考虑中心节点的设置。

平案比最大的节点是 358，其次为 352、356、355；而最小的节点依次为 332、362、344、337。其中平案比越大表示交巡警服务平台日处理的案件量少。

综上可得，D 区交巡警区域设置上明显不合理。

Ｄ区节点域量最大的节点是 358，其次为 353、346、347。而最小的是节点 330 其次为 329、337、339。其中节点域量越大表示该域日案发量越多。又由题知只有节点 320~328 设有交巡警服务平台。因此 D 区的交巡警服务平台未充分考虑中心节点的设置。

平案比最大的节点是 358，其次为 352、356、355；而最小的节点依次为 332、362、344、337。其中平案比越大表示交巡警服务平台日处理的案件量少。

综上可得，D 区交巡警区域设置上明显不合理

E 区节点域量最大的节点是 376，其次为 375、424、429。而最小的是节点 474 其次为 471、469、464。其中节点域量越大表示该域日案发量越多。又由题知只有节点 373~386 设有交巡警服务平台。因此 E 区的交巡警服务平台未充分考虑中心节点的设置。

平案比最大的节点是 375，其次为 424、429、430；而最小的节点依次为 387、390、420、378。其中平案比越大表示交巡警服务平台日处理的案件量少。

综上可得，E 区交巡警区域设置上明显不合理

F 区节点域量最大的节点是 543，其次为 544、542、532。而最小的是节点 582 其次为 578、523、574。其中节点域量越大表示该域日案发量越多。又由题知只有节点 475~385 设有交巡警服务平台。因此 F 区的交巡警服务平台未充分考虑中心节点的设置。

平案比最大的节点是 544，其次为 521、545、543；而最小的节点依次为 582、574、541、575。其中平案比越大表示交巡警服务平台日处理的案件量少。

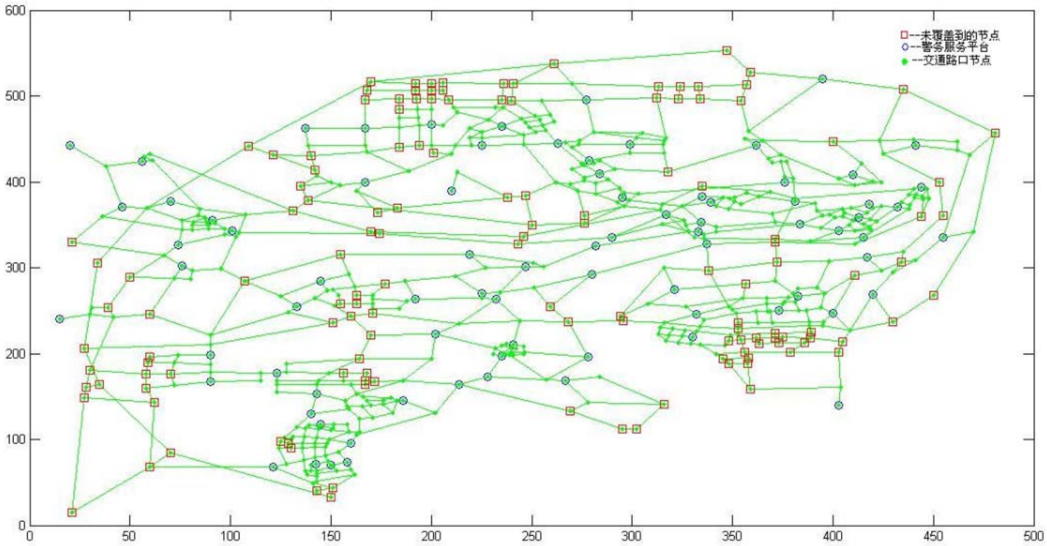
综上可得，F 区交巡警区域设置上明显不合理

5.4.3 对全市快速处警、人口分布平台区域设置评价

由附件 1 可知，A、B、C、D、E、F 区的未处警率分别为 5.38%、5.27%、23.45%、13.27%、28.31%、25.18%。平均为处警率为 16.81%。从而可知该市各区之间的出警率分布严重不均衡。

由各区的人口和面积分布，得出个区的人口密度分别为：A 区，2.727 人 / 公里;B 区 0.204 人 / 公里;C 区 0.222 人 / 公里;D 区 0.191 人 / 公里;E 区 0.176 人 / 公里;F 区 0.193 人 / 公里。各区得人均可享有交巡警服务平台个数分别为：A 区，0.333 个;B 区 0.381 个;C 区 0.347 个;D 区 0.123 个;E 区 0.197 个;F 区 0.208 个。显然，A 区的人口密度远远大于其他区，但 A 区的人均享有数却只有 0.333 个，仅次于 B 区和 C 区。因而 A 区的人口密度与人均享有个数严重不成正比。

运用 matlab 编程，得到的全市的未覆盖的节点（即 3 分钟内各区未能处警的节点）和现有警务平台设置点的分布情况如下图所示：



由图，可以直观的看出，未覆盖点主要分布在各区的边缘地带，而有的交巡警服务平台设置在最边缘，如图图中左上角的三个平台，因而这种设置是很不合了的。

5.4.4 各区交巡警服务平台优化设置模型

由交巡警平台设置合理性分析，知设置不合理主要是在不能快速处警、发案高频地平台设置少、平台设置与区域人口密度不成正比。

对于本题有两种大的改进方案；第一种，增加交巡警服务平台，再对现有的交巡警服务平台做必要的调整。第二种，不增加交巡警服务平台，只对现有的交巡警服务平台做必要的调整。考虑到增加交巡警服务平台需要增加大量得财和人了，实际操作起来较为困难，因而本文采用第二种方案。

对于第二种方案又有两优化方案；第一种，全市统一管理，统一对所有交巡警服务平台进行合理分配；第二种，分区管理，只对该区内的交巡警服务平台进行合理分配。因为就现实情况分析，全市统一管理难度大，不便于调配。因而本文从第二种方案对交巡警服务平台的设置进行优化。

由于交巡警服务平台的遵旨是服务群众，因而首先从人民的利益出发，即首先解决事故的快速处警问题.

在现有交巡警服务平台个数不变的情况下，对各区的交巡警服务平台的位置进行重新设置。建立快速处警的 0 － 1 规划优化模型如下：

$$\min z=\sum_{k=1}^{x}\sum_{i=1}^{x}B_kC_{ki}A_{ki}$$

$$\text{s.t}\quad \sum_{k=1}^{x}A_{ki}=1$$

$$\sum_{i=1}^{x}A_{ki}\ge 1 \tag{5}$$

$$\sum_{k=1}^{x}B_k=y$$

$$A_{ki}=0\quad \text{or}\quad 1\qquad i,k\in(1,2,3.......,x)$$

注：A_{ki}=1,表示第i个节点被第k个节点处警；A_{ki}=0,表示第i个节点不被第k个节点处警。

B_k=1,表示第k个节点设置了交巡警服务平台；B_k=0,表示第k个节点未设置交巡警服务平台。

C_{ki}表示第i个节点到第k个节点得最短路程。

x表示个区的节点数；y表示各区交巡警服务平台设置个数。

要使交巡警服务平台个数与发案域量成正比，也就是使得个交巡警服务平台的工作量尽可能均等，模型如下：

$$\min z=\sum_{j=1}^{y}\left(\sum_{i=1}^{x}D_{ij}Q_i-S\right)$$

$$\text{s.t}\quad \sum_{j=1}^{y}D_{ij}=1$$

$$\sum_{i=1}^{x}D_{ij}\ge 1 \tag{6}$$

$$D_{ij}=0\quad \text{or}\quad 1\qquad i\in(1,2,3.......,x),\quad j\in(1,2,3.......,y)$$

注：D_{ij}=1 表示第i个节点由第j个交巡警服务平台处警。D_{ij}=0 表示第i个节点不由第j个交巡警服务平台处警。

Q_i表示第i个节点日案发次数。S表示所对应区所有节点的平均日案发次数。

x表示个区的节点数；y表示各区交巡警服务平台设置个数。

5.4.5 对各区交巡警服务平台优化设置模型求解

快速处警的 0 － 1 规划优化模型如下：

以所有交巡警服务平台到所处警的范围内所有点的路程的总和为目标函数；

$$\min z=\sum_{k=1}^{x}\sum_{i=1}^{x}B_kC_{ki}A_{ki}$$

要满足任何一个节点有且仅被一个交巡警服务平台处警，即表述为：

$$\sum_{k=1}^{x}A_{ki}=1$$

任何一个交巡警服务平台至少处警一个节点，即：

$$\sum_{i=1}^{x} A_{ki} \geq 1$$

交巡警服务平台数为 y，即：$\sum_{k=1}^{x} B_k = y$

$$A_{ki} = 0 \quad \text{or} \quad 1 \qquad i,k \in (1,2,3.......,x)$$ 。

A_{ki} =1,表示第 i 个节点被第 k 个节点处警；A_{ki} =0,表示第 i 个节点不被第 k 个节点处警。

要使交巡警服务平台个数与发案域量成正比，也就是使得个交巡警服务平台的工作量尽可能均等，模型如下：

以工作量的方差最小为目标函数：

$$\min z = \sum_{j=1}^{y} \left(\sum_{i=1}^{x} D_{ij} Q_i - S\right)$$

满足要满足任何一个节点有且仅被一个交巡警服务平台处警，任何一个交巡警服务平台至少处警一个节点，即：$\sum_{j=1}^{y} D_{ij} = 1$，$\sum_{i=1}^{x} D_{ij} \geq 1$

$$D_{ij} = 0 \quad \text{or} \quad 1 \qquad i \in (1,2,3.......,x)，\quad j \in (1,2,3.......,y)$$

注：D_{ij} =1 表示第 i 个节点由第 j 个交巡警服务平台处警。D_{ij} =0 表示第 i 个节点不由第 j 个交巡警服务平台处警。

Q_i 表示第 i 个节点日案发次数。S 表示所对应区所有节点的平均日案发次数。

运用 lingo8 对以上两个优化模型求解得各区的交巡警设置平台节点如表 5：

表 5

A	B	C	D	E	F
1	3	5	27	2	2
2	24	14	32	31	58
3	25	58	33	32	62
43	36	59	34	33	69
44	37	60	35	36	70
45	38	61	37	37	71
46	39	62	38	38	72
54	43	108	40	39	73
55		109	41	40	79
63		110		41	80
64		111		67	81
65		112		74	
66		113		75	
67		114		83	
68		116		84	
69		126			
70		131			
75					
76					

77					

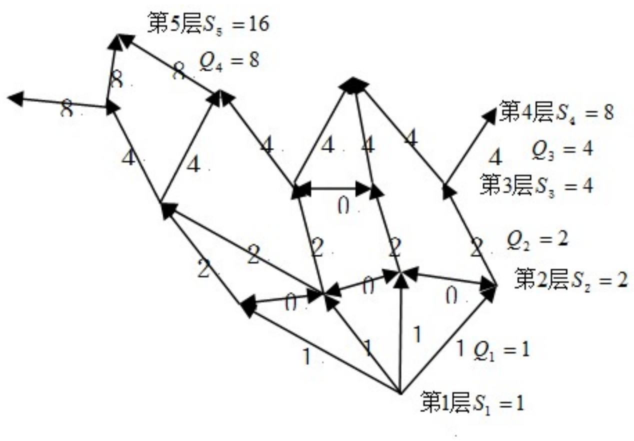
5.5 最佳围堵方案模型的建立与求解

由题知，犯罪嫌疑人的逃离方向存在不定的概率因素，即在地图上存在一定的概率分布情况，且范围大小随时间动态变化。逃犯在三分钟的时间内已经逃至距事发案件 P 点 S 距离时，警察开始施行围堵。由假设知逃犯的逃离速度与警察的围堵速度相同，那么令交巡警服务平台点为 $Q_j(j=1,2,3...)$，逃犯在三分钟以后可以到达的节点为 $P_i(i=1,2,3...)$，则围堵成功的条件可以转化为：$\exists Q_i$ 到 P_i 路程小于 P_i 到三分钟时所到达地点的路程，且各个可能节点至少能有一个交巡警服务平台进行围堵。对于最佳围堵方案，因为时间的变化，利用逃离范围的动态性，建立树杈元素传递算法，求解在时间不断增加情况下，即犯罪嫌疑人可能到达范围的不断扩大，最先到达围堵条件的解。

5.5.1 树杈元素传递算法

树杈元素传递即元素沿树杈传递，随经过节点的个数，元素按一定的规律花生变化。

以起始元素 1 的传递为例，即树杈最低层节点上的元素 $S_1=1$，则第 1 层与第 2 层各节点之间的传递量为 $Q_1=S_1=1$（Q_k 表示第 k 层节点与第 $k+1$ 层节点之间的传递量；S_k 表示第 k 层节点元素；$S_k=S_{k-1}+Q_{k-1}$）。从而可得第二层节点的元素 $S_2=2$，从而易知同层之间相邻节点之间的传递量为 0。由此原理，沿树
杈向上传递，传递示意简略图表示如下：



由传递性，若两点之间传递的元素等于 0，则此树枝不具传递性，因此可以认为此两点间无连接。

运用树杈元素传递算法对本题进行分析。把树杈图中的每个树杈看成题中的每个节点。围堵流程如下：

一、 找到可围堵节点，不妨设逃犯三分钟到达的点为第一层，那么第二层的所有点位逃犯三分钟不能到达的点，即第二层的点为可围堵点。

二、 根据要使在可围堵点上围堵到逃犯的条件，即各个可围堵节点至少能有一个交巡警服务平台进行围堵。若在第二层上能实现全部围堵，则围堵结束，否则延伸到第三层进行围堵。

三、 关于已延伸到下层围堵情况下，该层哪些围堵点优先围堵。为了尽可能的动用最少警力，即避免重复围堵。根据元素树杈算法，一节点向下一层传递的路径越多，则说明它传递的元素量越大，因而对下一层的影响越大。因此，尽可能围堵传递元素量大的节点。

四，　　根据二三的原理，往复流转，直至实现围堵。

5.5.2 最佳封锁模型的求解：

根据题目要求，求解在 32 号节点 P 发生案件后围堵封锁的最佳方案，假定逃犯的逃离速度也是 60 km/h 与警车速度相同。带入以上模型当中，首先确定三分钟的路程的起始分析范围，通过树杈分层理论原则确定将要到达三分钟路程的节点为第一层节点，之后的依次类推。利用 matlab 编写递推算法程序确定每层能够的围堵的节点，在第一层当中能否全部被警务平台围堵，通过 matlab 计算得：第一层当中存在可以一次性围堵点，即 5 号，6 号，10 号，55 号，3 号。之后继续递推第二层节点当第二层节点存在两点同时可以有两个平台可以管理时，再通过计算其二者的最小路径确定其最佳分配，按此逻辑运用 matlab 编程逐一求解。最后确定其最小生成树网络，即可得最短时间安排的围堵方案，解得其编号结果如下表 6：

表 6

节点号	3	4	5	6	10	15	16	40	41	55
对应去管辖的平台号	2	1	5	6	10	15	16	17	18	3
节点号	60	171	234	240	244	246	248	370	371	561
对应去管辖的平台号	4	170	168	169	172	171	167	321	320	480

根据模型所求得的最佳围堵方案的路线计算其最慢到达封锁节点的服务平台所花费的时间为：8.79 分钟。所得时间即为封锁路口所需要的最短时间。
对以上数据画出其示意图如下：

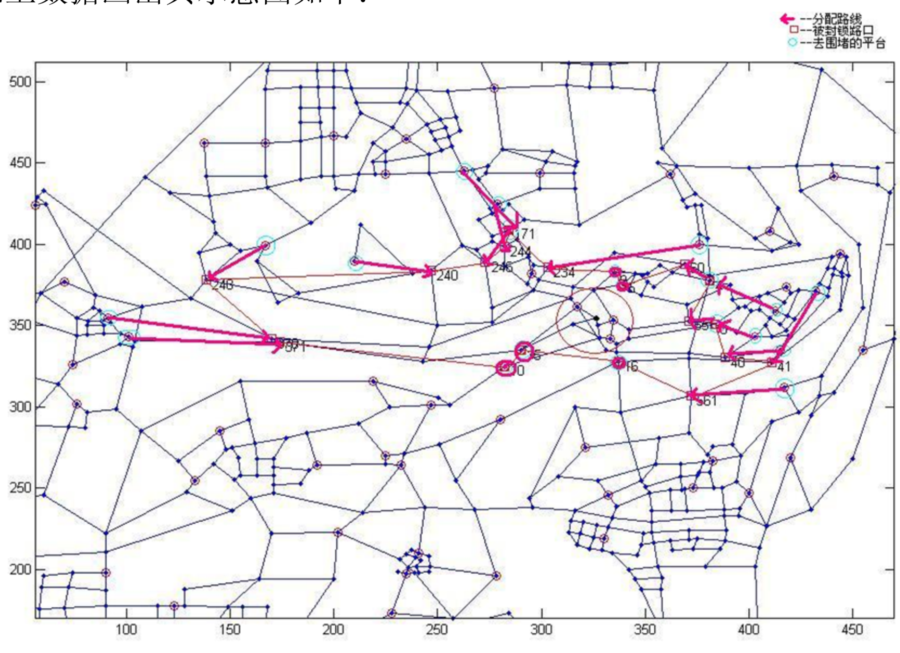


图 2 封锁示意图

图形注解：如图中所示给出了最佳围堵方案，其中标有节点序号的是封锁点，每个封锁点都存在一一对应的平台去围堵，由圆圈所画出的即为需行动的服务平台，红色箭头标明了围堵目标和方向。

6 模型评价与推广

6.1 模型评价

6.1.1 优点分析

本文建立的模型具有较强的适应性，针对不同种实际问题，都能给出很好的解决方案。对于图论分析的最短路径模型，在要求能够在最短时间到达或最快速度赶到的条件下，可以很合理的解决分配管理范围问题。而且在忽略其他细微因素的影响下，能够给出最优的解决方案。

基于非方阵指派问题的调度模型的建立，使得有关资源分配调度问题的解决，变得更加高效快速合理。本文在解决快速封堵交通要道的警员调配问题上，使用此模型准确合理的给出了快速封堵要道的时间和方案。

对于针对全市现有交巡警服务平台设置方案的合理性评价模型，充分的从各个方面考虑其合理性准则，能全面的对现有分布情况进行合理的分析。之后可以根据模型分析得出的结果，利用 0-1 规划优化其分布。

最后建立的树杈元素传递模型，能非常直观明了的解决此类最优围堵问题，利用树杈传递层层分析其可能存在的方案。

6.1.2 缺点分析

本文的模型存在一定的条件性，忽略了很多现实生活中可能存在的影响因素。对于模型一考虑的条件少，只要到达时间尽可能少的条件下分配问题，如果将工作量考虑进去则可能模型会更优。

6.2 模型推广

本文的模型不仅仅只在交巡警的调配和管理问题上起作用，而且也能解决公交车的调配问题，能运用第一模型去求解最佳路线分配问题。

参考文献

[1]杨丽英等，非方阵指派问题的求解，信息与控制，第 38 卷第 6 期，2009.12

[2]姜启源编，数学模型（第二版），高等教育出版社，1993.8

[3]谢金星、薛毅编，优化建模与 LINGO/LINDO 软件，清华大学出版社，2005.7

[4]芦群等，城市公共交通网络均衡分配模型与算法，交通与计算机，2004(04)

附件

附件 1：floyd 算法通用程序

```
%floyd 算法通用程序，输入 a 为赋权邻接矩阵
%输出为距离矩阵 D,和最短路径矩阵 path
function [D,path]=floyd(a)
n=size(a,1);
D=a;
path=zeros(n,n);
for i=1:n
    for j=1:n
        if D(i,j)~=inf
            path(i,j)=j;
        end
    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);
                path(i,j)=path(i,k);
            end
        end
    end
end

%配合 floyd 算法的后续程序,s 为源点,t 为宿点
%L 为长度,R 为路由
function [L,R]=router(D,path,s,t)
L=zeros(0,0);
R=s;
while 1
if s==t
L=fliplr(L);
L=[0,L];
return
end
L=[L,D(s,t)];
R=[R,path(s,t)];
s=path(s,t);
end
```

附件 2：问题一第 1 小问求解

```
clc,clear
%数据处理
load shuju.mat %存放节点坐标变量 node，街道（边）端点变量 bian
```

```
load D.mat    %D.mat 变量 D 存放城市所有节点发案率
load PT.mat    %存放所有平台节点标号
DD=[];%存放各街道长度
kc=92;%考察节点数 kc
chk=13;%区域出口数
P=[1:20];%平台节点矩阵
LP=setdiff([1:kc],P);%非平台节点
pt=length(P);%平台数
for i=1:length(bian(:,1))
DD(i)=pdist([node(bian(i,1),:);node(bian(i,2) ,:)]); %求各街道长度
end
DD=DD';
for k=1:kc %绘制 1~kc 节点图
    plot(node(k,1),node(k,2),'*');
    text(node(k,1)-3,node(k,2)+3,num2str(k));%标注每个点标号
    hold on
    if sum(ismember(P,k))~=0 %平台点画圈标注
    plot(node(k,1),node(k,2),'ro');
    end
end
for j=1:length(bian(:,1))%绘制相邻节点连线
    if bian(j,1)<=kc&bian(j,2)<=kc
    plot([node(bian(j,1),1),node(bian(j,2),1)],[node(bian(j,1),2),node(bian(j,2),2)]);
    hold on
    end
end
%floyd 算法求最短路
a=inf*ones(kc,kc);%建立邻接矩阵 a
for i=1:length(bian(:,1))
    if bian(i,1)<=kc&bian(i,2)<=kc
        a(bian(i,1),bian(i,2))=DD(i);
        a(bian(i,2),bian(i,1))=DD(i);
    end
end
for jj=1:kc
    for kk=1:kc
        if(jj==kk)
            a(jj,kk)=0;
        end
    end
end
[Dd,path]=floyd(a);%调用 floyd 函数求最短路
LL=[];%存放普通节点分别到各平台的最短路值，为（kc-pt）*pt 矩阵
for v=1:kc-pt
for n=1:pt
    [L,R]=router(Dd,path,LP(v),P(n));%调用辅助函数求指定两节点最短路
    LL(v,n)=L(end);
end
```

```
end
RR=[];%存放普通节点，最近的平台节点及其最短路程值
rr=[];
for nn=1:length(LL(:,1))
RR(nn)=P((find(LL(nn,:)==min(LL(nn,:)))));%普通节点由距离最近的平台管辖
rr(nn)=min(LL(nn,:));
end
RR=[LP',RR',rr'];%存放普通节点，所属平台及最短路距离值
```

附件 3： 问题一第 2 小问调度方案求解

```
load Aout.mat%Aout 为城区 A 出口节点标号
load Qd.mat%所有节点两两间最短路值
pt=20;%平台数目
LL2=[];%存放的效益矩阵(即路程矩阵）cij
for v=1:length(Aout)
for n=1:pt
	LL2(v,n)=Qd(Aout(v),n);
end
end
GG=[];%存放第 i 个出口节点，及到第 j 个平台的最短路距离值
for uu=1: length(Aout)
	for jj=1:pt
```

```
            gg=[uu,jj,LL2(uu,jj)];
            GG=[GG;gg];
        end
end
[qq ww]=sort(GG(:,3));
gg=[];%按最短路程从小到大排列
for i=1:length(ww)
        g=[GG(ww(i),1:2),qq(i)];
        gg=[gg;g];
end
m=1;%m 从 1 开始逐步增大，直到恰能够找到饱和 13 个出口，的匹配方案为止
while 1      %循环寻找恰能实现饱和匹配 13 个出口所需最小的 m 值
mm=length(Aout);
A=zeros(n,mm);%构建关联矩阵 A
for i=1:m
            A(gg(i,2),gg(i,1))=1;
end
[M]=ZDPP(n,mm,A);%调用最大匹配子程序，求最大匹配矩阵
if min(sum(M))~=0 %当每个出口均至少分配到 1 个平台即结束循环
        break;
end
m=m+1;
end
[Aout(gg(m,1)),gg(m,2:3)]    %最后一个被封锁的出口情况
Tmax=gg(m,3)/10                 %全部封锁所需时间
```

附件 4： 问题一第 2 小问求最大匹配子程序

```
%n、m 表示两边各节点数，A 表示二分图一边到另一边的弧 01 矩阵（多余节点补零）
function [M]=ZDPP(m,n,A)
M(m,n)=0;
for(i=1:m)for(j=1:n)if(A(i,j))M(i,j)=1;break;end;end        %求初始匹配 M
   if(M(i,j))break;end;end    %获得仅含一条边的初始匹配 M
while(1)
   for(i=1:m)x(i)=0;end    %将记录 X 中点的标号和标记*
   for(i=1:n)y(i)=0;end    %将记录 Y 中点的标号和标记*
   for(i=1:m)pd=1;         %寻找 X 中  M 的所有非饱和点
      for(j=1:n)if(M(i,j))pd=0;end;end
      if(pd)x(i)=-n-1;end;end    %将 X 中 M 的所有非饱和点都给以标号 0 和标记*，程序中用 n+1 表示 0 标号，标号为负数时表示标记*
   pd=0;
   while(1)xi=0;
      for(i=1:m)if(x(i)<0)xi=i;break;end;end        %假如 X 中存在一个既有标号又有标记*的点，则任取 X 中一个既有标号又有标记*的点 xi
```

```
        if(xi==0)pd=1;break;end    %假如 X 中所有有标号的点都已去掉了标记*，算法终止
        x(xi)=x(xi)*(-1); %去掉 xi 的标记*
        k=1;
        for(j=1:n)if(A(xi,j)&y(j)==0)y(j)=xi;yy(k)=j;k=k+1;end;end    %对与 xi 邻接且尚未给标号的 yj 都给以标号 i
        if(k>1)k=k-1;
           for(j=1:k)pdd=1;
              for(i=1:m)  if(M(i,yy(j)))x(i)=-yy(j);pdd=0;break;end;end    %将 yj 在 M 中与之邻接的点 xk (即 xkyj∈M)，给以标号 j 和标记*
              if(pdd)break;end;end
           if(pdd)k=1;j=yy(j);      %yj 不是 M 的饱和点
              while(1)P(k,2)=j;P(k,1)=y(j);j=abs(x(y(j)));      %任取 M 的一个非饱和点 yj，逆向返回
                 if(j==n+1)break;end    %找到 X 中标号为 0 的点时结束，获得 M-增广路 P
                 k=k+1;end
              for(i=1:k)if(M(P(i,1),P(i,2)))M(P(i,1),P(i,2))=0;    %将匹配 M 在增广路 P 中出现的边去掉
                 else  M(P(i,1),P(i,2))=1;end;end  %将增广路 P 中没有在匹配 M 中出现的边加入到匹配 M 中
              break;end;end;end
    if(pd)break;end;end    %假如 X 中所有有标号的点都已去掉了标记*，算法终止
```

附件 5：问题一第 3 小问，增加平台设置方案求解程序

```
clc
clear
load RR
load zd
%寻找之前分配中无平台 3 分钟内可以达到的节点
knode=RR(find(RR(:,3)>30),1);
%寻找到这些点距离在 30mm 以内的点
fend={};
for i=1:length(knode)
fend{i}=find(zd(knode(i),:)<=30);
end
fend=fend';
d1=length(fend{1});
d2=length(fend{3});
d3=length(fend{5});
d4=length(fend{6});
Kxj=[];%存放所有可行新增节点分布可能情况
for i1=1:d1
        for i2=1:d2
                for i3=1:d3
                        for i4=1:d4
                                                Kxj=[Kxj;[fend{1}(i1),fend{3}(i2),fend{5}(i3),fend{6}(i4)]];
                        end
```

```
            end
        end
end
save Kxj %所有可能的增加节点位置
load shuju.mat%存放节点坐标变量 node，街道（边）端点变量 bian
load Qd   %城市任意两节点间的 最短路程矩阵
load D
kc=92;%节点数
chk=13;%区域出口数
JH=[];%存放各种方案下的工作量方差值
D=D([1:kc]);
for su=1:length(Kxj(:,1))
P=[1:20,Kxj(su,:)];%平台节点矩阵
LP=setdiff([1:kc],P);%非平台节点
pt=length(P);%平台数
LL=[];
for v=1:kc-pt
for n=1:pt
    LL(v,n)=Qd(LP(v),P(n));%最短路
end
end
RR=[];%存放普通节点，最近的平台节点及其最短路程值
rr=[];
for nn=1:length(LL(:,1))
RR(nn)=P(find(LL(nn,:)==min(LL(nn,:))));
rr(nn)=min(LL(nn,:));
end
RR=[LP',RR',rr'];%存放普通节点，所属平台及最短路距离值
%计算工作量方差
jh=0;
        for i=1:pt
            jh=jh+(sum(D(RR(find(RR(:,2)==i),1)))+D(i)-sum(D)/pt)^2;
        end
        JH=[JH jh];
end
JH=JH';
Kxj(find(JH==min(JH)),:)%此即为增加平台分布方案
```

附件 6： 问题二第 1 小问 lingo 程序

```
model:
title A区情况;
sets:
  node/1..92/:B;
  bian(node,node):c,A;
endsets
data:
c=@OLE('e:\acij.xls','cij');
@OLE('acij.xls','A','B')=A,B;
enddata
min=@sum(bian(k,i):B(k)*c(k,i)*A(k,i));
@for(node(i):@sum(node(k):A(k,i))=1);
@for(node(k):@sum(node(i):A(k,i))>=1);
@sum(node:B)=20;
@for(node:@bin(B));
@for(bian:@bin(A));

End
```

附件 7：问题二第 2 小问

```
%求节点 Pp 的相邻节点集合文件 Qlju.m
function lju=Qlju(Pp)
load zd
load shuju
lju=[bian(find(bian(:,1)==Pp),2);bian(find(bian(:,2)==Pp),1)];

%----------------------------------------------------------------
%以 P(32)为中心点向外围拓展的第一批节点
clc,clear
load Qd
load shuju
load path
load PT
kc=582;%节点数
chk=17;%区域出口数
P=PT;%平台节点矩阵
LP=setdiff([1:kc],P);%非平台节点
pt=length(P);%平台数
Pp=32;
circle(2,node(Pp,1),node(Pp,2));
L=Pp;%存放前续节点
L2=Pp;%存放最顶端节点
JD=[];%存放上层节点中超程节点
for j=1:6
    Lju=[];
for i=1:length(L2)
    lju=Qlju(L2(i));
    Jn=intersect(L,lju);
   lju=setdiff(lju,Jn);
  Lju=[Lju;lju];
  for j=1:length(lju)
      plot([node(L2(i),1),node(lju(j),1)],[node(L2(i),2),node(lju(j),2)],'r');
      hold on
  end
end
L2=unique(Lju)
L=[L;L2]

ZD=[];
for jj=1:length(L2)
```

```
        ZD=[ZD Qd(Pp,L2(jj))];
end
jd=L2(find(ZD>30));
L2=setdiff(L2,jd);
JD=[JD;jd]
ZD
end
Ju=[];
for i=1:length(JD)
      Ju(i)=Qd(Pp,JD(i))-30;
end
Ju=Ju';%剩余距离
SS=[];%外点到各平台距离
kxp={};%可达要求的平台点
tzd=[];%需要扩展搜寻点
for j=1:length(JD)
        for k=1:pt
                SS(j,k)=Qd(JD(j),P(k));
        end
    kxp{j}=PT(find(SS(j,:)<=Ju(j)));
    if isempty(kxp{j})
    tzd=[tzd JD(j)];
    end
end
kxp=kxp';

%任意外围节点继续向下一级节点拓展求解
lc,clear
load Qd
load shuju
load path
load PT
kc=582;%节点数
chk=17;%区域出口数
P=PT;%平台节点矩阵
LP=setdiff([1:kc],P);%非平台节点
pt=length(P);%平台数
Pp=39;
circle(2,node(Pp,1),node(Pp,2));
L=Pp;%存放前续节点
L2=Pp;%存放最顶端节点
JD=[];%存放上层节点中超程节点
for j=1:1
        Lju=[];
for i=1:length(L2)
        lju=Qlju(L2(i));
        Jn=intersect(L,lju);
      lju=setdiff(lju,Jn);
    Lju=[Lju;lju];
```

```
    for j=1:length(lju)
          plot([node(L2(i),1),node(lju(j),1)],[node(L2(i),2),node(lju(j),2)],'r');
          hold on
    end
end
L2=unique(Lju)
L=[L;L2]

ZD=[];
for jj=1:length(L2)
        ZD=[ZD Qd(Pp,L2(jj))];
end
ZD
end
Ju=[];
JD=L2;
for i=1:length(JD)
      Ju(i)=Qd(32,JD(i))-30;
end
Ju=Ju';%剩余距离
SS=[];%外点到各平台距离
kxp={};%可达要求的平台点
tzd=[];%需要扩展搜寻点
for j=1:length(JD)
        for k=1:pt
                SS(j,k)=Qd(JD(j),P(k));
        end
    kxp{j}=PT(find(SS(j,:)<=Ju(j)));
    if isempty(kxp{j})
    tzd=[tzd JD(j)];
    end
end
kxp=kxp';
```