

基于融合蚁群-A*算法的多目标路径规划方法

李永真[†], 黄学功, 张志安

(南京理工大学 机械工程学院, 江苏 南京 210094)

摘 要:针对传统蚁群算法在二维栅格地图下存在搜索时间长、收敛速度慢、考虑因素单一等问题,提出了一种融合蚁群-A*算法。首先将启发式方法的思想融入到蚁群算法,优化蚁群算法的搜索效率;其次引入最大最小蚂蚁系统,提出一种精英蚂蚁信息素更新规则;同时增加考虑转向次数、转向角度等因素,在启发式信息中加入弯曲抑制算子,减少弯曲次数和累积弯曲角度,避免算法以优化路径长度作为单一目标;最后提出一种改进撤回机制,解决算法死锁问题。仿真表明,在相同地图环境中,改进的蚁群算法在路径长度、路径拐点以及收敛速度都有了显著提升,更适用于复杂环境。

关键词:蚁群算法; A* 算法; 最大最小蚂蚁系统; 路径规划

中图分类号: TP242

文献标识码: A

Multi-target Path Planning Method Based on Fused Ant Colony- A* Algorithm

LI Yongzhen[†], HUANG Xuegong, ZHANG Zhian

(School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China)

Abstract: Traditional ant colony algorithm has problems such as long search time, slow convergence speed, and single consideration factor in two-dimensional grid maps. A fusion algorithm of ant colony and A* was proposed to address the problems. First, the idea of heuristic methods was integrated into ant colony algorithm to optimize its search efficiency. Second, the max-min ant system was introduced to propose an elite ant pheromone updating rule. Meanwhile, the factors such as number of turns and turning angle were considered to increase a bending suppression operator in the heuristic information to reduce the number of bends and cumulative bending angles, avoiding the algorithm that optimize path length as a single objective. Finally, an improved recall mechanism was propose to address algorithm deadlock issues. Experiments show that in the same map environment, the improved ant colony algorithm significantly reduces path length, path inflection points, and accelerates convergence speed, making it more suitable for complex environments.

Key words: ant colony algorithm; A* algorithm; max-min ant system; path planning

移动机器人的路径规划技术是在环境地图已知的情况下,规划出一条不发生碰撞的路径。传统的路径规划算法有 Dijkstra^[1]、A*^[2]、RRT^[3]、蚁群算法^[4]、遗传算法^[5]等。Dijkstra 算法遍历节点

过多,时间复杂度大。RRT 算法规划好的路径不一定是最优路径。相对来说蚁群算法具有较好的求解能力,易与多种算法结合起来提高性能等优点,但同时存在收敛速度慢、冗余节点多等问题。针对

算法存在的缺陷,大量学者通过研究不断地对算法进行改进,提高了算法的性能与适用性。封声飞等^[6]差异化分配初始的信息素,并在状态转移函数中考虑路径转折因素,提高了路径平稳性,但求解质量不高;王星宇等^[7]设计一种自变性挥发因子,增强算法搜率能力,但在较复杂环境下存在路径拐点多、平稳性差等问题;马军等^[8]将 A* 算法的搜索思想融入蚁群算法中,提高算法的指向性,降低算法停滞的风险,但存在考虑因素过于单一的问题;鲁飞等^[9]调整启发函数的权重因子,完善信息素挥发系数;赵天亮等^[10]引入 A* 算法的估价函数,提出新的路径信息素更新方式。李二超等^[11]提取障碍物的特征点进行预处理,基于特征点采用信息素不均分布和双向平行路径搜索提高求解构建效率,利用不同复杂度和比例的网格图,验证了算法的优越性。李晨阳等^[12]提出一种基于高斯分布信息素挥发机制的蚁群算法,降低了搜索导航过程中与障碍物的碰撞概率,提高了移动机器人的搜索效率和避让能力。

已有的文献研究在一定程度上提高了算法的效率和平稳性,但仍存在无用拐点过多、路径质量差、收敛慢等问题,且未考虑蚁群的死锁问题。本文基于以上文献对蚁群算法的改进研究,针对上述问题做了以下改进:首先将 A* 算法借助启发函数引导的思想结合到传统蚁群算法中,提高算法的收敛速度;其次添加弯曲抑制算子,减少弯曲次数和减小累积转弯角度;同时引入最大最小蚂蚁系统,设置信息素限制区间,避免算法早熟;最后提出撤回机制,解决蚂蚁在复杂环境中陷入死锁状态的问题。

1 基本蚁群算法

1.1 基本思想

蚁群算法是受蚂蚁觅食启发的一种仿生学算法。建立蚂蚁释放的信息素量与路径长度的关系规则,让信息素在较短路径上的累积浓度与时间成正比,吸引更多蚂蚁选择该路径,从而实现最优寻食路径的发现。

1.2 状态转移规则

在传统的蚁群算法中,蚂蚁选择下一个节点的概率是依据状态转移概率公式的,在 t 时刻,某只蚂蚁 k 从栅格点 i 爬行到栅格点 j 的状态转移概率 $P_{ij}^k(t)$ 为:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allow}_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta} & s \in \text{allow}_k \\ 0 & s \notin \text{allow}_k \end{cases} \quad (1)$$

$$n_{ij} = \frac{1}{d_{ij}} \quad (2)$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

式中, τ_{ij} 为路径 i 到 j 的信息素浓度值, η_{ij} 为路径 i 到 j 的启发式信息。 α, β 分别为信息素和启发式信息对状态转移概率的影响力因子。 d_{ij} 表示节点 i 和 j 的距离。 (x_i, y_i) 和 (x_j, y_j) 为 i, j 的坐标。 allow_k 表示节点下阶段可选路径的集合。

蚁群算法通过轮盘赌轮法确定蚂蚁路径方向,不断重复,直到到达目标点。式(4)为每个个体的累积概率 q_j 。

$$q_j = \sum_{j=1}^l P_{ij}^k \quad (4)$$

q_j 相当于转盘上的跨度,跨度越大的区域越容易选到, l 代表下一步可选路径的数量之后通过 rand 函数随机生成一个 $(0, 1)$ 的小数 r , 比较 q_j 与 r 的大小,选出大于 r 的最小的 q_j , 对应的就是第 k 只蚂蚁在第 i 条路径时下一步要选择的目标点。

1.3 信息素更新规则

传统算法在每次迭代完成后,依据路径规划的长度对全局信息素进行更新。式(5)为每个周期结束时对每条路径上的信息素浓度的更新公式。

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \end{cases} \quad (5)$$

式中, n 为蚂蚁数量, ρ 是信息素的蒸发速率。 $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在节点 i 与 j 之间释放的信息素,定义如下:

$$\Delta\tau_{ij}^k(k) = \begin{cases} \frac{Q}{L_k}, & \text{第 } k \text{ 只蚂蚁经过}(i, j) \\ 0, & \text{其他} \end{cases} \quad (6)$$

式(6)为蚁周模型, Q 是一常数,为迭代一次过程中的信息素总量。 L_k 为第 k 只蚂蚁所走路径的长度。

2 改进蚁群算法

传统蚁群算法邻接栅格之间的启发式值差异性不足,初始信息素相同,存在搜索时间长、路径不平滑等问题。当地图模型更复杂时,可能会陷入死锁状态,且在栅格地图中,传统的路径规划可能有

更多的弯曲次数和较大的累积弯曲角度。针对以上问题,本文对启发函数、信息素、撤回机制等方面做了以下改进。

2.1 改进启发函数

A* 算法是在 Dijkstra 算法的基础上发展起来的,避免了盲目搜索,具有较高的搜索效率。A* 算法的启发式代价用估计函数方程 $f(n)$ 如下:

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

本文利用 A* 算法出色的寻优能力和搜索效率,将其启发智能搜索思想融入到蚁群算法的启发式函数中来搜索最优路径,借助预估移动代价引导蚂蚁更快到达目标点;同时在启发式中添加弯曲抑制算子,以减少弯曲次数和减小累积转弯角度。改进的启发式函数如下:

$$n_{ij}(t) = \frac{Q}{g(n) + h(n) + C(n)} \quad (8)$$

$$C(n) = \text{phi1} \times \text{bend} + \text{phi2} \times \text{angle} \quad (9)$$

$$g(n) = \sum_{i=1}^n S_{(i)} \quad (10)$$

$$h(n) = \sqrt{(f_x - n_x)^2 + (f_y - n_y)^2} \quad (11)$$

式中, Q 是一个大于 1 的常数; $C(n)$ 是弯曲抑制因子; bend 是从前一个节点到下一个节点的路径弯折数; angle 为拐角的夹角; phi1 、 phi2 分别是转弯次数和转弯程度的转换系数; $g(n)$ 是节点 n 距起点的代价; $S_{(i)}$ 表示节点 i 的父节点到 $i-1$ 的移动距离; $h(n)$ 是节点 n 距终点的估计代价; (n_x, n_y) 为当前节点的坐标; (f_x, f_y) 为目标节点坐标。

2.2 改进信息素更新方式

2.2.1 初始信息素改进

传统蚁群算法初始信息素浓度各处相同,导致最开始的蚂蚁觅食无方向性,从而收敛速度较慢。本文设计一种新的初始信息素浓度分布方式,指导蚂蚁走最优路径。由于最优路径更可能分布在起点和目标点连线的附近范围,故本文在设置初始信息素浓度时,增加了该范围内的信息素浓度来指导蚂蚁寻优,调整后的初始信息素浓度如下:

$$\tau_{ij}(0) = \frac{Q}{d(j,l)} + c \quad (12)$$

式中, $\tau_{ij}(0)$ 为初始信息素浓度, c 为一常数, $d(j,l)$ 为邻接栅格 j 与目标点和起点连接线 l 的距离。距离连线 l 越近的栅格点,初始信息素浓度越高,开始阶段蚁群越倾向于选择。

2.2.2 信息素更新规则

传统蚁群算法每次迭代是将所有蚂蚁的信息素进行更新,收敛速度较慢,同时传统蚁群算法信息素更新只考虑路径长度的因素,存在考虑因素过于单一等问题。本文综合考虑路径长度、转弯次数和转弯角度等多方面因素,提高路径选择的合理性,减少路径不必要的转折,同时引入最大最小蚂蚁系统更新最优路径信息素思想,提出一种基于多因素判断最优路径方式的精英蚂蚁信息素更新规则。据此本文提出的改进信息素更新规则如式(13)所示。

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{\text{best}} \\ \Delta\tau_{ij}^{\text{best}} = \frac{Q_1}{L_{\text{best}}} + \frac{Q_2}{C_{\text{turn}}^{\text{best}}} \end{cases} \quad (13)$$

$$C_{\text{turn}}^{\text{best}} = x\text{Cal}(l) + y\text{Turn}(l) \quad (14)$$

式中, Q_1 、 Q_2 为大于 1 的常数; L_{best} 表示当前的最优路径; $\text{Cal}(l)$ 表示最优路径上累计转弯角度之和; $\text{Turn}(l)$ 是最优路径上的转弯次数之和; x 、 y 为转弯角度和转弯次数的权重占比系数。

2.2.3 信息素限制

随着信息素的不断更新迭代,可能会导致部分路径上信息素过多、信息素浓度差异过大,引发算法重复构建同一解。为了避免传统蚁群算法因信息素积累陷入局部最优解而失去进一步搜索空间能力的情况,本文设置一个信息素限制区间,将信息素限制在 $[\text{Tau}_{\min}, \text{Tau}_{\max}]$ 。式(15)为信息素的限制公式。同时将初始信息素设置为一个大于 $\text{Tau}_{\max}$ 的一个非常大的值,由于区间限制的存在,可以增加算法的探索性,避免算法早熟。

$$\text{Tau} = \begin{cases} \text{Tau}_{\min}, \text{Tau} < \text{Tau}_{\min} \\ \text{Tau}, \text{Tau}_{\min} \leq \text{Tau} \leq \text{Tau}_{\max} \\ \text{Tau}_{\max}, \text{Tau} > \text{Tau}_{\max} \end{cases} \quad (15)$$

2.3 死锁问题处理策略

当运动环境更加复杂时,由于禁忌表的存在,蚂蚁下一步可能没有节点进行选择,从而陷入死锁。如图 1,当蚂蚁从节点 T 到节点 S 时,下一个可选节点仅有 C ,蚂蚁将陷入死锁。对于死锁问题,文献[13]采用将陷入死锁状态的蚂蚁杀死的策略进行处理。然而,当较多蚂蚁被消灭时,不利于算法寻找最佳路径。文献[14]提出了借用禁忌表来标记死锁节点的方法。文献[15]提出让陷入死锁的蚂蚁返回起点重新出发,该方法能避免蚂蚁数量减少,但存在计算量大的问题。为克服以上缺

陷,本文提出一种新的处理策略,当节点只有一个可行的相邻节点时,直接把该节点标记为障碍节点并进行撤回,该方法能有效避免蚂蚁陷入 U 型陷阱。此外,在寻路过程中,当蚂蚁无可行节点时,令其直接死亡,保证算法不停滞且不增加计算量。

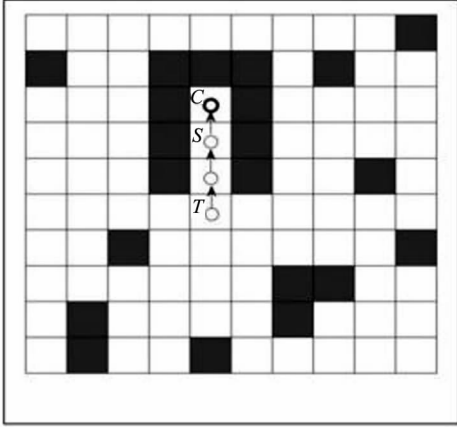


图 1 死锁状态图

2.4 改进算法执行步骤

改进算法主要为以下几步:

Step1: 建立环境地图模型,给出移动机器人的起点和目标点。

Step2: 初始化蚂蚁系统。设置蚂蚁数量 n , 参数 α 决定信息素路径的相对影响, 参数 β 决定启发式值, 全局信息素挥发系数 ρ , 信息素强度 Q 等相关参数。

Step3: 将蚂蚁 k ($k = 1, 2, \dots, n$) 放置在当前节点上, 并在禁忌表中更新已经遍历过的节点。

Step4: 根据禁忌表, 判断蚂蚁是否陷入了死锁状态。如果蚂蚁处于死锁状态, 则采用改进处理机制。

Step5: 根据式(1)计算出蚂蚁状态转移概率, 使用轮盘赌方法选择下一路径方向。若蚂蚁此时已经到达终点, 则进行下一步操作, 否则返回 Step3。

Step6: 判断蚂蚁是否到达目标点, 重复之前步骤, 直至 n 只蚂蚁在当前迭代过程中全部完成搜索。

Step7: 更新信息素。如果迭代次数满足不等式 $N > N_{\max}$, 则不再计算, 就会输出最终结果, 否则将转到 Step3。

算法流程如图 2 所示。

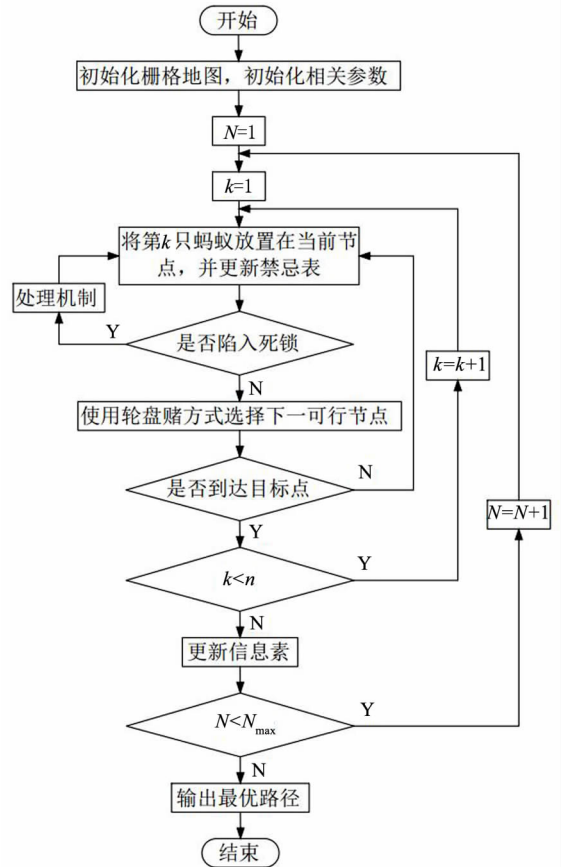


图 2 算法流程图

3 仿真分析

3.1 地图模型

利用栅格模型建立工作环境, 将机器人的工作空间划分为 $N \times N$ 个正方形。如图 1 所示, 黑色单元为障碍, 白色单元为可通行。为转换成机器语言, 白色单元格标记为 0, 黑色单元格标记为 1。栅格地图使用方法和创建方式简单, 具有较强的适应性, 计算机处理和存储方便等优点。

将栅格地图模型置于二维坐标系中, 然后采用序列号法对每个网格进行标记。在图中 $N \times N$ 网格映射中, 位置坐标 (x, y) 对应于号码为 T 的网格, 其横坐标和纵坐标的计算公式如下:

$$\begin{cases} x = \begin{cases} \text{mod}(T, N) - 0.5, & \text{mod}(T, N) \neq 0 \\ N + \text{mod}(T, N) - 0.5, & \text{其他} \end{cases} \\ y = N + 0.5 - \text{ceil}(T/N) \end{cases} \quad (16)$$

式中, mod 是取余操作, ceil 是向上取整操作。

3.2 改进蚁群算法仿真

为保证实验结论的准确性, 在比较三种算法的

性能时,应尽量保持算法的公共参数一致。经分析了大量实验数据并参考了相关文献后,本文的实验参数初始值设置如表 1 所示。

表 1 实验参数		
参数	符号	参数值
蚂蚁数量	n	50
迭代次数	K	100
r 信息素蒸发系数	ρ	0.3
信息素重要程度参数	α	1
启发因子重要程度参数	β	7
小方格边长	a	1
信息素增强系数	Q	1
路径长度信息素增强系数	Q_1	1.2
弯曲因子信息素增强系数	Q_2	1.1
转弯角度转换系数	x	$10/\pi$
转弯次数转换系数	y	2

本文将传统蚁群算法、文献[8]算法以及本文的改进算法放到 Matlab 仿真平台上进行对比分析。由于轮盘赌法的不确定性,在保证地图环境一致的前提下,将 3 种算法进行相同次数的仿真,最后取综合指标进行对比。为使实验结果更有说服力,本文使用 Matlab 仿真软件先后在 2 种不同的环境中进行 10 次仿真实验。

环境一仿真的运动轨迹和迭代次数收敛曲线如图 3 和图 4 所示,仿真结果数据如表 2 所示。环境二仿真的运动轨迹和迭代次数收敛曲线如图 5 和图 6 所示,仿真结果数据如表 3 所示。

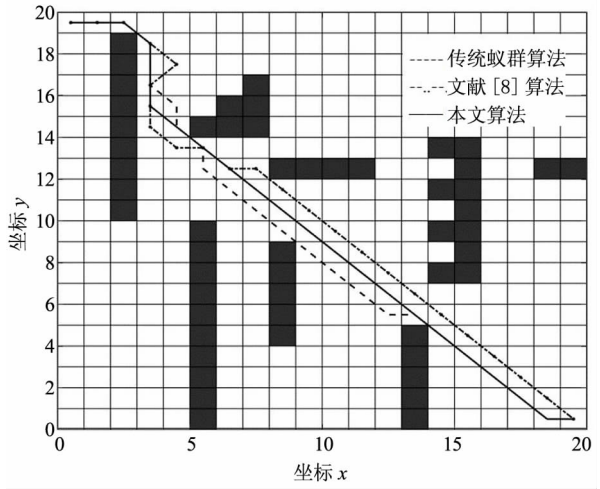


图 3 环境一轨迹对比

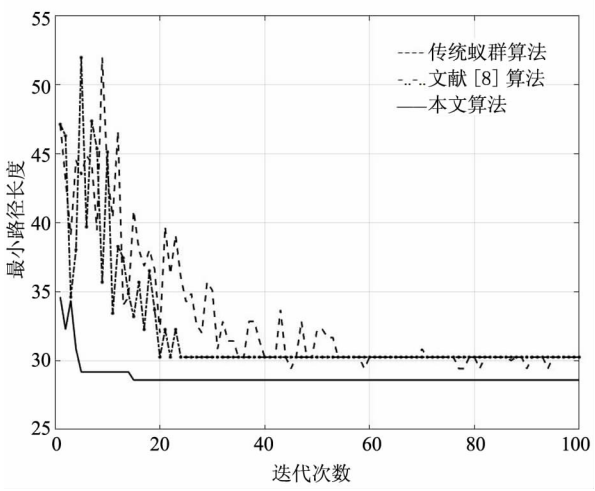


图 4 环境一收敛曲线对比

表 2 环境一仿真数据			
指标	传统算法	文献[8]	本文算法
最优路径长度	30.0416	29.4558	28.6274
最少转弯次数	10	8	4
最小迭代次数	52	22	6
平均程序运行时间/s	4.0563	2.7896	2.4024

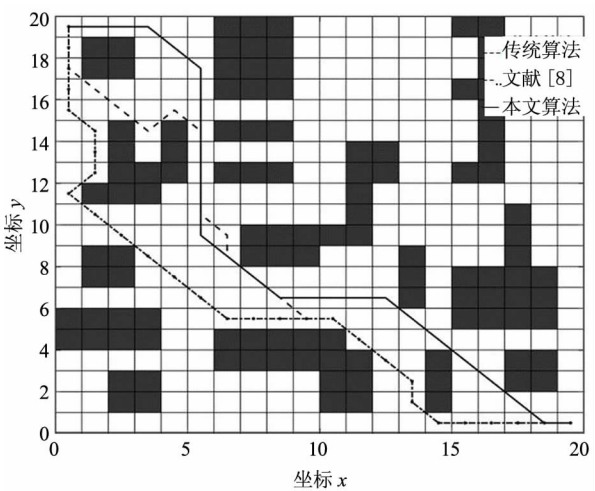


图 5 环境二轨迹对比

表 3 环境二仿真数据			
指标	传统算法	文献[8]	本文算法
最优路径长度	33.2748	32.1421	31.1042
最少转弯次数	12	9	6
最小迭代次数	88	43	10
平均程序运行时间/s	4.0742	2.9233	2.4656

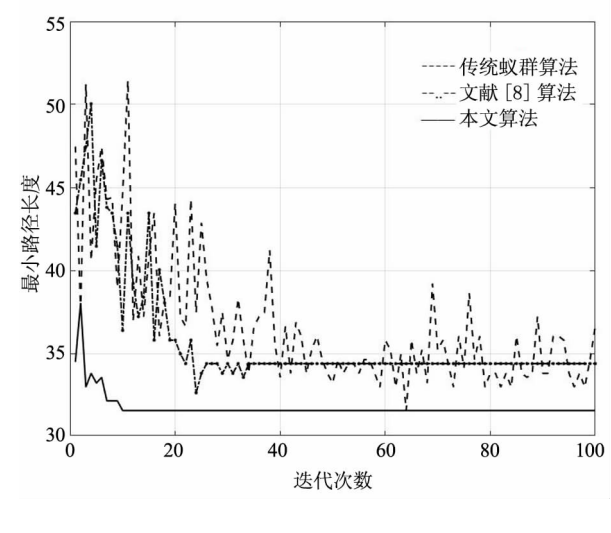


图 6 环境二收敛曲线对比

从两种环境的仿真数据表 2 和表 3 可知,在复杂的地图环境中,本文的改进算法在最优路径长度、转弯次数、收敛速度以及程序运行时间等方面较传统蚁群算法和文献[8]当中的算法都有显著的提升,体现了本文改进算法的有效性和优越性。

3.3 改进撤回机制的测试

为更好地体现本文改进撤回机制的优劣,在其他参数不变的前提下,将算法中的蚂蚁数量减少为 20 只,迭代次数减少为 50 次。图 7 和图 8 为原算法和采用改进撤回机制两种算法的轨迹与收敛曲线。

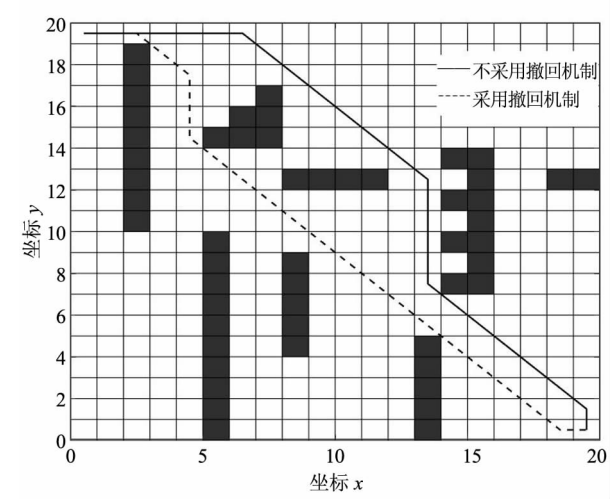


图 7 撤回机制路径对比

为验证算法的稳定性和可靠性,对每种算法重复进行 10 次实验并记录数据,实验统计数据如表 4 和表 5 所示,从两表中可以看出采用撤回机制的算法找到最优路径的概率更大,且算法达到收敛所需迭代次数更少。

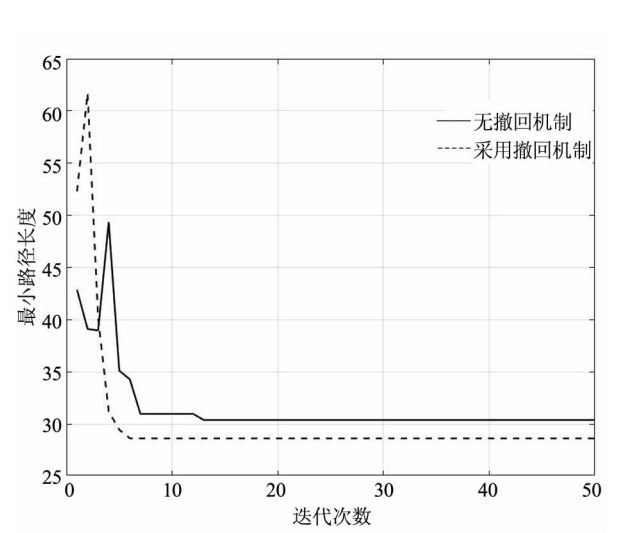


图 8 撤回机制收敛曲线对比

表 4 路径长度对比数据

次数	路径长度	
	无撤回机制	采用撤回机制
1	30.3848	29.7990
2	30.6274	30.3848
3	31.5563	28.6274
4	30.9706	30.3848
5	30.9706	29.7990
6	29.7990	29.7990
7	30.3848	28.6274
8	29.7990	28.6274
9	30.3848	29.7990
10	30.3848	28.6274

表 5 迭代次数对比数据

次数	迭代次数	
	无撤回机制	采用撤回机制
1	17	9
2	26	9
3	10	6
4	9	6
5	17	10
6	22	9
7	24	13
8	21	12
9	16	9
10	13	7

4 结 论

针对原始蚁群算法在较复杂的二维栅格地图下存在搜索时间长、收敛速度慢、找寻不到最优路径、考虑因素单一等问题,提出一种改进蚁群算法。首先在原蚁群算法的基础上融入 A* 算法的计算优先级函数,增加考虑转向因素,在启发式信息中加入弯曲抑制算子,优化了蚁群算法的启发函数,提高了算法的收敛速度;其次引入最大最小蚂蚁系统,提出新的信息素更新规则,设置信息素上下限,避免算法早熟;最后引进改进的撤回机制,解决算法死锁问题,增强算法的全局搜索能力。仿真实验结果表明,相同地图环境中,本文改进算法与基本蚁群算法相比,路径长度更短、转弯次数更少、收敛速度更快。

参考文献

[1] 陈智康,刘佳,王丹丹,等.改进 Dijkstra 机器人路径规划算法研究[J].天津职业技术师范大学学报,2020,30(3):30—35.

[2] 鲁毅,高永平,龙江腾. A* 算法在移动机器人路径规划中的研究[J].湖北师范大学学报(自然科学版),2022,42(2):59—65.

[3] 符强,蓝星辉,任风华,等.一种双向目标 RRT 路径规划算法研究[J].计算机仿真,2023,40(1):447—454,461.

[4] 黄永良,李燕,张小龙.基于改进蚁群算法的机器人路径规划[J].电脑与信息技术,2023,31(2):1—4.

[5] 宋俊福,徐炳辉,张岩,等.基于改进自适应遗传算法的机器人路径规划[J].信息技术,2022,46(11):49—53+60.

[6] 封声飞,雷琦,吴文烈,等.自适应蚁群算法的移动机器人路径规划[J].计算机工程与应用,2019,55(17):35—43.

[7] 王星宇,胡燕海,徐坚磊,等.基于改进蚁群算法的机器人路径规划方法[J].电子技术应用,2023,49(1):75—80.

[8] 马军,宋栓军,韩军政,等.融合蚁群-A* 算法的移动机器人路径规划[J].西安工程大学学报,2020,34(1):72—77.

[9] 鲁飞,鲁照权,牛晨,等.基于改进蚁群算法的三维路径规划研究[J].传感器与微系统,2022,41(1):45—49.

[10] 赵天亮,张小俊,张明路,等.基于改进融合蚁群算法的机器人路径规划方法研究[J].制造业自动化,2022,44(5):185—190.

[11] LI E C, QI K K. Ant colony algorithm for path planning based on grid feature point extraction[J]. Journal of Shanghai Jiaotong University (Science), 2023, 28(1): 86—99.

[12] LI C Y, LIU Q J, SONG S, et al. Path planning for mobile robots based on an improved ant colony algorithm with gaussian distribution [J]. Journal of Physics: Conference Series, 2022, 2188(1): 012005.

[13] WANG D S, YU H F. Path planning of mobile robot in dynamic environments[C]//2011 2nd International Conference on Intelligent Control and Information Processing. IEEE, 2011, 2: 691—696.

[14] LIN W. Path planning for unmanned wheeled robot based on improved ant colony optimization[J]. Measurement and Control, 2020, 53(5—6): 1014—1021.

[15] 任红格,胡鸿长,史涛.基于改进蚁群算法的移动机器人全局路径规划[J].华北理工大学学报(自然科学版),2021,43(2):102—109.