

DOI:10.19651/j.cnki.emt.2108151

基于栅格地图改进的智能路径选择算法*

瞿新豪 丁云飞 谢亚琴

(南京信息工程大学电子与信息工程学院 南京 210044)

摘要: 交通和物流领域的发展,对路径规划算法的时间复杂度与精度提出了越来越高的要求。针对该问题,提出了一种基于栅格地图改进的智能路径选择算法。该算法利用方向信息来对算法进行优化设计,基于原有地图带有的始末位置的方向信息,定义权值矩阵与方向权重向量,以此作为路径中下个栅格的选择依据,并采取奖励与惩罚措施来提高算法的收敛速度。此外,算法还采取了多余栅格的优化措施来提高算法性能。通过对于典型仓储 AVG 问题模型的仿真来验证算法的有效性。结果表明,路径长度收敛到满足条件的较优解所需的时间,智能路径选择算法较 A* 算法及传统 Dijkstra 算法可以减少 10% 以上,具有较低的时间复杂度和较高的寻优效率。

关键词: 路径规划;栅格地图;智能路径选择算法;奖励与惩罚机制;障碍物率

中图分类号: TP24 **文献标识码:** A **国家标准学科分类代码:** 520.2099

Improved intelligent path selection algorithm based on grid map

Qu Xinhao Ding Yunfei Xie Yaqin

(College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China)

Abstract: With the development of transportation and logistics, higher requirements are put forward for the time complexity and accuracy of path planning algorithm. Aiming at this problem, proposes an improved intelligent path selection algorithm based on grid map which uses the direction information to optimize the algorithm. Based on the direction information of the beginning and end positions in the original map, the weight matrix and direction weight vector are defined as the basis for the selection of the next grid in the path, and reward and punishment measures are taken to improve the convergence speed of the algorithm. In addition, the algorithm also takes the optimization measures of redundant grid to improve the performance of the algorithm. The effectiveness of the algorithm is verified by the simulation of a typical storage AVG problem model. Simulation results show that the time required for the path length to converge to the optimal solution can be reduced by more than 10% compared with A* algorithm and traditional Dijkstra algorithm. The proposed intelligent path selection algorithm has lower time complexity and higher optimization efficiency as well.

Keywords: path planning; grid map; intelligent path selection algorithm; reward and punishment mechanism; obstacle rate

0 引 言

随着城市化进程的不断加快,交通和物流系统逐渐趋于智能化,其中智能交通系统(intelligent transportation system, ITS)与自动引导车(automatic guided vehicle, AGV)是典型代表。城市智能交通系统(ITS)建立在完善设备的基础上,采用了计算机网络的数据传输技术、信息控制技术和通信技术电子监控调度手段、信息传感技术等现代科学技术,是一种城市交通管理系统^[1]。AGV 由于具有

自动化程度高、灵活性高等优点,被广泛应用于柔性制造物流系统。多 AGV 的调度和路径规划,对降低物流运输成本,提升物流效率起着至关重要的作用^[2]。然而,上述技术发展的核心问题都包含了路径规划问题。

路径规划问题的主要部分是单源路径规划问题,即求解某指定点到网络中所有其他节点之间的最短距离。传统的路径规划算法包括 Dijkstra 算法^[3]、A* 算法以及各类仿生算法如蚁群算法^[4-5]和遗传算法^[6]。这些算法由于算法逻辑的不同,因而算法效率与路径质量各不相同。例如

收稿日期:2021-10-21

* 基金项目:国家自然科学基金(62001238)、江苏省高等学校大学生创新创业训练计划(202010300061Y)项目资助

Dijkstra 算法存在时间、空间复杂度过高的问题, A^* 算法由于启发函数的限制对于非凹障碍型地图的适用性不高, 而遗传算法存在包括“顶端优势”在内的局部最优情况和迭代进化过程缓慢等局限性。这些不足越来越难以应对日益高并发的调度规划需求。

针对上述算法的局限性, 国内外学者提出了很多相应的改进措施。文献[7]采用邻接表法存储数据, 以优化存储结构, 并使用优先队列法排序, 提高算法执行。文献[8]采用全局规划启发函数以克服限制, 并通过变步长 A^* 算法以优化用时。文献[9]采用结合禁忌搜索的思想以优化规划, 增强全局搜索能力。

虽然上述改进措施都有效地提升了算法的优化能力, 但是这些改进主要是针对路径长度的优化, 而在算法的时间复杂度上考虑较少。在当前对于高并发路径规划的需求下, 算法效率与路径质量之间的平衡是重中之重。本文提出一种基于栅格地图改进的路径规划算法。本算法利用栅格地图带有的方位信息, 引导寻路方向, 同时利用对多余栅格在算法中的优化, 加快收敛速度, 同时通过采取奖励与惩罚等反馈来避免陷入寻路中的局部最优, 从而在保证算法精度的情况下提高算法效率。文中将该算法应用在典型实例——仓储 AVG 路径规划问题, 仿真结果显示了该算法具有高效性。

本文各节的内容安排如下: 第1节给出了栅格地图模型的构建, 第2节提出了基于栅格地图改进的路径规划算法及其算法实现过程中的若干关键技术, 第3节给出了仿真实验结果及分析, 第4节为本文的结束语。

1 栅格地图模型

在构建路径算法时, 地图的选取十分重要。地图的表示方法有很多种, 每一种地图所展示的信息种类都不相同。例如在地铁路线图和景点图中, 每一个地点用一个点来表示, 通过显示点与点之间的边以及点的方位提供信息, 该种类型的地图称为拓扑地图(topological map)^[10]; 在世界地图等大型地图中, 用经纬度或者横纵坐标来表示每一个地点, 这种类型称为尺度地图(metric map)^[11]; 还有一种地图称为语义地图(semantic map)^[12], 里面的每一个地点和道路都是用标签的集合表示, 信息比较详细和具体。在地理系统中, 拓扑地图常用于路径规划, 尺度地图通常用于地图的构建和定位, 而语义地图则用于人机交互之中。栅格地图^[13](grid map)、特征地图和点云地图等属于典型的尺度地图。不同种类地图之间也可以通过信息的增减和推导相互转化。栅格地图是基于一套行列组成的方格数据模型, 使用一组方格描述地理要素, 每一个方格的值代表一个现实的地理要素。由于栅格地图的构建、表示和保存相对较容易, 且位置具有唯一性, 对于短路径的规划更加方便, 因此本文选择栅格地图进行仿真与研究。

1.1 栅格地图

栅格地图是指将地图栅格化, 将一块区域看作方正的矩形, 并将矩形内部均分成块状的小栅格。每个栅格都带有自己的权值或其他信息, 即对于每个栅格的状态, 要么占用, 要么空闲, 要么未知(即初始化状态)。本文中每个栅格上所带的信息代表着该栅格被占据的概率, 若该栅格中有障碍物或存在干扰项等因素时, 则该栅格被占据的概率为0, 在寻路过程中, 该栅格会被自动排除; 若栅格中不存在任何的干扰, 那么该栅格可能被占据。栅格地图在引入启发函数的 A^* 算法、蚁群算法和遗传算法等仿生算法中均有着很重要的应用。

1.2 栅格地图的建立

对于一个矩形路径规划场地, 可以根据栅格边长 N 将场地划分为多个矩形栅格, 使原地图成为 row 行 col 列的栅格地图。地图中每一个栅格都代表一个节点, 由相应的序号标注。第 i 行第 j 列栅格的序号 $s(i, j)$ 可以表示为式(1):

$$s(i, j) = i + (j - 1) \times j, i \in [1, row], j \in [1, col] \quad (1)$$

每一个栅格的权值由权值矩阵 $P(m)$ 表示, $P(m)$ 的值表示第 m 号栅格被占据的概率。根据地图初始信息, 栅格可以分为障碍栅格与可通行栅格。根据上文所述, $P(m)$ 的初始值设置为式(2):

$$P(m) = \begin{cases} 0, & \text{栅格 } m \text{ 有障碍} \\ 1, & \text{栅格 } m \text{ 可通行} \end{cases} \quad (2)$$

在算法中, $P(m)$ 值的变化代表着栅格与周围栅格的关联程度。该值越大, 栅格与周围栅格的关联程度越大, 该栅格被占据的概率就越大。

从栅格地图的划分可以看出, 一个节点最多有8个方向。所以对于每一个具体的栅格, 其与周围栅格的序号关系可由矩阵 N 表示。规定方位顺序沿逆时针方向依次为: 左上、左、左下、下、右下、右、右上、上。依照上文式(1)的栅格序号的标号, 可得 N 为式(3):

$$N = [-row - 1, -row, -row + 1, +1, row + 1, row, row - 1, -1] \quad (3)$$

由此, 可得栅格 s 的周围栅格的栅格序号 $cand(s)$ 满足式(4):

$$cand(s) = s + N \quad (4)$$

但是这只是处于地图内部的自由栅格, 而对于地图边界的边界栅格, 由于存在部分方向超出地图界限, 根据式(4)进行计算就会存在错误。上边界栅格条件为 $\text{mod}(s, row) = 1$, 下边界栅格条件为 $\text{mod}(s, row) = 0$, 左边界栅格条件为 $1 \leq s \leq row$, 右边界栅格条件为 $row \times (col - 1) + 1 \leq s \leq row \times col$ 。规定满足上述条件的栅格对应方向的栅格序号为0, 以便后续操作时识别该方向不存在栅格。

1.3 多余栅格的优化

栅格地图是根据实际地图进行划分的,含有方向信息,因此只需要按指定方向检索即可,不会绕弯路。但是由于栅格的面积大小是一定的,会产生很多多余的栅格,寻路时运算时间上会造成浪费。以图 1 为例,整块区域中灰色的部分代表障碍物,若要从 A 格走到 B 格,根据 A 格和 B 格的方位信息,算法检索可以从起始点的右下方检索,而 G 格方向可以忽略。但是,在寻路过程中,D 格、E 格两个栅格很明显是多余的,不需要对这两个栅格进行判断,若删去多余栅格的判断时间,效率就会得到提高。这就需要对栅格地图做一些转变,在整个区域中,除了目标格点和起始格点,可将剩下的格点分为节点格点和多余格点。若一个格点除了后退方向只有一个方向可走,可视为多余格点,例如图 1 中的 D 和 E;若有多个方向可走,可视为节点格点,例如图 1 中的格点 C 和 F。在寻路过程中,可以直接跳过多余格点,得到简化后的栅格图如图 2 所示。

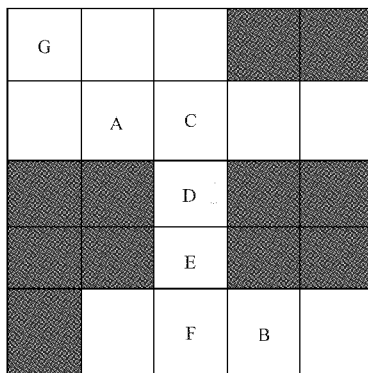


图 1 栅格图示例

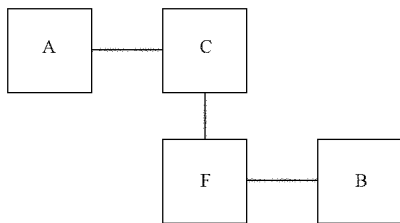


图 2 化简后的栅格

2 路径选择算法

2.1 算法思想

本文所提出的算法的基本思路是利用方向信息来对算法进行优化设计,促进算法收敛。基于原有地图带有的始末位置的方向信息,定义权值矩阵与方向权重向量,根据当前栅格的周围的有效栅格的被选择概率的大小,选取概率最大的栅格作为新的当前栅格。若存在多个概率最大的栅格,算法则可能随机选择栅格作为新的当前栅格。每次更新当前栅格,就重复进行选择概率的计算与选择,循环执行,直到当前栅格为目标栅格,完成一次有效的路径规划。

若一次规划未能得到有效路径,则采取惩罚措施,避免再次陷入此循环。若本次规划优于上次规划,则对本次规划结果进行奖励。路径规划中对多余栅格进行优化,以提高规划效率。每次规划结束之后,与前几组结果进行对比,判断算法是否已经收敛。

2.2 方向权重向量

栅格地图的方向信息可以体现在方向权重向量中,方向权重向量在刚开始的路径规划中起引导作用。利用当前栅格和目标栅格的位置关系,给每一个周围栅格赋予相应的方向权重。权重值范围为 $[0,1]$,若某一方向的方向权重为 1,则表示目标栅格位于当前栅格的这个方向上;若有两个方向的方向权重为 1,目标栅格在这两个方向之间;若某一方向的方向权重为 0,则目标栅格在相反方向;若某一方向权重在两者之间,表示该方向偏离目标栅格方向,有可能被选取。方向权重向量存于矩阵 W 中,顺序与矩阵 N 相同,依次为左上,左,左下,下,右下,右,右上,上。例如:若目标栅格在当前栅格的右侧,则 $W=[0.25,0,0.25,0.5,0.75,1,0.75,0.5]$ 。由于具有 8 种情况,因而存在 8 个权值矩阵。矩阵中除上述已规定外的权重值为经验值设定。

2.3 路径选择概率

对于第 i 组第 j 次路径规划下的当前栅格为 s 。其周围的有效邻近可到达栅格为 a 个,将这些栅格构成待选集合 V 。对于待选集合内每一个栅格 v ,其选择概率依式(5)计算:

$$Pro_{s,t} = \begin{cases} \frac{P(t)W(k_t)}{\sum_{z \in V} P(z)W(k_z)}, & t \notin Trace \cap t \in V \\ 0, & \text{其他} \end{cases} \quad (5)$$

式中: $Pro_{s,t}$ 表示从栅格 s 到栅格 t 的概率, $P(t)$ 为栅格 t 的关联值, $W(k_t)$ 为栅格 t 对应的方向向量的值, z 为待选集合 V 中的栅格, $P(z)$ 为栅格 z 的关联值, $W(k_z)$ 为栅格 z 对应的方向向量的值。矩阵 $Trace$ 为本组规划中已规划栅格集合,此集合可以排除之前规划栅格重复参与规划,以避免回退、绕圈等陷入死循环的情况,也可使在同组不同次路径规划中不重复单一路径规划,保证同组内规划情况的多样性,以便区别比较。

待选栅格 t 与当前栅格 s 的序号如式(6)所示。

$$N(k_t) = t - s, 1 \leq k_t \leq a \quad (6)$$

计算概率之后,更新当前栅格为被选择概率最大的栅格,并针对更新后的当前栅格计算其周围栅格的被选择概率。若存在多个最大概率栅格,本算法规定从中任意选取 1 个栅格更新为当前栅格。

2.4 权值矩阵的修正

奖励和惩罚机制是对权值矩阵进行修正的主要形式。路径规划中陷入障碍物或已走栅格围成死路的现象十分常见,因而需要对结果进行修正。在第 1 次寻路时,先由方向权重向量引导下一个栅格的走向,直到与目标格点重合,生

成一条路径 L_n ，该路径所对应的矩阵中存放着该路径所包含的格点的序号。在之后的寻路中会将其中部分格点排除。最后找出所有可能的路径放入数组 $path$ 中， $path$ 中存放的是每一条路径 L_n 所经过格点的序号，每一条路径相对应的总权值放入数组 $value$ 中。 $value$ 中的值会一一进行比较，直到找出最优解。比较过程中，会将较优解所对应的路径的序号提取出来，启用奖励机制，增加每个格点被占据的概率，其余的启用惩罚机制，最后在相应的记忆矩阵上进行修正。每一组结束后，都会得到一个修正后的记忆矩阵。根据最后的记忆矩阵可以选择出最优的路径。

修正中有两个重要的参数，分别是奖励系数 A 和惩罚系数 B ，分别代表着对路径奖励或惩罚的程度。若第 n 条路径 L_n 优于前一条路径 L_{n-1} ，则会在权值矩阵相应的格点上加上 $A \times value(n)$ ；若第 n 条路径不如上一条路径 L_{n-1} ，则会在权值矩阵相应的格点上减去 $B \times value(n)$ 。若出现陷入死路，未完成规划，则在权值矩阵对当前路径所有经过的栅格按惩罚系数 B 减少。在数值设定上， A 的值要远远大于 B ，但 A 不是越大越好，如果 A 值大到一定程度时，会出现“回走”的现象，就会进入一个死循环。所以 A 和 B 的值要根据实际权值来调整：如果实际权值较大，应该适当调大 A 值，调小 B 值；如果实际权值较小，应该适当调小 A 值，调大 B 值。此外，也可以给“修正”加一个阈值，防止“回走”现象的出现。此外，若当前栅格的权值减小到一个设定的大于 0 的极小值时，规定不再减小，固定为该极小值，既能保证再次规划时该栅格被选中的概率极低，又将其与障碍物栅格相区别。

2.5 组数选择

当前组路径规划完成后，与前几组完成规划的规划结果进行对比，比较结果是否相同。若规划完全相同，则认为算法已经收敛，算法停止，输出该路径规划结果；否则继续执行该算法。

2.6 算法流程

综上所述，本文的智能路径规划算法的算法流程如图 3 所示。

如算法 1 所示，基于栅格地图进行改进的智能路径选择算法包含如下 8 个步骤。

3 仿真与分析

为验证本改进算法在路径选择中的可行性与有效性，本文选择以典型的仓储自动引导车路径规划问题为例，采用 MATLAB 进行仿真测试。实验中，首先将仓库简化为一个边长为 N 的正方形，分别针对边长为 10×10 、 100×100 和 200×200 三种不同大小的仓库，将其栅格化获得不同尺寸的栅格地图。引导车需要将物品从 P 点运输到 Q 点，P 点为地图上左上角栅格，Q 点为地图右下角栅格。在仿真中采用多种算法进行路径规划，以进行对比。表格中数据均采取 10 次算法仿真所得到的平均值。每一

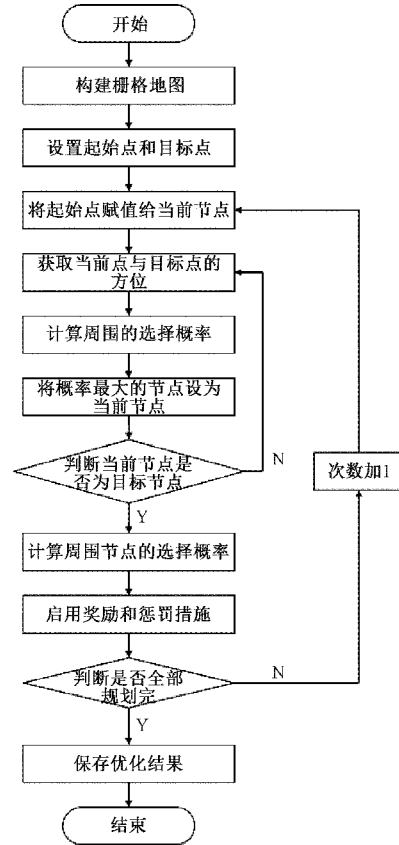


图 3 算法流程

算法 1 基于栅格地图改进的智能路径选择算法

- 1) 构建栅格地图，并初始化算法参数，包括组数 m 和次数 n 、奖励系数 A 与惩罚系数 B 、地图行数 row 与列数 col 、起始栅格与目标栅格等。
- 2) 设置起始栅格为当前栅格。
- 3) 将当前栅格放入已规划栅格 $Trace$ 中，并判断有效的待选相邻栅格，放入集合 $cand$ 中。
- 4) 由当前栅格与目标栅格的位置关系，确定方向向量，并计算选择概率。
- 5) 若所有选择概率均为 0，则转至步骤 7)；否则，选择概率最大的栅格作为当前栅格。
- 6) 判断当前栅格是否为目标栅格。若否，则转至步骤 3)，继续进行路径规划；若是，则保存路径，并计算路径长度，与之前规划路径进行对比，更新权值矩阵，进行奖励与惩罚措施。
- 7) 判断本组路径规划是否全部完成。若否，则转至步骤 2) 进行下一次路径规划；若是，且 n 次路径规划均为无效规划，则更新权值矩阵，进行惩罚措施；若至少有一次有效规划，则对本组规划中最优规划进行存储。
- 8) 若本次存储的规划与前几组路径规划结果完全相同，或 m 组路径规划完成，则对保存的最短路径进行优化，输出规划结果。

次移动的距离定义为从一个栅格的中心到另一个栅格的中心(即栅格边长)的长度,并规定该长度为单位长度。

3.1 10×10 栅格地图

图 4 所示为栅格地图大小为 10×10,障碍物率为 21% (图中黑色方块所示区域为障碍物)的情况下路径规划结果,表 1 所示为算法的运行时长及路径长度的实验对比。将本文所提出的算法与文献[14]所提算法以及当前常用的传统 Dijkstra 算法和 A* 算法分别进行了对比。

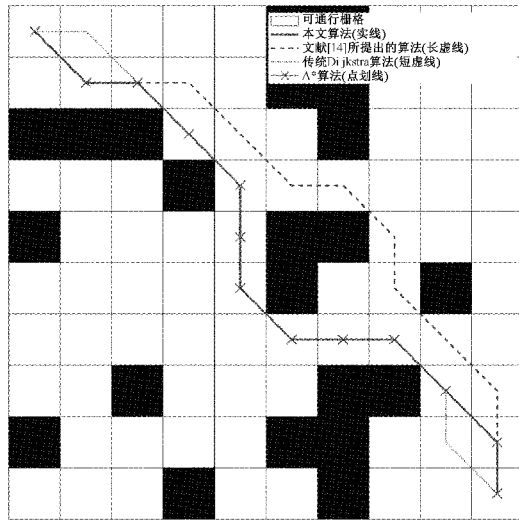


图 4 10×10 路径规划对比图

表 1 算法时长与路径长度对比

算法选择	算法用时/s	路径长度
本文算法	0.746	14.485 2
文献[14]所提算法	0.792	14.485 2
A* 算法	0.815	14.485 2
传统 Dijkstra 算法	0.929	14.485 2

由图 4 和表 1 可明显看出,算法在小地图小障碍物率下可以准确地获得最优路径,且在 4 个算法都能取得最优路径的情况下,与传统 Dijkstra 算法相比,本文所提出的算法用时节约 19.69%,与 A* 算法相比,用时节约 10.38%,与文献[14]所提算法相比,用时节约 5.81%。算法总体优化十分理想。

3.2 100×100 栅格地图

100×100 的栅格地图如图 5 所示,该栅格地图对应的障碍物率为 58.65%,其中障碍物率为障碍物栅格(图中黑色区域)占总栅格的比例。本文所提出的算法与文献[14]所提出的算法以及当前常用的传统 Dijkstra 算法和 A* 算法进行对比。所得结果如图 5 所示。

图 5 直观地展示了 4 种不同的算法在 100×100 的栅格地图中所得到的路径,不同算法的算法时长和路径长度对比如表 2 所示。由表 2 可以看出,与其余 3 种算法相比,本文所提出的算法用时明显减少,与传统 Dijkstra 算法相

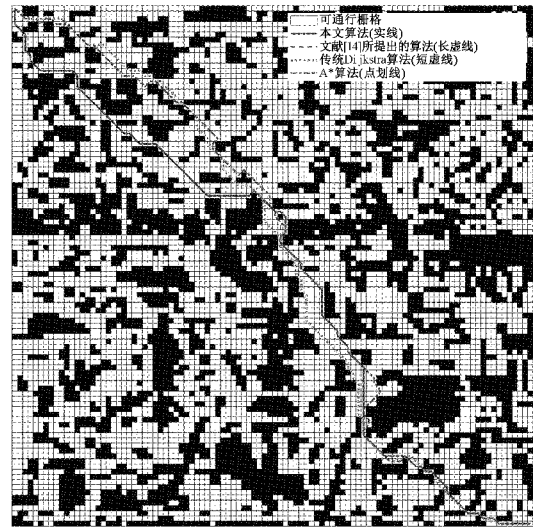


图 5 100×100 的栅格地图下不同算法的路径规划对比图

比,用时节约 33.54%,与 A* 算法用时相比节约了 12.16%,与文献[14]所提出的算法相比用时节约了 6.23%。但是从该表中也能看到,在路径长度方面,与传统的 Dijkstra 算法以及文献[14]所提出的算法相比,本文所提算法得到的路径长度略有增加,与传统的 Dijkstra 算法相比,路径长度增加了 2.89%,这说明算法还存在局部的绕路现象。

表 2 算法时长与路径长度对比

算法选择	算法用时/s	路径长度
本文算法	5.633	163.338 1
文献[14]所提算法	5.984	161.923 9
A* 算法	6.318	167.036 6
传统 Dijkstra 算法	8.476	158.752 3

利用上述运行结果,对算法的迭代情况进行了分析。事实上,由于算法具有一定的随机性,算法运行结果会存在一定的浮动,然而在实际情况下,可以接受一个与最优解相差不大的范围作为优解域^[15],实验中设置算法求出的最小值的 1.1 倍作为优解阈值。最后路径迭代效果如图 6 所示。

从图 6 可知,算法在接近 90 次迭代后基本完成了路径规划的任务,关键的是仅仅 4 次迭代就将结果收敛到优解域内,迭代次数少。在算法效率上,改进后的算法能够实现更快速的搜索,在一定程度上减少了搜索时间,提升了算法效率。

仿真同时针对奖励系数 A 与惩罚系数 B 在算法中的影响进行了研究。在测试时更改了起始点与目标点位置,起始栅格设置为 5 601,终点栅格设置为 4 700。

首先控制惩罚系数 B=0.003,奖励系数 A 设置范围为 0~21.6,每个奖励系数 A 值测试 108 组数据,A 的增长步长设置为 0.2,路径长度随 A 值变化的趋势如图 7 所示。

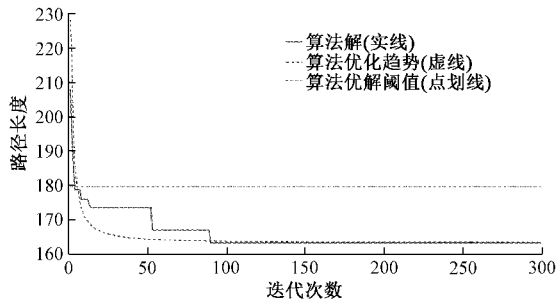


图6 算法优化迭代趋势图

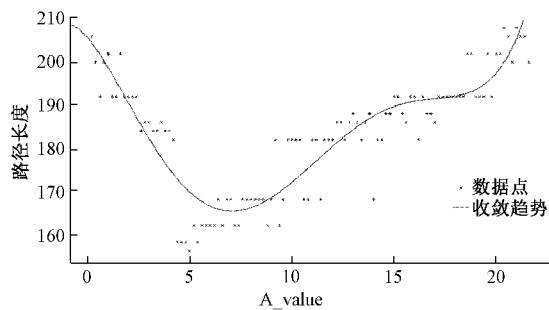


图7 路径长度随A值的变化趋势

随后控制奖励系数 $A=5$, 惩罚系数 B 设置范围为 $0 \sim 0.0108$, 每个惩罚系数 B 值测试 108 组数据, B 的增长步长为 0.0001 , 路径长度随 B 值变化的趋势如图 8 所示。

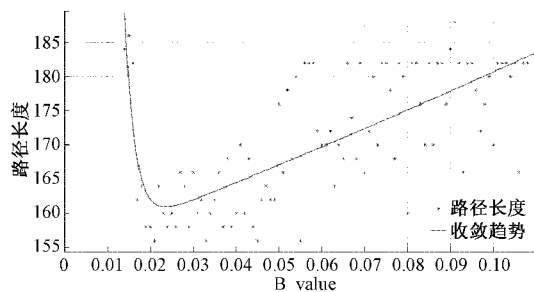


图8 路径长度随B值的变化趋势

由图 7 可知, 路径最小值在 $A=5$ 附近出现, A 值在 $5 \sim 7$ 的范围内收敛效果都处于优解域内, 而在其他范围内算法的收敛性都较差; 由图 8 可得, 算法在 B 很小时无法找到解, 之后算法快速收敛至最小值附近, 然后逐渐发散。

综合上图 7 和 8 可得, A 值变化对路径长度的影响呈现较为连续的分布, 说明 A 值在优化上作用更为关键; B 值的变化对路径长度的影响结果更明显出现了成块的阶梯分布, 说明 B 值的变化会对算法的收敛速度产生一定的影响。

由此可知, 惩罚系数 B 的设置对算法结果的影响要远大于奖励系数 A 的设置。所以在确定系数时应当优先考虑惩罚系数 B , 以保证算法的收敛, 体现惩罚的效果。而 A 值的设置需要比 B 大, 但是也需要满足一定的条件, A 不可过大, 否则会出现“回走”的现象, 从而陷入死循环。

3.3 200×200 栅格地图

200×200 的栅格地图如图 9 所示, 该栅格地图对应的障碍物率为 55.11% 。本文所提出的算法与文献[14]所提出的算法以及当前常用的传统 Dijkstra 算法和 A^* 算法进行对比。所得结果如图 9 所示。

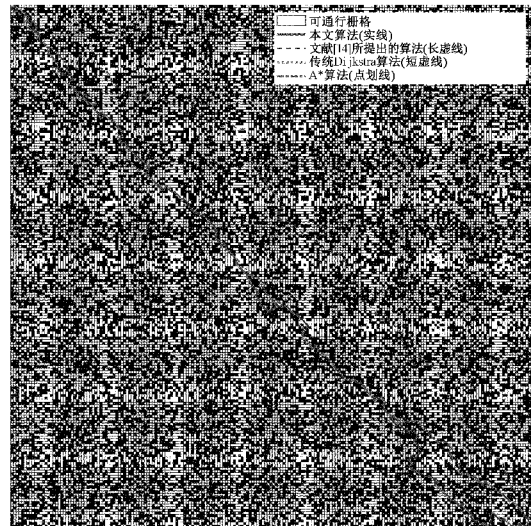


图9 200×200的栅格地图下不同算法的路径规划对比图

图 9 直观地展示了 4 种不同的算法在 200×200 的栅格地图中所得到的路径, 不同算法的算法时长和路径长度对比如表 3 所示。由表 3 可以看出, 与其余 3 种算法相比, 本文所提出的算法用时明显减少, 与传统 Dijkstra 算法相比, 用时节约 33.80% , 与 A^* 算法用时相比节约了 17.45% , 与文献[14]所提出的算法相比用时节约了 11.74% 。但是从该表中也能看到, 在路径长度方面, 与传统的 Dijkstra 算法以及 A^* 所提出的算法相比, 本文所提算法得到的路径长度略有增加, 与传统的 Dijkstra 算法相比, 路径长度增加了 11.01% , 这说明算法还存在局部的绕路现象。

表3 算法时长与路径长度对比

选择算法	算法用时/s	路径长度
本文算法	12.629	353.801 1
文献[14]所提算法	14.310	375.943 2
A^* 算法	15.298	332.901 6
传统 Dijkstra 算法	19.077	318.717 8

3.4 不同障碍物率情况下算法用时及路径长度比较

障碍物率是描述地图复杂程度的一个重要指标, 其大小会对算法的存在很大的影响。由于障碍物比率过小的地图缺乏算法规划的必要, 而障碍物比率过大的地图很大程度上并不存在一条能够到达目的地的路线, 因而设置障碍物比率的研究范围为 $25\% \sim 75\%$ 。地图使用 100×100 的

栅格地图。测试采用随机生成地图进行仿真,比较各算法用时变化。各算法用时比较图如图 10 所示。

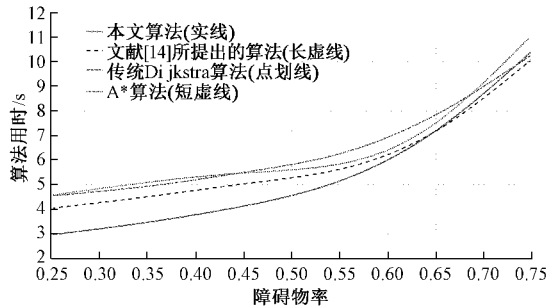


图 10 算法用时随障碍物率变化

由图 10 的仿真结果可以看出,随着障碍物比率的增大,4 个算法的执行时间都随之增大。本文所提出的算法在低障碍物率的情况下,明显优于另外 3 个算法,仅在障碍物率高于 64% 的情况下,算法执行时间急剧增大,与传统算法执行时间相当。这是由于随着障碍物率的增大,栅格地图的复杂度也随之增大,仅依靠方位这一信息不足以较好地收敛到最短径,算法执行时难以快速收敛,进行了很多的无效搜索,从而导致性能下降。

在不同障碍率情况下,路径长度的比较如图 11 所示。

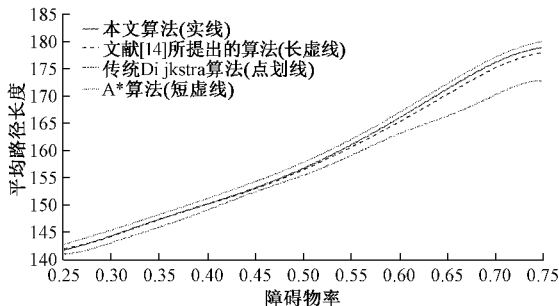


图 11 路径长度随障碍物率变化

由图 11 的仿真结果可以看出,随着障碍物率增加,各算法的平均路径长度均随之增加,传统 Dijkstra 算法在大障碍率的情况下增长趋势略小于其他 3 种算法。综合图 10 和 11 可以看出,本文所提算法在障碍物率较大的情况下也也能正常工作。

3.5 仿真结果分析

由上述仿真结果可知,本文所提出的算法在处理存在较多节点的地图时,收敛速度快,能够有效节省搜索时间。但本文算法相比于传统 Dijkstra 算法,随着节点变多,所求得的最短路径也会相应变长;算法的执行效率依赖于奖励和惩罚系数值的设定;在障碍物率不超过 64% 的情况下,算法的执行性能优于传统的路径选择算法;在障碍物率超过 64% 的情况下,性能略有降低。综上可得,本文算法在小障碍物率情况下算法用时更优;在小地图中所求最短路径较优;在大地图中最短路径相对较长,但算法的用时的相对更短。这说明算法在应对小尺度高并发的路径规划时有

很好的效率,而在大尺度地图的路径规划中也有不错的优势,算法能较为有效地应对日益增长的智能规划需求,体现了算法的有效性 with 高效性。

4 结 论

在现有的对全局路径进行规划的算法研究中,算法效率和路径质量两者的平衡是重中之重。本文基于栅格地图的算法思路,从方向上对寻路范围进行改进,将搜索区域从整圆缩小为一个扇形。在处理存在大量节点的路径规划问题时,能够提高算法执行效率。但是,该算法也存在许多不足,例如奖励和惩罚系数不能自适应设置,需要根据经验自行设置,对于障碍物率较大时的地图,执行效率不高等问题,后续还有很大的优化空间。

参 考 文 献

- [1] 艾峰. 城市智能交通系统的发展现状与趋势[J]. 智慧城市, 2021, 7(20):140-141.
- [2] 姜辰凯,李智,盘书宝,等. 基于改进 Dijkstra 算法的 AGVs 无碰撞路径规划[J]. 计算机科学, 2020, 47(8):272-277.
- [3] 王树梅,黄石,臧禹顺. 基于 Dijkstra 算法的城市物流公交系统优化[J]. 计算机技术与发展, 2021,31(10): 179-183,189.
- [4] RAMAMOORTHY R, THANGAVELU M. An improved distance-based ant colony optimization routing for vehicular ad hoc networks[J]. International Journal of Communication Systems, 2020, 33(14): e4502.
- [5] PANDA R K, CHOUDHURY B B. An effective path planning of mobile robot using genetic algorithm[J]. 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015: 287-291.
- [6] 杜映峰,陈万米,范彬彬. 群智能算法在路径规划中的研究及应用[J]. 电子测量技术, 2016,39(11):65-70.
- [7] 张本俊,周海娇,刘淑琴. 改进 Dijkstra 算法在公共交通出行的研究[J]. 物联网技术, 2018, 8(11):45-48.
- [8] ERKE S, BIN D, YIMING N, et al. An improved A-Star based path planning algorithm for autonomous land vehicles[J]. International Journal of Advanced Robotic Systems, 2020, 17(5):172988142096226.
- [9] 吴方圆. 基于改进蜂群算法的工业机器人路径规划研究[J]. 电子测量技术, 2019,42(7):13-17.
- [10] 王永胜,罗禹贡,黄晨,等. 基于拓扑地图的自主泊车路径协调与优化策略[J]. 中国公路学报, 2021,34(1): 177-187.
- [11] 宗琴,姜树辉,彭荃. 多尺度矢量地图数据库中模糊相似关系的图谱构建[J]. 测绘科学, 2020, 45(3): 117-121.

- [12] 金泰宇,黄劲松. 一种基于语义地图的激光雷达定位方法[J/OL]. 测绘地理信息, 2020: 1-7, DOI: 10.14188/j.2095-6045.2020148.
- [13] 潘杰. 智能交通系统中最优路径选择的设计与实现[D]. 大连:大连理工大学, 2013.
- [14] 程向红,祁艺. 基于栅格法的室内指示路径规划算法[J]. 中国惯性技术学报, 2018, 26(2):102-106,133.
- [15] 李宝磊,吕丹桔,张钦虎,等. 基于多元优化算法的路径规划[J]. 电子学报, 2016, 44(9):2242-2247.

作者简介

瞿新豪,本科,主要研究方向为无线通信、无线定位技术。
E-mail:LorisXinhaoQu@126.com

丁云飞,本科,主要研究方向为信息的获取与处理,电子设备与信息系统的的设计、开发、应用和集成。

E-mail:dingyunfei0908@163.com

谢亚琴,工学博士,副教授,主要研究方向为无线资源管理、定位技术和路径规划算法。

E-mail:xyq@nuist.edu