



计算机工程与应用  
Computer Engineering and Applications  
ISSN 1002-8331, CN 11-2127/TP

## 《计算机工程与应用》网络首发论文

题目: 改进蚁群算法在机器人路径规划中的应用  
作者: 何雅颖, 范昕炜  
网络首发日期: 2021-04-14  
引用格式: 何雅颖, 范昕炜. 改进蚁群算法在机器人路径规划中的应用. 计算机工程与应用. <https://kns.cnki.net/kcms/detail/11.2127.TP.20210413.1744.030.html>



**网络首发:** 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

**出版确认:** 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

# 改进蚁群算法在机器人路径规划中的应用

何雅颖, 范昕炜

中国计量大学 质量与安全工程学院, 杭州 310018

**摘要:** 针对传统蚁群算法在移动机器人路径规划问题中存在的易陷入局部最优与收敛速度慢等问题, 提出一种改进的蚁群算法。首先根据起点到终点距离和地图参数构建全局优选区域, 提高该区域内初始信息素浓度, 避免算法初期盲目搜索; 其次利用局部分块优化策略分别对各个子区域进行寻优并更新区域内最优路径信息素, 增强局部搜索能力加快收敛速度; 后对全局路径进行寻优, 更新全局最优路径信息素。且在信息素更新公式中引入信息素增强因子, 加强最优路径信息素含量。应用反向学习优化信息素, 改进状态选择概率, 提高算法寻优能力。实验结果表明, 改进后的算法明显提高了收敛速度, 同时寻优能力更强。

**关键词:** 蚁群算法; 路径规划; 局部分块优化策略; 增强因子; 反向学习

文献标志码: A 中图分类号: TP242 doi: 10.3778/j.issn.1002-8331.2012-0548

## Application of Improved Ant Colony Optimization in Robot Path Planning

HE Yaying, FAN Xinwei

College of Quality and Safety Engineering, China Jiliang University, Hangzhou 310018, China

**Abstract:** Aiming at the problems of traditional ant colony algorithm in mobile robot path planning, such as easy to fall into local optimum and slow convergence speed, an improved ant colony algorithm is proposed. Firstly, the global optimization region is constructed according to the distance from the starting point to the end point and the map parameters, so as to improve the initial pheromone concentration in the region, and avoid blind search at the beginning of the algorithm. Secondly, the local block optimization strategy is used to optimize each sub region and update the optimal path information in the region, so as to enhance the local search ability and speed up the convergence speed. Finally, the global path is optimized and updated Global optimal path pheromone. The pheromone enhancement factor is introduced into the pheromone updating formula to enhance the pheromone content of the optimal path. The opposition-based learning is applied to optimize pheromone, improve the probability of state selection, so as to improve the optimization ability of the algorithm. The experimental results show that the improved algorithm significantly improves the convergence speed and has stronger optimization ability.

**Key words:** ant colony optimization; path planning; local block optimization strategy; enhancement fact; opposition-based learning

**基金项目:** 国家重点研发计划 (2017YFF0209700)。

**作者简介:** 何雅颖(1995-),女,硕士研究生,研究方向为路径规划、智能控制,E-mail: heyayinga@163.com; 范昕炜(1973-),男,博士,高级工程师,研究方向为机器人、模式识别与智能控制、数据挖掘。

## 1 引言

机器人路径规划是指根据已知障碍环境, 自行规划出一条从起始点到达终止点无碰撞的最优路径<sup>[1]</sup>。其中传统方法包括栅格法、A\*算法<sup>[2]</sup>、人工势场算法<sup>[3]</sup>等。随着智能群算法的发展, 遗传算法<sup>[4]</sup>、粒子群<sup>[5]</sup>算法、蚁群算法等也开始应用于路径规划中。

蚁群算法(Ant Colony Optimization)是由意大利学者 M.Dorigo<sup>[6]</sup>等提出的一种仿生蚂蚁觅食的智能群算法, 具有正反馈、并行性、强鲁棒性和较好适应性的特点, 但同时也具有易陷入局部最优、收敛速度较慢和易陷入死锁等问题。其中信息素对算法结果具有重要影响, 不少学者通过信息素不同的设置方式来优化算法。文献[7]提出非均匀信息素分布划定起点到终点两个点为顶点的矩形区域为有利区域, 提高该区域的初始值, 该方法能有效防止蚂蚁在初期向目标点反方向搜索, 但是区域内信息素并没有差别, 改进效果有限。文献[8]提出双层蚁群算法先通过外层算法求取最优解延伸其解空间, 后利用内层算法进行局部寻优, 如果在此空间内找到更优路径则执行信息素二次更新, 该方法虽然提高了最优解的可能性但是仅限于有限空间探索, 算法可能陷入局部最优。文献[9]引入了最优解与最差解改进信息素更新规则, 增强当前最优路径的信息素, 减弱最差路径的信息素, 该方法有效的加快了收敛速度, 但同时也减少了种群的多样性不利于最优解的获取。文献[10]提出在算法搜索过程中加入人工势场局部搜索寻优算法, 将人工势场中力因素转换为局部扩散信息素, 算法有效减少了交叉路径与蚂蚁“迷失”数量, 提高了蚁群对障碍物的预避障能力, 但算法结构设计较为复杂, 环境适应能力有待提高。文献[11]将遗传算法与蚁群算法相融合, 提出利用遗传算法生成初始路径, 将较优路径作为蚁群算法初始信息素的参考信息, 减少初期算法盲目性, 后利用蚁群算法对初始路径进一步优化, 该改进算法能避免局部最优但算法运行时间较长。

综上所述, 信息素的设置能有效提高收敛速度但同时也有可能降低算法寻优能力, 不利于获取最优解, 为平衡寻求最优解与算法收敛速度, 提出改进的蚁群算法, 其中包括: 对初始信息素进行非均匀分配, 提高收敛速度; 提出局部分块优化策略优化子区域路径提高求解质量; 利用信息素增强因子保留更优解, 避免陷入局部最优; 引入反向学习对信息素进行优化, 改进状态选择概率, 提高算法全局寻优能力。

## 2 相关知识

### 2.1 蚁群算法

#### 2.1.1 转移概率

在  $t$  时刻蚂蚁  $m$  从节点  $i$  到节点  $j$  由公式(1)计算转移概率  $p_{ij}^m(t)$  决定, 并根据轮盘赌法则对下一路径点进行选择。

$$p_{ij}^m(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed_m} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & j \in allowed_m \\ 0, & j \notin allowed_m \end{cases} \quad (1)$$
$$\eta_{ij}(t) = 1/d(i, j) \quad (2)$$

式中:  $\tau_{ij}(t)$  为路径  $(i, j)$  间的信息素含量,  $\alpha$  为信息素启发因子。 $\eta_{ij}(t)$  为期望启发项, 通过式(2)计算可得; 其中  $d(i, j)$  为路径  $(i, j)$  间的欧式距离,  $\beta$  为期望启发式因子。 $\alpha$  和  $\beta$  分别代表着信息素、期望启发项的重要程度。 $allowed_m$  表示蚂蚁  $m$  所有下一可行节点集合。

#### 2.1.2 信息素更新

每只蚂蚁在行走的过程中都会留下一一定量的信息素, 随着迭代次数的增加, 信息素不断地增加, 同时也在不断挥发。所有蚂蚁完成一次迭代后, 会对残留信息素进行更新, 根据式(3)、(4)更新信息素。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{m=1}^w \Delta\tau_{ij}^m(t) \quad (3)$$
$$\Delta\tau_{ij}^m(t) = \begin{cases} Q/L_m, & \text{如果第 } m \text{ 只蚂蚁经过路径 } (i, j) \\ 0, & \text{其他} \end{cases} \quad (4)$$

式(3)中  $\rho$  为挥发系数,  $\Delta\tau_{ij}$  为本次迭代中路径  $(i, j)$  的信息素增量;  $w$  为蚂蚁总数,  $\Delta\tau_{ij}^m(t)$  为本次迭代中第  $m$  只蚂蚁在路径  $(i, j)$  留下的信息素。式(4)中  $\Delta\tau_{ij}^m(t)$  值的计算采用的更新方式为蚁周模型, 除此之外还有蚁量模型和蚁密模型。式中  $Q$  是一个常数, 表示信息素强度值,  $L_m$  表示第  $m$  蚂蚁在本次迭代中所走路径总长度。

### 2.2 反向学习

反向学习的概念于 2005 年由 Tizhoosh<sup>[12]</sup>提出, 后被成功用于算法的进化, 解决优化问题。以下为反向解定义:

**定义 1** 假设  $x$  是一维空间的一个实数,  $x \in [a, b]$ , 因此  $x$  反向解为<sup>[13]</sup>:

$$\bar{x} = a + b - x \quad (5)$$

**定义 2** 假设  $P = (x_1, x_2, \dots, x_D)$  为  $D$  维空间中一点, 其中  $x_i \in [a_i, b_i]$ ,  $\forall i \in \{1, 2, \dots, D\}$  则其反向解为  $\bar{P} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_D)$ , 其中  $\bar{x}_i$  定义为<sup>[13]</sup>:

$$\bar{x}_i = a_i + b_i - x_i \quad (6)$$

### 3 改进蚁群算法

#### 3.1 非均匀初始信息素分布

蚁群算法中, 蚂蚁主要根据栅格间信息素的差异寻求最优路径, 而在传统的蚁群算法中, 初始信息素采用的是均匀分布, 因此每只蚂蚁的状态转移概率只取决于期望启发项, 而栅格间期望启发项差异较小因此初期算法因为缺少信息素引导具有盲目性, 且搜索速度较慢, 最终影响解的质量与收敛速度<sup>[14]</sup>。因此对初始化信息素进行非均匀分布处理, 提高栅格间信息素的差异能有效提高搜索效率。为避免改进后的初始信息素过度或者错误引导后续蚂蚁探索最优解, 信息素的非均匀分布仅针对区域进行差异处理而非单个栅格点, 即只提高有利区域内的初始信息素, 降低算法陷入局部最优的可能。由于最优路径多集中于起点  $S$  至目标点  $E$  的连线附近区域, 其区域范围受障碍物影响, 因此以起点  $S$  和目标点  $E$  连线作  $x$  轴、以连线中点作为原点建立坐标系, 以起点至终点距离  $d_{SE}$  为长轴, 全局优选区域距离  $d_{diff}$  为短轴建立椭圆, 其计算公式如式(8)。在该椭圆内的区域为全局优选区域  $R$ , 并提高优选区域  $R$  内的路径点初始信息素, 而不在该范围内的路径点则保持原初始信息素值  $\tau_0$ 。非均匀初始信息素分布规则如式(7)所示:

$$\tau_{ij}(0) = \begin{cases} C + \tau_0, & j \in R \\ \tau_0, & j \notin R \end{cases} \quad (7)$$

$$d_{diff} = (1 + \xi) \cdot \min(r, c) / 2 \quad (8)$$

式(7)中  $\tau_0$  为信息素初始值;  $C$  为全局优选区域信息素增量; 式(8)中  $d_{diff}$  与地图障碍物有关,  $\xi$  为环境障碍物占比,  $\xi$  越高, 则  $d_{diff}$  越大, 全局优选区域范围更广;  $r$  和  $c$  分别表示矩阵地图的行数与列数。

#### 3.2 改进信息素更新规则

信息素的更新包括对原有路径上信息素的挥发

与新增路径信息素的积累, 因此提出局部信息素分块优化策略对信息素的更新规则进行了改进。

在传统蚁群算法中, 所有的蚂蚁完成当前迭代后将对蚂蚁路径进行更新。路径的好坏评价取决于全局路径的长度, 而忽略了对局部区域路径的评价。图 1 中实线路径与虚线路径分表代表历史最优路径和当前迭代最优路径, 横坐标  $x \in [0, 5]$  时, 当前迭代最优路径更短; 而在  $x \in [5, 20]$  时, 历史最优路径更短, 因此提出局部信息素分块优化策略。由于算法初期并不知道路径将会经过哪些路径点, 无法利用具体坐标作为分割依据, 因此将矩阵地图按  $x$  坐标以一定间隔步长  $n$  纵向切割均匀划分为多个区域,  $n$  越小容易造成路径曲折较多, 不利于寻求最佳路径,  $n$  较大则达不到分块优化的效果, 如图 1 所示矩阵地图被以步长  $n = 5$  划为了四个区域, 有多个点都经过同一切分线时, 以最后经过点作为终止点。随后引入精英蚂蚁<sup>[15]</sup>的策略, 当所有蚂蚁完成当前迭代后, 对各区域内的最优局部路径信息素进行更新。全局最优路径本质上是由各个局部最优路径组合而成的, 对局部最优路径信息素的更新旨在引导后续蚂蚁行驶至该区域内的路径选择, 一旦蚂蚁行驶至该区域最优路径附近则能通过此策略更大概率选择最优路径点。但局部区域路径优劣如果仅从区域内路径长度考虑, 很容易导致区域内水平方向的路径被认为是最优而非更接近终点的总距离, 如式(9)所示。同时由于非均匀初始化信息素的改进, 对局部路径长度的选择也更有利。

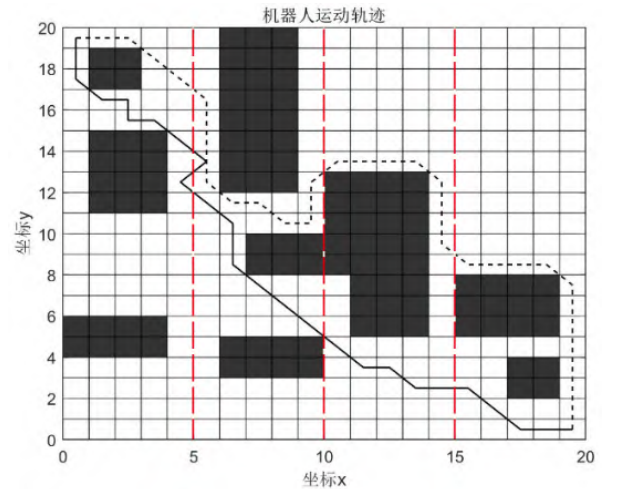


图 1 信息素分块优化

Fig.1 Pheromone block optimization

因为局部更新只是针对局部区域, 缺乏全局信息素引导, 因此对每个区域内最优路径更新后, 再对蚂



蚁的全局信息素进行更新。鉴于此前已经对局部最优路径进行了更新，为了避免信息素重复叠加，只对全局最优路径信息素进行信息素更新。局部最优路径更新规则与全局最优路径更新规则相同，如式(9)和式(10)所示。

$$L_{best} = L_{dist} + L_{ds} + L_{de} \quad (9)$$

$$\Delta \tau_{ij}(t) = \begin{cases} Q / L_{best}, & \text{路径}(i, j) \text{为全局最优路径或局部最优路径} \\ 0, & \text{其他} \end{cases} \quad (10)$$

式中：当计算局部最优路径信息素时， $L_{best}$  为各个区域内最优路径长度， $L_{dist}$  为各局部区域内最优路径段长度， $L_{ds}$  为路径段第一个路径点到起点距离， $L_{de}$  为路径段最后一个路径点到终点距离。在计算全局最优路径信息素时， $L_{ds}$  和  $L_{de}$  都为 0，则  $L_{best}$  则为全局最优路径长度。

如果只对最优路径进行更新，随着迭代次数增加，最优路径上的信息素累计越来越多，高浓度的信息素含量会导致即使迭代后期出现了比历史更优路径，这条新路径上释放的信息素仍然远远少于现有最优路径上的信息素，容易陷入局部最优且可能在迭代收敛时丢失该解<sup>[16]</sup>。针对该问题，在信息素更新规则中引入信息素增强因子  $r$ ，当出现比历史的更优路径时，通过增强因子提高此路径上的信息素，如式(11)所示。增强因子  $r$  随着当前迭代次数  $N_c$  的改变而发生改变，初期  $r$  值较小避免信息素过量叠加防止算法陷入局部最优，后期值较大加快收敛，公式为式(12)。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + (1+r)\Delta\tau_{ij}(t) \quad (11)$$

$$r = \begin{cases} e^{\frac{N_c-1}{N_{max}}}, & \text{如果 } L_{best} < L \\ 0, & \text{其他} \end{cases} \quad (12)$$

式中： $N_{max}$  为迭代总数， $L_{best}$  为当前局部区域或全局最优路径， $L$  为历史最优路径。

### 3.3 改进状态转移概率

反向学习的主要思想在于对一个可行解，计算同时评估其反向解，对比后选择较优解。由于蚁群算法的正反馈机制不适用于完全解<sup>[17]</sup>，因此反向学习更适用于扩展蚁群算法解空间而非评估反向解，通过反向学习增加解空间覆盖率达到提高求解质量的效果。文献[12]提出基于反向学习的蚁群系统用于解决 TSP 问题，文献[16]提出反向蚁群优化算法解决故障诊断监控参数优化问题。

参考以上文献，引入反向信息素改进蚁群算法状

态转移概率，用于路径规划寻求最优解。根据反向学习的定义，反向信息素  $\overline{\tau}_{ij}$  计算如下式：

$$\overline{\tau}_{ij} = \begin{cases} \tau_{ij}(0) + \frac{1}{L_{gbest}} - \tau_{ij}, & \text{if } N_c > 1 \\ \tau_{ij}(0), & \text{if } N_c = 1 \end{cases} \quad (13)$$

式中：此时的信息素  $\tau_{ij}$  为局部和全局更新完毕后的信息素，分布优化策略已结束，因此式(13)中  $L_{gbest}$  为全局最优路径长度。由于第一次迭代过程中不会产生  $L_{gbest}$ ，因此第一次迭代时  $\overline{\tau}_{ij}$  的值为初始信息素。其中确定反向信息素的值为初始信息素  $\tau_{ij}(0)$  和最优路径长度  $L_{gbest}$ ，因为信息素的积累由初始信息素和最优路径长度共同决定。

当蚂蚁  $m$  在  $t$  时刻进行路径点选择时，其转移概率为：

$$P_{ij}^m(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed_m} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & \text{if } \lambda < \lambda_c \\ \frac{[\overline{\tau}_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed_m} [\overline{\tau}_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & \text{if } \lambda > \lambda_c \\ 0, & j \notin allowed_m \end{cases} \quad (14)$$

$$\eta_{ij}(t) = 1/d(j, e) \quad (15)$$

式中： $\lambda_c$  为反向概率， $\lambda_c \in (0, 1)$ ， $\lambda$  为随机数；当  $\lambda < \lambda_c$  时，使用反向信息素计算选择概率，当  $\lambda > \lambda_c$  时，选择概率为原公式计算其值。其中  $d(j, e)$  为下一节点  $j$  到终点  $e$  的欧氏距离。

### 4 改进蚁群算法的应用步骤

步骤 1 建立矩阵栅格地图，设置起始位置起点  $S$  与目标点  $E$ ；初始化参数，包括：信息素启发因子  $\alpha$ ，期望启发式因子  $\beta$ ，挥发系数  $\rho$ ，蚂蚁总数  $W$ ，迭代总数  $N_{max}$ ，信息素强度值  $Q$ ，反向概率  $\lambda_c$ ，信息素初始值  $\tau_0$  与初始信息素增量  $C$ 。

步骤 2 初始信息素非均匀处理。通过公式(8)确定  $d_{diff}$ ，计算全局优选区域  $R$  的范围，并根据式(7)对初始信息素非均匀处理得出  $\tau_{ij}(0)$ 。

步骤 3 路径点选择。将  $W$  只蚂蚁放置于起点处，初始化禁忌表、路径点、路径长度。并将起点加入禁忌

表中。根据公式(14)和(15)计算每一只蚂蚁下一路径点选择概率,通过轮盘赌选择下一个节点,并将该节点加入禁忌表中。

步骤4 判断蚂蚁是否到达了终点,如果没有,则返回到步骤3直到到达目标点为止。否则执行步骤5。

步骤5 局部信息素分块更新。当所有蚂蚁完成当前迭代后,对所有地图进行分块处理,并分别计算每个区域的最佳局部路径,按式(9)、(10)和式(11)分别对每个区域进行信息素更新,此时 $L_{best}$ 为每个区域最优局部解。当出现更优解时,按照式(12)计算增强因子并带入式(11)中进行计算, $L$ 为每个区域历史最优路径。

步骤6 全局信息素更新。在局部区域信息素更新完毕后,按式(9)、(10)和式(11)对全局信息素进行更新。此时 $L_{best}$ 全局最优解。当出现更优解时,按照式(11)计算增强因子并代入式(11)中进行计算, $L$ 为全局历史最优路径。

步骤7 反向信息素更新。按照式(13)对反向信息素进行更新,式中 $L_{best}$ 为全局最优解。

步骤8 判断是否达到最大迭代数。如果达到最大迭代数则输出当前最优全局路径,如果没有,则清空禁忌表,并将迭代次数加一,重复步骤3到步骤7,直到达到最大迭代数。

## 5 模拟实验和分析

为验证改进蚁群算法的有效性,在 MATLAB 2018b 上进行仿真实验,操作系统为 Win10 (64 bit) 操作系统,Core™i5-10210U 处理器,16GB 运行内存。仿真实验采用两组对比实验,分别为 20\*20、30\*30 的栅格地图,并在相同地图下将本文改进蚁群算法与传统蚁群算法、文献[8]算法结果进行对比分析。

### 5.1 参数选取

蚁群算法参数多利用经验值取值,并利用多次对比结果选取最佳参数,但算法参数还涉及局部分块优化的步长  $n$ ,因此基于以上参数,取分割步长  $n = 1, 2 \dots c$  ( $c$  为地图的列数) 在随机 20\*20、30\*30 地图进行实验。其余仿真参数设置为:  $\alpha = 1$ ,  $\beta = 7$ ,  $\rho = 0.2$ ,  $W = 50$ ,  $N_{max} = 100$ ,  $\lambda_c = 0.05$ ,  $Q = 1$ ,  $\tau_0 = 1$ ,  $C = 3$ 。

从 20\*20 仿真结果图 2(a)来看  $n \in [4, 6]$  时,即

将地图分为 4~5 个区域效果较好,当  $n$  较小时,迭代次数比不分块更高,算法信息素分布分散,难收敛,无法找到较优值,因此路径长度也比不分块更长。随着  $n$  的增加,虽然迭代收敛次数有时能较快收敛但是算法也容易陷入局部最优。从 30\*30 的仿真结果图 2(b)也可以看出这一点,当  $n \in [4, 7]$  时,对应将地图分为 4~5 效果较好,  $n$  过长或过短都将影响收敛与最终结果,造成收敛速度与局部最优解的矛盾。基于此,对 20\*20 和 30\*30 都分为四块,20\*20 地图取  $n = 5$ , 30\*30 地图取  $n = 6$ 。

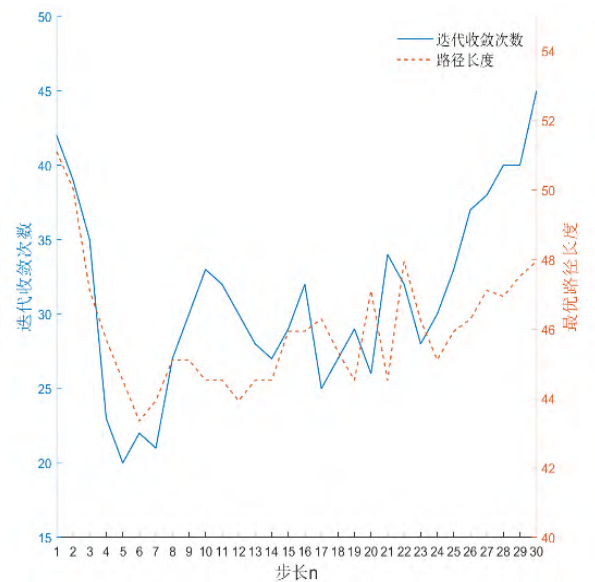
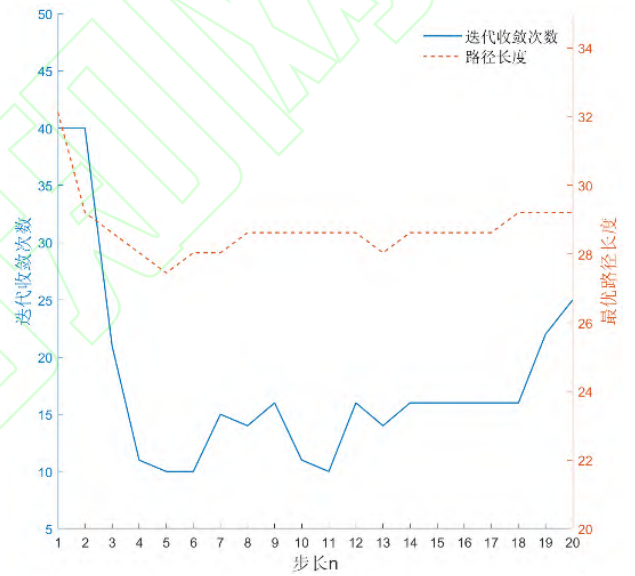


图2  $n$  取值对结果的影响

Fig.2 The influence of  $n$  value on the result

5.2 20\*20 地图实验仿真

在 20\*20 栅格地图下，分别对传统蚁群算法、文献[8]、本文改进算法进行仿真，路径仿真分别如图 3 和图 4 所示。最终实验数据如表 1 所示：传统算法路径规划长度为 32.866，文献[8]规划长度为 33.452，本文改进算法规划长度为 28.038，本文算法较另外两者算法路径规划长度更短，拐点更少，对比传统算法与文献[8]算法，路径长度分别缩短了 14.6% 和 16.2%，拐点分别减少 66.6% 和 75%。传统蚁群算法收敛次数为 21 次,文献[8]为 5 次，本文改进算法收敛次数为 12 次。虽然文献[8]收敛次数为 5，但是其求解质量明显低于本文改进算法，这也说明本文在算法收敛速度与求解全局最优解上平衡的更好。传统蚁群算法在迭代后期丢失最优解，最后收敛为局部最优值，而由于本文引入了增强因子  $r$  避免了这一问题，保留了最优解，避免了陷入局部最优。

为进一步验证各个改进环节的有效性，分别对每个环节改进和本文整体改进在 20\*20 栅格地图下进行独立实验，实验结果如表 2 所示，初始化与分块策略这一改进环节对比传统蚁群算法有效减少了迭代次数的同时找到了更优解。反向学习信息素的引入对比传统算法也有有效的提高了算法的全局寻优能力，避免陷入了局部最优，但也由于缺乏信息素引导导致迭代次数较多。而本文的改进方法综合以上步骤，结合了收敛速度的提升与全局寻优能力上的改进，能在较少的迭代次数内找到更优解，同时曲线更平滑路径拐点

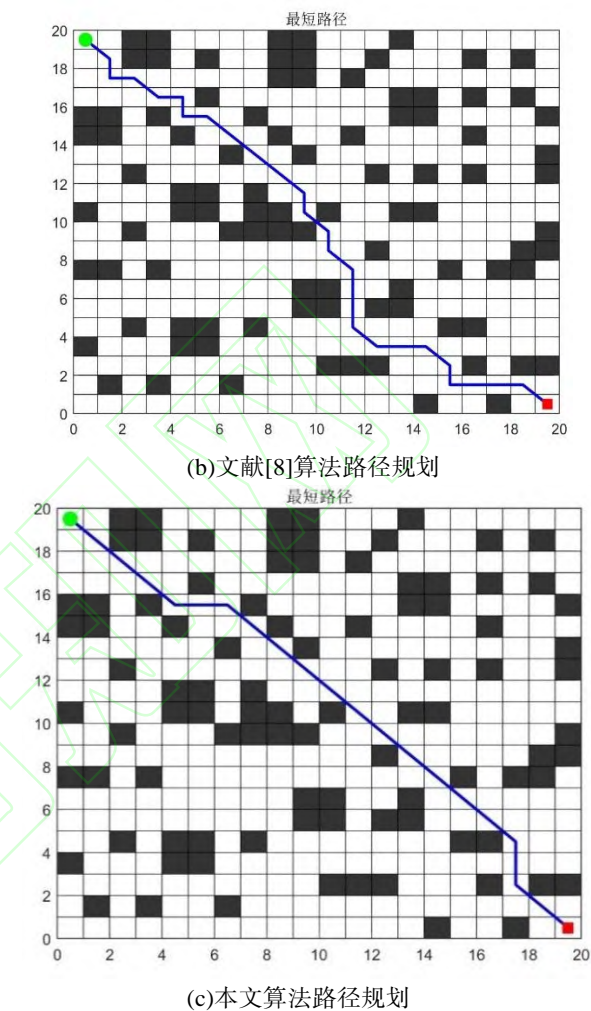
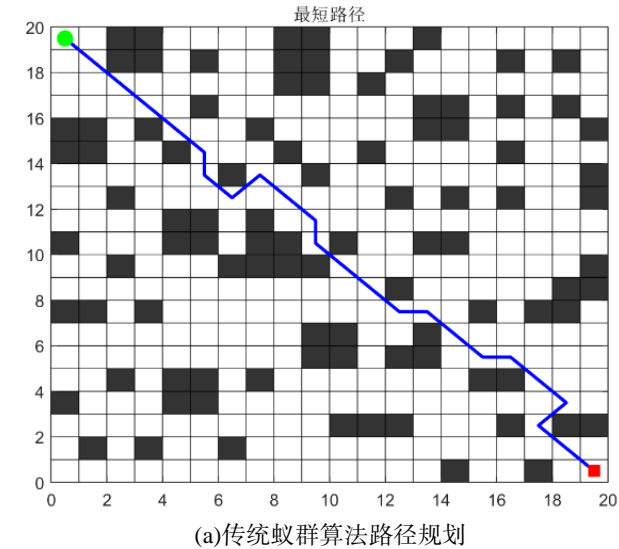


图 3 20\*20 环境下四种算法仿真结果  
Fig.3 Simulation results of four algorithms in the 20\*20 environment

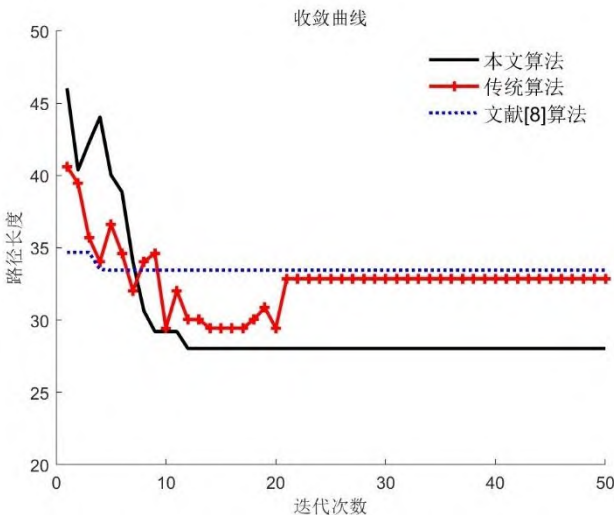


图 4 各算法路径收敛曲线  
Fig.4 Convergence curve of each algorithm path



表 1 20\*20 环境各个算法结果对比

Table 1 Comparison of the results of various algorithms in the 20\*20 environment

路径性能指标	传统蚁群算法	文献[8]算法	本文算法
路径长度	32.866	33.452	28.038
拐点个数	12	16	4
收敛迭代次数	21	5	12

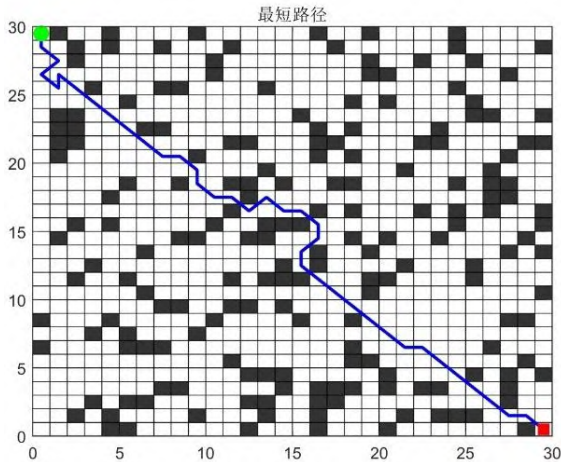
表 2 20\*20 环境各改进环节结果对比

Table 2 Comparison of the results of improvement steps in the 20 \* 20 environment

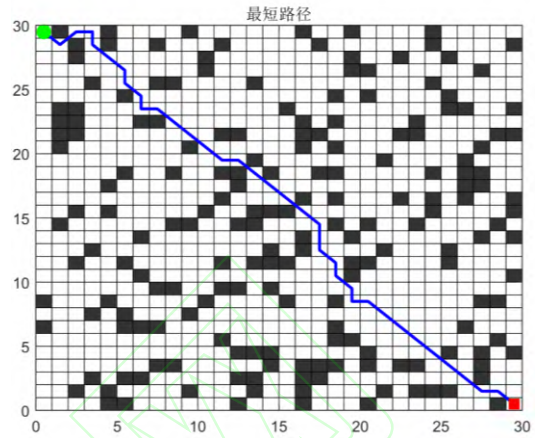
路径性能指标	非均匀初始化 与信息素更新	反向信息素	本文算法
路径长度	28.452	28.629	28.038
拐点个数	10	6	4
收敛迭代次数	6	12	12

### 5.3 30\*30 地图实验仿真

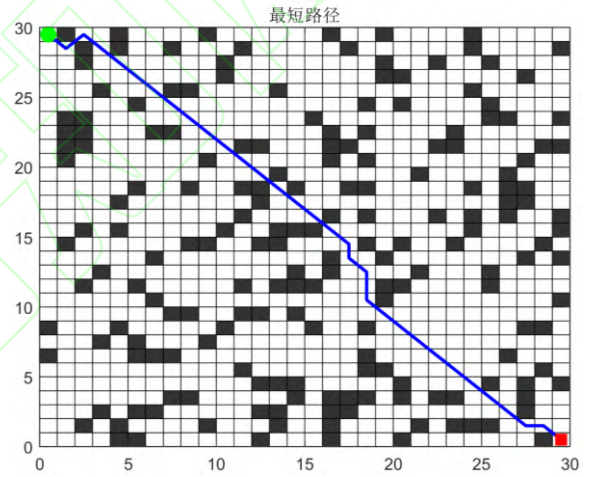
在 30\*30 复杂环境下, 实验仿真结果如图 5 到图 6 所示, 随着地图的复杂度增加, 三种算法收敛迭代次数都有所增加, 传统蚁群算法收敛曲线波动较大, 算法稳定性较差, 且依然陷入了局部最优。文献[8]虽然对比传统算法结果更优, 但由于该算法仅针对于最优解拓展出来的小区域进行局部优化, 缺少全局探索, 因此也陷入局部最优, 最终实验数据如表 3 所示, 综合指标来看, 本文改进算法在复杂环境下也有不错效果。最优路径长度上来看, 本文改进算法路径长度对比传统蚁群算法和[8]分别缩短了 14.1% 和 7.9%, 拐点个数减少 63.3% 和 60%, 迭代次数减少 23% 和 39.3%。在 30\*30 地图上也对各个步骤的改进进行了验证如表 4, 其结果与 20\*20 地图结果类似, 再一次验证了本文算法的有效性与优越性。



(a)传统蚁群算法路径规划



(b)文献[8]算法路径规划



(c)本文算法路径规划

图 5 30\*30 环境下四种算法仿真结果

Fig.5 Simulation results of four algorithms in the 30\*30 environment

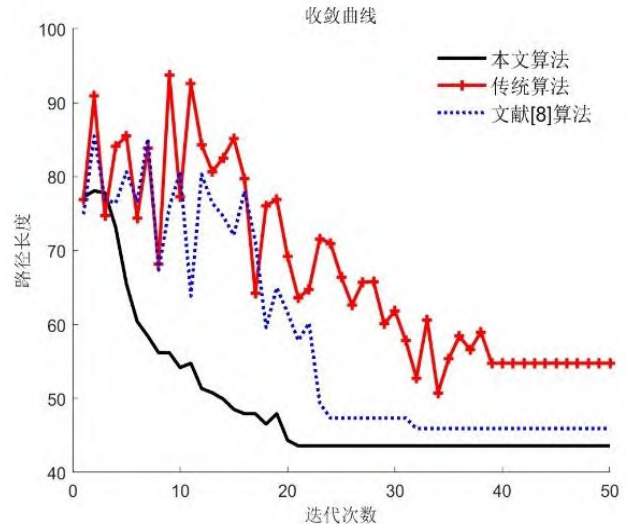


图 6 各算法路径收敛曲线

Fig.6 Convergence curve of each algorithm path



表3 30\*30 环境各个算法结果对比

Table 3 Comparison of the results of various algorithms in the 30\*30 environment

路径性能指标	传统蚁群算法	文献[8]算法	本文算法
路径长度	50.764	47.350	43.592
拐点个数	22	20	8
收敛迭代次数	39	33	20

表4 30\*30 环境各改进环节结果对比

Table 4 Comparison of the results of improvement steps in the 30 \* 30 environment

路径性能指标	非均匀初始化 与信息素更新	反向信息素	本文算法
路径长度	44.178	45.936	43.592
拐点个数	15	10	8
收敛迭代次数	19	29	20

## 6 结束语

本文针对传统蚁群算法存在易陷入局部最优、收敛速度慢等不足,提出改进蚁群算法。首先对根据地图的障碍物占比率以及矩阵地图大小构建全局优选区域,对初始信息素进行差异化处理,提高区域内信息素初始浓度而非具体栅格点信息素浓度,避免陷入局部最优的同时提高了收敛速度;利用局部信息素分块优化策对矩阵地图进行分块处理,分别对各个子区域进行寻优并更新最优路径信息素,后对全局路径进行寻优,更新最优路径信息素。为避免只更新最优路径信息素,丢失更优解,在信息素更新公式中引入增强因子,保留更优解,避免收敛于次优解。最后应用反向学习优化信息素,改进状态选择概率,拓展了解空间,提高了全局搜索能力。实验结果表明,改进后的算法明显提高了收敛速度,寻优能力更强。对初始化非均匀分布的处理、信息素更新规则的改进以及反向学习在蚁群路径规划上的应用对比传统算法均有明显的改进,进一步的验证了算法的有效性与优越性。

## 参考文献:

- [1] 朱大奇,颜明重.移动机器人路径规划技术综述[J].控制与决策,2010,25(7):961-967.  
ZHU D Q,YAN M C.Survey on technology of mobile robot path planning[J]. Control and Decision, 2010, 25(7): 961-967.
- [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):903-910.  
ZHAO X,WANG Z,HUANG C K,et al.Mobile Robot

Path Planning Based on an Improved A\* Algorithm[J]. Robot,2018,40(6):903-910.

- [3] 张希闻,肖本贤.基于人工势场法的移动机器人动态路径规划[J].山东农业大学学报(自然科学版),2018,49(6):937-940.  
ZHANG X W,XIAO B X.The planning for dynamic path of a mobile robot based on artificial potential field method[J].Journal of Shandong Agricultural University(Natural Science Edition),2018,49(6):937-940.
- [4] 刘二辉,姚锡凡,蓝宏宇,等.基于改进遗传算法的自动引导小车路径规划及其实现平台[J].计算机集成制造系统,2017,23(3):465-472.  
LIU E H,YAO X F,LAN H Y,et al.AGV dynamic path planning based on improved genetic algorithm and its implementation[J].Computer Integrated Manufacturing Systems,2017,23(3):465-472.
- [5] 王娟,吴宪祥,郭宝龙.基于改进粒子群优化算法的移动机器人路径规划[J].计算机工程与应用,2012,48(15):240-244.  
WANG J,WU X X,GUO B L.Robot path planning using improved particle swarm optimization[J].Computer Engineering and Applications,2012,48(15):240-244.
- [6] Dorigo M, Birattari M, Stutzle T. Ant colony optimization[J]. IEEE Computational Intelligence Magazine, 2006, 1(4): 28-39.
- [7] 江明,王飞,葛愿,等.基于改进蚁群算法的移动机器人路径规划研究[J].仪器仪表学报,2019,40(2):113-121.  
JIANG M,WANG F,G E Y,et al. Research on path planning of mobile robot based on improved ant colony algorithm[J].Chinese Journal of Scientific Instrument, 2019, 40(2): 113-121.
- [8] 许凯波,鲁海燕,黄洋,等.基于双层蚁群算法和动态环境的机器人路径规划方法[J].电子学报,2019,47(10):2166-2176.  
XU K B,LU H Y,HUANG Y,et al.Robot path planning based on double-layer ant colony optimization algorithm and dynamic environment[J].Acta Electronica Sinica,2019,47(10):2166-2176.
- [9] 陈志,韩兴国.改进蚁群算法在移动机器人路径规划上的应用[J].计算机工程与设计,2020,41(8):2388-2395.  
CHEN Z,HAN X G.Application of improved ant colony algorithm in mobile robot path planning[J].Computer Engineering and Design,2020,41(8):2388-2395.
- [10] 刘建华,杨建国,刘华平,等.基于势场蚁群算法的移动机器人全局路径规划方法[J].农业机械学报,2015,46(9):18-27.  
LIU J H,YANG J G,LIU H P,et al.Robot global path planning based on ant colony optimization with artificial potential field [J].Transactions of the Chinese Society for

- 
- Agricultural Machinery,2015,46(9):18-27.
- [11] Wang L,Luo C,Li M,et al.Trajectory planning of an autonomous mobile robot by evolving ant colony system[J]. International Journal of Robotics and Automation, 2017, 32(4): 406-413.
- [12] Tizhoosh H R.Opposition-based learning: A new scheme for machine intelligence [C].Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automatic, 2005: 695-701.
- [13] 马海平,阮谢永,金宝根.反向蚁群算法在故障监控中的应用[C].中国自动化学会控制理论专业委员会,第二十九届中国控制会议论文集,2010,4:3895-3898.
- MA H P,RUAN X Y,JIN B G.Oppositional ant colony optimization algorithm and its application to fault monitoring[C].Plenary Speakers at the 29th Chinese Control Conference,2010,4:3895-3898.
- [14] 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.
- FENG S F,LEI Q,WU W L,et al.Mobile robot path planning based on adaptive ant colony algorithm[J].Computer Engineering and Applications,2019,55(17):35-43.
- [15] Yousefikhoshbakht M, Didehvar F, Rahmati F. An efficient solution for the VRP by using a hybrid elite ant system[J]. International Journal of Computers Communications & Control,2014,9(3):340-347.
- [16] Yang H,Qi J,Miao Y,et al. A new robot navigation algorithm based on adouble-layer ant algorithm and trajectory optimization[J].IEEE Transactions on Industrial Electronics,2019,66(11) :8557-8566.
- [17] Alice R Malisia, Hamid R Tizhoosh. Applying opposition-based ideas to the ant colony system [C].Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automatic, 2005: 695-701.