

基于凸策略优胜劣汰蚁群算法的机器人路径规划^①

鲍义东^{1,2}, 夏栋梁¹, 赵伟艇¹

¹(平顶山学院 软件学院, 平顶山 467000)

²(哈尔滨工业大学 机器人技术与系统国家重点实验室, 哈尔滨 150080)

摘要: 针对机器人在已知静态工作环境中运动路径的快速选择和判优, 提出一种改进蚁群优化算法. 该算法首先对机器人的静态工作空间进行凸策略处理, 从环境上降低了搜索的盲目性和落入陷阱的可能性, 并在此基础上加入一种优胜劣汰策略, 进一步提高了算法的时间性能、最佳性能和鲁棒性. 实验结果表明改进的蚁群优化算法不仅克服了易于陷入局部最优解及运算量大的缺陷, 而且也大幅度提高了算法的运算效率.

关键词: 路径规划; 蚁群算法; 凸处理; 优胜劣汰策略

Robot Path Planning Based on Ant Colony Optimization Convex Fittest Strategy

BAO Yi-Dong^{1,2}, XIA Dong-Liang¹, ZHAO Wei-Ting¹

¹(School of Software, Pingdingshan University, Pingdingshan 467000, China)

²(State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150080, China)

Abstract: An improved ant colony optimization based on convex policy and the survival of the fittest strategy for robot path planning is proposed in this paper. This algorithm first use convex policy to process the robot's static workspace, and it can reduce the blindness of searching and the possibility of falling into the trap in environment. In addition, a survival of the fittest strategy is added in the ant colony optimization to further improve the performance of the algorithm time, optimum performance and robustness. Experimental results show that the improved ant colony optimization not only can overcome the defect of easy to fall into local optimal solution, but also can improve the operation efficiency.

Key words: path planning; ant colony algorithm; convex treatment; survival of the fittest strategy

机器人路径规划问题是机器人导航和控制的基本问题,关键是要找到一个从起点到目标点机器人能够安全绕过障碍且没有碰撞的过程^[1]. 当前有许多传统方法解决机器人路径规划问题,如Petri网络算法,基于数据融合的模糊规划、人工势场法、可见的图顶点法和多边形法图的方法等^[2]. 但这些方法的计算效率通常比较低. 因此,提出了几种针对启发式方法的的解决问题方案,具有良好的性能,例如人工势场、模糊逻辑、神经网络、遗传算法、蚁群优化等^[3].

蚁群优化算法是元启发算法,也是一种仿生进化算法. 它适合解决离散的组合优化问题,具有并行计算,自适应,积极反馈,和良好的鲁棒性等特点^[1].

近年来,蚁群优化算法吸引了学者们的高度关注,

提出很多改进的蚁群算法来解决各种非确定性问题,如二次分配问题^[4],作业调度问题^[5],多目标跟踪^[6],动态生产调度^[7],车辆路径问题^[8],系统识别^[9]等.

移动机器人在复杂环境下只有极少的先验知识,而采用不同的全局搜索和局部优化方法,实现在复杂环境中机器人搜索目标的有效路径规划,这个过程需要学习并积累知识. 在蚁群优化算法中,蚂蚁通过感应连接路径走向目的地^[10,11]. 蚂蚁移动过程是从当前位置移到对象或目标的获取的最短路径信息. 如果最短路径被障碍物阻塞且不能克服,而蚂蚁的当前位置不包含结点,则就出现了蚂蚁朝那个方向移动直至终点的问题. 因此,机器人路径规划问题可以采用蚁群优化算法来解决^[12,13].

① 基金项目:河南省教育厅科学技术研究重点项目(14B520039);校青年科研基金项目(PXY-QNJJ-2014004)

收稿时间:2014-11-25;收到修改稿时间:2015-01-12

高尚^[14]提出了一种基于网格模型的改进蚁群算法,并离散网格节点上的信息素,来解决连续空间优化问题. Chen^[15]通过在节点信息素沉积改进蚁群优化算法.该算法引入信息素挥发因子的自适应策略来提高算法的性能,并应用于机器人路径规划问题. 陈杰等^[16]通过信息量更新方式和使用局部最优搜索策略来改进蚁群算法的收敛性,并提出了采用天折算法来解决蚂蚁落入陷阱问题.

上述采用的方法,在搜索过程中具有较高的盲目性和落入陷阱的可能性. 本文提出了一种在基本蚁群算法的基础上提出了优胜劣汰策略与凸处理策略相结合的算法. 算法首先对机器人的静态工作空间进行凸策略处理,接着融合蚁群算法的优胜劣汰策略的处理. 该算法能进一步从环境上降低了搜索的盲目性和落入陷阱的可能性,进一步提高了算法的时间性能、最佳性能和鲁棒性.

1 基本蚁群算法模型

基本的蚁群算法蚂蚁状态转移概率如下:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\beta(t)}, & j \in allowed_k \\ 0, & \text{其他} \end{cases} \quad (1)$$

式中, m 是蚂蚁数量; $b_i(t)$ 表示在 t 时刻位于某个城市的蚂蚁数量, 记作 $m = \sum_{i=1}^n b_i(t)$; d_{ij} 表示两城市 i 和 j 的距离; η_{ij} 为启发函数, 一般取 d_{ij} 的倒数; τ_{ij} 表示路径 (i, j) 上的信息素轨迹强度; $\Delta\tau_{ij}$ 表示在边 (i, j) 上, 蚂蚁 k 留下的单位长度轨迹信息素量; p_{ij}^k 表示蚂蚁 k 在城市 i 上选择城市 j 的转移概率. α 和 β 分别表示信息启发式因子和期望启发式因子^[17].

用一个数据结构禁忌表 $tabu_k$ 来表示每个蚂蚁必须经过所有城市的这个约束条件. $tabu_k$ 中记录了当前时刻某个蚂蚁已经走过的城市名称, 且需要保证在本次循环中不允许该蚂蚁再重复走过这些城市. 当本次循环结束后, 式中 $allowed_k = \{1, 2, \dots, n\} - tabu_k$ 表示蚂蚁 k 下一步允许选择的城市.

在经过 n 个时刻后, 蚂蚁完成一次循环, 各路径上信息素量根据下式进行调整

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t, t+n) \quad (2)$$

$$\Delta\tau_{ij}(t, t+n) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+n) \quad (3)$$

$$\Delta\tau_{ij}^k(t, t+n) = \begin{cases} \frac{Q}{L_k}, & \text{如果在本次循环中蚂蚁} k \text{ 经过路径}(i, j) \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中, $\Delta\tau_{ij}^k(t, t+n)$ 表示在路径 (i, j) 上, 第 k 只蚂蚁在本次循环中留下的信息素量. $\Delta\tau_{ij}^k(t, t+n)$ 表示路径 (i, j) 上, 本次循环信息素的增量; ρ 为路径上信息素残留因子, $1-\rho$ 为路径上信息素的衰减系数; L_k 为在本次循环中第 k 只蚂蚁走的路径长度; Q 为信息素强度.

上述蚁群算法模型的算法步骤如下^[16]:

Step1:初始化蚁群相关参数. 设时间 $t=0$, 循环初始次数为 $N_c=0$, 最大循环次数为 N_c_max . 设图中每条边 (i, j) 的初始信息素量为 $\tau_{ij}(t) = \text{const}(\text{const}$ 为常数), 初始时刻的信息素增量 $\Delta\tau_{ij}(0)=0$, 确定启发函数 η_{ij} , 并设置禁忌表为空;

Step2: 将 m 只蚂蚁随机放置到 n 个城市上, 然后把每只蚂蚁的初始城市置到当前禁忌表中;

Step3: 循环次数从 $N_c \leftarrow N_c + 1$;

Step4: 蚂蚁序号 $k \leftarrow k + 1$;

Step5: 根据式(2.1)计算出概率, 蚂蚁根据概率选择下一个城市 j ;

Step6: 根据新的城市移动蚂蚁, 修改禁忌表;

Step7: 是否遍历完所有城市, 即 $k < m$, 是则跳转到 Step3, 否则执行 Step7;

Step8: 根据公式(2),(3)和(4), 更新每条路径上的信息素;

Step9: 若循环次数 $N_c < N_c_max$, 清空禁忌表, 并跳转到 Step2; 否则, 结束循环并输出程序结果.

2 凸处理与优胜劣汰策略

机器人路径规划在选择使用基本蚁群算法时, 当环境中存在一些复杂障碍物的时, 某些搜索路径的蚂蚁就有可能陷入这些障碍物, 从而出现无后继栅格可供选择的现象, 从而使得整个算法停滞.

91	92	93	94	95	96	97	98	99	100
81	82	83	84	85	86	87	88	89	90
71	72	73	74	75	76	77	78	79	80
61	62	63	64	65	66	67	68	69	70
51	52	53	54	55	56	57	58	59	60
41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

(a)普通栅格

91	92	93	94	95	96	97	98	99	100
81	82	83	84	85	86	87	88	89	90
71	72	73	74	75	76	77	78	79	80
61	62	63	64	65	66	67	68	69	70
51	52	53	54	55	56	57	58	59	60
41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

(b)凸处理之后的栅格

图 1 栅格法普通图与带有凸处理策略的视图

解决这个问题一般采用凸处理方法，在初始化栅格环境时，对其中复杂障碍物做特定的人工假设或处理，使栅格环境中所有复杂障碍物都为凸的形状，以消除算法的停滞现象。用凸处理方法虽然可以在一定程度上减小算法停滞现象的发生，但并没有消除障碍物与环境边界之间生成的陷阱。如在图 1(b)中，左下角的障碍物与环境边界生成了一个隐藏的陷阱栅格，蚂蚁没有后继选择栅格，使整个算法停滞。

由上述可知采用凸处理会产生障碍物与环境边界之间所生成的陷阱，为了解决这一问题并进一步提高算法性能，提出一种优胜劣汰策略算法。传统蚁群算法在一次循环中从初始位置移动到目标位置的蚂蚁是正常死亡的，退出自己的生命周期，结束路径搜索过程。而本文的优胜劣汰策略是让那些在搜索路径的过程中落入陷阱的蚂蚁和落在蚁群倒数几位的蚂蚁不再参与路径搜索，提前退出自己的生命周期。在信息素更新阶段，被淘汰的蚂蚁所走过的路径不进行任何信息素更新。

优胜劣汰策略的蚁群算法实现路径规划的过程如下：

- (1) 设置参数并初始化蚁群；
- (2) 将蚁群中所有蚂蚁放置起始位置，并将 g_1 加入到所有蚂蚁的禁忌表中；
- (3) 从第一只蚂蚁开始，利用状态转移规则选择移动到下一个栅格，并将下一个栅格加入到蚂蚁的禁忌表中；
- (4) 判断该蚂蚁是否还有后继栅格可供选择，如果无后继栅格可选，则说明蚂蚁落入陷阱，则淘汰此蚂蚁，使其退出本次路径搜索过程；
- (5) 当所有蚂蚁都选择好下一个栅格后，本次蚁群中未淘汰的蚂蚁开始下一步移动；
- (6) 如此反复，直至已有接近半数(实践值)的蚂蚁

已经到达目标点，则淘汰所有剩下未到达目标点的蚂蚁，则蚁群的本次路径搜索结束。

在正常死亡的蚂蚁所走过的路径上进行信息素更新，被淘汰的蚂蚁不进行信息素更新。出动下一拨蚂蚁，用同样的方法进行路径搜索过程并进行信息素更新。如此反复，直至满足循环次数或者某些终止条件。

当环境中特殊障碍物比较多时会有相当数量的蚂蚁在达到路径规划目标位置之前被淘汰，一定程度上减少了有效搜索最优路径的蚂蚁的数量，但因为优胜劣汰策略的实现非常简单，只需让蚁群前半数的蚂蚁正常死亡即可，所以带优胜劣汰策略的蚁群算法是可行的。

另一方面来说，由于在路径搜索过程中被淘汰的蚂蚁不需要参与接下来的路径搜索，即不用像其它蚂蚁那样需要连续不断地选择下一个要移动的栅格直至到达路径规划目标位置为止，所以优胜劣汰策略的引入实际上简化了蚁群搜索路径的过程，降低了算法的复杂度。

在信息素更新阶段，带优胜劣汰策略的蚁群算法采取了与基本蚁群算法不同的信息素更新策略，即只对一次循环中前几个完成整个路径搜索过程的蚂蚁所走过的路径进行信息素更新。因为蚂蚁或者是在落入陷阱而无后继栅格可选的情况下或者就是选择了一个比最优路径较差的路径才被淘汰的，所以被淘汰的蚂蚁所走过的路径必定不是最优路径，同时为了避免以后的蚂蚁再次落入同样的陷阱，对被淘汰的蚂蚁所走过的路径不进行任何信息素更新。

优胜劣汰算法如图 2 所示。

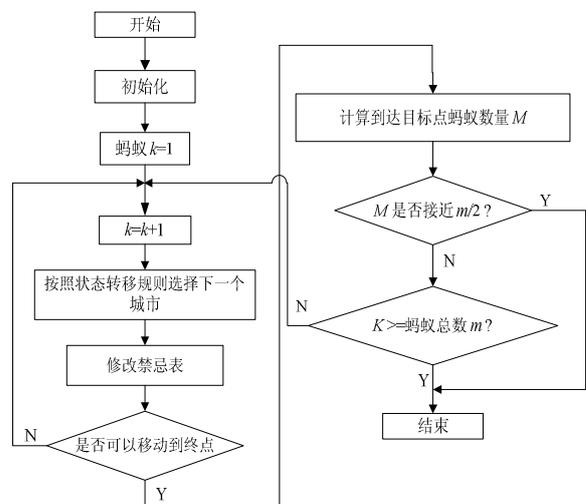


图 2 优胜劣汰算法流程图

3 凸处理与优胜劣汰策略蚁群算法

为了解决传统蚁群算法的算法停滞和在复杂环境下频繁落入陷阱的问题, 本文将凸处理策略与优胜劣汰策略结合. 融合算法以 10×10 栅格和 20×20 栅格为例, 给出算法的执行过程.

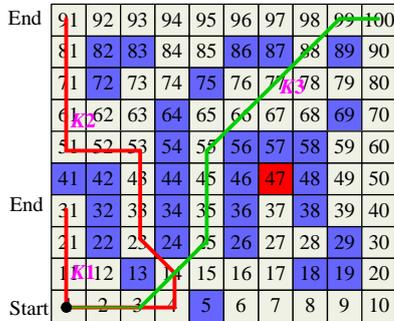


图 3 基于凸处理与优胜劣汰策略的融合的 10×10 栅格蚂蚁行走示意图

图 3 中, 栅格 g_{47} 通过凸处理策略处理后变成障碍物, 蚂蚁 K_1 由于落入陷阱 g_{31} 被淘汰, 蚂蚁 K_2 由于优胜劣汰策略的存在大大落后于最优路径或者接近最优路径的蚂蚁而在 g_{91} 栅格被淘汰. 蚂蚁 K_3 属于本次循环中的正常死亡. 从而循环往复就可以得出最优路径.

融合算法 20×20 栅格示意图:

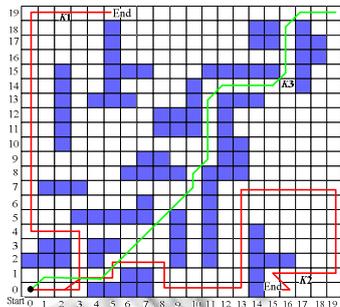


图 4 基于凸处理与优胜劣汰策略的融合的 20×20 栅格蚂蚁行走示意图

随着地图的复杂性增加, 蚂蚁所走的路径的选择结果也就跟着增加, 本图列出了其中三种机器人路径规划情况, 蚂蚁 K_1 和 K_2 是非正常死亡的情况, 是由于:

- (1) 蚂蚁进入陷阱而无法继续行走;
- (2) 该蚂蚁所走的路径远大于理论最优路径.

蚂蚁 K_3 所走路径是蚂蚁正常死亡的路径选择的一种, 以此方法的不断循环从而可以找到最优路径.

另外, 蚁群算法陷入局部最优问题上, 由于在迭代次数内蚁群探索路径时选择最佳路径的概率和次数均不可能为 0, 而正是由于凸处理和优胜劣汰融合策略以及信息素更新原则使得蚁群选择最佳路径的概率和基值变得越来越大, 直到通过优胜劣汰策略淘汰掉不是最佳路径的局部最优路径, 进而得出最佳路径. 因此, 本文的算法策略是可以避免算法陷入局部最优情况的.

将凸处理策略与优胜劣汰策略融合后的改进的蚁群算法应用到机器人路径规划的算法流程图(如图 5).

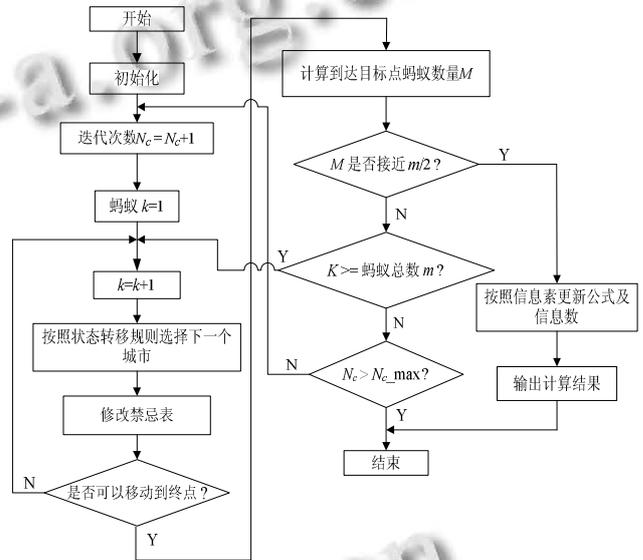


图 5 基于凸处理优胜劣汰蚁群算法流程图

4 实验与结果分析

4.1 参数设置

(1) 蚂蚁数量 m 的选择

蚂蚁数量的选取至关重要. 如果选取蚂蚁数量多, 则可以提高蚁群算法的稳定性和全局搜索能力, 会使信息正反馈作用减弱, 但收敛速度减慢. 而过少的蚂蚁数量加快了收敛速度, 但算法全局搜索的随机性减弱, 导致稳定性差. 本算法选取蚂蚁数量 $m = 10$.

(2) α , β 和 ρ 的组合设置

α 为信息启发式因子, 反映了蚂蚁在运动过程中所积累的信息量. α 越大, 搜索的随机性减弱, 但 α 过小时, 容易使得搜索过早陷入局部最优.

β 为期望启发式因子, 反映了蚂蚁在运动过程中启发式信息. β 越大, 算法的收敛速度越快, 但搜索随机性减弱, 易于陷入局部最优.

ρ 为信息素残留因子, 反映了蚂蚁个体之间相互影响强弱程度. ρ 越大, 算法的随机性能与全局搜索能力越强, 但会使算法的收敛速度降低; 但 ρ 过小时, 前期搜索过的路径被再次选择的可能性就过大, 影响了算法的随机性与全局搜索能力.

实际上, α , β 和 ρ 的作用是紧密耦合的. 文献[18]认为, 这三个参数的最佳取值范围是: $0 \leq \alpha \leq 5, 0 \leq \beta \leq 5, 0.1 \leq \rho \leq 0.99$. 本文确定三个值的过程是: 先确定蚂蚁数目, 然后对参数进行微调, 对取值范围较大的一些参数, 如信息启发因子 α , 期望启发因子 β , 以及信息素强度 Q 等参数进行调整. 最后对取值范围较小的信息素挥发因子 ρ 进行微调. 经过多次试验和数据的统计分析. 最后在本文算法中三个参数取值为 $\alpha=1, \beta=5, \rho=0.5$.

(3) Q 的确定

Q 为信息素强度, 表示一次循环后蚂蚁在经过路径上所释放在的信息素总量. Q 越大, 路径上信息素的累积速度就越快, 算法收敛速度越高. 但 Q 过大时, 全局搜索能力变差, 容易陷入局部最优解. 本算法取 $Q=100$.

(4) 循环次数 N_c 的设置

为保证所提算法在设定的循环次数内收敛到最优路径, N_c 的值应尽量取大一些. 在栅格环境地图大小为 10×10 的情况下, 本算法选取 $N_c=100$.

4.2 仿真结果分析

(1) 10×10 栅格环境仿真

依照算法基本性能指标的定义, 并按照图 6 仿真结果可以计算出 10×10 栅格环境下本算法的最佳性能指标为:

$$E_0 = \frac{c_b - c^*}{c^*} \times 100\% = \frac{15.06 - 15.06}{15.06} \times 100\% = 0.00\%$$

时间性能指标为:

$$E_T = \frac{I T_0}{I_{max}} \times 100\% = \frac{30 \times 0.0055}{100} \times 100\% = 0.165\%$$

鲁棒性能指标为:

$$E_R = \frac{c_a - c^*}{c^*} \times 100\% = \frac{15.0712 - 15.06}{15.06} \times 100\% = 0.07\%$$

结合基本蚁群算法和带凸处理和优胜劣汰机制的蚁群算法的实现过程, 从图 6 中可以发现最优路径长度为 14.49, 改进后的蚁群算法在迭代 5 次找到最优路径, 优于传统蚁群和文献[16]算法.

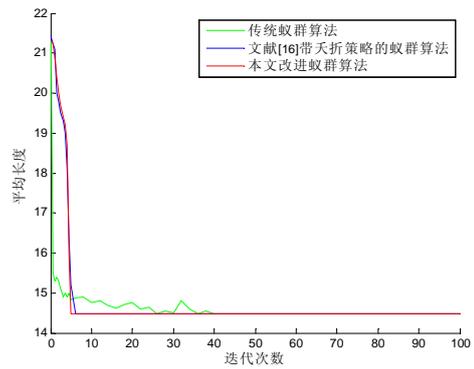


图 6 10×10 栅格不同算法迭代平均路径长度比较

表 1 10×10 栅格本文改进蚁群算法与带天折蚁群算法性能比较

算法策略	迭代次数	最短路径	最佳性能指标(%)	时间性能指标(%)	鲁棒性能指标(%)
本文改进蚁群算法	5	14.4853	0.00	0.165	0.07
带天折蚁群算法	6	14.4853	0.00	1.35	0.15

表 1 表示本文改进算法和文献[16]算法性能比较. 由表可以得出本蚁群算法在蚂蚁搜索最优路径的时候不仅可以在时间上容易搜索到最短路径而由于淘汰机制的使用不会陷入局部最优, 且需要的迭代次数比较少.

(2) 20×20 栅格环境下的仿真

为了比较改进算法的优越性, 采用上述同样的 20×20 栅格环境得出 MATLAB 仿真结果, 如图 7.

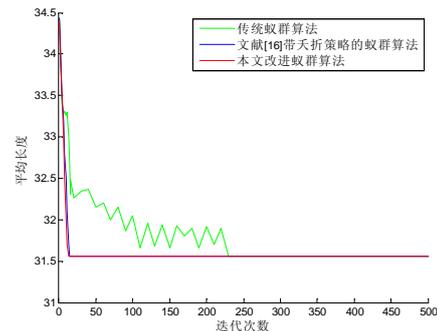


图 7 20×20 栅格不同算法迭代平均路径长度

依照算法基本性能指标的定义, 由仿真结果可以计算出 20×20 栅格环境下本算法的最佳性能指标为:

$$E_0 = \frac{c_b - c^*}{c^*} \times 100\% = \frac{31.56 - 31.56}{31.56} \times 100\% = 0.00\%$$

时间性能指标为:

$$E_r = \frac{I_a T_0}{I_{\max}} \times 100\% = \frac{220 \times 0.015}{500} \times 100\% = 0.66\%$$

鲁棒性性能指标为:

$$E_R = \frac{c_a - c^*}{c^*} \times 100\% = \frac{31.60 - 31.56}{31.56} \times 100\% = 0.127\%$$

结合基本蚁群算法和带凸处理和优胜劣汰机制的蚁群算法的实现过程,从图 7 中可以发现最优路径长度为 31.56,改进后的蚁群算法在迭代 5 次找到最优路径,优于传统蚁群和文献[16]算法.表 2 表示本文改进算法和文献[16]算法性能比较.

表 2 20×20 栅格本文改进蚁群算法与带天折蚁群算法性能比较

算法策略	迭代次数	最短路径	最佳性能指标(%)	时间性能指标(%)	鲁棒性能指标(%)
本文改进蚁群算法	13	31.5	0.00	0.66	0.127
带天折的蚁群算法 ^[16]	15	31.5	0.00	1.06	0.27

5 结论

传统蚁群算法在解决这个问题上算法复杂且容易落入陷阱,使用凸处理对障碍物与障碍物之间产生的陷阱进行了人工填充和假定,而在用凸处理策略中存在着不能处理障碍物环境边界之间产生的陷阱.为了改善这个缺陷及进一步提高整个算法的性能,提出了优胜劣汰策略,并将两者结合成改进蚁群算法的算法,通过 MATLAB 仿真,可以得出改进的算法解决了传统的蚁群算法易于陷入局部最优解和落入环境陷阱,和带天折的算法相比,具有最佳性能、时间性能和鲁棒性的整体优势.

参考文献

- Deng XY, Zhang LM, Luo L. An improved ant colony optimization applied in robot path planning problem. *Journal of Computers*, 2013, 8(3): 585–593.
- Abe Y, Shikann M, Fokuda T. Vision based navigation system by variable template matching for autonomous mobile robot. *Proc. of the IEEE International Conference on Robotics & Automation*. Leaven. 1998. 952–957.
- Weerayuth N, Chaiyaratana N. Closed-loop time-optimal path planning using a multi-objective diversity control oriented genetic algorithm. *Proc. of the IEEE International Conference on Man and Cybernetics*. 2002, 6. 16–20.
- Maniezzo V, Colorni A, Dorigo M. The ant system applied to the quadratic assignment problem[Technical Report]

- IRIDIA/94-28, IRIDIA, Université Libre de Bruxelles, Belgium, 1994.
- Gao QL, Luo X, Yang SZ. Stigmergic cooperation mechanism for shop floor control system. *International Journal of Advanced Manufacturing Technology*, 2005, 25(7): 743–753.
- Bozdogan AO, Efe M. Improved assignment with ant colony optimization for multi-target tracking. *Expert Systems with Applications*, 2012, 38: 9172–9178.
- Xiang W, Lee HP. Ant colony intelligence in multiagent dynamic manufacturing scheduling. *Engineering Applications of Artificial Intelligence*, 2008, 21: 73–85.
- John EB, Patrick RM. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 2004, 18: 41–48.
- Wang L, Wu QD. Linear system parameters identification based on Ant system algorithm. *Proc. of the IEEE Conference on Control Application*. 2001. 401–406.
- Sariff NB, Buniyamin O. Ant colony system for robot path planning in global static environment. *Selected Topics in System Science And Simulation in Engineering*, 2011: 192–197.
- Buniyamin N, Sariff N, Wan Ngah WAJ, Mohammad Z. Robot global path planning overview and a variation of ant colony system algorithm. *International Journal of Mathematics and Computers in Simulation*, 2011, 5(1): 9–16.
- Chakraborty S. Ant colony system: A new concept to robot path planning. *International Journal of Hybrid Information Technology*, 2013, 6(6): 11–30.
- Mei H, Tian Y, Zu L. A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment. *International Journal of Information Technology*, 2006, 12(3): 78–88.
- 高尚,钟娟,莫述军.连续优化问题的蚁群算法研究. *微机发展*, 2003, 13(11): 12–13.
- Chen X, Yuan Y. Novel ant colony optimization algorithm for robot path planning. *Systems Engineering and Electronics*, 2008, 30(5): 952–955.
- 陈杰.基于蚁群算法的机器人路径规划研究[硕士学位论文].南京:南京理工大学,2009.
- 李士勇.蚁群算法及其应用.哈尔滨:哈尔滨工业大学出版社,2004.
- 段海滨.蚁群算法原理及其应用.北京:科学出版社,2005.