

承诺书

赛区评阅编号（由赛区组委会评阅前进行编号）：

2013 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）：

摘 要

本文就某市交巡警服务平台的设置与调度问题分别建立了最短路径覆盖模型、基于模拟退火算法的方差最小优化模型、基于 0-1 规划的快速封锁模型、方差最小的数学优化模型、模糊综合评价模型和基于二分法和匈牙利算法的最优搜索模型。

针对服务平台管辖分配问题, 本文首先根据 floyd 算法计算各路口间的最短路径, 根据可被平台管辖的条件将路口节点分为三类, 每一类对应各自的被管辖方式。首先按照就近管辖的管辖方式给出一组解 (见表 1); 考虑到这种分配方式下工作量方差过大, 本文建立了方差最小的优化模型, 按照优化算法逐步调整管辖分配方式; 为得到全局最优解, 本文最后又建立了基于模拟退火算法的方差最小优化模型, 给出了更优的管辖分配方式 (见表 5), 此时方差为 11.415, 管辖方式较为均衡。

为了解决 20 个平台对 13 个交通要道的快速全封锁问题, 本文引入了 0-1 变量, 建立了指派问题的 0-1 整数规划模型。首先将最短封锁时间作为目标函数, 求出最短封锁时间为 8.015 min, 然后将此结果转化为约束条件, 将最短总封锁时间作为目标函数, 重新求解得到封锁时间最短而且总任务量最小的最优封锁调度方案 (见表 8)。

在新增平台提升 A 区服务平台设置合理性问题时, 本文选择在 3 min 内无法到警的那一类路口节点中增加新的平台, 建立了以工作量方差最小为目标的优化模型。将不满足到警时间约束的 6 个节点分为 4 组, 每一组内建立一个新的平台, 通过在各组内局部使用模拟退火算法, 找到最优的新建平台方案, 得到新增加的平台为标号为 29, 40, 48, 92, 此时的工作量方差为 6.28, 各平台间均衡程度有了较大的改进。

在对该市现有服务平台设置方案合理性分析的问题上, 本文引入了隶属度函数, 建立了模糊综合评价模型。通过计算综合评价向量得到区域 E 是该城市交巡警平台设置最不合理的区域。本文通过问题一中的优化算法计算出区域 E 中应当增加的平台个数与位置 (见表 13) 优化后各评价指标都有大幅提升。本文还对其他具有明显不合理性的区域作出了同样的优化 (见表 14、15)。

在对嫌疑人的围堵问题上, 本文建立了基于二分法和匈牙利算法的最优搜索模型。通过二分法控制时间的变化, 找出不同时间下嫌疑人的存在区域, 使用匈牙利算法计算全市交巡警平台对该区域边界点的最大匹配, 根据匹配方式计算封锁各边界点的时间, 同时计算嫌疑人从 P 点到达边界点的时间, 当各平台所用时间小于嫌疑人时, 就保证围捕成功。通过本文建立的最优搜索模型, 得出需要 7.36 min 可以完成围捕, 得到具体参与围捕的平台调度方式 (见表 16)。

关键词: floyd 算法 模拟退火算法 0-1 整数规划 模糊综合评价 匈牙利算法

一、问题的重述

1.1 交巡警平台设置与调度问题的背景

“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、服务群众四大职能。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。

交巡警制度的改革整合了警力资源，是将刑事执法、治安管理、交通管理以及服务群众四大职能有机融合的新型防控体系，代表了未来的发展方向，是脱离原始、改变粗放和走向动态警务运行模式的显著标志，必将产生强大的司法制衡力，社会治安的驾驭力和打击犯罪的冲击力。

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。

1.2 问题的提出

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（详见附件）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、问题的分析

2.1 问题一的分析

2.1.1 交巡警服务平台管辖范围分配问题的分析

这是一个复杂的指派问题，要求我们给出 A 区 20 个平台对 92 个路口节点的管辖方案。本文除了考虑到警时间的约束外，还要考虑在满足时间的约束下，尽量减小各平台工作量的不均衡程度。

由于题目已经给出了全市的路口节点标号和所有的道路起、终点编号，我们使用 floyd 算法可以很容易的计算出任何两个节点之间的距离以及路径。由于交巡警服务平台的首要职责是在管辖区内的路口需要时及时到达路口位置，本题目中的要求是在 3 min 内到达路口，所以我们只要找出与各平台距离 3 km 以内的路口节点，使他们归这些平台管辖即可。

根据本题目的数据，A 区中的道路节点可以分为三类：3km 内只可被某一个服务平台管辖的节点、3km 内可被多个平台管辖的节点和 3km 内不被任何一个平台管辖的节点。对于第一、三种道路节点，我们只能选择距离最近的平台来管辖他们，对于第二类道路节点，我们首先将他们归给距离最近的平台来管辖，然后通过计算工作量的方差来衡量这种管辖范围分配下的均衡程度，在不均衡的情况下可以通过改变一些平台与节点见的管辖关系来降低方差，增加均衡程度。当方差降低到一定程度时，我们认为此时的管辖范围分配是较优的，停止对之前结果的修正。

除了利用最短路径和均衡程度来确定管辖范围的分配外，我们还采用模拟退火算法等智能算法来实现方差最小的最优解的搜索。通过多次执行即可得到最优解。

2.1.2 区域全封锁的平台调动问题的分析

这还是一个典型的指派问题，题目要求由 20 个平台中选出 13 个去封锁出入 A 区的 13 个路口，并且达到最快封锁。

区域全封锁完成的时间取决于最后一个被封锁的路口的耗时，但是我们要在达到最快封锁时间的条件下，尽量保证平均封锁用时最小。我们首先以最大耗时最小为目标函数，将各平台和各出入路口满足的约束条件列出，建立 0-1 整数规划问题，求出最大耗时及相关的封锁方案。虽然我们得到了封锁所用的最短时间，但是我们并没有得到最佳封锁方案，这是因为之前的目标函数只能保证耗时最长的封堵方案达到最小值，并没有对其他封堵的方案进行限制。我们再将得到的最大耗时作为约束条件，将各平台耗费时间之和作为目标函数，约束条件不改变，重新求解 0-1 规划问题，得出平均封堵用时最短的最优封锁方案。

2.1.3 增加交巡警服务平台问题的分析

通过 2.1.1 的分析，我们得到了 6 个点不能在 3 min 内到警，而我们讨论如何增加交巡警服务平台，就是为了解决这些节点不能及时到警的问题。所以我们在安放这些新的交巡警平台的位置时，主要考虑这 6 个点以及能够在 3 min 内管辖到这 6 个点的其他路口节点。在考虑具体安放位置时，我们还要综合考虑新平台建立时自身管辖范围内的工作量以及对附近其他平台工作量的影响，优先选择那些在建立后降低了平台工作量方差的路口节点。

在处理本问题时，我们首先将那 6 个不能及时到警的节点，按照互相能否在 3 min 内到达划分成几组，在每一组内只要新建一个交巡警平台即可满足及时到警的要求。如果某组内的节点与其它已被管辖的节点之间距离小于 3 km，那么新建平台的位置就在他们中选择。在选择好新的平台后，要更新管辖分配方式，将可被新平台管辖的节点修改为该新平台管辖，如果引起了方差变大，则不作此修改，如果方差减小，则继续按照这种试探性的方式修改管辖方式，直到方差不能再变小为止。通过这种对所有可能情况的遍历，我们就可以得到增加交巡警服务平台问题的最优解。

同时我们注意到，交巡警平台并不一定是新建越多越好，由于新建平台客观上存在一定的成本，如果我们将实现 3 min 内到警作为主要目标，那么只要所有的节点都已经满足在 3 min 内可到警，我们优先选择新建平台数最小的那种方案。

2.2 问题二的分析

2.2.1 全市交巡警服务平台设置合理性问题的分析

我们可以对该市现有交巡警服务平台设置方案进行分区考虑，即分别关注各区平台设置方案的合理性，如果平台设置显示在区域交界处有过多重复，再考虑平台间的协管或者重新划分区域。

仿照问题一，我们计算了全市各区所有节点中在 3 min 内无法到警的路口节点数量，并分区计算了所占比例，同时根据附件中的人口数量和区域面积计算人均警力配比等可用来描述区域交巡警平台设置是否合理的统计量。如果某区的路口 3 min 内到警率较低，人均警力匹配值较低，或是平台工作量方差较大，那么我们可以考虑在这一区中增加一定数量的新平台，具体增加数量及位置可以仿照问题一中对 A 区的增加方式来完成。

我们根据附件数据可以计算出各区的 3 min 到警率、日均处理案件数、平台服务人口总数和各区工作量的方差值，通过引入隶属度函数建立模糊评价矩阵，进行模糊综合评价，最终得到综合评价向量，对矩阵中综合评价值较低和较高的区域详细分析，并对综合评价值最低的区域提出解决方案。

2.2.2 全市交巡警服务平台最佳围堵方案问题的分析

当该市某路口 P 发生重大刑事案件时，犯罪嫌疑人已逃跑，由于在案发 3 min 后巡

警才接到报警，并且不知道嫌疑人的逃跑方向，嫌疑人在这 3 min 内可能逃窜在以 P 点为中心，3 km 为路程的一个放射状区域内。为了快速搜捕嫌疑人，将调度全市的交巡警服务平台来对嫌疑人所有可能的出逃方向上的路口节点进行封堵。假设某时间嫌疑人可能存在的区域为 Q ，记 Q 的边界节点集为 ∂Q ，则能否围堵成功就取决于警方能否比嫌疑人早 3 min 到达 ∂Q 中所有的节点。

我们计算全市所有平台对 ∂Q 中各节点完成全封锁各项时间，若每项时间比嫌疑人到达 ∂Q 中相同节点的最短时间都少 3 min，我们就认为封堵方案是成功的，此时平台对各路口节点封堵时间中最长的即为完成封堵所需的时间。如果有一个节点的封堵时间不满足这个约束，即认为围堵方案失败，我们需要计算下一个时间段内嫌疑人可能存在的区域，按照相似的条件去判断能否围堵成功。

在时间的求解上，我们给时间设置一个上限，利用二分法逐步逼近完成封堵需要的最短时间，并且得出此时的封堵方案。

三、模型的假设

- 1、假设每个巡警服务台的职能和警力配备相同；
- 2、把道路交叉路口看做图的节点，路口之间的道路看作图的边；
- 3、警车的速度保持固定的 60km/h，不受交通状况的影响；
- 4、事故发生在路口节点处，路段上不发生案件；
- 5、交巡警平台的日工作量由管辖路口处与平台的距离和发案率决定；
- 6、同一平台管辖范围内没有多起案件同时发生的情况；
- 7、增设一个新的交巡警平台需要一定的成本，成本与平台所在地段无关；
- 8、犯罪嫌疑人驾车速度也是 60km/h；
- 9、交巡警服务平台接到报警后能立即出警，中间没有延误。

四、符号说明

符号	说 明
d_{ij}	路口 i 到路口 j 间的最短路程
p_j	路口 j 的案发率
w_i	平台 i 的日工作量
x_{ij}	0-1 变量, 平台 i 是否管辖路口 j
I	初始评价矩阵
$\mu(x)$	隶属度函数
R	模糊关系矩阵
ω	综合评价指标权重向量
P	综合评价向量
Q	嫌疑人逃跑后可能的所在区域
∂Q	嫌疑人逃跑后可能所在区域的边界点集
T_i	全市平台封锁 ∂Q 中节点的单项封锁用时
t_i	嫌疑人到达 ∂Q 中节点的单项用时

注：其它符号将在下文中给出具体说明

五、模型的建立与求解

5.1 基于最短路的交巡警服务平台管辖范围分配模型

5.1.1 模型的建立

我们首先引入工作量的定义：

定义 某交巡警平台每天的出警距离，等于该交警平台管辖区域内所有路口案发率（次数）与路口到平台距离的乘积之和，称为该平台的日均工作量。

用数学符号 w_i 表示工作量，即

$$w_i = \sum_j d_{ij} * p_j, i=1,2,...,20, j \text{ 为 } i \text{ 管辖的节点标号}$$

工作量是衡量一个平台日均出警次数的指标，工作量越大，意味着该平台的警务人员越辛苦。如果多个平台的工作量差别较大，说明当前的管辖分配方式不均衡。

下面是本文确定平台管辖范围分配方式的算法：

- (1) 利用 floyd 算法求出所有节点任意两点间的距离矩阵；
- (2) 以平台 3 min 能否到达路口为筛选条件，把时间大于 3 分钟的点选出来；
- (3) 以各服务平台为中心，以巡警 3 min 内可达距离为条件，计算各服务平台所能管辖的路口节点；
- (4) 将所有的路口节点分成三类：只可被某一个服务平台管辖的节点、可被多个平台管辖的节点和不被任何一个平台管辖的节点；
- (5) 将不同类别节点采用不同的方式确定其归属。

其中的第一类点已经确定被某个平台管辖，这里我们仅需对后两类点的归属进行讨论，具体管辖方式为：对于可归属两个或两个以上平台的节点，我们优先考虑交巡警能够尽早到达，通过查找距离矩阵的方法，将这些点逐一地归属到距离它们最近的平台管辖；对于被任何一个平台管辖的节点（共 6 个），这类点的特点是在一次寻优中不归属任何一个平台。由于这类点不能实现突发事件发生时交巡警 3 分钟内赶到，但也不能因此无人管辖，所以我们也把这些点归属到离它最近的平台管辖。

5.1.2 模型的求解

我们首先使用 floyd 算法计算出各路口节点间的距离和路径，按照相距最短的分配原则将所有的路口节点分配给与之相距最近的平台，这样就确定了最初的管辖分配方式。

经过程序计算，根据最短路确定的的管辖分配结果如下表（表 1）：

表 1 根据最短路确定的的管辖分配方式表

交巡警 平台位 置标号	平台管辖路口节点标号	3 km以外的 路口标号	总案发 率	工作量
1	1, 67, 68, 69, 71, 73, 74, 75, 76, 78		10.3	8.47
2	2, 39, 40, 43, 44, 70, 72	39	10.9	12.87
3	3, 54, 55, 65, 66		5.6	5.85
4	4, 57, 60, 62, 63, 64		6.6	6.12
5	5, 49, 50, 51, 52, 53, 56, 58, 59		9.7	10.09
6	6,		2.5	0
7	7, 30, 32, 47, 48, 61	61	9.6	9.30
8	8, 33, 46		5	2.27
9	9, 31, 34, 35, 45		8.2	6.27
10	10,		1.6	0.00

11	11, 26, 27		4.6	2.39
12	12, 25		4	2.86
13	13, 21, 22, 23, 24		8.5	8.88
14	14,		2.5	0
15	15, 28, 29	28, 29	4.8	14.16
16	16, 36, 37, 38	38	3.8	4.87
17	17, 41, 42		5.3	2.57
18	18, 80, 81, 82, 83		6.1	3.26
19	19, 77, 79		3.4	1.15
20	20, 84, 85, 86, 87, 88, 89, 90, 91, 92	92	11.5	11.81

通过最短路算法得到上表数据，我们同时计算了各平台工作量的平均值、极差和方差，数值见下表（表2）：

表2 最短路算法得到的管辖分配方式下的统计量表

算法	平均值	极差	方差
最短路分配	5.66	14.16	19.808

5.1.3 结果分析

我们发现标号为 6, 10 和 14 的三个服务平台只管辖本身，分析可知这主要是由于这些服务平台距离其他顶点均较远，而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。同样也可以看出来，平台 1, 2, 5, 20 等管辖路口节点数量较多，它们的工作量较大。计算中同时发现存在有 6 个路口节点，标号分别为 28, 29, 38, 61, 39, 92，与他们距离最近的交巡警服务平台都超过了 3 km，但我们仍将这 6 个节点分配给了距它们最近的平台管辖。管辖分配结果中，最长出警时间约为 5.7 min，几乎为要求的 3 min 的两倍。

每一个平台管辖范围内的总案发率和工作量差别较大，说明这种按照距离就近的管辖范围分配方式很不均衡，结果并不理想。所以我们考虑在改进算法时，不完全按照距离就近的原则进行管辖分配，而是当距离在允许被管辖的范围内，综合考虑每一个路口节点被管辖时对相应平台的工作量产生的负担，分配时要使各平台工作量尽量均衡。

5.2 基于方差最小的交巡警服务平台管辖范围分配模型

5.2.1 模型的建立

本模型是对 5.1 中给出的结果进行优化，优化过程以方差减小为目标。在 5.1 的求解基础上，按照 5.1.1 的路口节点分类方式，将第二类点（可被多个平台管辖的节点）中分配方式不合理的节点分配给其他平台节点管辖，其中第二类节点和可管辖他们的平台统计见附录一。

我们给出方差最小的管辖范围分配模型的优化算法：

- (1) 找出 5.1 给出结果中工作量大于均值且最大的平台 i ；
- (2) 找出属于平台 i 管辖的且在附录一中给出的节点，计算所有这些节点中对平台 i 工作量，记工作量最大的路口节点为 j ；
- (3) 将节点 j 调整给可管辖 j 的工作量最小的平台，重新计算所有平台的工作量和方差；
- (4) 若方差减小，则返回第(2)步，若方差增大，记此 j 为不可调整的节点，并将节点 j 调回给平台 i ，将各平台工作量和方差修改回原值，返回第(2)步，若方差不再减小，记 i 为不需调整的平台，返回(1)步；
- (5) 当方差不再减小时，停止算法，得到最优解。

5.2.2 模型的求解

我们按照 5.2.1 中的优化算法调整了表 1 中的部分数据，调整后的管辖分配结果见下表（表 3）：

表3 根据方差最小确定的的管辖分配方式表

交巡警 平台位 置标号	平台管辖路口节点标号	3 km以外的 路口标号	总案发率	工作量
1	1, 68, 69, 71, 73, 74, 75, 76, 78		8.7	7.17
2	2, 39, 44, 72	39	5.4	7.48
3	3, 54, 55, 65, 66		5.6	5.85
4	4, 57, 60, 62, 63, 64		6.6	6.12
5	5, 49, 53, 56, 58, 59		7.2	7.18
6	6, 50, 51, 52		5	5.41
7	7, 30, 48, 61	61	6.5	5.54
8	8, 32, 33, 46, 47		8.1	7.61
9	9, 31, 34, 35, 36, 45		9.3	7.29
10	10,		1.6	0
11	11, 26, 27		4.6	2.39
12	12, 25		4	2.86
13	13, 21, 22, 23, 24		8.5	8.88
14	14,		2.5	0
15	15, 28, 29	28, 29	4.8	14.16
16	16, 37, 38	38	3.9	4.20
17	17, 40, 41, 42, 43		8.7	10.18
18	18, 80, 81, 82, 83, 84, 85, 88		9.2	9.49
19	19, 67, 70, 77, 79		5.1	5.78
20	20, 86, 87, 89, 90, 91, 92	92	8.4	8.94

我们再次计算了各平台工作量的平均值、极差和方差，数值见下表（表 4）：

表4 方差最小算法得到的管辖分配方式下的统计量表

算法	平均值	极差	方差
方差最小优化	6.33 km	14.16 km	11.523

通过表 4 我们直观的看到，工作量的方差数值减小了很多，这说明我们对 5.1 结果的部分优化调整是正确的和有效的。这时大部分平台的工作量都处在平均值的附近，剩余的可调整的路口节点再做调整对总体方差的影响不定，所以我们不再进行优化处理。我们在基于方差最小的目标下得到的管辖分配方案即为表 3 对应的管辖分配。

5.2.3 结果分析

平台的工作量平均值有所上升，这是必然的，因为我们是在基于最短路分配方式的基础上修改管辖分配方式的，总距离的增加必然会带来工作量的增加。极差的并没有减小，这是因为路口 28, 29 不处在任何平台的 3 km 距离内，而且相距最近的平台标号为 15，所以平台 15 对路口 28, 29 的管辖关系不可改变，故其工作量 14.16 km 使极差依然很大。方差由 5.1 中的 19.808 减少为 11.523，这说明我们的优化调整减少了各平台间的不平衡程度，新的管辖分配方式均衡了各平台的工作量，这对交巡警服务平台的正常运转是极为重要的。

5.3 基于模拟退火的交巡警服务平台管辖范围分配模型

5.3.1 模型的建立

由于 5.2 中对管辖范围的分配优化算法在方差满足一定标准时就停止了，并不一定找到使得方差最小的全局最优解，本文利用模拟退火算法，在使用 5.2 中给出的解作为初始解的条件下，随机产生一组扰动，即改变若干组管辖匹配方式，计算新的管辖方式下的工作量方差，若方差减小就接受新的解，若方差增大就重新生成新的解，在满足一定次数的扰动后，即可得到最优解。

5.3.2 模型的求解

我们通程序的多调试运行，得到了基于模拟退火算法的交巡警服务平台管辖范围分配方案，具体分配方案见下表（表 5）：

表5 基于模拟退火算法的管辖分配方式表

交巡警平台位置标号	平台管辖路口节点标号	3 km以外的路口标号	总案发率	工作量
1	1, 64, 68, 71, 73, 74, 75		5.9	8.51
2	2, 39, 43, 70	39	7.1	16.25
3	3, 54, 55, 66, 67		5.9	3.31
4	4, 57, 60, 62, 63, 65		6.2	4.58
5	5, 49, 51, 52, 53, 56, 59		7.7	5.54
6	6, 47, 50, 58		5.3	9.02
7	7, 30, 48, 61	61	6.2	6.05
8	8, 32, 35, 45		6.8	9.14
9	9, 31, 33, 36, 46		8.8	6.54
10	10,		2.1	0
11	11, 26, 27		4.4	5.60
12	12, 25		3.7	0
13	13, 21, 22, 23, 24		9.5	8.88
14	14,		2.2	0
15	15, 28, 29	28, 29	4.6	14.16
16	16, 34, 37, 38	38	4.7	7.55
17	17, 40, 41, 42, 72		8.2	7.91
18	18, 77, 78, 79, 80, 82, 83, 84		9.4	15.34
19	19, 69, 76, 81		4.4	11.23
20	20, 85, 86, 88, 89, 90, 91, 92	92	10.6	10.69

我们再次计算了各平台工作量的平均值、极差和方差，并和前两种方法的各项数值做了对比，数值见下表（表6）：

表6 三种算法得到的管辖分配方式下的统计量表

算法	平均值	极差	方差
最短路径分配	5.66	14.16	19.808
方差最小优化	6.33 km	14.16 km	11.523
模拟退火算法	7.515 km	16.25 km	11.415

我们还根据表5作出了A区交巡警服务平台管辖范围分配示意图（图1），图中相同形状和颜色的符号代表由某平台管辖的所有路口节点。

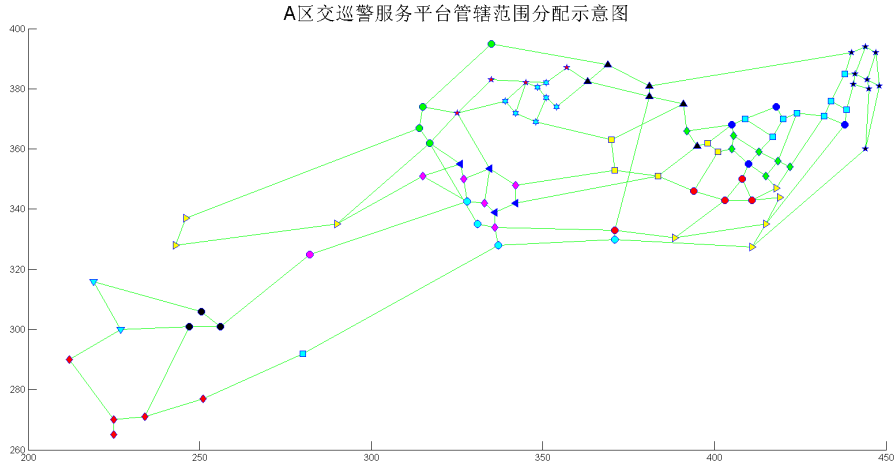


图 1 A 区交巡警服务平台管辖范围分配示意图

5.3.3 结果分析

从表 6 我们可以看出，在使用模拟退火算法得到的管辖分配方式下的平台工作量方差较 5.2 中的有所减小，但是减小的程度很小，极差增大并且平均值有所增加。这是因为本题目中的一些较大的且不可改变的管辖方式（如 15 号平台管辖 28, 29 号路口节点）使得方差不可能很小，在方差稳定在 12 左右且平均值小于方差时，减小方差的方式就变成了增大平均值。具体分析某些管辖分配方式（如 19 号平台管辖 69, 76, 81 号路口节点），对照 5.2 的结果，我们发现一些平台更多地去管辖距离它 3 km 内的较远一些的路口节点，这正是平均值增大的原因。

我们可以通过模拟退火算法的本质来解释这个现象：模拟退火算法对我们给出的初始解随机产生一个扰动，比如更改一组管辖方式，计算方差是否变小，如果不变小，则产生一组新的扰动，如果变小，在变小的基础上继续产生扰动，直到完成一定次数的扰动停止。我们在设定所有可能的扰动时，只选择那些可以被多个平台管辖的路口节点，将它们从被某个平台管辖更换至被另一个平台管辖。模拟退火算法并不像我们手动调整不合理的管辖分配时会优先对明显不合理的一组匹配进行修正，而是完全随机求最优解，所以在求得的结果中会有一些地理分布上不合理的管辖区域，但是这是完全符合我们建立的模型的。在实际确定管辖匹配方式时，可以在这个解的基础上做出一定的调整，使得解符合实际情况。

5.4 区域全封锁的平台调动模型

5.4.1 模型的建立

因为这是图论问题中的一个典型的指派问题，本文考虑引入 0-1 变量 x_{ij} ，表示是否由交巡警平台 i 对出口 j 完成封锁，考虑到完成封锁需要的时间取决于最后一个路口被封锁上所花费的时间，我们将封锁方案中单项最长耗时最小作为目标函数：

$$obj. \min \max_{\substack{i=1,\dots,20 \\ j=1,\dots,13}} \{x_{ij}d_{ij}\}$$

共有 13 个离开 A 区的路口需要封锁，且每一个路口只需一个交巡警平台封锁，得到约束条件：

$$\sum_{i=1}^{20} x_{ij} = 1, j = 1, 2, \dots, 13$$

每一个平台只可能封锁一个离开 A 区的路口，且总共只有 13 个平台参与了封锁路口的任务：

$$\begin{cases} \sum_{j=1}^{13} \sum_{i=1}^{20} x_{ij} = 13 \\ \sum_{j=1}^{13} x_{ij} \leq 1, i = 1, 2, \dots, 20 \end{cases}$$

根据上面的目标函数和约束条件，我们建立了针对封锁路口问题的 0-1 整数规划模型 (1)：

$$obj. \min \max_{\substack{i=1,\dots,20 \\ j=1,\dots,13}} \{x_{ij}d_{ij}\}$$

$$st. \begin{cases} \sum_{i=1}^{20} x_{ij} = 1, j = 1, 2, \dots, 13 \\ \sum_{j=1}^{13} \sum_{i=1}^{20} x_{ij} = 13, \\ \sum_{j=1}^{13} x_{ij} \leq 1, i = 1, 2, \dots, 20 \\ x_{ij} = 0 \text{ 或 } 1 \end{cases} \quad (1)$$

我们可以使用 lingo 软件对这一类问题进行求解。

5.4.2 模型的求解

在使用 lingo 软件求解 (1) 后，我们得到的初步封锁调度方案如下表（表 7）：

表 7 初步封锁调度方案表

交巡警平台位置标号	2	3	5	7	9	10	11	12	13	14	15	16	19
封锁路口节点标号	62	16	30	29	48	12	22	24	23	21	28	14	38

注：表中加粗的数据表示最长耗时出现在该组对应的封锁匹配中。

在这种封锁分配方案下，我们计算总封锁时间为 66.529min，平均耗时 5.118min，最大封锁时间为 7 号平台对 29 号路口节点的封锁，耗时 8.015min。

由于我们在考虑封锁问题时采用的目标函数单项封锁的最长时间最短，没有考虑总

时间最短，即平均用时最短。而在实际问题中，除了最长时间最短外，我们认为总耗时越短越好，因为总耗时越小，交警平台的人力耗费就越小，所以我们将总封锁时间最短列为目标函数，并将(1)中得到的时间结果作为约束条件：

$$x_{ij} * d_{ij} \leq 8.015, i = 1,2,...,20, j = 1,2,...,13$$

这样我们得到一个新的求总工作量最小的 0-1 规划问题(2)：

obj.

$$\min = \sum_{j=1}^{13} \sum_{i=1}^{20} x_{ij} * d_{ij}$$

st.

$$\left\{ \begin{array}{l} x_{ij}d_{ij} \leq 8.015, i = 1,2,...,20, j = 1,2,...,13 \\ \sum_{i=1}^{20} x_{ij} = 1, j = 1,2,...,13 \\ \sum_{j=1}^{13} \sum_{i=1}^{20} x_{ij} = 13, \\ \sum_{j=1}^{13} x_{ij} \leq 1, i = 1,2,...,20 \\ x_{ij} = 0 \text{或} 1, \end{array} \right.$$

(2)

在使用 lingo 软件求解(2)后，我们得到的最优封锁调度方案如下表（表 8），根据表 8 作出了最优的快速全封锁调度方案示意图（图 2）：

表 8 最优封锁调度方案表

交巡警平台位置标号	2	4	5	7	8	9	10	11	12	13	14	15	16
封锁路口节点标号	38	62	48	29	30	16	22	24	12	23	21	28	14

注：表中加粗的数据表示最长耗时出现在该组对应的封锁匹配中。

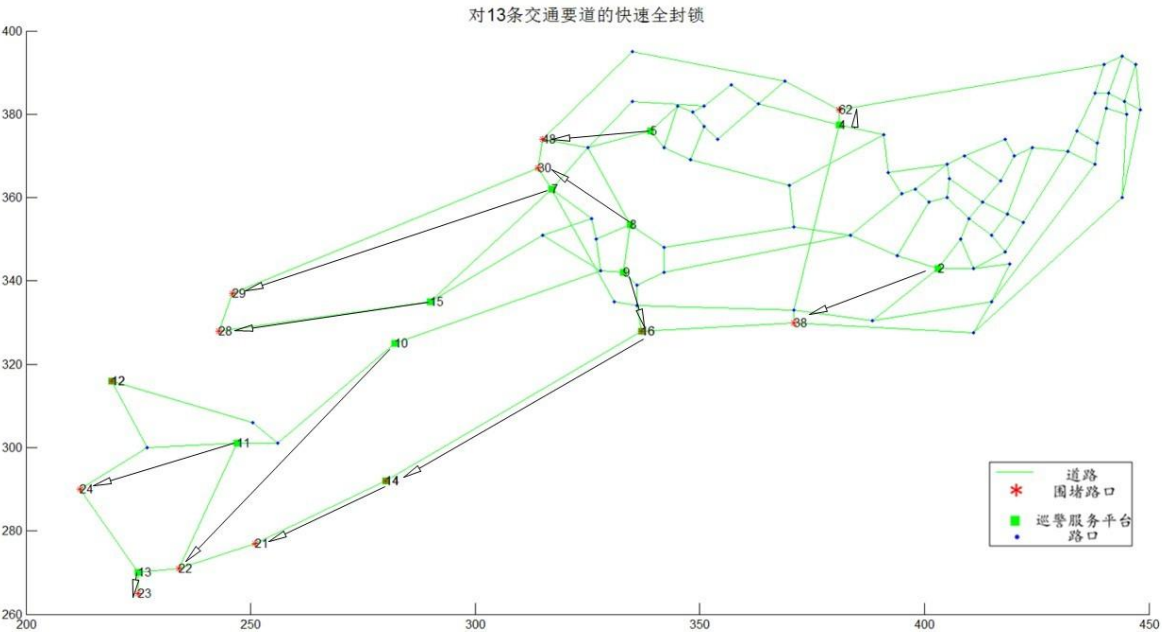


图 2 最优的快速全封锁调度方案示意图

我们发现在这种封锁分配方案下，我们计算总封锁时间为 46.188 min，平均耗时 3.553 min，最大封锁时间仍为 7 号平台对 29 号路口节点的封锁，耗时为 8.015 min。

5.4.3 结果分析

我们对两种模型的求解结果做了进一步的数据处理，求出总耗时和平均耗时等数据，并将两个模型的结果放在一起对比，见下表（见表 9）：

表 9

模型	总耗时 (min)	平均耗时 (min)	最长耗时 (min)	最长耗时对应封锁匹配
模型 (1)	66.529	5.118	8.015	7--29
模型 (2)	46.188	3.553	8.015	7--29

对比两个模型的求解结果我们发现，最长耗时都是由平台 7 封锁路口 29 造成的 8.015 min，但模型 (2) 的求解结果中平均封锁耗时为 3.553 min，比模型 (1) 的 5.118 min 有了很大的减少，通过观察封锁示意图我们看到，14 和 16 号两个平台虽然自身也为 A 区出口，但他们并不封锁自己，这并不说明我们的新的求解结果是不合理的，正是因为我们目标函数更改为总封锁时间最小，这种封锁分配的调整对封锁总时间降低带来的影响大于对个别平台时间增加的影响，就可能使程序给出了让一些平台去封锁较远的路口的结果，这说明我们将两个模型联合求解的思路是正确的。

5.5 确定增加交巡警服务平台个数和位置的模型

5.5.1 模型的建立

通过 5.1 中的求解，我们已经得到了标号为 28, 29, 38, 39, 61, 92 的 6 个节点距离任何一个平台的时间均超过了 3 km，所以我们增加新平台的主要目标就是在管辖这些节点时，使得到警时间均在 3 min 之内，其次在增加平台时，我们还要考虑新平台的建立对其他平台工作量的影响程度，我们尽可能的选取那些可以使各平台工作量的方差降低的位置修建新的平台。这样既可以使所有的路口节点都在 3 min 内可到警，又可以使管辖分配方式更加均衡。

在选取新建平台位置时，我们不能只考虑这 6 个特殊节点，距这些节点 3 km 内的其它路口节点均可以考虑，通过筛选我们得到了所有的可新建平台节点见下表（表 10）：

表 10 可新建交巡警平台的路口节点标号表

到警时间超过 3 min 的路口标号	与该点距离小于 3 km 的节点标号	到警时间超过 3 min 的路口标号	与该点距离小于 3 km 的节点标号
28	29	39	38, 40
29	28	61	48
38	39, 40	92	87, 89, 90, 91

5.5.2 模型的求解

根据上面的分析，我们将所有可新建交巡警平台的点分为以下四组（表 11）：

表 11 可新建交巡警平台的路口节点分组示意表

组别	1	2	3	4
道路节点标号	28,29	38,39,40	48,61	87,89,90,91,92

由表11我们看到所有的可新建平台节点可分为4组，为了满足所有路口节点均在3 min内可到警的条件，每一组至少要修建一个新平台，所以我们至少要新建4个交巡警平台。

在每组内部选择哪个节点作为新建交巡警平台位置时，我们在该组区域内局部使用模拟退火算法，找到使总工作量方差减小最多的节点，则该节点即可作为该分组区域内的新建平台位置。

我们不将各分组内每一节点作为新建平台下的具体管辖关系和方差计算结果列出，只列出我们得出的每一组内部最佳新建平台位置和相应的工作量方差（见表12）。

表 12 最优新建交巡警平台的路口节点示意表

组别	可修建平台标号	最优新建 平台编号	新平台管辖路口 节点标号	新平台 工作量	平均工 作量	方差
1	28, 29	29	28, 29	1.23	6.39	9.94
2	38, 39, 40	40	38, 39, 40	4.96	6.88	10.34
3	48, 61	48	30, 48, 61	1.74	6.58	12.21
4	87, 89, 90, 91, 92	92	88, 89, 92	10.18	6.72	10.12

5.5.3 结果分析

通过表 12 我们看到，在逐个新建平台时平均工作量和 workload 方差都有一定程度的减小。我们将上表中的数据整合到 A 区的所有平台管辖方式中，计算了在增加此 4 个平台后 A 区的管辖分配方式。经过模拟退火算法计算得，此时方差为 6.28，平均值为 5.08 min，这时的数据已经较 5.3 中的结果有了很大的改进，说明在我们的这种新建平台方案下，整个 A 区的交巡警服务质量有了巨大的提升。此时 A 区的全部管辖分配方式见附录二。因为我们已经达到了 3 min 内到警率为 100% 的目标，并且方差已经减小到了一个很可观的范围，再增加第五个平台时，对 workload 方差的降低作用并不会很明显，反而会

有成本的增加，所以我们不考虑增加第五个平台的方案。

5.6 交巡警服务平台设置合理性的评价模型

5.6.1 模型的建立

本问题要求我们对各区域的服务平台设置的合理性进行分析，我们建立了模糊综合评价模型。通过该模型得到的每个地区的综合评分，得分最低的地区，即交巡警服务平台方案设置最不合理的地区。

本文给出的模糊综合评价模型的评判指标如下：

- (1) 各区交巡警 3 min 内到达路口节点的比率；
- (2) 各区交巡警平台日均处理案件总数（次）；
- (3) 各区交巡警平台平均服务人口总数（万人）；
- (4) 各区平台间工作量方差。

针对这四个评价指标，我们首先得到一个初始矩阵 $I_{6 \times 4}$ ，矩阵 I 中的元素就是各区域评判指标的实际计算数值。然后我们为四个评判指标分别定义隶属度函数 $\mu_1(x), \mu_2(x), \mu_3(x), \mu_4(x)$ ，根据隶属度函数，我们可以计算出模糊关系矩阵 $R_{6 \times 4}$ 。然后通过变异系数加权法给出四个评价指标在决策中占的权重，记权重向量为 $\omega = (\omega_1, \omega_2, \omega_3, \omega_4)$ 。

于是得到所有区域的综合评价为 $P = \omega R^T$ ，其中 $P = (p_1, p_2, p_3, p_4, p_5, p_6)$ 为综合评价向量，通过比较 P 中元素的大小就可以得到综合评价最差的区域，即为我们分析出的最不合理的区域。

其中变异系数加权法的公式如下：

$$\begin{cases} V_i = \frac{\sigma_i}{\bar{x}_i}, & i = 1, 2, \dots, n \\ \omega_i = \frac{V_i}{\sum_{i=1}^n V_i}, & i = 1, 2, \dots, n \end{cases}$$

式中， V_i 是第 i 项指标的变异系数，也称为标准差系数； σ_i 是第 i 项指标的标准差； \bar{x}_i 是第 i 项指标的平均数， ω_i 是各项指标的权重。

5.6.2 模型的求解

我们首先对全市的 B, C, D, E, F 五个区采用 floyd 算法计算出各区域中的交巡警范围平台到各路口节点的最短距离，统计出其中不满足 3 min 内到警的路口节点数，并且算出各区的 3 min 到警率；同时我们根据题目附件中的各区人口数量和各节点日均发案率的数据，计算出了各区域的日均处理案件数量以及各区域平台服务人口总数；然后我们采用了 5.3 中的模拟退火算法对这五个区域分别作出了最优的管辖分配方案，计算了各区的工作量方差值，这些统计量见下表（表 13）：

表 13 全市各区域评判指标表

区域	3 min 到警率	日均处理案件数（次）	每平台服务人口总数（万人）	工作量方差
A	0.9348	6.23	3.00	11.415
B	0.9178	8.30	3.48	8.27
C	0.6948	11.01	2.88	192.23
D	0.7692	7.53	8.11	75.69
E	0.6893	7.96	5.07	299.82
F	0.6759	9.93	4.82	132.31

根据表 13 我们得到了初始矩阵

$$I = \begin{bmatrix} 0.9348 & 6.23 & 3.00 & 11.415 \\ 0.9178 & 8.30 & 3.48 & 8.27 \\ 0.6948 & 11.01 & 2.88 & 192.23 \\ 0.7692 & 7.53 & 8.11 & 75.69 \\ 0.6893 & 7.96 & 5.07 & 299.82 \\ 0.6759 & 9.93 & 4.82 & 132.31 \end{bmatrix},$$

然后我们确定隶属度函数，3 min 到警率是正向指标，越大越好，所以我们设定该项指标的隶属度函数为

$$\mu_1(x) = x,$$

日均处理案件数表示该地区的案件发生多少，越高说明该地区越不安全，是负向指标，所以我们设定该项指标的隶属度函数为

$$\mu_2(x) = 1 - \frac{x}{15},$$

平台服务人口总数表示该地区的警力匹配质量，数量越高说明质量越低，为负向指标，所以我们设定该项指标的隶属度函数为

$$\mu_3(x) = 1 - \frac{x}{10},$$

各地区工作量的方差衡量某地区的平台管辖分配的均衡性，方差越小说明分配越均衡，是负向指标，所以我们设定该项指标的隶属度函数为

$$\mu_4(x) = 1 - \frac{x}{300}.$$

根据隶属度函数，我们计算得到模糊关系矩阵

$$R = \begin{bmatrix} 0.9348 & 0.5847 & 0.7000 & 0.9620 \\ 0.9178 & 0.4467 & 0.6520 & 0.9724 \\ 0.6948 & 0.2660 & 0.7120 & 0.3592 \\ 0.7692 & 0.4980 & 0.1890 & 0.7477 \\ 0.6893 & 0.4693 & 0.4930 & 0.0006 \\ 0.6759 & 0.3380 & 0.5180 & 0.5590 \end{bmatrix}.$$

我们使用变异系数加权法来计算四项评价指标的权值，代入公式得到权重向量为：

$$\omega = (0.132 \quad 0.211 \quad 0.268 \quad 0.389),$$

然后计算综合评价向量

$$P = \omega R^T = (0.8086 \quad 0.7684 \quad 0.4784 \quad 0.5481 \quad 0.3224 \quad 0.5168),$$

所以我们得出该市六区域的综合评价排名为： $A > B > D > F > C > E$ ，区域 E 是综合评价价值最小的，所以我们认为 E 区是该城市中交巡警平台分布最不合理的区域，需要对交巡警平台的设置进行优化。

5.6.3 结果分析

我们分析权重矩阵 $\omega = (0.132 \quad 0.211 \quad 0.268 \quad 0.389)$ ，发现第四项指标“工作量方差”的权重最大，再对照初始矩阵 I 发现区域 E 的方差较其它区间相比大很多，通过观察该市交通网络区域示意图，我们发现 E 区面积较大，跨度很大，而且单条道路都比较长，其中 3 min 内不可到警的节点数多达 32 个，而且 E 区交巡警平台数所占比例较小，这些节点极大地增加了平台管辖时的工作量方差，这正是区间 E 综合评价价值最小的主要原因。

同时我们发现区域 A 和 B 的综合评价价值位于前两名，通过观察该市交通网络区域示意图，我们可以明显看到 A 和 B 区面积较小并且道路紧凑，属于城市的中心城区，交通非常发达，通过分析计算结果我们看到这两个区域的四项评价指标都较高，“3 min 到警率”和“工作量方差”都明显比其他几个区域优，其他指标没有明显的短板，这正是 A 和 B 两个区域综合评价价值最高的原因。

5.6.4 解决方案

对不合理方案的修改是相当复杂的，需要考虑实际因素，既不能将已有设置全部推翻重来，也不能不计成本地加入很多的新平台。我们在考虑对不合理的区域进行优化时，只要优化后的该区域各项指标相较原来有了很大的改进即可。基于这种想法，我们采用类似于 5.5 中的优化方法给 E 区增加新的交巡警平台，这样可以有效优化四个评价指标，尤其是减小工作量方差。我们通过模拟退火算法给出向 E 区增加新的平台方案以及增加后的各项指标的变化表（见表 14）：

表 14 E 区增加交巡警平台前后的各项评价指标表

增加平台标号	增加平台个数	3 min到警率	日均处理案件数（次）	每平台服务人口总数（万人）	工作量方差
无	0	0.6893	7.96	5.07	299.82
388, 392, 402, 411, 418	5	0.8766	7.96	3.8	49.9161

通过表 14 我们可以看到 E 区的 3 min 到警率、工作量方差和每平台服务人口总数均

比原来的值有了明显改进。我们做出此时 E 区的交巡警平台分布示意图（图3）：

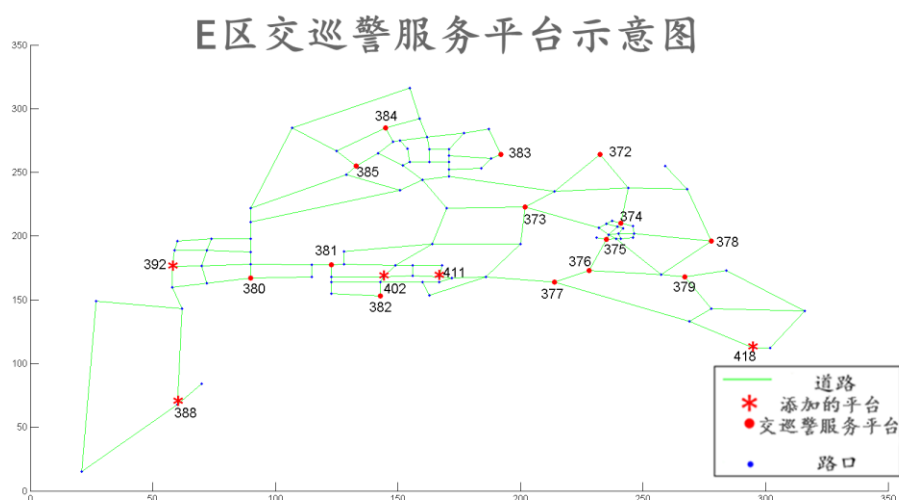


图3 优化后的 E 区的交巡警平台分布示意图

与处理 E 区相类似的方法，我们同样对方差大于100的其他两个区域 C 和 F 进行增加交巡警平台的处理，它们增加后的各项指标的变化表（见表15，表16）：

表15 C 区增加交巡警平台前后的各项评价指标表

增加平台标号	增加平台个数	3 min到警率	日均处理案件数（次）	每平台服务人口总数（万人）	工作量方差
无	0	0.6948	11.01	2.88	192.23
207, 239, 262, 286, 303, 313	6	0.8252	11.01	2.13	37.9410

表16 F 区增加交巡警平台前后的各项评价指标表

增加平台标号	增加平台个数	3 min到警率	日均处理案件数（次）	每平台服务人口总数（万人）	工作量方差
无	0	0.6759	9.93	4.82	132.31
491, 509, 518, 525, 574	5	0.9074	9.93	3.31	32.8513

我们在所有给出的新建平台全部建成的条件下，重新根据模糊综合评价的方法对这六个区进行综合评价，经计算，我们得到综合评价矩阵为：

$$P = (0.7990 \quad 0.7482 \quad 0.6061 \quad 0.3290 \quad 0.5619 \quad 0.6107),$$

这时 A, B 两区域仍然是综合评价值最高的两个区域，区域 D 成为了综合评价值最低的区域，是目前交巡警服务平台设置安排最不合理的区域，区域 C, E, F 在增加新的平台后，

综合评价值有了很大的提升，这正是我们所期待的，也说明我们的改进措施较为合理。

5.7 全市交巡警服务平台最佳围堵模型

5.7.1 模型的建立

考虑到本问题涉及最大匹配和最优搜索问题，本文建立了基于二分法、匈牙利算法的最快围捕方案搜索算法，算法如下：

- (1) 给定一个时间上下限 $[a,b]$ min;
- (2) 计算嫌疑犯在时间 $\tau = (a+b)/2$ 内可能逃窜到的区域 Q ，并记区域 Q 的边界为 ∂Q ;
- (3) 由匈牙利算法计算全市所有平台封锁 ∂Q 中所有节点的最大匹配，并记每一项匹配耗时为 T_i ;
- (4) 由 floyd 算法计算嫌犯从 P 节点到达 ∂Q 中各节点的时间，记到达各节点的耗时分别为 t_i ;
- (5) 若 τ 的精度足够大，则停止算法， $T = \max\{T_i\}$ 即为完成围堵的最短时间，否则，进入下一步;
- (6) 若 $T_i \geq t_i$ ，则令 $b = \tau$ ，返回第 (2) 步; 若 $T_i < t_i$ ，则令 $a = \tau$ ，返回第 (2) 步。

通过这个最快围捕方案搜索算法，在得到最快围堵时间的同时，我们也可以得到最佳围堵方案。

5.7.2 模型的求解

我们根据 5.7.1 中给出的最快围捕方案搜索算法，设定时间上下限为 $[3,10]$ ，通过 MATLAB 程序运行，得到的结果如下表（表 17）：

表 17 最快围捕方案表

路口标号	平台标号	嫌疑人到达时间	平台封锁时间	路口标号	平台标号	嫌疑人到达时间	平台封锁时间
70	1	6.39	1.04	227	170	6.84	2.59
43	2	6.45	0.80	217	171	6.45	2.21
68	3	6.13	2.68	218	172	6.22	0.47
62	4	6.73	0.35	239	173	9.48	6.57
26	11	7.04	0.90	215	174	6.03	4.86
14	14	6.16	0	273	182	7.15	2.10
29	15	12.89	5.70	371	320	9.81	7.36
17	17	7.50	0	558	475	8.17	2.05
41	18	7.65	5.54	549	476	11.64	3.03
76	19	6.24	1.43	562	480	7.72	1.94
168	168	6.15	0	487	481	8.10	4.56
240	169	8.25	7.05	482	482	8.40	0

注：表中的加粗字体为围捕中耗时最长的单项封锁

我们从表 17 中得出，完成围捕所需最短时间为 7.36 min，是由 320 号交巡警平台封锁 371 号路口节点使用的。我们画出这种围堵方案示意图（见图 4）：

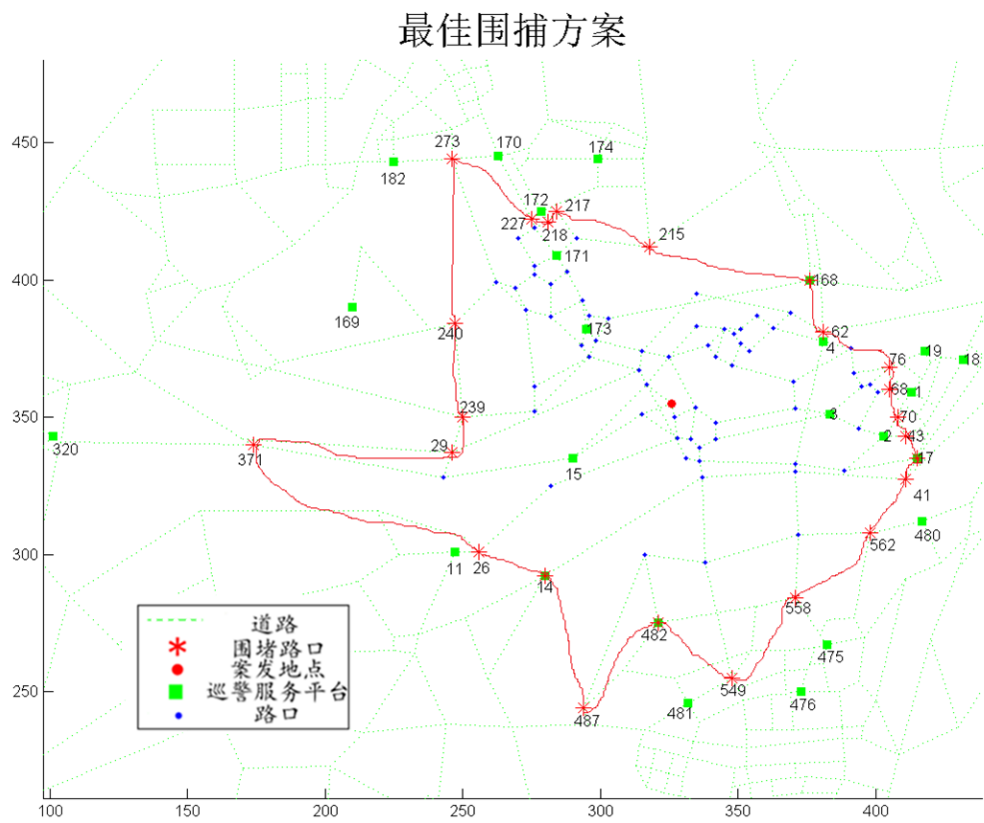


图 4 最快围捕方案示意图

5.7.3 结果分析

我们通过最快围捕方案示意图看到，该嫌犯的所有可能逃离该城市的道路确实已经被封堵完全，该次围堵总共动员了全市的 24 个交巡警平台，其中需耗时最长的是 320 号平台对 371 号节点的封锁，耗时为 7.36 min。

六、模型的评价

本文中建立的模型有效的解决了合理分配交巡警平台的管辖范围问题，警力资源的分配以及对各路口的有效封锁问题。整个模型的建立思路清晰，可操作性强，该模型求解了在较为理想状态下交巡警平台的最优设置，降低了各平台工作量的方差，使得各平台工作量较为均衡。考虑围堵封锁问题时，在满足时间约束的条件下继续对结果进行优化，使得各平台工作量尽量均衡。但该模型也有一定的局限性，如模型建立在理想化的环境中，如道路的畅通性，出警车辆和人员配备的可行性等忽略了生活中存在的不定

因素。

优点：

1. 本文建立的模型很好的解决了交巡警服务平台的设置与管辖范围的分配问题，有助于对警力资源分配的优化，同时提高治安水平和工作效率，在实际问题中具有重要的参考意义；
2. 本文将实际问题抽象成了图论模型、规划模型和优化模型，简化了问题的难度，科学正确地分析和解决了问题；
3. 本文引入的模拟退火算法适用性很强，可以有效求出较优解；
4. 在评价六个城区平台设置方案的合理性问题时，计算使用到的几个评价指标对实际问题中不同地区都具有普适性，我们引入的模糊综合评价模型适用性强。

缺点：

1. 本文的部分模型假设过于理想化，比如道路通行能力、案件发生位置等在实际问题中会更为复杂，在使用时需要作出调整；
2. 考虑嫌疑人逃跑时，仍按照 60 km/h 的速度，没有考虑不同的速度情况；
3. 本文在对某区的平台设置进行优化时，没有考虑跨区管辖的情况；

七、模型的改进和推广

本题目是一个分配调度问题，分配调度问题是运筹学的一个重要分支，它在解决城市建设、公共设施安置、工业生产选址等中都发挥着重要的作用。

本文中建立的“交巡警平台管辖范围分配”算法具有很强的适用性，可以推广至超市、医院、消防、物流中心和志愿者服务站等城市基础服务部门的设点问题，本文中建立的诸多模型能够对已知相关数据的区域或路网进行合理规划，实现城市基础设施设置的合理性。

本文中引入的模拟退火算法在解决建模问题中有很大的作用，其智能化的计算方法在解决优化问题时有很大的优势。但是该算法可能只能求出局部最优解，因为在使用时对初值的质量要求较高。

八、参考文献

- [1]姜启源，谢金星. 数学模型（第三版）[M]. 北京：高等教育出版社，2003：82-130.
- [2]袁新生，邵大宏. LINGO 和 EXCEL 在数学建模中的应用[M]. 北京：科学出版社，2007：50-62.
- [3]谢金星，薛毅. 《优化建模与 LINDO/LINGO 软件》，清华大学出版社 2011.
- [4]刘卫国. 《MATLAB 程序设计与应用》，高等教育出版社. 2012
- [5]甘应爱等. 《运筹学》，清华大学出版社. 2005
- [5]高尚. 模拟退火算法中的退火策略研究[J]. 航空计算技术，2002，32(4):20-26.

附 录

附录一：可被多个平台管辖的节点及可管辖它们的平台标号表

可被多个平台管辖的道路节点标号	可管辖该节点的平台标号	可被多个平台管辖的道路节点标号	可管辖该节点的平台标号
25	11, 12	67	1, 2, 3, 19
31	7, 8, 9, 15	68	1, 2, 3, 19
32	7, 8, 9	69	1, 2, 19
33	7, 8, 9, 16	70	1, 2, 3, 17, 19
34	7, 8, 9, 16	71	1, 2, 18, 19
35	8, 9, 16	72	1, 2, 17, 18
36	8, 9, 16	73	1, 2, 18, 19
37	8, 9, 16	74	1, 2, 18, 19
40	2, 17	75	1, 2, 19
42	1, 2, 17	76	1, 2, 3, 19
43	1, 2, 3, 17	77	1, 18, 19
44	1, 2, 3	78	1, 2, 18, 19
45	8, 9, 16	79	1, 18, 19
46	8, 9, 16	80	1, 18, 19
47	5, 6, 7, 8	81	18, 19, 20
48	5, 6, 7	82	18, 19, 20
50	5, 6	83	18, 19, 20
51	5, 6	84	18, 20
52	5, 6	85	18, 20
56	5, 6	87	18, 20
58	4, 5	88	18, 20
59	5, 6	89	18, 20
64	1, 3, 4, 19	90	18, 20
65	1, 3, 4, 19	91	18, 20
66	1, 2, 3, 4, 19		

附录二：新建交巡警服务平台后的 A 区管辖分配方式表

交巡警平台位置标号	平台管辖路口节点标号	工作量	交巡警平台位置标号	平台管辖路口节点标号	工作量
1	1, 71, 74, 75, 76, 78	3.71	13	13, 21, 22, 23, 24	0
2	2, 42, 44, 68, 70	12.7	14	14,	0
3	3, 54, 55, 66, 67	6.12	15	15, 31	0.11

4	4, 57, 60, 62, 63, 64	6.55	16	16, 35, 36, 37	5.57
5	5, 49, 51, 53, 56, 58, 59	7.94	17	17, 41, 43, 72	9.43
6	6, 50, 52	4.92	18	18, 73, 80, 81, 82, 84, 85	3.44
7	7, 47	2.01	19	19, 65, 69, 77, 79	6.54
8	8, 33, 45	7.79	20	20, 83, 86, 87, 90, 91	10.2
9	9, 32, 34, 46	0	29	28, 29	1.74
10	10,	2.39	40	38, 39, 40	4.96
11	11, 26, 27	2.86	48	30, 48, 61	1.23
12	12, 25	8.88	92	88, 89, 92	3.71

附录三：lingo 程序代码

```

model:
!20个警察局, 13个路口的分配问题;
sets:
police/w1..w20/;
cross/j1..j13/;
links(police,cross): cost,volume;
endsets
!目标函数;
min=@max(links: cost*volume);
!有13个匹配;
@sum(links:volume)=13;
@for(links:@bin(volume));
!每个警察局最多只能管一个路口;
@for(police(I):
@sum(cross(J): volume(I,J))<=1;
);
!每个路口只能有一个警察局管;
@for(cross(J):
@sum(police(I): volume(I,J))=1;
);
obj_value1=@sum(links: cost*volume);
data:
cost=@ole('cost_volume.xls',cost);
@ole('cost_volume.xls')=volume,obj_value1;
enddata
end

```

附录二：部分 MATLAB 程序代码

```

%模拟退火
%添加平台后求解新的分配方式
%建立初始解, 都找最近的平台
jiapingtai=[92,48,40,29];%添加的平台
col=20;
pingtai=[1:20,jiapingtai];
[M,pingtaishu]=size(pingtai);
jianshi2=ones(1,92);%添加平台后的监视方案

```

```

for i=1:92
    jianshi2(i)=pingtai(1);%假设与第一个警察局最近
    for j=1:pingtaishu %对于第 i 个没警察局的点，寻找最近的警察局
        if(a(jianshi2(i),i)>a(pingtai(j),i))
            jianshi2(i)=pingtai(j);
        end
    end
end
end
multijpt0=zeros(pingtaishu,col);%初始解，都找最近的平台
for i=1:92
    [j,k]=find(pingtai==jianshi2(i));
    [M,N]=size(find(multijpt0(k,:)==0));
    multijpt0(k,N+1)=i;
end

e=0.1^30;L=20000000;at=0.9999;T=1;
multi01=zeros(pingtaishu,col);
multi00=zeros(pingtaishu,col);
multi01=multijpt0;%multi01 为新生成的安排方式
multi00=multijpt0;
gzltemp=zeros(1,pingtaishu);
for i=1:pingtaishu %求工作量
    for j=1:col
        if(multi00(i,j)~=0)
            gzltemp(i)=gzltemp(i)+a(pingtai(i),multi00(i,j))*afcs(multi00(i,j));
        end
    end
end
end
gzltemp=gzltemp*0.1;
fc0=var(gzltemp);%求出初始方差记为 fc0

%get multi_temp2 & multi2
%duopingtshu=55;
%multi_temp2=zeros(1,duopingtshu);

jilupingtaishu=zeros(1,92);%计算每个点的可用平台数

```

```

for ii=1:92
    for kk=1:pingtaiashu
        if(a(pingtai(kk),ii)<30)
            jilupingtaishu(ii)=jilupingtaishu(ii)+1;
        end
    end
end
jj=1;
for ii=1:92
    [M,N]=size(find(pingtai==ii));%N 为 0 时 表示 ii 不是平台
    if(~N && jilupingtaishu(ii)>1)%不是平台而且可用平台数大于 1 则记录
        multi_temp2(jj)=ii;
        jj=jj+1;
    end
end
[temp,duopingtaishu]=size(multi_temp2);
multi2=zeros(duopingtaishu,col);
for ii=1:duopingtaishu %49
    for jj=1:pingtaiashu %21
        if(a(multi_temp2(ii),pingtai(jj))<30)%如果 ii 点离 pingtai(jj)距离小于 30，把 pingtai(jj)
加进来
            [M,N]=size(find(multi2(ii,:)));
            multi2(ii,N+1)=pingtai(jj);
        end
    end
end
end

%退火过程
for k=1:L
    if rand>0.5 %换一个

    index=ceil(rand*duopingtaishu);%第 index 个城市
    sign=multi_temp2(index);%城市编号
    [M,N]=find(multi01==sign);
    multi01(M,N:col)=[multi01(M,N+1:col),0];%在原位置去掉那个城市
    %multi01(i,j:col)=[multi01(i,j+1:col),0];
    for kk=1:col

```

```

        if(multi2(index,kk)~=0)
            N=kk;
        end
    end
    %to=multi2(index,ceil(rand*N));%to 为随机找个可以管的警察局行丢过去~
    [temp,to]=find(pingtai==multi2(index,ceil(rand*N)));
    for jj=1:col
        if(multi01(to,jj)~=0)%N 表示该行的数据个数
            N=jj;
        end
    end
    multi01(to,N+1)=sign;

else %换两个

index=ceil(rand*duopingtaishu);%第 index 个城市
sign=multi_temp2(index);%城市编号
[M,N]=find(multi01==sign);
multi01(M,N:col)=[multi01(M,N+1:col),0];%在原位置去掉那个城市
%multi01(i,j:col)=[multi01(i,j+1:col),0];
for kk=1:col
    if(multi2(index,kk)~=0)
        N=kk;
    end
end
%to=multi2(index,ceil(rand*N));%to 为随机找个可以管的警察局行丢过去~
[temp,to]=find(pingtai==multi2(index,ceil(rand*N)));
for jj=1:col
    if(multi01(to,jj)~=0)%N 表示该行的数据个数
        N=jj;
    end
end
multi01(to,N+1)=sign;

index=ceil(rand*duopingtaishu);%第 index 个城市
sign=multi_temp2(index);%城市编号
[M,N]=find(multi01==sign);

```

```

multi01(M,N:col)=[multi01(M,N+1:col),0];%在原位置去掉那个城市
%multi01(i,j:col)=[multi01(i,j+1:col),0];
for kk=1:col
    if(multi2(index,kk)~=0)
        N=kk;
    end
end
%to=multi2(index,ceil(rand*N));%to 为随机找个可以管的警察局行丢过去~
[temp,to]=find(pingtai==multi2(index,ceil(rand*N)));
for jj=1:col
    if(multi01(to,jj)~=0)%N 表示该行的数据个数
        N=jj;
    end
end
multi01(to,N+1)=sign;

end

gzltemp=zeros(1,pingtaishu);
for i=1:pingtaishu %求工作量
    for j=1:col
        if(multi01(i,j)~=0)
            gzltemp(i)=gzltemp(i)+a(pingtai(i),multi01(i,j))*afcs(multi01(i,j));
        end
    end
end
gzltemp=gzltemp*0.1;
fc1=var(gzltemp);%求出新方差记为 fc1

if(fc1<fc0)%新方差较小，接受新结果
    multi00=multi01;
    fc0=fc1
elseif(exp((fc0-fc1)/T)>rand)
    multi00=multi01;
    fc0=fc1;
end

```

```

        T=T*at;
        if T<e
            break;
        end
    end
end
sum(gzltemp)
sum(gzltemp)/pingtaishu

```

```

%计算新的覆盖率
atemp=a2(372:474,372:474);
afcstemp=afcs(372:474);
pingtaishu=15;
lukoushu=103;
col=35;

```

```

jiapingtai=[17 21 40 47 31];%添加的平台
pingtai=[1:pingtaishu,jiapingtai];
[M,pingtaishu]=size(pingtai);

```

```

dutym=zeros(103);
for i=1:pingtaishu %遍历平台
    for j=1:103 %遍历路口
        % [temp,ii]=find(pingtai==i-474);
        ii=pingtai(i);
        if(atemp(ii,j)<30)
            dutym(ii,j)=1;
        end
    end
end
end

```

```

inneed=find(sum(dutym)==0);%寻找没警察管的点，用行向量 inneed 记录
[m,n]=size(inneed)

```

```

%匈牙利

```



```

function [assignment,cost] = munkres(costMat)

assignment = zeros(1,size(costMat,1));
cost = 0;

validMat = costMat == costMat & costMat < Inf;
bigM = 10^(ceil(log10(sum(costMat(validMat))))+1);
costMat(~validMat) = bigM;

validCol = any(validMat,1);
validRow = any(validMat,2);

nRows = sum(validRow);
nCols = sum(validCol);
n = max(nRows,nCols);
if ~n
    return
end

maxv=10*max(costMat(validMat));

dMat = zeros(n) + maxv;
dMat(1:nRows,1:nCols) = costMat(validRow,validCol);

minR = min(dMat,[],2);
minC = min(bsxfun(@minus, dMat, minR));

zP = dMat == bsxfun(@plus, minC, minR);

starZ = zeros(n,1);
while any(zP(:))
    [r,c]=find(zP,1);
    starZ(r)=c;
    zP(r,:)=false;
    zP(:,c)=false;
end

```

```

while 1
    if all(starZ>0)
        break
    end
    coverColumn = false(1,n);
    coverColumn(starZ(starZ>0))=true;
    coverRow = false(n,1);
    primeZ = zeros(n,1);
    [rIdx,                                cIdx]                                =
find(dMat(~coverRow,~coverColumn)==bsxfun(@plus,minR(~coverRow),minC(~coverColumn)));
    while 1

        cR = find(~coverRow);
        cC = find(~coverColumn);
        rIdx = cR(rIdx);
        cIdx = cC(cIdx);
        Step = 6;
        while ~isempty(cIdx)
            uZr = rIdx(1);
            uZc = cIdx(1);
            primeZ(uZr) = uZc;
            stz = starZ(uZr);
            if ~stz
                Step = 5;
                break;
            end
            coverRow(uZr) = true;
            coverColumn(stz) = false;
            z = rIdx==uZr;
            rIdx(z) = [];
            cIdx(z) = [];
            cR = find(~coverRow);
            z = dMat(~coverRow,stz) == minR(~coverRow) + minC(stz);
            rIdx = [rIdx(:);cR(z)];
            cIdx = [cIdx(:);stz(ones(sum(z),1))];
        end
        if Step == 6

```

```

[minval,rIdx,cIdx]=outerplus(dMat(~coverRow,~coverColumn),minR(~coverRow),minC(~coverColumn));
    minC(~coverColumn) = minC(~coverColumn) + minval;
    minR(coverRow) = minR(coverRow) - minval;
else
    break
end
end

rowZ1 = find(starZ==uZc);
starZ(uZr)=uZc;
while rowZ1>0
    starZ(rowZ1)=0;
    uZc = primeZ(rowZ1);
    uZr = rowZ1;
    rowZ1 = find(starZ==uZc);
    starZ(uZr)=uZc;
end
end

rowIdx = find(validRow);
colIdx = find(validCol);
starZ = starZ(1:nRows);
vIdx = starZ <= nCols;
assignment(rowIdx(vIdx)) = colIdx(starZ(vIdx));
pass = assignment(assignment>0);
pass(~diag(validMat(assignment>0,pass))) = 0;
assignment(assignment>0) = pass;
cost = trace(costMat(assignment>0,assignment(assignment>0)));

function [minval,rIdx,cIdx]=outerplus(M,x,y)
ny=size(M,2);
minval=inf;
for c=1:ny
    M(:,c)=M(:,c)-(x+y(c));
    minval = min(minval,min(M(:,c)));
end
[rIdx,cIdx]=find(M==minval);

```