

# 几类路径问题算法探索

陈昱龙

上海海事大学理学院, 上海

收稿日期: 2023年3月17日; 录用日期: 2023年4月21日; 发布日期: 2023年4月28日

## 摘要

通过仿真模拟老鼠觅食的行为, 先用极限逼近的思想寻找老鼠在 $3 \times 3$ 的格子中找到食物的期望时间, 从枚举法开始寻找老鼠在 $3 \times 3$ 的格子中找到食物的最短路径, 再使用极限逼近方法寻找最短路径, 随后将此情景进行推广, 例如 $n \times n$ 的格子或者更复杂的情况, 说明该方法可以适用于任意一个网络找任意节点间最短路, 举出一个在实际生活的例子中, 应用极限逼近的方法找到其所需的最短路径。得出当一个问题的实质是最短路时, 都可以使用极限逼近的思想。

## 关键词

仿真, 最优路径, 极限逼近

# Exploration of Algorithms for Several Path Problems

Yulong Chen

School of Science, Shanghai Maritime University, Shanghai

Received: Mar. 17<sup>th</sup>, 2023; accepted: Apr. 21<sup>st</sup>, 2023; published: Apr. 28<sup>th</sup>, 2023

## Abstract

By simulating mice's foraging behavior, the idea of limit approximation can be used to work out the expected time for mice finding food in a  $3 \times 3$  grid. The author firstly employs enumeration method to look for the shortest path for mice's foraging in  $3 \times 3$  grids, and then uses the limit approximation method to find the shortest path. Subsequently, the author extends this scenario, for example, trying to find the shortest path in an  $n \times n$  grid or more complex cases. It is shown that this method can be applied to any grids to find out the shortest path between any nodes. An example is given in real life where the method of limit approximation is used to find the shortest path required. When the essence of a problem is to find out the optimal path, the idea of limit approximation can be always employed.

## Keywords

Simulation, Optimal Path, Limit Approximation

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

最短路问题的研究起源于 20 世纪 50 年代末, 是图论中一个经典的问题。寻找最短路就是在给定的网络中两节点间寻找到一条距离最短的路。基于研究的问题背景不同, 寻找到最短路的意义也不同。如在海运过程中, 海运公司就会考虑让船舶航行到终点的距离尽可能的短, 以减少运输的成本, 而让到终点距离尽可能的短, 就是一个寻找最短路的问题。同样的在人们的日常出行中也会多地选择能让人们最快到达的方案。

但需要注意的是, 在寻找最短路的过程中, 如果是实地进行实验, 不仅会花费大量的时间, 也会花费大量的金钱, 甚至会因为一些外在因素如天气等, 从而导致得到的结果出现误差。在此基础上, 我们可以运用计算机仿真技术, 这样既可以减少研究所用的经费支出, 也可以模拟在实际情况下的实验过程。在研究最短路问题[1]中我们针对结点较少的网络, 可以使用枚举法, 枚举法需要将每一条路都计算出来并比较相互之间的长度, 随着给定的网络中节点越来越多, 枚举法会变得越来越复杂。在寻找最短路中著名的算法有 Dijkstra (迪杰斯特拉)算法和 Floyd (弗洛伊德)算法等[2]。然而随着给定的网络中节点越来越多, 由于算法的时间复杂度, 运算的时间会越来越长。

本文从简单的  $3 \times 3$  的格子中老鼠觅食模型出发, 运用一种极限逼近的思想, 找到老鼠觅食的期望时间, 以此思想为基础逐渐地将模型复杂化, 最后将这极限逼近的思想用于寻找实际生活中的最短路。

## 2. 实现让老鼠成功的找到一次食物

为了让小老鼠在一个  $3 \times 3$  的格子中成功的从起点(1号)到终点(9号)找到一次食物, 我们要研究每一个格子之间的关系, 通过每个格子之间的关系确定当小老鼠处于任何一个格子的时候, 小老鼠接下来可以往哪里去, 但要保证小老鼠不会原地不动。为此, 我们可以设一个数列, 由于小老鼠的起点和终点是固定的, 对应到数列中就是首项和末项是固定的, 由于小老鼠在行走时会走到哪一个格子不确定, 存在走到重复格子的情况, 对应到数列中就是除了首项和末项, 中间项的个数是不确定的。

如图 1 所示:

1	2	3
4	5	6
7	8	9

Figure 1. The mouse position initial figure

图 1. 老鼠位置初始图

令九个格子分别为:

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$$

用一个数列表示一次找到食物路径为:

$$k = \{a_1, a_2, a_3, \dots, a_6, a_9\}$$

数列的第  $i$  个元素代表小老鼠第  $i$  步在那个位置。

令 9 个格子在欧式平面上的坐标为:

$$(x_i, y_i) (i=1, 2, 3, \dots, 9) (x_i, y_i \geq 0)$$

每一个格子代表一个欧式平面上的坐标令:

$$a_i = (x_i, y_i) (i=1, 2, 3, \dots, 9)$$

用一个  $9 \times 9$  矩阵  $M$  [3] 表示各个点之间的关系:

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

其中矩阵的元素  $M_{ij}$  的下标代表:

从  $a_i \sim a_j$  的距离

除对角线元素外, 其他值为 0 的元素代表无法直接到达该点。

在  $M_{ij} \neq 0$  下各个格子间坐标的关系

$$\hat{a} = (\hat{x}, \hat{y}) = ((x_i - x_j), (y_i - y_j))$$

在  $M_{ij} \neq 0$  并且确保两点之间可以直接到达下我们可以建立一个关于相互之间可以到达格子的递推公式:

$$\begin{aligned} a_j &= a_i + \hat{a} \\ (x_j, y_j) &= (x_i, y_i) + (\hat{x}, \hat{y}) \\ 1 \leq i, j \leq 9 \end{aligned}$$

例如: 令  $a_1$  的坐标为 (1.5, 7.5),  $a_1$  可以由  $a_2, a_4$  到达, 坐标分别为 (1.5, 4.5), (4.5, 7.5)。

则公式可以表示为:

$$a_1 = \begin{cases} a_2 + \hat{a} \\ a_4 + \hat{a} \end{cases}$$

对于数列第  $n$  个元素和第  $n+1$  个元素  $k_n, k_{n+1}$  有:

$$k_{n+1} = k_n + \hat{a}$$

那么由上述分析可知在一次寻找到食物的过程中：

数列可以有  $n$  个元素：

$$k = \{a_1, a_2, a_3, \dots, a_6, a_9\}$$

路径长度为：

$$n-1$$

至此，我们实现了让老鼠成功找到一次食物。

### 3. 实现让老鼠成功的找到多次食物并求其找到食物的期望时间

通过前面的分析已经知道在寻找到一次食物中，一个数列里面可以有  $n$  个元素，路径长度为  $n-1$ ，我们并不能将小老鼠每一次找到食物的路径都找出来，因为小老鼠找食物的路径可以有无穷多种，但为了能将最后的结果与真实的结果比较之后差别不大，我们需要进行足够多次的寻找，例如我们进行  $m$  次实验，那么我们可以给出在这  $m$  次寻找的过程中小老鼠找到食物的期望时间。

进行  $m$  次寻找故有  $m$  个数列：

$$k^1 = \{a_1, a_2, a_3, \dots, a_6, a_9\}$$

$$k^2 = \{a_1, a_2, a_3, \dots, a_8, a_9\}$$

⋮

$$k^m = \{a_1, a_2, a_3, \dots, a_8, a_9\}$$

那么可以通过一个数列里面元素之间的相互关系，得到走一步所需要的长度，相加就可以得到该次找到食物所用路径长度，有  $m$  个数列每一个数列都有一个路径长度：

$q_1-1, q_2-1, \dots, q_m-1$  其中分别代表第一次找到食物的路径长度，第二次找到食物的路径长度……

通过上面的分析在  $m$  次寻找中的期望时间  $t$  为：

$$t = \frac{\sum_{i=1}^m (q_i - 1)}{m}$$

为了得到与最精确的结果相差不大的答案，实际上我们需要进行无穷多次寻找食物的过程，但现实生活中这样是做不到的，所以我们对期望时间  $t$  求极限，此时期望时间等于：

$$t = \lim_{m \rightarrow +\infty} \frac{\sum_{i=1}^m (q_i - 1)}{m}$$

通过程序进行运算得到了如下的结果，结果如图 2 所示[4]。

通过图 2 可以知道，进行了 10,000 次的寻找食物的过程，最终在 10,000 次的查找下得到的期望时间约是 18 秒，则：

$$t = \frac{\sum_{i=1}^{10000} (q_i - 1)}{10000} = 18$$

每一次实验都可以得到一次找到食物的时间，期望时间又等于目前寻找次数的路径长度之和除以寻找次数，所以也可以得到一个数列，里面的元素是每一次查找的期望时间。

$$T = \{t_1, t_2, \dots, t_{m-1}, t_m\}$$

$$t_n = \frac{\sum_{i=1}^n (q_i - 1)}{n}$$

但由图中数据走势可以看出，大概从查找到 2000 次时开始，期望时间趋于稳定，几乎是一条直线，所以可以得出结论，小老鼠找到一次食物所用的期望时间是 18 秒，即：

$$t = \lim_{m \rightarrow +\infty} \frac{\sum_{i=1}^m (q_i - 1)}{m} = 18$$

亦可直观看出数列最后收敛于 18，即寻找了无穷多次后小老鼠找一次食物所需要的期望时间为 18 秒。

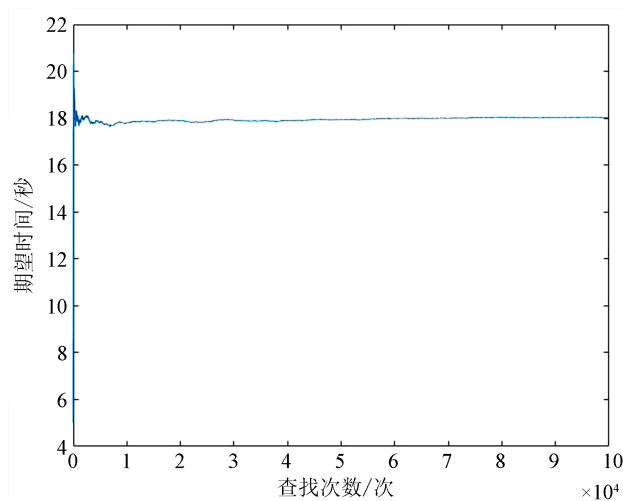


Figure 2. Plot of expected time result  
图 2. 期望时间结果图

#### 4. 寻找最短路的两个原则

原则 1：在寻找最短路的过程中，最短路径方案所含的节点个数  $\leq$  总的节点个数  
如图 3 [5] 的例子

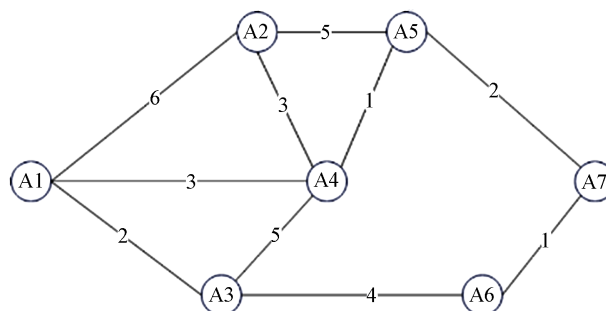


Figure 3. Example  
图 3. 例图

寻找 A1 到 A7 的最短路径长度以及方案。

首先我们要明确最短路径的方案中每个相邻节点是可以直接到达的且是没有重复节点的，以图 3 为例子，如果出现了以下情况：

$$\{A1, A4, A5, A4, A3, A6, A7\}$$

重复节点是 A4，此情况说明从 A4 走到 A5 之后，立刻又从 A5 返回 A4，说明多走了 A4 到 A5 的距离，由于每个相邻节点是可以互相直接到达的  
所以：

$$\{A1, A4, A3, A6, A7\}$$

所以在寻找最短路的过程中，最短路径方案所含的节点个数  $\leq$  总的节点个数。

**原则 2：最短路并不等于选择当前节点到下一个节点的最短距离**

如图 4：

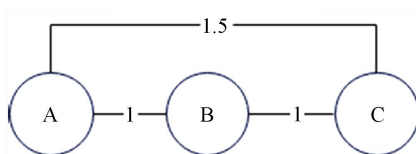


Figure 4. Example  
图 4. 例图

寻找 A 到 C 的最短路。

由图 4 知，若每一步选择最短的那么 A 到 B 到 C 距离为 2，但是选择 A 到 C 距离是 1.5。所以最短路并不等于选择当前节点到下一个节点的最短距离。

### 5. 使用枚举法求小老鼠最短路

直接对在  $3 \times 3$  的格子中对老鼠找到食物的路径方案枚举。

通过枚举法之后得到各个结果是：

- $\{a_1, a_2, a_3, a_6, a_5, a_4, a_7, a_8, a_9\}$
- $\{a_1, a_2, a_3, a_6, a_5, a_8, a_9\}$
- $\{a_1, a_2, a_3, a_6, a_9\}$
- $\{a_1, a_2, a_5, a_4, a_7, a_8, a_9\}$
- $\{a_1, a_2, a_5, a_6, a_9\}$
- $\{a_1, a_2, a_5, a_8, a_9\}$
- $\{a_1, a_4, a_5, a_2, a_3, a_6, a_9\}$
- $\{a_1, a_4, a_5, a_6, a_9\}$
- $\{a_1, a_4, a_5, a_8, a_9\}$
- $\{a_1, a_4, a_7, a_8, a_5, a_2, a_3, a_6, a_9\}$
- $\{a_1, a_4, a_7, a_8, a_5, a_6, a_9\}$
- $\{a_1, a_4, a_7, a_8, a_9\}$

一共 12 种方案。

统计每一个方案的路径长度，我们可以得到以下方案的最短路径长度都是最短的：

$$\begin{aligned} & \{a_1, a_2, a_3, a_6, a_9\} \\ & \{a_1, a_2, a_5, a_6, a_9\} \\ & \{a_1, a_2, a_5, a_8, a_9\} \\ & \{a_1, a_4, a_5, a_6, a_9\} \\ & \{a_1, a_4, a_5, a_8, a_9\} \\ & \{a_1, a_4, a_7, a_8, a_9\} \end{aligned}$$

以上 6 种方案都是最短路径的方案，最短路径长度为 4。

## 6. 使用极限逼近的思想求小老鼠最短路

由前面矩阵  $M$  [5]:

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

其中矩阵的元素  $M_{ij}$  的下标代表:

从  $a_i \sim a_j$  的距离

以及对于数列第  $n$  个元素和第  $n+1$  个元素  $k_n, k_{n+1}$  的递推公式有:

$$k_{n+1} = k_n + \hat{a}$$

通过原则 1 的限制，我们可以得到一次寻找到食物时的路径方案:

$$k = \{a_1, a_2, a_5, a_6, a_9\}$$

其中该路径方案的长度为 4。

在原则 1 的限制下，我们寻找食物  $m$  次，我们可以得到  $m$  个数列，和这些数列的方案的路径长度:

$$\begin{aligned} & k^1, k^2, \dots, k^m \\ & d^1, d^2, \dots, d^m \end{aligned}$$

在这  $m$  个数列中，可以存在路径方案一样的数列。

我们建立一个数列  $D$ ，里面元素是当前寻找  $m$  次下，里面  $m$  个路径长度中的最短值:

$$D = \{D_1, D_2, \dots, D_m\}$$

其中第  $n$  个元素为:

$$D_n = \min(d^1, d^2, \dots, d^n) (1 \leq n \leq m)$$

则数列里面第  $n+1$  个元素和第  $n$  个元素的递推公式是:

$$D_{n+1} = \min(d^1, d^2, \dots, d^{n+1})$$

$$D_n = \min(d^1, d^2, \dots, d^n)$$

$$D_{n+1} = \min(D_n, d^{n+1})$$

即:

$$D = \{\min(d^1), \min(d^1, d^2), \dots, \min(d^1, d^2, \dots, d^m)\}$$

此时, 我们效仿求期望时间的模式, 对数列  $D$  在趋近于无穷时求极限, 则可以得到理论上真实结果的最短路径长度。

$$d = \lim_{m \rightarrow +\infty} D$$

$$\lim_{m \rightarrow +\infty} D = \lim_{m \rightarrow +\infty} \{\min(d^1), \min(d^1, d^2), \dots, \min(d^1, d^2, \dots, d^m)\}$$

$$\lim_{m \rightarrow +\infty} D = \min(d^1, d^2, \dots, d^m)$$

$$d = \min(d^1, d^2, \dots, d^m)$$

结果由图 5 [4]:

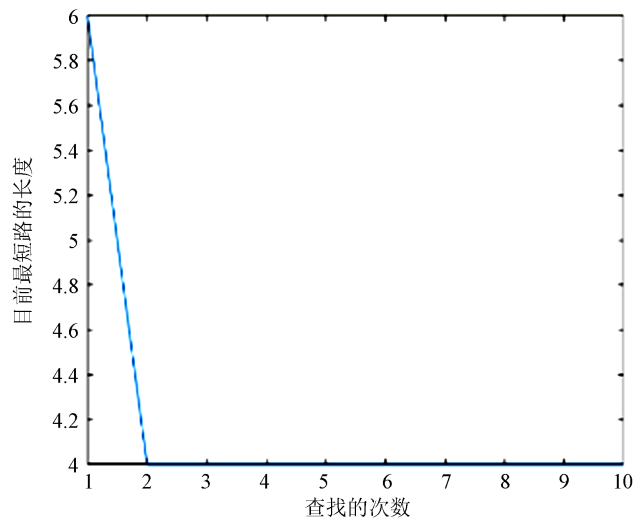


Figure 5. The shortest circuit length under the current search count

图 5. 当次查找次数下最短路径长度

由图 5 可以知道, 该图意思是一共进行了 10 次实验, 在第 2 次查找完成时, 我们发现真实情况下(枚举出的所有方案)的最短路径长度已经等于我们第 2 次查找完成时的最短路径长度。

$$d = \lim_{m \rightarrow +\infty} D$$

$$d = \min(d^1, d^2, \dots, d^m) = 4$$

通过与前面枚举法寻找老鼠觅食的最短路与图 5 得到的结果进行对比, 可以知道运用极限逼近思想找到的最短路径长度, 与枚举法寻找的最短路径的长度一致。

至此, 可以得出结论, 运用极限逼近思想找最短路径的方法是可行的。成功的用极限逼近的思想得到的老鼠觅食的最短路径, 其长度为 4。



## 7. 对极限逼近思想求最短路由推广的解释

由图 6:

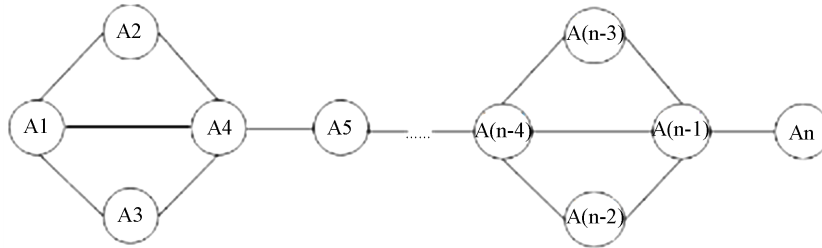


Figure 6. Example  
图 6. 例图

由图 6 看出中间可以省略无穷多个点，也就是可以组成任意一种情况，但由上面所述，任然可以使用上面解决老鼠觅食最短路的步骤，得到图 6 中 A1 到 An 的最短路长度，即适用于任意网络中任意节点间的最短路问题。

## 8. 使用极限逼近的思想进行拓展解决实际问题

举出下面一个实际问题

一款智能小车，小车只能由指挥中心统一调度，无法自行获得其它信息。已知每辆小车的最大线速度 10 厘米每秒并且小车在运动过程中严禁碰撞。现有 36 辆小车在一个边长为 2 米的正方形区域内排列成指定的图案。如图 7 所示，初始状态下小车以 20 厘米的横向及纵向间距均匀分布在正方形左下方的一个等腰直角三角形区域内。每辆小车视为半径 1 厘米的圆盘。

请设计一种调度方案，使得小车以尽可能短的时间排成如图 8 所示的图案，并且小车在曲线上的分布尽量均匀。请以 1 秒为时间步长的快照形式对运动过程进行展示。

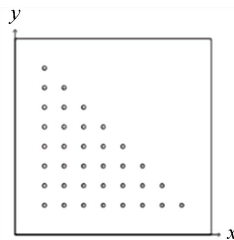


Figure 7. Initial figure  
图 7. 起始图

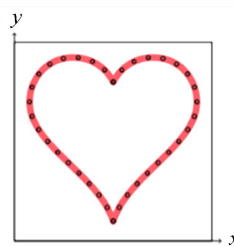


Figure 8. The target figure  
图 8. 目标图

该题需要设计一种调度方案，在短时间内使其从图 7 变到图 8。首先，我们明确每一个小车在从图 7 到图 8 的过程中，在某一方案中，该小车有可能停留在原地的情况，若该小车移动，那么该小车的目标点，就不可以有其他小车去。由于小车的速度一样，并且需要考虑小车在运动过程中是否会发生碰撞又要求短时间内，那么这任然是一个求最短路的问题。任然可以使用极限逼近的思想，但这时最短路不在是总路程最短。由于小车一起移动，所以在一个方案中的用时是找某一小车到其目标点最大距离，作为该方案的用时。再将多个方案用时对比，找到用时最短的方案。

首先给 36 个小车在图 7 位置编号：

$$a_1, a_2, a_3, \dots, a_{35}, a_{36}$$

再给 36 个小车在图 8 位置编号：

$$b_1, b_2, b_3, \dots, b_{35}, b_{36}$$

再由题目可知，以编号代表小车在欧式平面上的坐标：

$$a_i = (x_i, y_i) (i=1, 2, 3, \dots, 36)$$

$$b_i = (x'_i, y'_i) (i=1, 2, 3, \dots, 36)$$

再由一个  $36 \times 36$  的矩阵  $M$  存放从  $a_i \sim b_j$  的距离：

$$M_{ij} = \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2}$$

此时我们寻找再不考虑碰撞下，1 号小车从起始点去目标点的方案：

$$a_1 \sim b_j (1 \leq j \leq 36)$$

我们可以知道由 36 种可能，那么对于 1 整个方案来说有：

$$n + (n-1) + \dots + 3 + 2 + 1 = \frac{(n+1)n}{2}$$

$$36 + 35 + \dots + 3 + 2 + 1 = \frac{(36+1)36}{2} = 666$$

一共有 666 种可能，由此我们可以知道，如果使用枚举法是十分麻烦的，尽管理论上可行，但会花费许多时间，故我们使用求期望时间的思路。

构造一个函数：

$$y = f_n(x)$$

其中定义域为：

$$x \in (a_1, a_2, \dots, a_{36})$$

值域为：

$$y \in (b_1, b_2, \dots, b_{36})$$

易知该函数中是单射且满射，但是  $x$  映射到  $y$  中哪一个元素不确定，所以我们可以有多个函数。

例如函数  $y = f_1(x)$  将定义域中元素代入得：

$$b_2 = f_1(a_1), b_7 = f_1(a_2), b_4 = f_1(a_3), \dots, b_{11} = f_1(a_{36})$$

通过该函数我们就可以得到一个调度方案，即起始点编号为 1 的点去目标点编号为 2 的点，起始点编号为 2 的点去目标点编号为 7 的点……

得到了某一个方案之后我们需要考虑，这个方案中是否会出现小车相互碰撞的情况所以需要进一步讨论。

计算该方案中每一个起始点到相应目标点之间的直线斜率  $k$ ，以及该直线方程：

$$k = \frac{y_i - y'_i}{x_i - x'_i}$$

$$y = kx + c$$

$$\text{其中 } y_1 = k_1 x_1 + c$$

$c$  是截距

$x_1$  是方案中第一个起始点

$y_1$  是方案中对应的目标点

当然存在，有的方案中起始点与目标点一样，或者起始点只用竖直移动就可以到达目标点的特殊情况：

$k$  不存在

起始点与目标点一致

先讨论起始点和对应目标点的位置关系：

$$\text{起始点 } a_i = (x_i, y_i) (i = 1, 2, 3, \dots, 36)$$

$$\text{目标点 } b_i = (x'_i, y'_i) (i = 1, 2, 3, \dots, 36)$$

当斜率  $k$  存在且不为 0：

$$\text{当 } x_i > x'_i, y_i > y'_i$$

$$v_x = -\sqrt{\frac{v^2}{1+k^2}}, v_y = -\sqrt{v^2 - v_x^2}$$

$$\text{当 } x_i < x'_i, y_i < y'_i$$

$$v_x = \sqrt{\frac{v^2}{1+k^2}}, v_y = \sqrt{v^2 - v_x^2}$$

$$\text{当 } x_i > x'_i, y_i < y'_i$$

$$v_x = -\sqrt{\frac{v^2}{1+k^2}}, v_y = \sqrt{v^2 - v_x^2}$$

$$\text{当 } x_i < x'_i, y_i > y'_i$$

$$v_x = \sqrt{\frac{v^2}{1+k^2}}, v_y = -\sqrt{v^2 - v_x^2}$$

当斜率  $k = 0$ ：

$$\text{当 } x_i > x'_i, y_i = y'_i$$

$$v_x = -v, v_y = 0$$

$$\text{当 } x_i < x'_i, y_i = y'_i$$

$$v_x = v, v_y = 0$$

当斜率  $k$  不存在：

$$\begin{aligned} & \text{当 } x_i = x'_i, y_i > y'_i \\ & v_x = 0, v_y = -v \\ & \text{当 } x_i = x'_i, y_i < y'_i \\ & v_x = 0, v_y = v \end{aligned}$$

当起始点和目标点从一开始就一致:

$$\begin{aligned} & \text{当 } x_i = x'_i, y_i = y'_i \\ & v_x = 0, v_y = 0 \end{aligned}$$

此时建立任意两小车关于时间  $t$  的距离函数  $m(t)$ :

$$\begin{aligned} & t \text{ 时刻小车坐标 } (x_{it}, y_{it}) \\ & x_{it} = v_{ix}t + x_i, y_{it} = v_{iy}t + y_i \\ & \text{另任意一小车 } t \text{ 时刻坐标 } (x_{jt}, y_{jt}) \\ & x_{jt} = v_{jx}t + x_j, y_{jt} = v_{jy}t + y_j \\ & \text{求两小车在 } t \text{ 时刻时之间的距离} \\ & m(t) = \sqrt{(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2} \\ & (1 \leq i < j \leq 36) \\ & (m(t))^2 = (x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2 \end{aligned}$$

代入

$$\begin{aligned} x_{it} &= v_{ix}t + x_i, \\ y_{it} &= v_{iy}t + y_i, \\ x_{jt} &= v_{jx}t + x_j, \\ y_{jt} &= v_{jy}t + y_j \end{aligned}$$

$$\begin{aligned} (m(t))^2 &= [(v_{ix} - v_{jx})^2 + (v_{iy} - v_{jy})^2]t^2 \\ &+ 2[(v_{ix} - v_{jx})(x_i - x_j) + (v_{iy} - v_{jy})(y_i - y_j)]t \\ &+ [(x_i - x_j)^2 + (y_i - y_j)^2] \end{aligned}$$

由于小车在一个边长为 2 米的正方形区域内, 则:

$$\begin{aligned} & \text{区域最长长度 } q \\ & 2 \text{ m} = 200 \text{ cm} \\ & q = \sqrt{200^2 + 200^2} \\ & \text{则最长时间 } p \\ & p = \frac{q}{10} \\ & \text{故时间 } t \in [0, p] \end{aligned}$$

则我们可以从  $m(t)$  看出, 距离函数的平方是一个关于时间  $t$  的一元二次函数。  
为了保证小车在运行的时候不撞车, 则:

小车半径  $r = 2$

$(m(t))^2 > r^2$ , 小车不碰撞

$(m(t))^2 \geq r^2$ , 小车碰撞

由距离函数, 可以得到 36 辆小车之间是否碰撞, 可以得到符合题目要求的小车的调度方案。由于小车都是同一时间移动, 所以该方案最终耗时是:

$$t_1 = \frac{\max(M_{1j}, M_{2j}, \dots, M_{36j})}{v},$$

$$b_j = f_1(a_i), j \text{ 是 } b_j \text{ 下标}$$

$$M_{ij} = \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2}$$

$$v = 10 \text{ cm/s}$$

同理, 我们可以得到类似这样的函数:

$$y = f_2(x), y = f_3(x), \dots, y = f_n(x)$$

也可以得到这些方案的最终耗时:

$$t_n = \frac{\max(M_{1j}, M_{2j}, \dots, M_{36j})}{v},$$

$$b_j = f_1(a_i), j \text{ 是 } b_j \text{ 下标}$$

$$M_{ij} = \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2}$$

$$v = 10 \text{ cm/s}$$

为了在尽可能短的时间里面从图 7 到图 8, 所以在不考虑碰撞的情况下在  $n$  个方案中用时最短的是:

$$\min t = \min(t_1, t_2, \dots, t_n)$$

$$t_n = \frac{\max(M_{1j}, M_{2j}, \dots, M_{36j})}{v}$$

$$b_j = f_1(a_i), j \text{ 是 } b_j \text{ 下标}$$

$$M_{ij} = \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2}$$

$$v = 10 \text{ cm/s}$$

这样我们就可以得到一个关于时间  $T$  的数列

$$T = \{\min(t_1), \min(t_1, t_2), \dots, \min(t_1, t_2, \dots, t_n)\}$$

对这个数列求趋近于正无穷的极限:

$$t = \lim_{n \rightarrow +\infty} T$$

$$\lim_{n \rightarrow +\infty} T = \lim_{n \rightarrow +\infty} \{\min(t_1), \min(t_1, t_2), \dots, \min(t_1, t_2, \dots, t_n)\}$$

$$\lim_{n \rightarrow +\infty} T = \min(t_1, t_2, \dots, t_m)$$

$$t = \min(t_1, t_2, \dots, t_m)$$

这样, 就得到了理论上在考虑碰撞条件下, 小车的调度方案用时最短的方案。最初的情况如图 9 [4]所示其中红色的点为目标点, 蓝色的点为起始点。

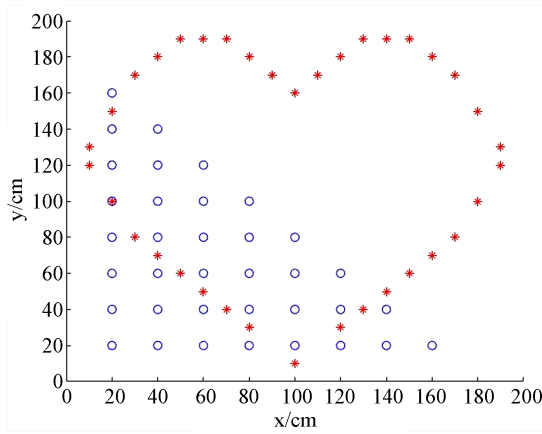


Figure 9. Contrast figure  
图 9. 对比图

图 10 [4]是经过一个不碰撞的调度方案, 最后 36 辆小车走到目标点的结果:

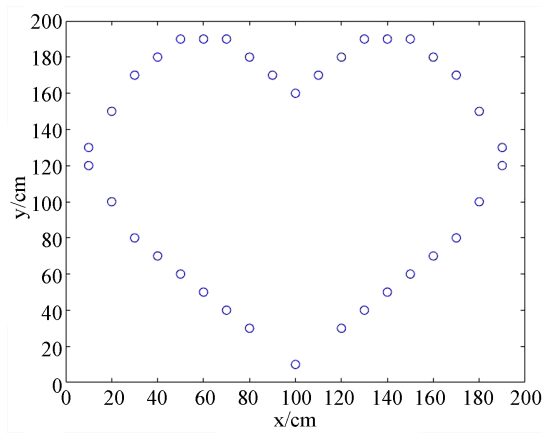


Figure 10. The figure of Results  
图 10. 结果图

图 11 [2]是运行一次程序后得出可能是用时最短的结果图 11。

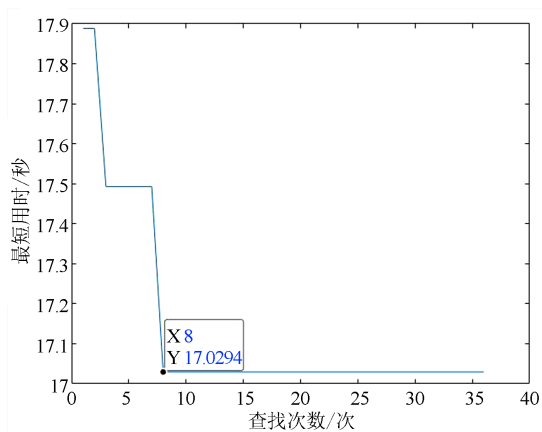


Figure 11. The shortest time graph considering collision  
图 11. 考虑碰撞最短用时结果图

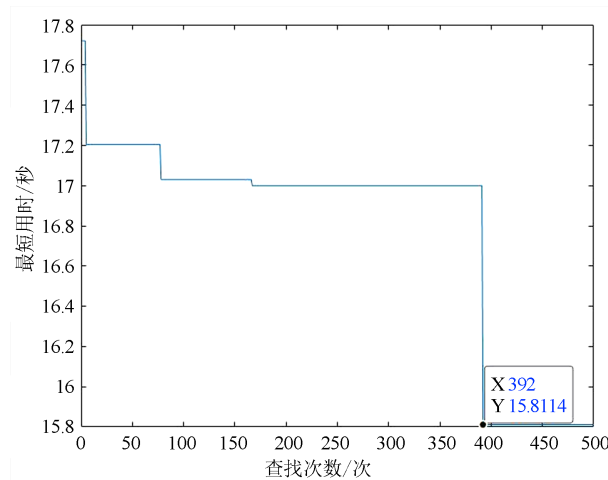
其中一个调度方案是：

[6, 25, 32, 36, 13, 20, 2, 16, 35, 8, 17, 21, 9, 4, 26, 27, 18, 12, 34, 28, 24, 22, 14, 15, 1, 19, 7, 29, 5, 11, 31, 3, 23, 10, 33, 30]

从上到下，从左到右，元素依次代表小车在起始点编号为 1 的去目标点编号为 6 的位置，依次类推。

需要注意的是，是先得出了一些方案(在不考虑碰撞的情况下得到的)再对这些得到的方案里面进行是否会碰撞分析。

如图 12 [4]所示是不考虑碰撞得到的



**Figure 12.** The shortest time graph without considering collision  
**图 12.** 不考虑碰撞最短用时结果图

通过图 12 我们可以知道，在不考虑碰撞的情况下，我们进行了 500 次查找 36 小车从图 7 到图 8 用时最快的调度方案。

由图 11 可知，我们在第 232 次查找中，得到了用时最快的调度方案的时间

那么可得，理论上查找次数趋近于无穷时调度方案用时最短时间是：

$$t = \lim_{n \rightarrow +\infty} T$$

$$\lim_{n \rightarrow +\infty} T = \lim_{n \rightarrow +\infty} \{ \min(t_1), \dots, \min(t_1, t_2, \dots, t_n) \}$$

$$\lim_{n \rightarrow +\infty} T = \min(t^1, t^2, \dots, t^m)$$

$$t = \min(t^1, t^2, \dots, t^m) = 16.4924$$

对比图 11 可知最初进行了 500 次实验，在这 500 次实验得到的方案基础上，判断是否会撞车，由图 11 可知，有 40 多次方案是不会碰撞的，而其中用时最短的是 17.02 s。但并不能列出所有的方案，这个值是否是最终用时最短的值，我们并不能确定。只能知道在试验次数趋于无穷时，可以得到真正的最短时间。

只能根据得出的结果，理论推导：

$$t = \lim_{n \rightarrow +\infty} T$$

$$\lim_{n \rightarrow +\infty} T = \lim_{n \rightarrow +\infty} \{ \min(t_1), \dots, \min(t_1, t_2, \dots, t_n) \}$$

$$\lim_{n \rightarrow +\infty} T = \min(t^1, t^2, \dots, t^m)$$

$$t = \min(t^1, t^2, \dots, t^m) = 17.02$$

## 9. 结语

从寻找老鼠找到食物的期望时间出发, 考虑到老鼠觅食中有可能出现一直找不到的情况, 为了找到确切的期望时间, 通过极限逼近的思想寻找老鼠找到食物的期望时间, 再通过此思想寻找老鼠找到食物的最短路, 最后将此思想应用于实际生活中, 得到了以下结论:

1) 面对出现在极端情况下有可能不知道老鼠觅食所需具体时间, 极限逼近的思想可以解决这一问题。

2) 在  $3 \times 3$  的格子中用极限逼近的思想寻找老鼠觅食最短路与枚举法得到的结果一致, 说明极限逼近这一方法可以应用于寻找最短路中。

3) 可以将极限逼近思想找最短路的方法运动到任意一个网格中, 寻找任意两个节点之间的最短路。

4) 枚举法面对复杂情况时不是很合适, 如经典的八皇后问题, 如果使用枚举法, 即使理论上能找到, 也会花费大量时间。

5) 极限逼近这一方法可以应用于生活中, 当面对一些人们指定最短出行路线、工厂如何让机器在最短时间上下物料、地铁乘务排班计划[6]等。只要判断出所面对的问题实质上是一个最短路问题, 都可以使用极限逼近的方法, 有一定实际意义。

6) 使用极限逼近思想得出的结果在实验次数较少时, 得到的可能不是最优解, 即使实验次数足够多, 也可能会出现不是最优解的情况, 对该方法得到的结果是否是最优解缺乏理论证明是本文的不足。

## 参考文献

- [1] 司守奎, 孙兆亮. 数学建模算法与应用[M]. 北京: 国防工业出版社, 2020: 40-42.
- [2] 张乃孝, 陈光, 孙猛. 算法与数据结构: C 语言描述[M]. 北京: 高等教育出版社, 2011: 309-315.
- [3] 姚慕生, 吴泉水, 谢启红. 高等代数学[M]. 上海: 复旦大学出版社, 2019: 57-58.
- [4] 刘卫国, 陈昭平, 张颖. MATLAB 程序设计与应用[M]. 北京: 高等教育出版社, 2002.
- [5] 屈婉玲, 耿素云, 张立昂. 离散数学[M]. 北京: 高等教育出版社, 2015: 303-304, 308-309.
- [6] 薛锋, 梁鹏, 李海, 陈崇双, 周天星. 地铁乘务排班计划优化的最短路快速算法[J]. 铁道科学与工程学报, 2022, 19(9): 79-87.