

基于启发式搜索算法的路劲规划

一、问题描述

左上角为坐标原点，水平向右为 x 轴方向，竖直向下为 y 轴方向。白色为自由栅格，黑色为障碍栅格，机器人只能在自由栅格中运动，并躲避障碍。每个栅格由唯一的坐标 (x, y) 表示。机器人一般有八个可移动方向。给出由初始位置 $(1, 1)$ 到目标位置 $(8, 6)$ 的最佳路线。

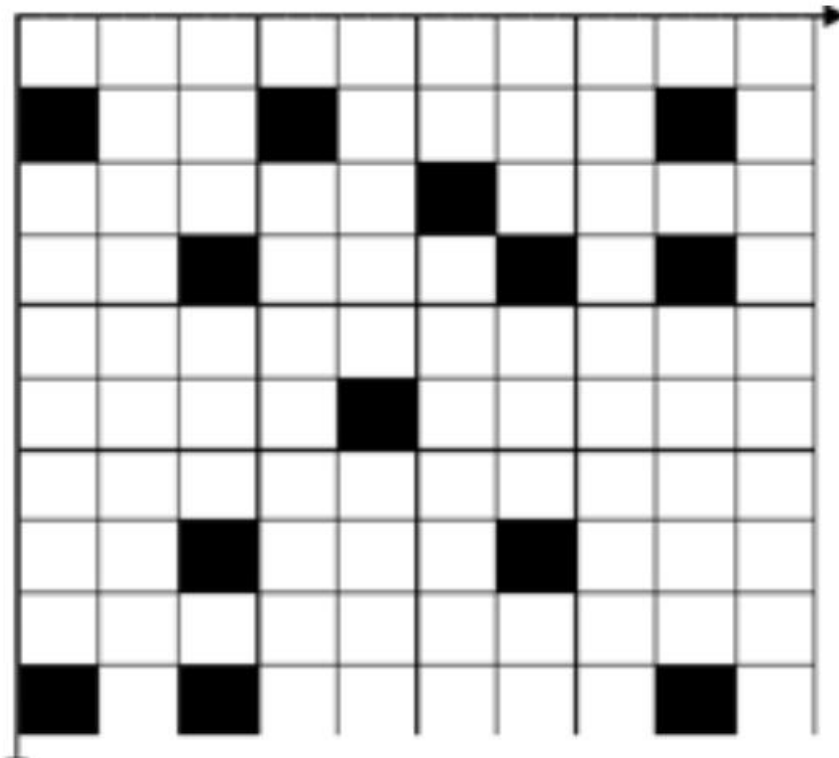


图 1 目标图像

二、状态空间表达和规则集

1. 状态空间表达

定义状态图是为 (S, F, G) ：状态 S 是点的坐标 (x, y) ，初始状态任意给定； F 是转换规则，这里存在八个移动方向表示八种规则； G 是目标状态，在搜索之前给定。

在表示图时，以 0 表示可以运动，1 表示障碍物，2 表示起始点，3 表示目标点，8 表示移动的机器人的路线。

2. 规则集

规则	操作	位置坐标变化
U	上方无墙→向上走一步	$(x, y) \rightarrow (x, y-1)$
D	下方无墙→向下走一步	$(x, y) \rightarrow (x, y+1)$

L	左方无墙→向左走一步	$(x, y) \rightarrow (x-1, y)$
R	右方无墙→向右走一步	$(x, y) \rightarrow (x+1, y)$
LU	左上方无墙→向左上走一步	$(x, y) \rightarrow (x-1, y-1)$
LD	左下方无墙→向左下走一步	$(x, y) \rightarrow (x-1, y+1)$
RU	右上方无墙→向右上走一步	$(x, y) \rightarrow (x+1, y-1)$
RD	右下方无墙→向右下走一步	$(x, y) \rightarrow (x+1, y+1)$

表 1 规则集

三、搜索过程

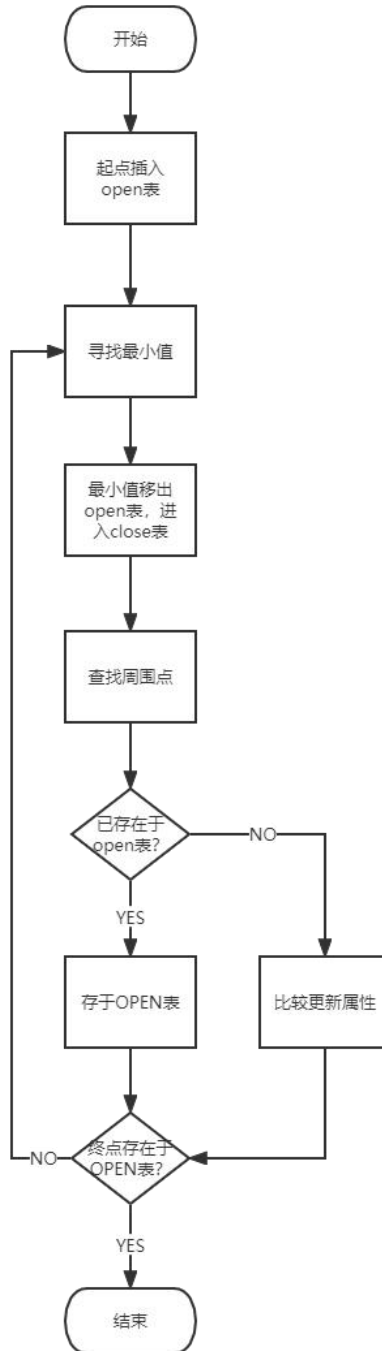


图 2 搜索过程

四、启发函数

代价函数 $g(n)$ 表示搜索深度：设定起始点的代价为 0，每往下拓展一层，代价加 1。

差异函数 $h(n)$ 表示当前状态和目标状态的“距离”。由于距离的定义函数有多种方式，比如欧氏距离，曼哈顿距离和切比雪夫距离等。因此，我在实验中，分别选取不同距离函数的定义作为差异函数，从而构建启发函数如下：

$$f(n) = g(n) + h(n)$$

1. 曼哈顿距离

曼哈顿距离是各坐标点的坐标差值的绝对值相加。二维平面上点 $a(x_1, y_1)$ 与点 $b(x_2, y_2)$ 之间的距离公式是：

$$d_{ab} = |x_1 - x_2| + |y_1 - y_2|$$

运行程序，由初始位置 (1, 1) 到目标位置 (8, 6)。结果拓展了 8 次，寻找路径为：(1, 1) → (2, 2) → (3, 3) → (4, 4) → (5, 5) → (6, 6) → (7, 6) → (8, 6)。由图可知，程序寻找的是最短路径。

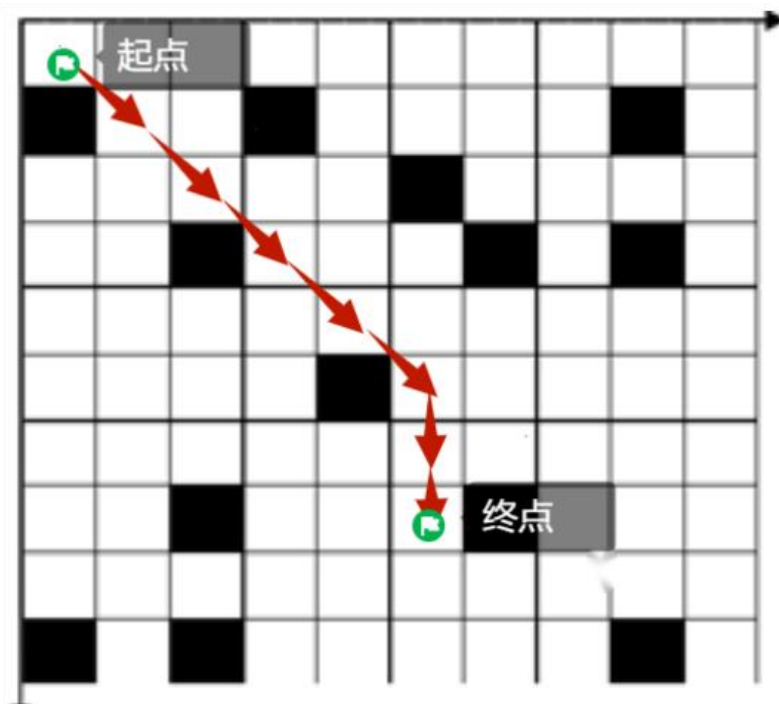


图 3 寻找路径-曼哈顿距离

2. 欧氏距离

欧式距离是两点之间的几何距离。即各坐标点的坐标之差的平方和相加，然后开根号(二范数)。二维平面上点 $a(x_1, y_1)$ 与点 $b(x_2, y_2)$ 之间的距离公式是：

$$d_{ab} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

运行程序，由初始位置(1, 1) 到目标位置(8, 6)。结果拓展了 8 次，寻找路径为：(1, 1) → (2, 2) → (3, 3) → (4, 4) → (5, 5) → (6, 6) → (7, 6) → (8, 6)。由图可知，程序寻找的是最短路径。

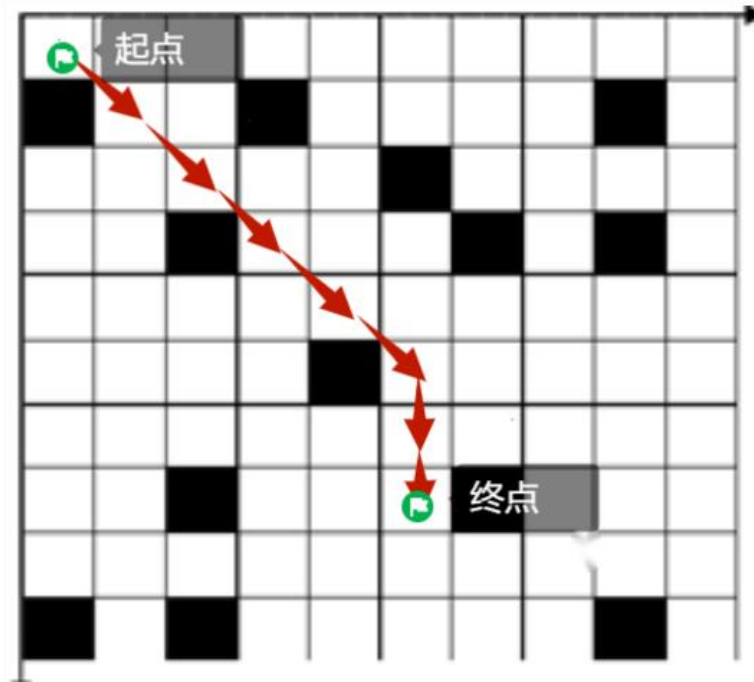


图 4 寻找路径-欧氏距离

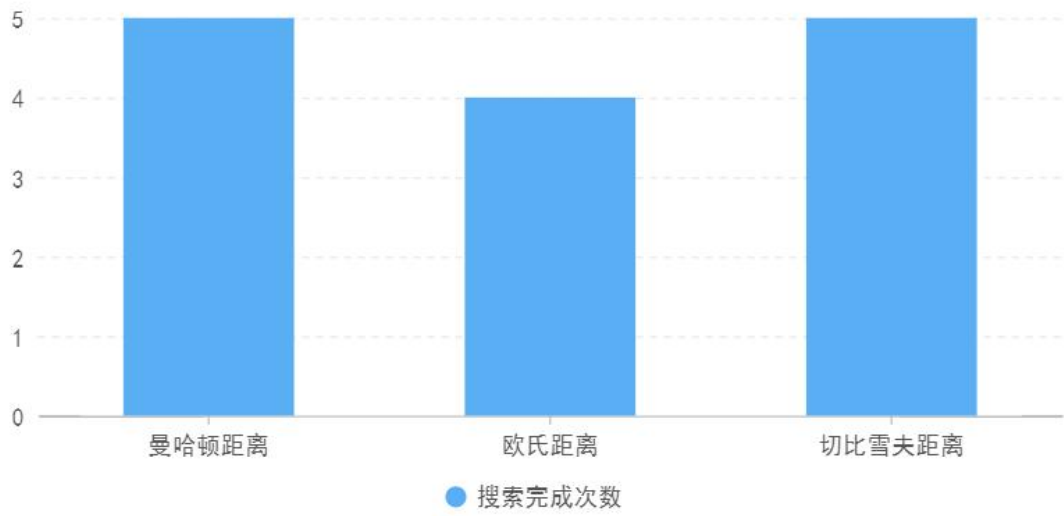
3. 切比雪夫距离

切比雪夫距离是各坐标的坐标差值中的最大值。二维平面上点 $a(x_1, y_1)$ 与点 $b(x_2, y_2)$ 之间的距离公式是：

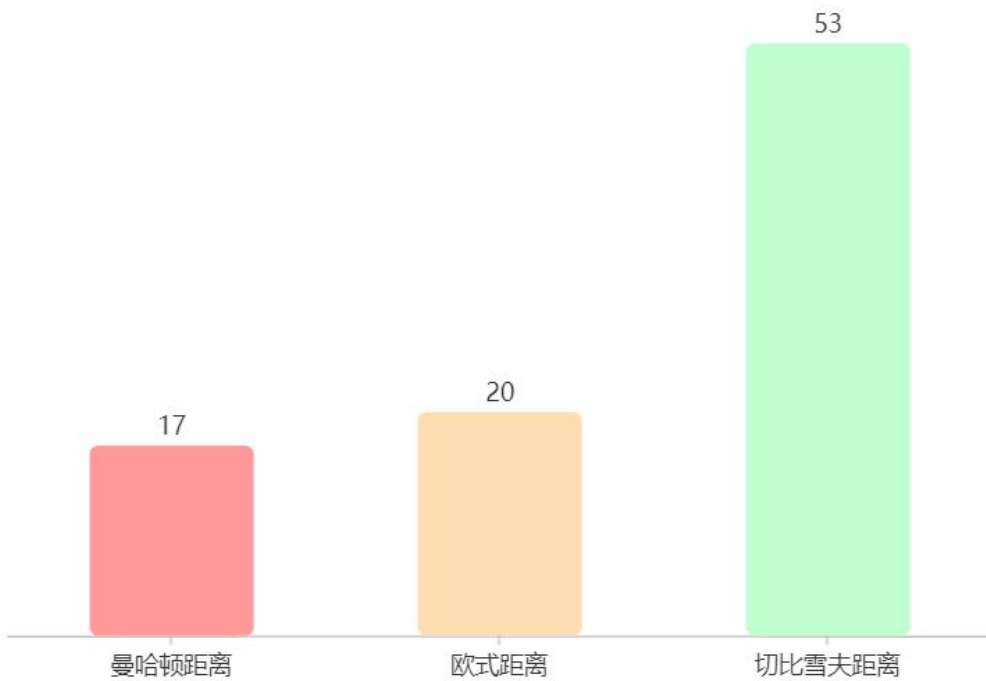
$$d_{ab} = \max(|x_1 - x_2|, |y_1 - y_2|)$$

运行程序，由初始位置(1, 1) 到目标位置(8, 6)。结果拓展了 49 次，寻找路径为：(1, 1) → (2, 2) → (3, 2) → (4, 2) → (5, 3) → (6, 4) → (7, 5) → (8, 6)。程序寻找的是最短路径。

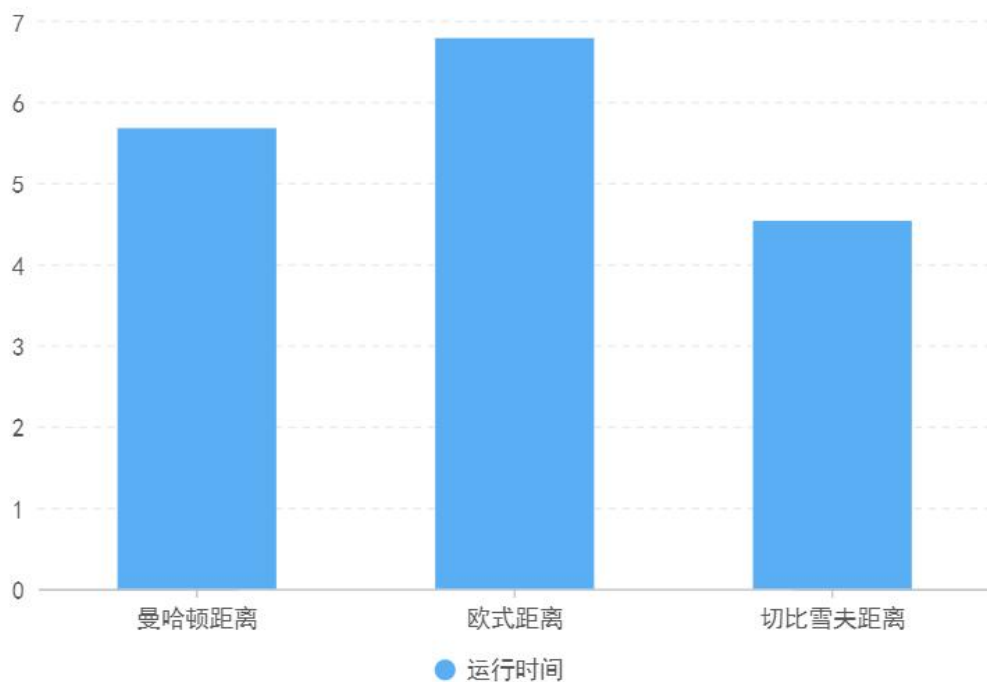
距离函数的搜索完成次数



距离函数的拓展次数



距离函数的运行时间



在过程中，发现初始位置(9, 1)到目标位置(5, 10)的最佳路线，采用欧氏距离的启发函数后，存在着扩展无法停止，找不到最优路径的情况，程序进入了“死循环”当中。分析可能原因为（1）移动到某一位位置可选择情况太多，程序走进了岔路 2）每次 h 变化太小，不能明显区分每个节点的“好坏”。

六、实验体会

本次实验主要是基于启发式搜索算法的路劲规划，并主要比较了不同距离函数对于搜索的影响。发现：采用曼哈顿距离在搜索过程中，拓展次数少，运行时间短，能较好地寻找到最短路径；采用欧式距离在搜索过程中，运行时间相对较长，且存在着扩展无法停止，找不到最优路径的情况；采用切比雪夫距离在搜索过程中，拓展次数多，但是每次计算差异函数 $h(n)$ 时运算相对简单，运行时间不会过长。综上所述，面对此种问题时，采用曼哈顿距离设计启发函数较好，采用切比雪夫距离次优。