

# 关于警察抓小偷的程序设计

张子晗

2019 年 6 月

## Contents

<b>1 问题重述</b>	<b>2</b>
1.1 基础问题 . . . . .	2
1.2 问题延伸 . . . . .	2
<b>2 问题解决</b>	<b>2</b>
2.1 基本模型 . . . . .	2
2.1.1 模型简介 . . . . .	2
2.1.2 模型思路 . . . . .	2
2.1.3 模型反思 . . . . .	3
2.2 拉格朗日乘数法 . . . . .	3
2.2.1 模型简介 . . . . .	3
2.2.2 模型思路 . . . . .	4
2.2.3 模型反思 . . . . .	6
2.3 权重图 . . . . .	6
2.3.1 模型简介 . . . . .	6
2.3.2 Dijkstra 算法 . . . . .	6
2.3.3 模型思路 . . . . .	7
<b>3 关于动态规划的一些思考</b>	<b>7</b>
3.1 基本模型 . . . . .	7
3.1.1 模型思路 . . . . .	7
3.1.2 模型反思 . . . . .	8
3.2 马尔科夫决策过程 . . . . .	8
3.2.1 模型介绍 . . . . .	8
3.2.2 模型思路 . . . . .	9
3.3 模型反思 . . . . .	10
<b>4 模型思路汇总</b>	<b>10</b>
<b>5 结果呈现</b>	<b>10</b>

# 1 问题重述

## 1.1 基础问题

假设在给定的  $10 \times 10$  方格中随机设定警察局的坐标为某格点上, 设有  $p_{total}$  个警察单位, 且方格边长为  $l$ 。若在特定时间点上同时爆发  $k$  起犯罪行为, 犯罪地点的坐标亦位于随机设定的格点上, 每处的罪犯数量为随机数生成的  $b_i, i \in [1, k], b_i \in [3, 10]$ 。与此同时, 警察从警察局出发前去控制犯罪, 选取最短路径行动, 且速度为常量  $v$ 。当警察到达现场后, 便对强盗展开抓捕行动。若在某个地点存在  $p$  个警察和  $b$  个强盗, 则抓捕行动持续  $\frac{b^2}{p}$  个时间单位。为量化强盗所造成的破坏, 在警察未到达犯罪现场的时间内, 每一时间单位中 1 个强盗可抓 2 个良民单位; 在抓捕行动的过程中, 1 个强盗可抓 1 个良民单位。请问如何通过最优化的警察分配方式, 可使得被抓取的总良民单位数量最小?

## 1.2 问题延伸

- (1) 如果地图并非两格点之间距离为定值的方格, 而是形状任意的加权图, 会对上述结论造成什么影响?
- (2)  $a, k, b_i$  均为随机数生成, 如何设置其边界值使其相对平衡并符合真实情况?
- (3) 如果犯罪行为非同时爆发而存在未知的时间间隔, 会对上述结论有什么影响?
- (4) 如某一犯罪地点的警察提前完成抓捕行动, 则其可派遣到其他地点进行协助抓捕。如此安排会对上述结论造成什么影响?

# 2 问题解决

## 2.1 基本模型

### 2.1.1 模型简介

该模型在定义域范围内利用随机数生成参数, 如警察局坐标, 犯罪地点坐标, 强盗数量等, 计算坐标之间相对曼哈顿距离之和的最小值, 来计算警察在行进过程中的损失最小值, 并结合迭代算法遍历符合限制条件 ( $\sum$  为定值) 的警察分布情况, 计算警察在抓捕过程中的损失最小值, 综合考量选取最优解。

### 2.1.2 模型思路

- (1) 假定  $p_{total} = 100, l = 1, v = 1, k = 5$ , 并在程序开头为变量赋值。<sup>1</sup>
- (2) 利用随机数生成警察局  $(x_0, y_0)$  及犯罪地点  $(x_i, y_i, i \in [1, k])$  的坐标值, 相互之间不重复, 存储到列表 `locations_set` 中 (其中 `locations_set[0]` 为  $(x_0, y_0)$ )。

<sup>1</sup>为提高用户体验度, 可将  $p_{total}, l, v, k$  的数值均设置为可让用户输入的指标

- (3) 计算警察局到诸犯罪地点的最短路径之和, 在本题的设定中, 即为  $\sum_{i=1}^k |x_i - x_0| + |y_i - y_0|$ , 赋值为 `total_dist`。同时记录警察局到每个犯罪地点的最短路径, 记录到列表 `shortest_path` 中, 可利用 `networkx` 库里 `shortest_path` 函数实现。
- (4) 利用随机数生成特定犯罪地点中的强盗数量, 存储到列表 `cri_set` 中, 其中的元素即为  $b_i, i \in [1, k], b_i \in [3, 10]$ 。
- (5) 定义长度为  $k$ , 每个元素均为 1 的列表 `police_set`, 其中的元素即为  $p_i, i \in [1, k], p_i \in [1, p\_total - k + 1]$ 。
- (6) 打印当前时间。
- (7) 利用迭代算法给  $p_i$  在定义域内赋值, 遍历所有情况, 并添加约束条件  $\sum_{i=1}^k p_i = 100$ 。<sup>2</sup>
- (8) 在所有满足约束的警察分配方案中, 计算每组所对应的综合损失值, 为  $total\_dist + \sum_{i=1}^k \frac{b_i^2}{a_i}$ , 存储到列表 `calc_cost` 中 (如不满足约束, 则在上一步直接跳出函数并计算下一种情况, 以降低运算时间, 提高效率)。
- (9) 同时在迭代的过程中进行大小比对, 如果新输出的结果比上次结果大则丢弃, 反之覆盖, 提高效率。由此执行便可选出最小的综合损失值, 将其赋值为 `min_cost` 中, 同时将其对应的警察分布情况最优解添加到空列表 `min_police_set`。
- (10) 打印当前时间, 计算运行过程中总耗时, 赋值为 `tick`。
- (11) 打印相关结果, 如列表 `calc_result` 的长度; 列表 `locations_set`, `crime_set`, 和 `min_police_set`; `total_dist` 和 `cri_set` 的数值等。
- (12) 根据 `shortest_path` 记录的元素, 结合 `matplotlib.pyplot` 在图像中画出路径, 颜色则由随机数生成的 `RGB` 代码表示。

### 2.1.3 模型反思

- (1) 在其他参数保持不变,  $k$  值逐渐增大时, 运算时间会呈现几何倍数的增长, 说明在整数拆分的问题上, 遍历选取最小值的解法具有局限性且效率过低 (如在  $p\_total = 100, l = 1, v = 1, k = 5$  时, `calc_result` 的长度为 4401033, 耗时为 86.2s, 但当  $k = 6$  时, `calc_result` 的长度则变为 91104908, 耗时为 1645.3s)。
- (2) 遍历的解法在数学意义上不够优美。

## 2.2 拉格朗日乘数法

### 2.2.1 模型简介

为提高运算速度, 可选取拉格朗日乘数法 (Lagrange Multiplier) 进行计算。其优点在于可以不通过遍历尝试, 而是利用数学方法 (微积分) 简化计算, 直接选取方程的最优结果, 可避免随

<sup>2</sup>没有用有限次 for 循环进行迭代在于其需要预知犯罪行动  $k$  的数值, 而且随着  $k$  值的变化, 代码也需要重新编辑, 灵活性和兼容性较低。

着  $k$  值不断增大导致运行时间过长的的问题。同时，该模型可利用 for 循环的方式具有很强的兼容性，对于不同  $k$  的取值均可适用。

### 2.2.2 模型思路

假设多元函数  $f(x_1, \dots, x_n)$  在  $A$  处存在极值  $\kappa$ ，且在  $A$  点的邻域内连续，则在  $A$  点处有：

$$f(x_1, \dots, x_n) = \kappa$$

同时存在常值函数：

$$g(x_1, \dots, x_n) = c$$

两函数在  $A$  点处的全微分为，

$$\begin{aligned} df &= \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n = 0 \\ dg &= \frac{\partial g}{\partial x_1} dx_1 + \frac{\partial g}{\partial x_2} dx_2 + \dots + \frac{\partial g}{\partial x_n} dx_n = 0 \end{aligned}$$

由于  $dx$  和  $dy$  均为选取的无穷小量，故线性方程组的系数成比例，

$$\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial g}{\partial x_1}} = \dots = \frac{\frac{\partial f}{\partial x_n}}{\frac{\partial g}{\partial x_n}} = -\lambda$$

故，

$$\begin{aligned} \frac{\partial f}{\partial x_1} + \lambda \cdot \frac{\partial g}{\partial x_1} &= 0 \\ &\dots \\ \frac{\partial f}{\partial x_n} + \lambda \cdot \frac{\partial g}{\partial x_n} &= 0 \end{aligned}$$

由此生成新函数，

$$\mathcal{L}(x_1, \dots, x_n, \lambda) = f(x_1, \dots, x_n) + \lambda \cdot g(x_1, \dots, x_n)$$

求原函数极值的问题就转化为求该函数极值的问题。

推广到多元的等式约束优化问题时，则有，

$$\begin{aligned} \min f(x_1, x_2, \dots, x_n) \\ s.t. h_k(x_1, x_2, \dots, x_n) = 0 \end{aligned}$$

令  $\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{k=1}^l \lambda_k h_k(\mathbf{x})$ ，函数  $\mathcal{L}(\mathbf{x}, \lambda)$  则为 Lagrange 函数，参数  $\lambda$  为 Lagrange 乘子，并使得以下方程组成立，

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_i} &= 0 (i = 1, 2, \dots, n) \\ \frac{\partial \mathcal{L}}{\partial \lambda_k} &= 0 (k = 1, 2, \dots, l) \end{aligned}$$

在本题目中，在  $total\_dist$  已知的情况下，即要求  $\min \sum_{i=1}^k \frac{b_i^2}{a_i}$ 。故，

$$f(a_1, \dots, a_k) = \sum_{i=1}^k \frac{b_i^2}{a_i}$$

$$g_0(a_1, \dots, a_k) = \sum_{i=1}^k a_i - 100 = 0$$

需要注意到，本题同时存在对于  $a_i (i \in [1, k])$  的不等式约束，即  $a_i \geq 1$ 。同时考虑到  $a_i$  为抽象出的警察单位，故不需要结合现实保证其为整数。为解决不等式问题，需要引入松弛变量将其转化为等式约束进行处理。

假设，

$$g_1(x) = x - b \geq 0$$

即可转换为，

$$h_2(x, b_1) = g_2(x) + b_1^2 = x - b + b_1^2 = 0$$

故在本题目中，新增了  $k$  个约束条件，即  $1 - a_k + x_k^2 = 0$ ，分别命名为  $g_1(a_1, \dots, a_k) \sim g_k(a_1, \dots, a_k)$ 。

此时新生成的拉格朗日函数为，

$$\mathcal{L}(a_1, \dots, a_n, \lambda, \mu_1, \dots, \mu_n, x_1, \dots, x_n) = \sum_{i=1}^k \frac{b_i^2}{a_i} + \lambda(\sum_{i=1}^k a_i - 100) + \sum_{i=1}^k \mu_i(1 - a_i + x_i^2)$$

同时保证该函数对于  $a_i, \lambda, \mu_i, x_i$  的偏导数均为 0，即

$$\frac{\partial \mathcal{L}}{\partial a_i} = \frac{\partial \mathcal{L}}{\partial \lambda} = \frac{\partial \mathcal{L}}{\partial \mu_i} = \frac{\partial \mathcal{L}}{\partial x_i} = 0$$

即可。

其中，

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial a_i} = -a_i^{-2} b_i^2 + \lambda - \mu_i = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{i=1}^k a_i = 100 \\ \frac{\partial \mathcal{L}}{\partial \mu_i} = 1 - a_i + x_i^2 = 0 \\ \frac{\partial \mathcal{L}}{\partial x_i} = 2x_i \mu_i = 0 \\ \mu_i \geq 0 \end{cases}$$

其中  $i \in \mathbb{Z}, i \in [1, k]$ 。 `fsolve()` 函数可以用于解此方程组。

在书写代码时，为提高运算速度，我没有使用 `sympy` 库里的 `diff()` 函数，因为其是利用解析的方式求函数对于特定变量的偏导，会根据函数种类的不同导致运算速度降低；于是我选择数值近似求解偏导数，

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\mathcal{L}(x_i + \Delta x_i) - \mathcal{L}(x_i - \Delta x_i)}{2\Delta x_i}$$

并将步长  $\Delta x = 10^{-4}$  以保证运算精准度。

值得注意的是，当 KKT(Karush-Kuhn-Tucker) 乘子  $\mu = 0$  时， $g_i(x) < 0$ ，在此时不等式约束不起作用，导致  $a_i$  的结果有可能为负数，与真实情况不相符。针对此问题，我选择为  $a_i$  设置初始值  $a_0 = 101 - k, a_1 \sim a_k = 1, \lambda = 0$  的方式作为替换，来界定参数定义域的范围，并由此逼近计算便可规避由于代数计算带来的异常值。

### 2.2.3 模型反思

- (1) 利用数学方法简化计算，具有极快的运算速度。不同于遍历算法，不会随着  $k$  值增大使得耗时显著增大。
- (2) 值得注意的是，当设置  $a_i = \frac{100}{\sum_{i=1}^k b_i} \cdot b_i, i \in [1, k]$ （即加权平均数）并对输出结果进行四舍五入时，便可得到与拉格朗日乘数法一致的结果。经过分析发现这是因参数设定所带来的巧合，

$$\begin{aligned}
 \min \sum_{i=1}^k \frac{b_i^2}{a_i} &= \min \sum_{i=1}^k \frac{b_i^2}{\frac{100}{\sum_{i=1}^k b_i} \cdot b_i} \\
 &= \min \sum_{i=1}^k \frac{b_i \cdot \sum_{i=1}^k b_i}{100} \\
 &= \min t \cdot \sum_{i=1}^k b_i \\
 &= \min \frac{(\sum_{i=1}^k b_i)^2}{100}
 \end{aligned}$$

存在最小值的情况。但如果修改前提条件  $cost = \frac{b_i^2}{a_i}$  便不一定能解出如此形式简单的解析解，故此方法不具有普遍性。

## 2.3 权重图

### 2.3.1 模型简介

在原有模型的基础上，我利用随机数生成成为相连接的节点赋上距离权重  $t \in [1, 10], t \in \mathbb{Z}^+$ ，并利用 Dijkstra 算法计算两点坐标之间的最短路径（曼哈顿距离），并由此计算整体过程中的损失最小值。

### 2.3.2 Dijkstra 算法

设  $G = (V, E)$  是一个带权有向图，把图中顶点集合  $V$  分成两组，第一组为已求出最短路径的顶点集合  $S$ （初始时  $S$  中只有一个源点，以后每求得一条最短路径，就将其加入到集合中，直到全部顶点都加入到  $S$  中），第二组为其余未确定最短路径的顶点集合  $U$ ，按最短路径长度的递增次序依次把第二组的顶点加入  $S$  中。在加入的过程中，总保持从源点  $v$  到  $S$  中各顶点的最短路径长度不大于从源点  $v$  到  $U$  中任何顶点的最短路径长度。此外，每个顶点对应一个距离， $S$  中的顶点的距离就是从  $v$  到此顶点的最短路径长度， $U$  中的顶点的距离，是从  $v$  到此顶点只包括  $S$  中的顶点为中间顶点的当前最短路径长度。

- (a) 初始时， $S$  只包含源点（警察局）， $v$  的距离为 0， $U$  包含除  $v$  外的其他顶点。若  $v$  与  $U$  中顶点  $u$  有边，则  $\langle u, v \rangle$  正常有权重，若  $u$  不是  $v$  的出边邻接点，则  $\langle u, v \rangle$  为  $\infty$ 。
- (b) 从  $U$  中选出距离最短的顶点  $k$ ，并将顶点  $k$  加入到  $S$  中；同时，从  $U$  中移除顶点  $k$ 。

- (c) 更新  $U$  中各个顶点到起点  $s$  的距离。之所以更新  $U$  中顶点的距离，是由于上一步中确定了  $k$  是求出最短路径的顶点，从而可利用  $k$  来更新其它顶点的距离；如， $(s, v)$  的距离可能大于  $(s, k) + (k, v)$  的距离。
- (d) 重复步骤 (b) 和 (c)，直到遍历完所有顶点。分别打印最短路径的轨迹和总长度。
- (e) 分别将不同的犯罪现场坐标作为终点，重复步骤 (a) ~ (d)。

### 2.3.3 模型思路

和 2.1.2 部分整体类似，只在计算  $total\_dist$  时选择使用 Dijkstra 算法的返回值，而非坐标的  $x$  值与  $y$  值差的绝对值之和。同时选用 2.2 介绍的拉格朗日乘数法来替代 2.1 中的遍历算法以提高运算速度。

## 3 关于动态规划的一些思考

### 3.1 基本模型

#### 3.1.1 模型思路

如果假设在某一犯罪地点的警察提前处理完犯罪，他便可以调配到其他地区去协助，从而减少综合损失值。不失一般性，假设最先处理完犯罪的  $a_1$  名警察处于  $(x_1, y_1)$ ，在余下的  $(x_2, y_2) \sim (x_k, y_k)$  的地点中选取最优解需要经过一下过程：

- (1) 假设在不经调配时， $k$  个犯罪地点单独完成抓捕行动所花时间为  $t_1 \sim t_k$ ，并且从  $(x_1, y_1)$  到  $(x_t, y_t)$  所花时间为  $\Delta t_{1t}$ ，则显然，最优解  $(x_i, y_i)$  一定符合条件，

$$0 < \frac{\Delta t_{1i}}{t_i - t_1} < 1$$

- (2) 将所有满足过程 (1) 的情况加入到列表 `time_worth_selection` 中。当  $a_1$  名警察迁移到  $(x_i, y_i)$  时，考虑到警察抓小偷的速度与时间无关，两地警察共同工作的时间为，

$$t_{work} = \frac{((\frac{t_i - (t_1 + \Delta t_{1i})}{t_i}) \cdot b_i)^2}{a_1 + a_i}$$

则显然由于迁移所带来的损失值减小量为，

$$saving\_cost = 1 \cdot (t_i - (t_1 + \Delta t_{1i} + t_{work}))$$

考虑到  $k$  值的取值不会很大而且该算法的复杂度不高，故选择遍历所有  $i \in (1, k]$  的情况便可有效解得最优解。选取使得  $saving\_cost$  最大的地点便是针对  $(x_1, y_1)$  的最佳迁移点。

- (3) 显然，当针对  $(x_1, y_1)$  点时，`time_worth_selection` 为空时，则  $saving\_cost = 0$ 。

为降低算法设计起的难度，设立以下假设简化问题：

- (1) 一处地区的警察只迁移一次。

- (2) 一处地区只能接收一次外来迁移的警察。例如，针对  $(x_i, y_i)$ ,  $max\ saving\_cost$  的地点选择为  $(x_t, y_t)$ 。如若与  $(x_1, y_1)$  的选择重合，则将  $(x_t, y_t)$  从 `time_worth_selection` 中剔除，并再次选择最优解。利用 `while` 循环重复该步骤，直到选择出未重合的地点。
- (3) 如果根据计算没有合适的地点可以迁移，则选择停留在原地，无论是 `time_worth_selection` 本来为空或是所有元素均已被剔除在外。

最后利用迭代算法，在遍历  $(x_1, y_1) \sim (x_{k-1}, y_{k-1})$  的  $k-1$  次迁移后，将原来计算的损失最小值分别减去每次计算的  $saving\_cost$  便是最优的结果。

### 3.1.2 模型反思

- (1) 参数设置是一大难点。经过不断尝试发现当  $\sum_{i=1}^k a_i = 50, v = 1, k = 10$  左右时，迁移情况出现较为频繁。如若令  $k$  逐渐增大，越多地点的警察趋向于停留在原地。迁移的频率与随机数生成的模式有关。
- (2) 此算法只针对计算局部最优解，并未考虑整体最优，故计算出的最小损失值并不准确。

## 3.2 马尔科夫决策过程

### 3.2.1 模型介绍

马尔科夫决策过程 (Markov Decision Process) 可被定义为以下数组  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ ，其中  $\mathcal{S}$  为描述状态的有限集合； $\mathcal{A}$  为描述动作的有限集合； $\mathcal{P}$  为迁移概率矩阵，为  $\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$ ； $\mathcal{R}$  为奖励函数； $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$ ； $\gamma$  为衰减系数， $\gamma \in [0, 1]$ ，引入此变量有数学表达的方便，避免陷入无限循环，远期利益具有一定的不确定性，符合人类对于眼前利益的追求，符合金融学上获得的利益能够产生新的利益因而更有价值。

在本题目中，状态由每个阶段中最优的警察数目分配表示，动作即为抓捕完成的警察向其他地区迁移的过程。概率矩阵在本题目中不占据显著作用，因朝各个方向迁移的概率是相等的，故相互抵消。

定义收获  $G_t$  为在一个马尔科夫奖励链上从  $t$  时刻开始往后所有的奖励的有衰减的总和，

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

因收获由每个阶段人员迁移带来的收益共同决定，且重要性不会随着状态的推移而降低，故  $\gamma = 1$ ，说明全局最优解为局部最优解的和。由于迭代次数已知，故不需要考虑函数是否收敛的问题

同时定义策略  $\pi$  为概率的集合或分布，在本题目中即为特定时间节点上最优的人员迁移策略，

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$$



定义状态价值函数  $v_\pi(s)$  表示从状态  $s$  开始，遵循当前策略时所获得的收获的期望，

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \\ &= \sum_{a \in \mathcal{A}} \pi(a|s) (\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')) \end{aligned}$$

highlightcolor

$q_\pi(s, a)$  为行为价值函数，表示在执行策略  $\pi$  时，对当前状态  $s$  执行某一具体行为  $a$  所能的到的收获的期望，

$$\begin{aligned} q_\pi(s, a) &= \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \\ &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a') \end{aligned}$$

highlightcolor

同时存在，

$$\begin{aligned} v_\pi(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s, a) \\ q_\pi(s, a) &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \end{aligned}$$

的等式关系。当，

$$\begin{aligned} v_*(s) &= \max_a v_\pi(s) \\ &= \max_a q_*(s, a) \\ &= \max_a \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s') \end{aligned}$$

highlightcolor

其从所有策略产生的状态价值函数中，选取使状态  $s$  价值最大的函数，即为本题所求。

### 3.2.2 模型思路

本部分与 3.1.1 的整体思路类似，为避免重复，只阐明两者间不同之处。

$S_1$  可用  $[a_{1,1}, a_{2,1}, \dots, a_{k,1}]^T$  表示。不失一般性，假设  $a_{1,1}$  为第一组处理完当地盗贼的警察数量。通过计算  $v_\pi$  计算本次动作的最优策略后， $S_2$  为  $[0, a_{2,2}, \dots, a_{k,2}]^T$ ，约定  $a_{i,2} \geq a_{i,1}, i \in [2, k]$ 。

在坐标集合中选中所有满足  $0 < \frac{\Delta t_{1i}}{t_i - t_1} < 1$  的坐标  $(x_t, y_t)$ ，并给其对应的  $a_{i,1}$  值赋上迁移量  $\Delta a_{i,1}$ ，并且设定定义域为  $0 \leq \Delta a_{i,1} \leq a_{1,1}$ ，使得，

$$a_{i,1} + \Delta a_{i,1} = a_{i,2}$$

同时定义目标函数  $v_*$  为，

$$v_\pi = \sum_{i=2}^k \left( \frac{(\frac{t_i - t_1 - \Delta t_{1i}}{t_i} \cdot b_i)^2}{a_{i,1}} - \frac{(\frac{t_i - t_1 - \Delta t_{1i}}{t_i} \cdot b_i)^2}{a_{i,2}} \right)$$

同时满足约束，

$$\sum_{i=2}^k (a_{i,2} - a_{i,1}) = a_{1,1}$$

当  $v_\pi$  取值到最大值  $v_*$  时，即为本次迁移的最优解（最优的  $\Delta a_{i,1}$  列表），将返回值定义为  $saving\_cost$ ，并将迁移量增加到  $S_1$  的对应数值上求得  $S_2$  的值。同时，没有被选中的坐标对应的警察为  $a_{k,1} = a_{k,2}$ ，即从  $S_1$  至  $S_2$  数值没有发生改变。综合二者便得到状态 2 下的所有数值。

最后利用迭代算法，在遍历  $(x_1, y_1) \sim (x_{k-1}, y_{k-1})$  的  $k-1$  次迁移，得到每状态下的人员分布情况。同时，每次迁移后分别更新该时间状态下各犯罪地点的警察，盗贼，和剩余处理时间的数值，并分别与坐标对应形成字典，以供下次迭代时调用参数。将原来计算的损失最小值分别减去每次计算的  $saving\_cost$  便是最优的结果。

### 3.3 模型反思

- (1) 在设定参数  $p\_total = 30, v = 1, k = 5$  时，通过多次尝试并总结经验，得到最优的人员分配方案往往是朝一处集体迁移，而非向多个地点发散式分配。此结论现阶段无法通过数学方式证明，可能是参数设置造就的偶然结果。

## 4 模型思路汇总

经过不断地算法提升后，最终得出完整模型的思路图，

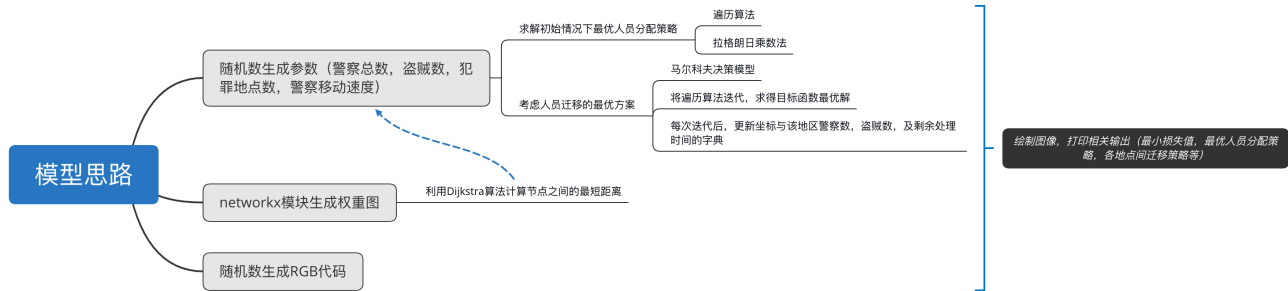


Figure 1: 模型思路图

## 5 结果呈现

通过演算求得相对成熟的算法后，我选择通过控制变量运行程序（即警车总数，地点总数，及盗贼随机数生成范围）的方式来计算各参数（即迁移后平均损失，迁移平均收益率<sup>3</sup>，及能迁

<sup>3</sup>通过迁移减少的损失值与未迁移时的损失值的比率，用来衡量迁移的效用

移占比<sup>4)</sup>之间的潜在关系。考虑到地点坐标，盗贼数量等数据均为随机数生成，为减小误差，我选择将每组环境下的程序各执行 500 次并分别计算其算术平均数。计算结果如下呈现：

500次	迁移后平均损失	迁移后损失标准差	迁移平均收益率（仅考虑能迁移的）	能迁移占比
10警察, 5地点, (7,10)盗贼	264.6175787	39.84757356	0.059312	74.60%
20警察, 5地点, (7,10)盗贼	203.5148989	36.3260613	0.021534	42.80%
30警察, 5地点, (7,10)盗贼	180.6133495	35.73158229	0.018656	37.20%
40警察, 5地点, (7,10)盗贼	167.9510108	34.41013572	0.0136116	32.20%
50警察, 5地点, (7,10)盗贼	158.4240975	36.22879434	0.013844	31.00%
60警察, 5地点, (7,10)盗贼	156.3481217	34.01195407	0.010414	28.60%
70警察, 5地点, (7,10)盗贼	150.2637151	33.04159689	0.008304	26.80%
80警察, 5地点, (7,10)盗贼	149.9104131	35.94680877	0.009662	24.20%
90警察, 5地点, (7,10)盗贼	148.9015341	34.95425847	0.006393	21.80%
100警察, 5地点, (7,10)盗贼	145.1915478	34.13703878	0.00712	21.40%
110警察, 5地点, (7,10)盗贼	144.7469291	34.58240305	0.006339	20.40%
120警察, 5地点, (7,10)盗贼	140.6276034	34.45304498	0.004813	23.60%
130警察, 5地点, (7,10)盗贼	141.5971831	31.42746228	0.004704	24%
140警察, 5地点, (7,10)盗贼	140.1062203	34.98980203	0.004	23.20%
150警察, 5地点, (7,10)盗贼	142.842162	37.15855055	0.004242	26.60%

Figure 2: 第一组数据

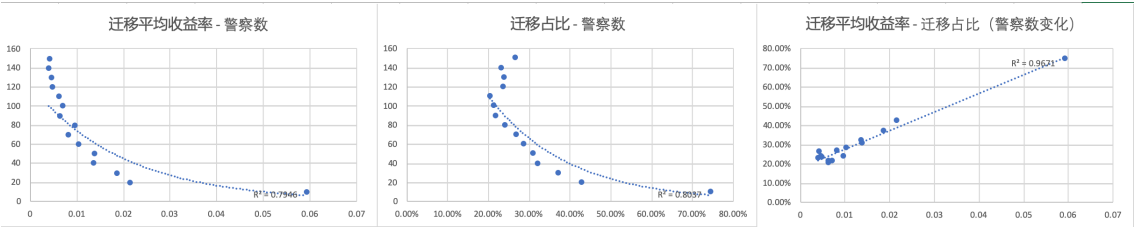


Figure 3: 第一组图表 (1)

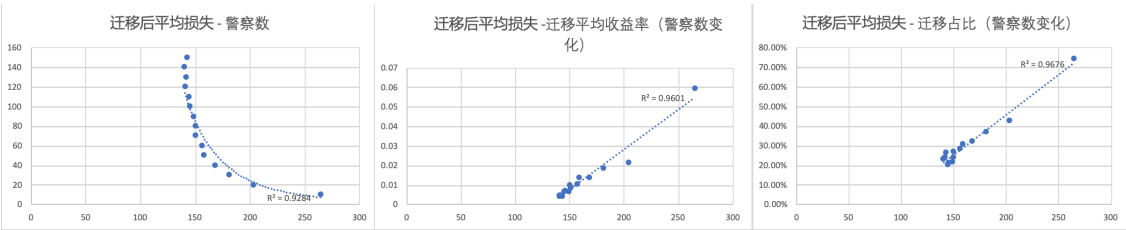


Figure 4: 第一组图表 (2)

<sup>4)</sup>能通过迁移减少损失的次数与总运行次数（500 次）的比率，用来衡量迁移行为出现的频率

500次	迁移后平均损失	迁移后损失标准差	迁移平均收益率 (仅考虑能迁移的)	能迁移占比
40警察, 1地点, (7,10]盗贼	26.848	12.26563441	0	0
40警察, 2地点, (7,10]盗贼	58.0050717	17.51272003	0.01769	5%
40警察, 3地点, (7,10]盗贼	89.65945496	25.36895406	0.014457	12.40%
40警察, 4地点, (7,10]盗贼	127.3703814	31.57286837	0.013349	15.20%
40警察, 5地点, (7,10]盗贼	168.4021118	35.95725907	0.0128	26.40%
40警察, 6地点, (7,10]盗贼	212.2001767	38.07162379	0.013873	38.40%
40警察, 7地点, (7,10]盗贼	260.7094202	47.21946628	0.013388	44.00%
40警察, 8地点, (7,10]盗贼	310.8207985	51.7830904	0.012221	50.60%
40警察, 9地点, (7,10]盗贼	361.8203254	55.17166144	0.01674	59.60%
40警察, 10地点, (7,10]盗贼	426.3891399	60.80975242	0.017787	64.40%
40警察, 11地点, (7,10]盗贼	488.1270482	64.14342794	0.018975	71.00%
40警察, 12地点, (7,10]盗贼	547.1770995	74.3269774	0.018662	74.80%
40警察, 13地点, (7,10]盗贼	616.5864287	81.1138028	0.02263	80.00%
40警察, 14地点, (7,10]盗贼	641.6273698	84.675123	0.02433	82.10%
40警察, 15地点, (7,10]盗贼	678.1231289	89.7623922	0.02541	84.92%

Figure 5: 第二组数据

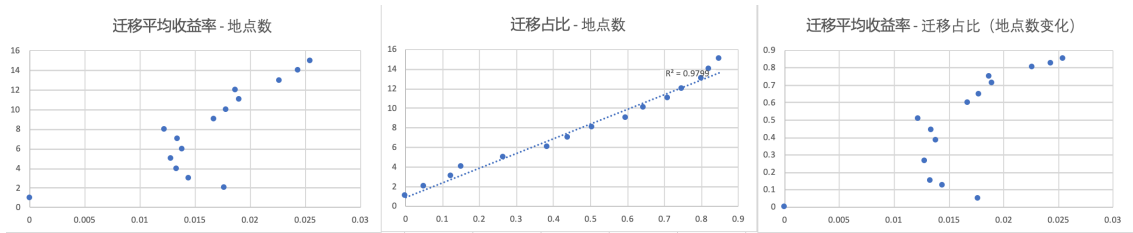


Figure 6: 第二组图表 (1)

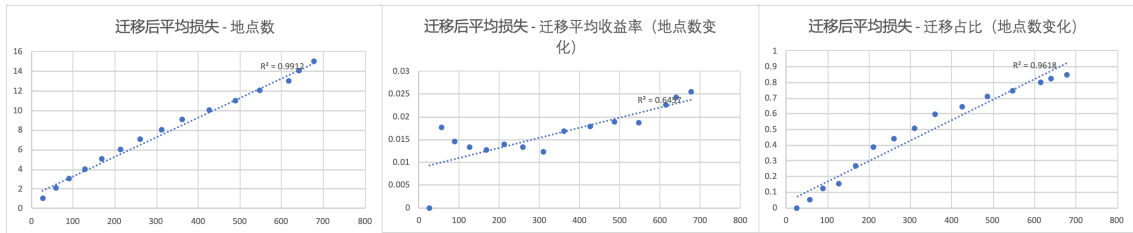


Figure 7: 第二组图表 (2)

500次	迁移后平均损失	迁移后损失标准差	迁移平均收益率 (仅考虑能迁移的)	能迁移占比
40警察, 5地点, [1,10]盗贼	147.0604638	34.89612408	0.019954	48.40%
40警察, 5地点, [2,10]盗贼	151.3225946	36.24281695	0.020107	40.40%
40警察, 5地点, [3,10]盗贼	151.1783729	36.23011147	0.019479	35.00%
40警察, 5地点, [4,10]盗贼	152.3668527	35.13517077	0.019609	36.80%
40警察, 5地点, [5,10]盗贼	161.336638	36.38106248	0.016491	39.00%
40警察, 5地点, [6,10]盗贼	160.0516108	34.71551671	0.016902	30.00%
40警察, 5地点, [7,10]盗贼	167.7751043	34.98958315	0.01254	27.40%
40警察, 5地点, [8,10]盗贼	170.7215806	34.9563939	0.008795	26.20%
40警察, 5地点, [9,10]盗贼	174.0080881	35.07921652	0.00949	20.60%
40警察, 5地点, [10,10]盗贼	180.656	31.94417827	0	0%

Figure 8: 第三组数据

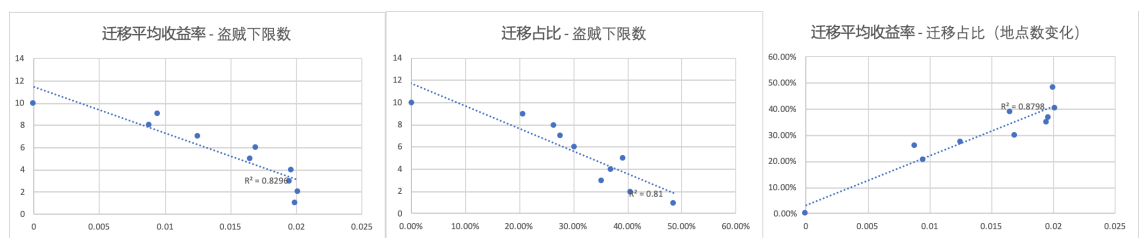


Figure 9: 第三组图表 (1)

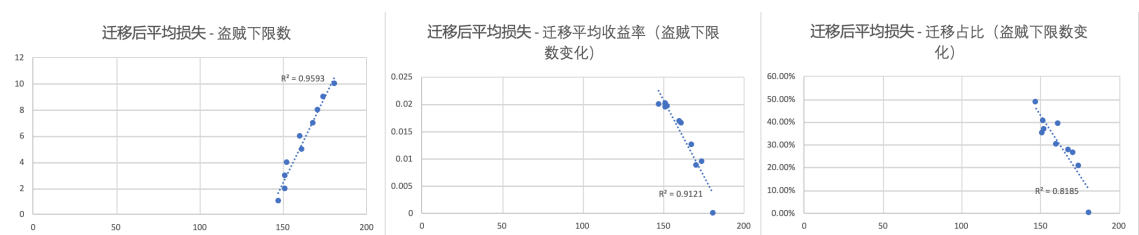


Figure 10: 第三组图表 (2)

各变量之间的关系大多可通过线性或指数函数的方式进行定性描述，并且  $R^2$  的数值整体处于较高的水平，说明拟合效果很好。唯有迁移平均收益率与地点数及迁移占比的数据之间离散程度较大而无法得到理想结果。