DOI:10.15923/j.cnki.cn22-1382/t.2019.2.07

改进 A 星算法移动机器人路径规划

宋 宇, 王志明*

(长春工业大学 计算机科学与工程学院, 吉林 长春 130012)

摘 要: 采用粒子群算法搜索每个栅格内的最优点坐标,将传统 A 星算法的搜索角度任意化。仿真表明,改进 A 星算法得到的路径缩短了 4%,平均转折角度减小了 54%。

关键词: 路径规划; A 星算法; 最短路

中图分类号: TP 301.6 **文献标志码**: A **文章编号**: 1674-1374(2019)02-0138-04

Path planning simulation based on improved A star algorithm

SONG Yu, WANG Zhiming*

(School of Computer Science & Engineering, Changchun University of Technology, Changchun 130012, China)

Abstract: Particle swarm optimization is used to search the best coordinates in each grid. The search angle based on traditional A star algorithm is unrestricted. Simulation indicates that the path length obtained by the improved algorithm is reduced by 4% and the average turning angle is reduced by 54%.

Key words: path planning; A star algorithm; shortest path.

0 引 言

路径规划问题在很多领域都具有广泛的应用,如机器人的自主无碰撞运行、无人机的避障突防飞行等。现有的路径规划方法包括蚁群算法、A 星算法、D 星算法、人工势场算法、模糊逻辑算法、神经网络算法、强化学习算法等。其中,A 星算法作为一种经典路径规划算法,被广泛应用于路径规划问题中,但由于在栅格环境下 A 星算法只选择周围栅格的中心点加入 open 表中,导致路径点的选择局限于栅格中心点,路径转折角度固定为一些特定的离散值,从而产生了无效冗余转折或找到的最终路径不是最优。文献[1]提出了扩展 Moore 型邻居结构 A 星算法;文献[2]提出了结合跳点算法的 A 星算法;文献[3]采用了二叉堆优化查找 open 表中 F 值的方法,加快了 A 星算法的执行速度;文献[4]提出了去除冗余点的 A 星算法;文献[5]采用 A 星算法初始化信息素,用蚁群算法进行了机器人路径规划;文献[6]用蚁群算法对 Hopfield 神经网络的参数进行了优化;文献[7-8]采用了优化算法进行路径规划。

文中通过建立映射矩阵的方式提出了一种任

收稿日期: 2018-12-21

基金项目: 吉林省青年科研基金资助项目(20160520020JH)

作者简介: 宋 宇(1969—),男,汉族,黑龙江呼兰人,长春工业大学教授,硕士,主要从事嵌入式系统及应用方向研究,E-mail:songyu@ccut.edu.cn. * 通讯作者:王志明(1991—),男,汉族,山西神池人,长春工业大学硕士研究生,主要从事路径规划方向研究,E-mail:120354157@qq.com.

意角度的 A 星算法，仿真实验表明，在栅格划分粗糙的情况下，改进算法效果显著。

1　相关算法

1.1　A 星算法

A 星算法是在 Dijstar 算法的基础上引入了启发函数，加快了算法收敛速度。A 星算法的估价函数为

$$f(x,y)=g(x,y)+h(x,y) \tag{1}$$

其中，$g(x,y)$ 为起始点到当前考察点 $X(x,y)$ 的实际距离值，$h(x,y)$ 为 $X(x,y)$ 点到目标点的估计距离值，距离值函数 $h(x,y)$ 一般取欧几里德距离（如式(2)）或绝对值距离（如式(3)），g_x、g_y 为目标点的 x 坐标与 y 坐标。

$$h(x,y)=\sqrt{(x-g_x)^2+(y-g_y)^2} \tag{2}$$

$$h(x,y)=|x-g_x|+|y-g_y| \tag{3}$$

A 星算法不断将当前位置周围 8 个栅格的中心点加入 open 表，并不断取 open 列表中 F 值最小的点作为当前位置，直到当前位置的临近点出现目标点或 open 表为空为止，A 星算法的流程如下：

1）初始化 open 表、opencost 表、close 表为空，将起点坐标加入 open 表，将起点的 cost 值 0 加入 opencost 表。

2）将 open 表中具有最小 F 值的点的坐标作为当前点，同时从 open 表与 opencost 表中删除这个点的信息，并把当前点的坐标加入 close 表。

3）依次计算当前点的邻居栅格中心的 8 个点的 g 值与 f 值，若这个点在 close 表中，则跳过这个点；若这个点既不在 open 表中，也不在 close 表中，则将这个点的坐标加入 open 表，同时将这个点的 g 值加入 opencost 表，并将这个点的箭头指向当前点。若这个点已经在 open 表中，则比较这个点之前的 g 值与通过当前点计算得到的 g 值的大小，若前者大，则修改这个点的 g 值为当前计算得到的 g 值，并将这个点的箭头方向指向当前点。

4）跳到 2），直到 open 表为空，或者 open 表中出现目标点为止。

1.2　粒子群算法

粒子群算法是一种模拟大自然群体寻优的优化算法，通过共享群体的适应度值大小来加速极值寻优过程。设 D 维粒子位置为 $x_i=(x_{i1},x_{i2},\cdots,x_{iD})$，粒子速度为 $v_i=(v_{i1},v_{i2},\cdots,v_{iD})$，种群中所有个体粒子目前为止适应度最优时的位置为 $p=(p_1,p_2,\cdots,p_D)$，到第 k 代为止种群中所有粒子的全局最优适应度值对应的粒子位置为 pb^k。则在每次迭代中粒子的速度更新公式为

$$V_{iD}^{k+1}=V_{iD}^{k}+c_1r_1(p_{iD}^{k}-x_{iD}^{k})+c_2r_2(pb^{k}-x_{iD}^{k}) \tag{4}$$

位置更新公式为

$$X_{iD}^{k+1}=X_{iD}^{k}+V_{iD}^{k+1} \tag{5}$$

通过不断迭代，最终算法保存了最优位置，即最优解。

2　改进 A 星算法

2.1　粒子群搜索

使用粒子群算法搜索当前栅格的邻居栅格内的最优粒子位置，如图 1 所示。

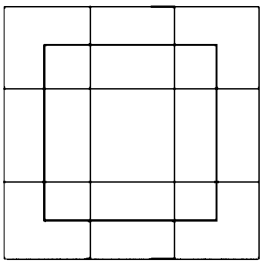

图 1　搜索坐标边界

最中间的格子代表当前点所在的格子，首先在当前点所在格子的 8 个邻居格子内的线段上搜索到 8 个最优点，目标函数为 F 值（当前栅格的 g 值与当前栅格内的实际点到搜索栅格内搜索点的距离与搜索点到目标点的距离的和），即搜索每个格子内直线上的 F 值最小的点，使用粒子群算法搜索时，粒子位置搜索范围限制在每个格子内的线段上。例如，粒子群算法在每个栅格内的给定范围搜索后得到的点如图 2 所示（黑色实心点）。

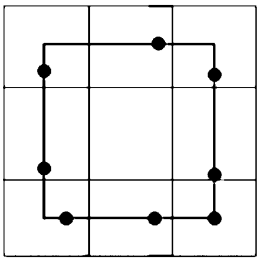

图 2　搜索得到的点

将搜索到的 8 个最优位置存入 $\boldsymbol{M}$ 矩阵中。同时将搜索到的点的 g 值与 f 值存入 cost 与 F 数组中。

2.2　邻居格子属于 open 表

如果邻居栅格序号已经在 open 表中，目标函数为从当前栅格的 $\boldsymbol{M}$ 矩阵（目前所有栅格内的点

的最优位置组成的矩阵)存储的当前栅格内的最优点到搜索栅格内点的距离与当前点的 g 值的和,上下左右栅格的搜索范围为栅格内黑色线段,如图3所示。

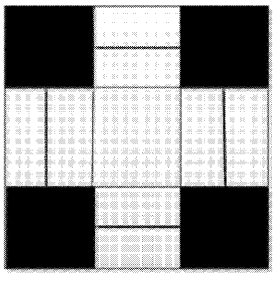

图3 搜索坐标边界

4个对角栅格的搜索范围为整个栅格之内(即若邻居栅格序号不在open表中,粒子群算法的搜索范围见图1,若栅格序号在open表中,粒子群算法的搜索范围见图3)。

比较被搜索栅格之前保存的 g 值与此次搜索到的点的目标函数值,若前者大,则将此邻居栅格的箭头指向当前栅格,并更新邻居栅格的 g 值与邻居栅格内 $\boldsymbol{M}$ 矩阵里存储的坐标值。

2.3 设置安全距离

为了避免规划出来的路径穿过障碍物,设置粒子的坐标距离最近障碍物中心坐标的阈值为 d,若粒子(搜索过程中点的坐标)的坐标与所有障碍物中距离最近障碍物的中心点的坐标距离小于 d,则将粒子群算法的适应度函数值额外增加100/distance,distance为机器人到障碍物中心的实际距离。若粒子的坐标距离最近的障碍物距离大于阈值 d,则不增加。

2.4 改进算法流程

1)以起点为当前点,以2.1所述方法搜索起点的邻居栅格内的8个最优位置,以及这8个位置的 g 值(即从起点到这8个位置的距离)与 f 值。将起点序号加入close表,将8个邻居栅格的序号加入open表。将搜索到的8个栅格内对应的8个最优位置存入矩阵 $\boldsymbol{M}$ 中。将搜索得到的8个 g 值与 f 值存入cost数组与 F 数组中。

2)从 F 数组中选择 f 值最小的栅格为当前栅格,同时将此栅格序号从open表移入close表。

3)以当前栅格为中心,依次考察当前栅格的8个邻居栅格。若邻居栅格序号在close表中,则跳过此栅格;若邻居栅格在open表中,则以2.2所述方法更新邻居栅格的 g 值和 f 值,以及最优点位置;若邻居栅格既不在open表中,也不在close表中,则以2.1所述方法更新 g、f 值与最优位置,同时将该栅格序号加入open表。

4)若终点不在open表中,跳到2),若终点在open表中,则算法结束。

3 仿 真

文中将粒子群算法迭代次数设置为10次,种群个数设置为10个。粒子群算法的速度最大、最小值不设限,粒子群搜索算法中的最大速度设置值为0.03,最小速度值为−0.03,阈值距离 d 设置为1.6。粒子群算法全局最优项前面的系数为2,局部最优项前面的系数为1.5。在Matlab2014a环境下进行了仿真。文中将平均转折角度定义为总的转折角度(每次转折时角度改变量的和)除以转折点的个数。由传统A星算法得到的路径和改进A星算法得到的路径分别如图4和图5所示。

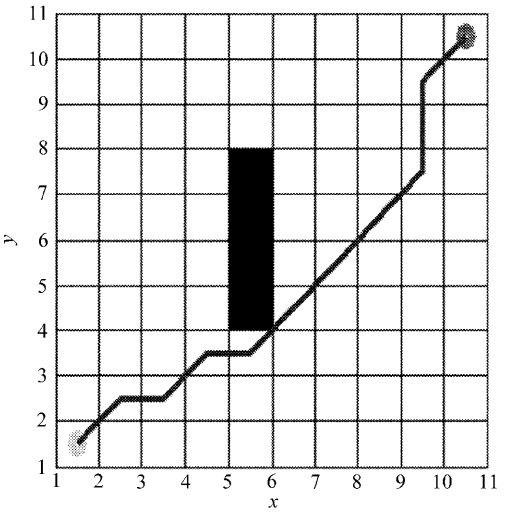

图4 A星算法路径

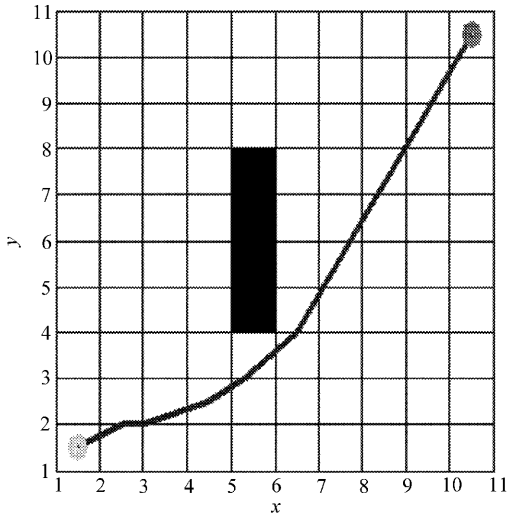

图5 改进算法路径

由图 4 和图 5 可知，改进算法找到的最优路径点较传统 A 星算法找到的路径点准确。

算法长度与转折角对比见表 1。

表 1　算法长度与转折角对比

算法	路径长度	平均转折角度
A 星算法	13.899 5	45.000
改进算法	13.321 2	2.019

障碍物较多情况下传统 A 星算法规划得到的路径和障碍物较多情况下改进算法规划得到的路径分别如图 6 和图 7 所示。

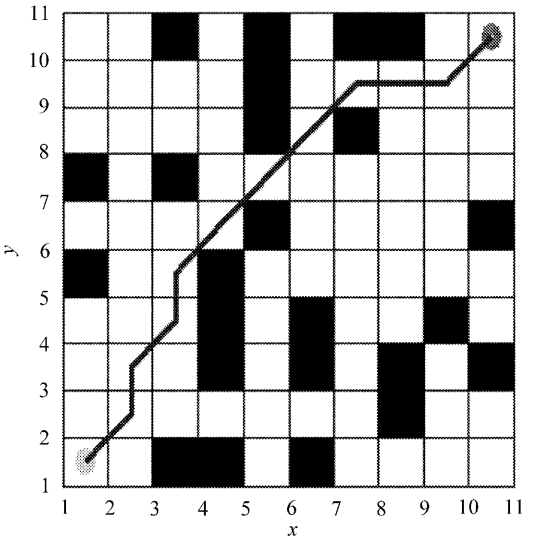

图 6　障碍物较多情况下 A 星算法路径

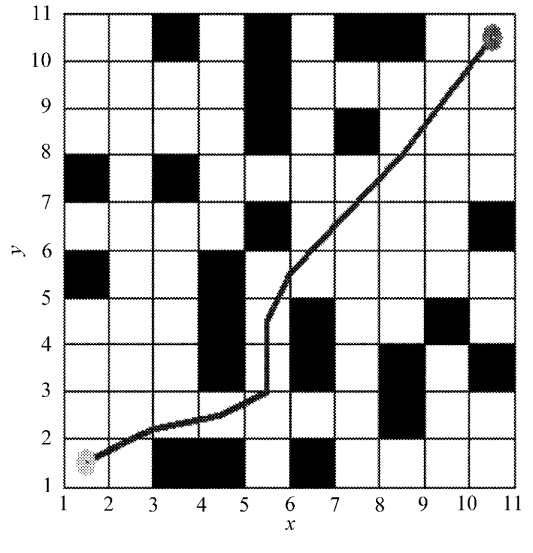

图 7　障碍物较多情况下改进算法路径

障碍物较多情况下算法长度与转折角对比见表 2。

表 2　障碍物较多情况下算法长度与转折角对比

算法	路径长度	平均转折角度
A 星算法	13.899 5	45.000 0
改进算法	13.657 8	20.758 7

由表 1 和表 2 可以看出，由文中算法得出的路径长度较短且路径较平滑，更符合实际机器人的运动规律。另一方面，从图 6 和图 7 也可以看出，由于文中算法加入了对安全距离的考虑，得出的路径尽量远离障碍物，而不是紧贴障碍物的尖点，故文中算法得出的路径较安全。

4　结　语

通过设置栅格内搜索到的点与栅格中心点的对应关系以及搜索方式得到了栅格内的最优点，而不是直接选择中心点作为备选点，通过设置安全距离使得生成的路径不经过障碍物。仿真结果表明，改进算法得出的路径长度较短且路径较平滑。

参考文献：

[1] 史久根，刘春霞，席海强.CA 模型下的改进 D＊路径规划算法[J].电子测量与仪器学报，2016，30(1)：30-37.

[2] 赵晓，王铮，黄程侃，等.基于改进 A＊算法的移动机器人路径规划[J].机器人，2018，40(6)：137-144.

[3] 姚雨，李庆，陈曦.优化的 A＊算法在航迹规划上的应用[J].微电子学与计算机，2017，34(7)：51-55.

[4] 王殿君.基于改进 A＊算法的室内移动机器人路径规划[J].清华大学学报：自然科学版，2012，52(8)：1085-1089.

[5] 蔡旻，刘士凡，陶重犇，等.基于改进 A＊算法的分拣搬运机器人路径规划[J].计算机测量与控制，2018，26(4)：164-166，170.

[6] 金飞虎，郭琦.基于蚁群算法的 Hopfield 神经网络在多空间站路径规划的应用研究[J].计算机应用研究，2010，27(1)：51-53.

[7] 魏勇，赵开新，王东署.改进粒子群算法在移动机器人路径规划中的应用[J].火力与指挥控制，2018，43(2)：41-43.

[8] 宋宇，王志明.基于 WDO 的无人机全局路径规划方法[J].长春工业大学学报，2017，38(6)：555-559.