

基于理论最短距离变权重 A* 算法的路径规划

刘源¹, 王海泉²

(1. 郑州航空工业管理学院 机电工程学院, 郑州 450015; 2. 中原工学院 电子信息学院, 郑州 450015)

摘要: 在栅格化的障碍物地图中, 将简单高效的 A* 算法引入解决路径规划问题; 为了提高路径规划效率, 减少搜索节点数量, 提出了一种在规定的椭圆区域内, 基于理论最短距离动态改变 A* 算法中估价函数权重的最短路径算法; 该算法将搜索范围限定在规定的椭圆区域内, 椭圆以起点和终点为焦点, 利用统计分析与路径中障碍物尺寸相结合的方法计算长轴参数; 将各节点实际代价权重赋予动态变化的权值, 以实际代价与起点 O 到终点 D 的直线距离的比值为该点权重, 且规定了上下限以保证搜索精度; 同时, 对节点估计代价赋予惩罚函数, 远离理论最短路径距离的节点将获得较大的惩罚值, 使最终路径靠近理论最短路径; 通过仿真实验证明, 该算法在保证搜索精度的前提下, 大大提高了搜索效率。

关键词: 理论最短距离; 路径规划; 椭圆搜索区域; 惩罚函数; 栅格法

Path Planning Based on Theoretical Minimum Distance of A* Algorithm

Liu Yuan¹, Wang Haiquan²

(1. School of Mechanotronics Engineering, Zhengzhou University of Aeronautical Industry, Zhengzhou 450015, China)

(2. School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450015, China)

Abstract: The paper introduces a simple and efficient A* algorithm into the path planning problem in a grid map. In order to improve the efficiency of path planning, this paper proposes a shortest path algorithm based on theoretical minimum distance, characterized by dynamically changing valuation function weights of A star algorithm with limited ellipse area. The start point and end point are two focal points of ellipse area, and the long axis of ellipse is calculated by statistical analysis and obstacle size between start point and end point. Actual cost of each node is given weight which dynamic change between upper and lower limits. The penalty cost is given to the penalty function which the farther away from the line, the greater the penalty. The final path is close to the shortest path in theory through penalty factor acting on estimated cost. The penalty factor is proportional to the shortest distance to the theory. The simulation results show that the shortest path algorithm can improve search efficiency greatly and ensure search accuracy.

Keywords: theoretical minimum distance; path planning; ellipse search area; penalty function; grid method

0 引言

路径规划是一个带约束的复杂优化问题, 是在具有障碍物的环境内按照一定的评价准则寻找一条从起点到终点的无碰撞路径, 最短通过路径是选择最优路径的重要准则^[1]。关于最短路径算法常用的有蜂群算法^[1]、蚁群算法^[2]、遗传算法^[3]、粒子群算法^[4]等启发式算法。这类基于种群的生物启发算法, 存在早熟收敛的缺陷, 容易陷入局部最优, 并且算法的函数较为复杂, 若初始种群较大, 则迭代次数增加, 收敛时间较长^[5]。A* 算法是一种带有启发函数的最短路径算法, 因其算法简单, 搜索效率高而广泛应用于城市道路路径规划^[6]。A* 算法以节点作为搜索对象, 只能在格子环境中应用, 如果将实际障碍物环境栅格化处理, 用节点坐标表示道路和障碍物信息, 就可以利用 A* 算法来规划最短路径。但是与城市道路不同, 栅格化处理的道路信息, 节点是等间距两两相连的, 因此, 其搜索最坏的时间复杂度为 $O(n^2)$, 当节点数 n 增多时, 时间复杂度也将大大增加。

为了提高 A* 算法的搜索效率, 优化途径大致分为 3 种, 一种是通过限制搜索区域减少搜索节点数量, 文献 [6] 根据起点到终点的欧式距离, 在两类不同大小的椭圆中寻找最短路径, 当欧式距离较远时, 可降低 33%~47% 的复杂度。但是若欧式距离较短, 则仅限制搜索区域对搜索效率提高并不显著。

第二种是通过改进启发函数减少算法遍历的节点数, 提高搜索效率。文献 [7] 以当前搜索节点到目的节点的距离作为估计权值, 通过改变启发函数权重来控制搜索效率和精度, 获得最优路径。文献 [8] 用威胁区域半径与航迹点到威胁区域的距离为权值, 改变估价函数权重规划出合理的飞行轨迹。上述通过改变启发函数权重, 可提高算法搜索效率, 但是得到的规划路径可能不是最短路径。

第三种是通过优化 open 表排序过程, 利用二叉堆存储 open 表中节点, 提高节点插入和删除效率, 缩短搜索时间^[9]。在栅格地图中, 由于节点数量较大, 绝大部分搜索时间用以考察节点, 优化 open 表排序对搜索效率的提高影响不大。

因此, 本文将实际障碍物地图栅格化后, 提出了在规定的搜索区域内基于理论最短路径的变权重 A* 算法。算法不仅以椭圆区域限定了搜索范围, 而且动态改变启发函数权重控制搜索效率和精度, 通过对估计代价施加惩罚函数, 优先选择靠近理论最短路径的节点, 保证了规划路径为最短路径, 同时也可将搜索范围控制在理论最短路径周围较小范围内。通过实例证明, 该算法在保证搜索精度的前提下, 明显提高了搜索效率, 且规划路径最接近理论最短路径。

收稿日期: 2017-07-05; 修回日期: 2017-07-29。

基金项目: 河南省科技攻关项目(172102210076)。

作者简介: 刘源(1985-), 女, 湖南怀化人, 硕士, 实验师, 主要从事车辆智能化和信息融合方向的研究。

王海泉(1981-), 男, 河南郑州人, 博士, 副教授, 主要从事智能算法及非线性控制方向的研究。

1 障碍物环境及搜索区域规划

1.1 障碍物环境栅格化处理

A* 算法要求搜索环境为网格, 因此利用栅格法对于实际障碍物环境建模, 将已知的实际障碍物环境通过栅格化处理建立二维栅格图^[10-11], 如图 1 所示, 在栅格图中路径行进规则为可按直线、对角线前进。将障碍物栅格设置为黑色, 自由栅格设置为白色, 每个栅格的序号值 m 与坐标位置 (x, y) 的关系为:

$$\begin{cases} x = \text{mod}(\frac{m}{n}) \\ y = \text{int}(\frac{m}{n}) \end{cases} \quad (m, c \in \mathbf{N}_+) \quad (1)$$

式中, mod 为取余运算; int 为取整运算; n 为每行栅格数。

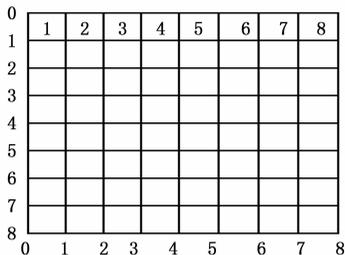


图 1 障碍物环境栅格化

本文要对实际障碍物及其环境进行栅格化处理, 由于实际障碍物的形状可能不规则, 若划分的栅格未被障碍物全部填满时, 该栅格也不能作为自由栅格通行, 因此, 利用正方形栅格表达障碍物边界时, 要将未被障碍物边界填满栅格全部做填满处理, 如图 2 所示。



图 2 障碍物边界栅格填充处理

从图中可知, 栅格尺寸越小, 填充后的障碍物边界越平滑接近实际障碍物边界, 获得的路径也会更加平滑, 同时节点数量也会大大增加, 降低搜索效率, 因此, 为了平衡精度和效率, 本文中栅格边长取值为 0.5。

1.2 搜索区域规划

要提高最短路径规划效率, 可通过限制搜索区域、减少搜索节点数量来实现。实验证明, 以起点 O 和终点 D 为焦点, 最短路径通过的节点 (置信区间 95%) 都位于类似椭圆形的区域内^[6]。因此, 可以用椭圆方程来限制搜索区域, 椭圆标准方程如下:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2)$$

$$c = \frac{\sqrt{(x_O - x_D)^2 + (y_O - y_D)^2}}{2} \quad (3)$$

在椭圆的搜索区域内各节点 i 到 O 、 D 点的距离之和都满足 $L_{iO} + L_{iD} \leq 2a$ 。 $2a$ 为椭圆长轴, 确定了该值就可以规划出确切的椭圆搜索范围。要求解 $2a$ 值, 可采用统计分析方法, 该方法要从栅格地图中随机抽取若干节点, 节点两两构成待求解最短路径的起点和终点, 求出两点间的最短路径记为 P_{ab} ,

两点间的直线距离记为 L_{ab} , P_{ab} 与 L_{ab} 的比值记为 r_{ab} , 选择计算出的 r_{ab} 最大值计算椭圆长轴:

$$2a = L_{OD} \times \max r_{ab} = \max \frac{P_{ab} \sqrt{(x_O - x_D)^2 + (y_O - y_D)^2}}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}} \quad (4)$$

在障碍物尺寸接近的环境中, 障碍物绕行所花费的路程较为接近, 因此通过统计分析得到的 r_{ab} 值彼此较为接近, 具有普遍性, 不会出现某一比值明显大于其他值的情况, 以此作为椭圆长轴计算依据较为可靠。但是, 当环境中有少数障碍物尺寸远大于其他障碍物尺寸, 则绕行该大型障碍物所花费的路程明显增加, 若随机抽取的节点构成的路径没有穿过该大型障碍物, 则由统计得出的 r_{ab} 值与实际绕行大型障碍物的 r_{ab} 相比偏小, 则该统计值 r_{ab} 不再具有典型性, 其规划的椭圆区域可能过小, 路径无法越过大型障碍物, 使得搜索无法继续, 导致寻路失败, 如图 3。

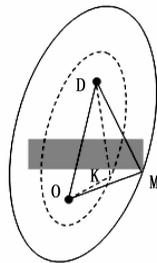


图 3 椭圆搜索区域

为此, 当椭圆边界点 K 到两焦点距离之和 $L_{OK} + L_{KD}$ 小于障碍物边界 M 到两焦点距离之和 $L_{OM} + L_{MD}$ 时, 应以 $L_{OM} + L_{MD}$ 与 L_{OD} 的比值代替 r_{ab} , 即:

$$2a = \begin{cases} \max \frac{P_{ab} \sqrt{(x_O - x_D)^2 + (y_O - y_D)^2}}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}, \\ L_{OK} + L_{KD} \geq L_{OM} + L_{MD} \\ \sqrt{(x_O - x_M)^2 + (y_O - y_M)^2} + \sqrt{(x_D - x_M)^2 + (y_D - y_M)^2}, \\ L_{OK} + L_{KD} < L_{OM} + L_{MD} \end{cases} \quad (5)$$

$$b = \sqrt{a^2 - c^2} \quad (6)$$

2 基于理论最短距离的变权重 A* 算法

2.1 经典 A* 算法估价函数

A* 算法是一种启发算法, 其搜索基本原理是: 对搜索中遇到的每个新节点, 按照启发函数计算代价估值, 将估值最小的节点作为当前节点, 继续搜索。

经典 A* 算法的估价函数为:

$$f(v) = g(v) + h(v) \quad (7)$$

在 A* 算法中, 扫描 open 表中所有节点并按照 $f(v)$ 值排序, 找到 $f(v)$ 值最小的节点作为当前节点。 $g(v)$ 为当前节点到起点的实际代价, $h(v)$ 为当前节点到终点的估计代价, 均采用欧几里德距离。估计代价 $h(v)$ 体现了算法的启发性, 其大小决定了路径搜索的方向, 当越接近终点时, 估计代价值越小, 其在估价函数中所占比例减小, 搜索的方向性变差, 效率降低。为了提高搜索效率, 可适当提高 $h(v)$ 在启发函数中的比重, 但又会造成在搜索初期搜索范围过小而不能找到最优路径。为了平衡搜索效率和搜索精度, 应该实现在搜索初期以扩

大搜索范围, 提高搜索精度为主, 搜索后期以快速接近终点为主的搜索策略。

2.2 变权重估价函数

为了在保证搜索精度的基础上提高搜索效率, 可动态改变实际代价的权重, 令 $g(v)$ 的权重为:

$$W = \begin{cases} H, & w < L \\ w, & L \leq w \leq H \\ L, & w > H \end{cases} \quad (8)$$

$$w = \frac{g(v)}{\sqrt{(x_0 - x_D)^2 + (y_0 - y_D)^2}} \quad (9)$$

H 为 $g(v)$ 权重的上限值, L 为 $g(v)$ 权重的下限值, (x_0, y_0) 为起点坐标, (x_D, y_D) 为终点坐标。

该计算方法以实际代价 $g(v)$ 与起点到终点直线距离 L_{OD} 的比值作为实际代价的权重, 在搜索前期该权重值过小, 搜索范围也小, 搜索精度低, 因此规定了权重的下限值 L , 保证搜索精度, 越接近终点该权重越大, 搜索精度高, 但搜索范围过大而效率降低, 因而规定了权重的上限值 H , 保证搜索效率。

两点间的直线距离为理论最短距离, 在当前节点 (x_i, y_i) 、起点 (x_0, y_0) 和终点 (x_D, y_D) 构成的三角形中, 节点到起点 O 和终点 D 的距离之和 $L_{Ox} + L_{Dy}$ 越接近起点到终点直线距离 L_{OD} , 则该节点位于最短路径上的可能性越大。为了更快搜索到直线距离附近节点, 对估计代价 $h(v)$ 设计惩罚函数 W' , 该值与节点到 L_{OD} 的距离成正比, 远离 L_{OD} 的点 W' 值大, 接近 L_{OD} 的点 W' 值小。 W' 的计算公式为:

$$W' = \frac{g(v_i) + \sqrt{(x_i - x_D)^2 + (y_i - y_D)^2}}{\sqrt{(x_0 - x_D)^2 + (y_0 - y_D)^2}} \quad (10)$$

为此, 本文提出了变权重估价函数:

$$f(v) = Wg(v) + W'h(v) \quad (11)$$

在搜索过程中, 靠近 L_{OD} 直线的节点排序靠前被优先选择, 因而算法的主要搜索区域自动控制理论最短路径 L_{OD} 直线附近。

2.3 改进 A* 算法实现步骤

本文提出的改进 A* 算法, 在经典算法基础上, 限制了节点搜索范围, 并实时改变估价函数权重, 在保证搜索精度的基础上, 加快了搜索速度。

- 1) 获得起点 O 和终点 D 坐标, 并计算两点间的直线距离 L_{OD} ;
- 2) 计算椭圆参数 a, b , 确定搜索区域;
- 3) 生成 open 列表和 closed 列表;
- 4) 判断节点是否在搜索区域内, 计算节点权重 W 和惩罚函数 W' 值, 找到 open 列表中估价值 f 最小的节点作为当前节点, 如果该点为终点, 则搜索结束, 返回搜索路径, 如果不是终点, 则向下执行;
- 5) 将当前节点移出 open 列表, 移入 closed 列表, 将当前节点相关节点加入 open 列表。
- 6) 循环 4) ~ 5) 步骤, 直到当前节点为终点, 结束搜索, 返回路径。

3 路径规划仿真和结果分析

本文利用 Matlab 进行了仿真实验, 验证了在多种障碍物的情况下, 本文提出算法的有效性。

实验中, 栅格地图尺寸为 100×100 , $g(v)$ 权重的上限值 H

$= 0.8$, 下限值 $L = 0.5$, 起点坐标 $(98, 58)$, 终点坐标 $(2, 40)$ 。以搜索节点数, 搜索时间、路径长度为主要对比参数。

第一类为障碍物尺寸较为均匀的障碍物环境, 在图中任取 20 对点计算 $r_{ab} = 1.45$; 第二类为障碍物尺寸相差较大的障碍物环境, 其搜索范围应包含图中最大障碍物, 计算得到 $r_{ab} = 1.71$ 。实验结果如图 4~11 所示。

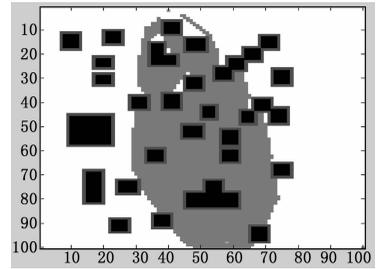


图 4 障碍物 1 下的经典 A* 算法搜索节点范围

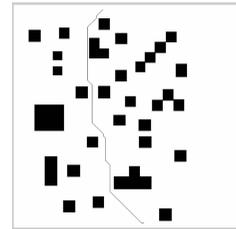


图 5 障碍物 1 下的经典 A* 算法搜索结果

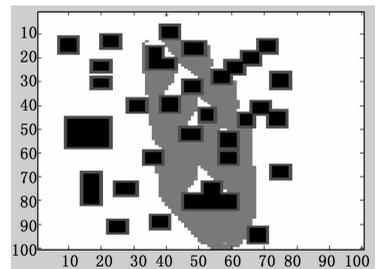


图 6 障碍物 1 下的改进 A* 算法搜索节点范围

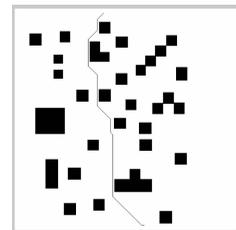


图 7 障碍物 1 下的改进 A* 算法搜索结果

表 1 障碍物 1 下的搜索参数

算法	搜索节点数	搜索时间/ms	路径长度/m
经典 A* 算法	2 243	11 128	549.2 031
改进 A* 算法	1 457	7 025	549.2 031

表 2 障碍物 2 下的搜索参数

算法	搜索节点数	搜索时间/ms	路径长度/m
经典 A* 算法	2 320	3 891	596.9 848
改进 A* 算法	653	1 078	596.9 848

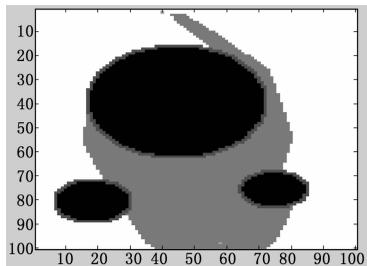


图 8 障碍物 2 下的经典 A* 算法搜索节点范围

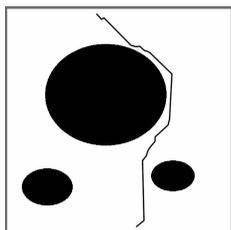


图 9 障碍物 2 下的经典 A* 算法搜索结果

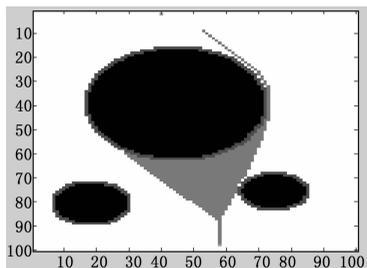


图 10 障碍物 2 下的改进 A* 算法搜索节点范围

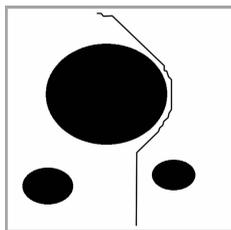


图 11 障碍物 2 下的改进 A* 算法搜索结果

表 1、表 2 对比了经典 A* 算法和改进 A* 算法在两种不同障碍物的地图中搜索结果。计算可知，在障碍物 1 的地图中，经典 A* 算法搜索节点数和搜索时间均为改进 A* 算法的 1.5 倍，并且改进后的 A* 算法搜索范围较小，选择搜索的节点更加靠近理论最短路径 L_{OD} 。对比两种算法得到的路径长度，可知，改进后的 A* 算法规划的路径依然是全局最短路径，但搜索效率大大提高。

在障碍物 2 的地图中，经典 A* 算法搜索节点数和搜索时间均为改进 A* 算法的 3.6 倍，该地图中存在尺寸较大的障碍物，经典 A* 算法在寻找越过障碍物路径阶段搜索范围很大，花费了主要的搜索时间，而改进后的 A* 算法，规划了一个包含最大障碍物在内的搜索椭圆，并且在惩罚函数的作用下，优先选择了靠近理论最短路径 L_{OD} 的节点，进一步降低了搜索的复杂程度。两种算法得到的路径长度一致。这说明改进后的 A* 算法缩小了搜索的区域，减少了搜索节点数，仍可以获得最

短路径搜索结果。

需说明的是，在上述仿真结果中，虽然两种算法规划的最短路径长度一致，但从图中实际路线来看，其选择的节点和路径是不完全一致的。导致这种结果的原因是，本算例中节点间距是均匀的，且限制对角线路径，所以相同长度的路径可能不止一条。

因此，综合考虑搜索精度和搜索效率，改进 A* 算法在保证较高搜索精度的前提下大大提高了搜索效率。

4 结论

本文将实际障碍物环境栅格化处理后，引入了改进 A* 算法规划最短路径。本文提出的算法在规定的椭圆区域内，基于理论最短路径动态改变 A* 算法中估价函数权重，在提高搜索效率的同时，保证了规划路径为全局最短路径。通过仿真实验证明，该算法缩小了节点搜索范围，通过对各节点 $g(v)$ 赋予不同权重平衡搜索精度和搜索效率，实现了前期以搜索精度为主，后期以搜索效率为主的搜索策略，并且通过对 $h(v)$ 施加惩罚函数，使实际规划路径更加靠近理论最短路径。与经典 A* 算法对比，本文提出的算法在保证搜索精度的前提下，大大提高了搜索效率。

参考文献：

- [1] 王海泉, 胡瀛月, 廖伍代, 等. 基于改进人工蜂群算法的机器人路径规划 [J]. 控制工程, 2016, 23 (9): 1407 - 1411.
- [2] 康 冰, 王曦辉, 刘 富. 基于改进蚁群算法的搜索机器人路径规划 [J]. 吉林大学学报 (工学版), 2014, 44 (7): 1062 - 1068.
- [3] Ahmed F, Deb K. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithm [J]. Soft Computing, 2013, 17 (7): 1283 - 1299.
- [4] 刘贵杰, 刘 鹏, 穆为磊, 等. 采用能耗最优改进蚁群算法的自治水下机器人路径优化 [J]. 西安交通大学学报, 2016, 50 (10): 93 - 98.
- [5] Andre J, Siarry P, Dognon T. An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization [J]. Advance in Engineering Software, 2011, 32 (1): 49 - 60.
- [6] 王世明, 邢建平, 张玉婷, 等. 典型城市路网中的椭圆最短路径算法 [J]. 系统工程理论与实践, 2011, 31 (6): 1158 - 1164.
- [7] Qi M J, Sun H N, Gao G F. Research on an improved algorithm for shortest path searching in urban traffic based on GIS [A]. Proceedings of Electrical and Control Engineering International Conference [C], 2011: 1184 - 1187.
- [8] 卢 青, 周 军, 呼卫军. 基于改进 A* 算法的滑翔飞行器轨迹规划 [J]. 系统工程与电子技术, 2016, 38 (12): 2758 - 2762.
- [9] 潘长安. 基于改进 A 星算法的城市交通寻径的研究 [D]. 福建: 华侨大学, 2012.
- [10] Klemm S, Oberlander J, Hermann A, et al. RRT* - Connect: Faster, asymptotically optimal motion planning [A]. IEEE Conference on Robotics and Biomimetics [C]. IEEE, 2015: 1670 - 1677.
- [11] Orlin J B, Madduri K, Subramani K, et al. A faster algorithm for the single source shortest path problem with few distinct positive lengths [J]. Journal of Discrete Algorithms, 2010, 8 (2): 189 - 198.