doi:10.6041/j.issn.1000-1298.2024.02.042

基于动态扩展邻域蚁群算法的移动机器人路径规划

潘玉恒　奥日格拉　鲁维佳　丛　佳　王世通　陈　阳
（天津城建大学计算机与信息工程学院，天津 300384）

摘要：针对蚁群算法易陷入局部最优、路径转折点多、收敛速度慢的问题，提出一种基于动态扩展邻域蚁群算法（Dynamic extended neighbourhoods ant colony optimization，DENACO）。在蚂蚁搜索方式上采用动态扩展邻域方法，并定义新的信息素计算方式和增量规则，在取得更优收敛路径长度的同时，减少路径转折点数量及路径节点数量；引入自适应调整因子改进启发函数，提高算法的全局搜索能力，并设定迭代阈值，提升算法的收敛速度；提出一种路径节点双优化策略，对规划好的路径进一步优化，提高路径综合质量。不同复杂度及不同规模栅格地图中的仿真实验表明，DENACO 算法所规划的路径更优，路径转折点数量减少，收敛速度加快，路径节点数量明显减少，表明算法具有更高的可行性和适用性。

关键词：移动机器人；蚁群算法；路径规划；动态扩展邻域；自适应启发函数
中图分类号：TP242.6　　**文献标识码**：A　　**文章编号**：1000-1298(2024)02-0423-10　　**OSID**：

Path Planning of Mobile Robots Based on Dynamic Extended Neighbourhoods Ant Colony Algorithm

PAN Yuheng　Aorigela　LU Weijia　CONG Jia　WANG Shitong　CHEN Yang
(*School of Computer and Information Engineering*, *Tianjin Chengjian University*, *Tianjin* 300384, *China*)

Abstract: To solve the problems of ant colony algorithm in complex grid environment, such as local optimization, many turning points and slow convergence, dynamic extended neighbourhoods ant colony optimization (DENACO) algorithm was proposed. Firstly, the method of dynamic extended neighborhoods was applied in the ant search mode to obtain the optimal convergence path length and reduce the number of inflection points and the number of path nodes. Meanwhile, a computational method and increment rule of pheromone were defined to reduce space costs, and the upper and lower limits of pheromone were set to avoid premature convergence of the algorithm to local optimality. Secondly, the adaptive adjustment factor and target point factor were introduced into the heuristic function, and a weight coefficient was set to improve the global search ability of the algorithm. Moreover, an iteration threshold of the algorithm was set. When the iteration exceeded the threshold, the pheromone concentration factor and heuristic factor values were updated to improve the convergence speed of the algorithm. Finally, a double optimal strategy of nodes of path was proposed. Two optimization methods were used to further optimize the planned path, and the best was taken as the final optimization result to improve the comprehensive quality of the path. Simulation experiments on raster maps of different complexities and scales showed that compared with the traditional ant colony algorithm and other comparison algorithms, the path planned by DENACO algorithm was superior. It had a shorter path length, reduced number of inflection points, accelerated convergence speed, and significantly fewer path nodes. These results indicated that the DENACO algorithm was highly feasible and applicable.

Key words: mobile robot; ant colony algorithm; path planning; dynamic extended neighbourhoods; adaptive heuristic function

收稿日期：2023-07-04　修回日期：2023-09-21
基金项目：国家自然科学基金项目（62204168）、天津市科技计划项目（20YDTPJC00160、21YDTPJC00780）、天津市教委科研计划项目（2019KJ101、2017SK027）、天津市研究生科研创新项目（2022SKYZ033）和天津城建大学教育教学改革与研究重点项目（JG-ZD-22035、JG-ZD-22038）
作者简介：潘玉恒（1978—），女，副教授，博士，主要从事人工智能和信号处理等研究，E-mail：panyuheng@tju.edu.cn
通信作者：鲁维佳（1980—），男，讲师，主要从事无人驾驶技术和嵌入式系统等研究，E-mail：luweijia@tcu.edu.cn

0 引言

随着现代产业向智能化推进,移动机器人在各领域得到更广泛的应用[1-3]。路径规划作为移动机器人自主导航的关键技术,能够为机器人规划出一条从起点到终点的无碰撞路径。传统移动机器人路径规划算法包括 Dijkstra 算法[4]、A* 算法[5]、人工势场法[6]、动态窗口法[7-8]等,具有简单、易于实现的优点,在解决简单环境下路径规划问题时具有一定适用性。相对于传统路径规划算法,智能仿生算法,如遗传算法[9]、粒子群算法[10]、蚁群算法(Ant colony algorithm,ACO)[11-12]等,在处理大规模复杂环境下的路径规划问题时表现出更高的可行性。

蚁群算法作为一种随机启发式搜索的群智能算法,对解决路径规划问题具有很强的适用性[13-15],但是蚁群算法存在易陷于局部最优、收敛速度慢和路径转折点较多等问题[16-17]。针对搜索方式的改进有:文献[18]以无限步长的搜索方式寻找路径,拓宽了蚂蚁视野,缩短了搜索路径长度。文献[19]对蚂蚁搜索方式由 8 方向 8 邻域扩展到 16 方向 24 邻域,扩大了蚂蚁搜索范围,获得了更优路径。搜索方向及步长的增加会使蚂蚁有更高的概率偏离目标点,使算法容易陷入局部最优。同时,搜索邻域的大量增加,也会使单步搜索的计算量倍长。针对启发式函数的改进有:文献[20]为了平衡算法的全局搜索能力和收敛速度,在启发式信息函数中引入自适应调整因子。文献[21]在启发式信息函数中加入下一节点与目标点的距离,提升算法早期搜索的全局性,同时引入阻尼系数,使收敛速度得到提高。通过改进启发式函数能够在避免算法陷入局部最优的同时,获得较快的收敛速度。针对信息素更新方式的改进有:文献[22]设计了次优路径抑制因子,将其用于信息素的更新公式中,且仅对较优路径进行信息素更新,避免算法陷入局部最优。文献[23]改进了最大最小蚂蚁系统,并将信息素设置上下限,防止算法过早收敛。对于传统蚁群算法的信息素的改进方法目前局限于固定邻域的搜索方式,针对其它搜索方式的信息素改进研究较少。针对算法规划的路径进行优化:文献[24]提出了一种双层蚁群算法,由两个独立连续运行的蚁群算法组成,用转折点优化算法对规划好的路径进行优化,减小了规划路径长度。文献[25]对改进后的蚁群算法生成的路径进行进一步的优化,去掉不必要的转折点,最终使得路径长度更短。删去路径节点可以有效缩短路径长度,但面对更复杂多变的场景,单一的删除路径节点方式往往得不到最优解。

针对以上改进算法存在的问题,本文提出一种动态扩展邻域蚁群算法。蚂蚁在搜索过程中向外扩展一个与上一步的搜索方向相同的邻域,减少蚂蚁搜索过程中产生的路径节点数量及路径转折点数量,解决算法易陷入局部最优问题。设计自适应启发式函数,在迭代前期引导蚂蚁走向目标点,而不是周围距离更近的栅格,在迭代超过阈值后改变信息素因子和启发式因子,使算法加速收敛。针对动态扩展邻域的搜索方式,使用新的信息素计算方式和信息素增量规则,使空间开销保持不变。提出路径节点双优化策略,对规划好的路径进行进一步优化,以提高路径平滑度,缩短路径长度。

1 问题描述

1.1 环境建模

为了模拟实际环境,采用栅格法建立 $M \times M$($M=10$)二维栅格图模拟机器人运行环境。网格按照由上到下、由左到右的顺序进行编号,编号分别为 1,2,…,100。如图 1 所示,其中黑色栅格为障碍物,白色栅格为自由栅格,蚂蚁只能在自由栅格中移动。

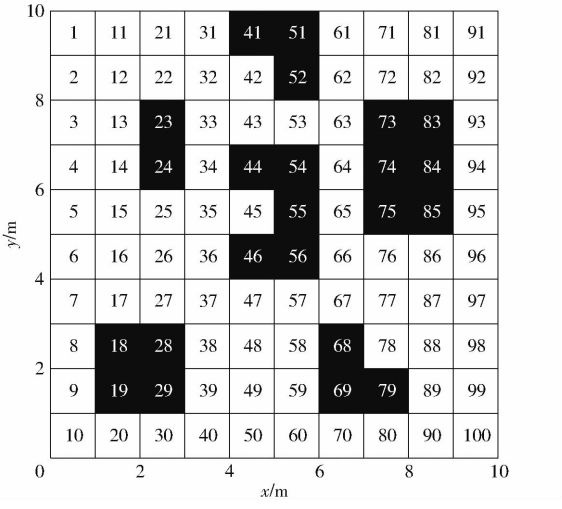


图 1 栅格环境图

Fig. 1 Grid map

栅格地图中各栅格坐标表示为其中心点坐标(x,y),栅格坐标和栅格编号之间的关系为

$$\begin{cases} x = \mathrm{ceil}\left(\dfrac{N}{M} - 1\right) + 0.5 \\ y = \begin{cases} M + 0.5 - \mathrm{mod}(N,M) & (\mathrm{mod}(N,M) \neq 0) \\ 0.5 & (\mathrm{mod}(N,M) = 0) \end{cases} \end{cases} \tag{1}$$

式中 mod( )——取模运算函数

N——栅格编号

M——地图规模

ceil( )——向上取整运算函数

1.2　蚁群算法

蚂蚁在走过的路径上释放出信息素，后续蚂蚁通过路径上信息素浓度选择行走方向，通常较短路径留下的信息素较多，因此蚁群会不断向此类路径聚集，最终找到全局最优路径。

1.2.1　状态转移概率

状态转移概率函数由信息素浓度和启发式信息两部分决定。在已知静态栅格环境中，t 时刻栅格 i 上的蚂蚁 $k(1,2,\cdots,m)$ 将移动到栅格 j。首先计算周围可移动自由栅格的转移概率 $p_{ij}^{k}(t)$，计算方法为

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{u\in C}\tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)} & (j\in C)\\ 0 & (j\notin C)\end{cases}\tag{2}$$

其中

$$\eta_{ij}(t)=\frac{1}{d_{ij}}\tag{3}$$

式中　C——蚂蚁下一步可移动的邻域集合

$\tau_{ij}(t)$——t 时刻从栅格 i 到栅格 j 路径上的信息素浓度

$\eta_{ij}(t)$——t 时刻从栅格 i 到栅格 j 的启发式信息

α——信息素浓度因子

β——启发式信息因子

d_{ij}——栅格 i 到栅格 j 的欧氏距离

然后使用轮盘赌来确定下一个移动栅格 j。

1.2.2　信息素更新规则

在蚁群探索路径过程中，同一条路径走过的蚂蚁越多，信息素积累也就越多，但这些信息素存在自然挥发的现象，其计算方法为

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}(t)\tag{4}$$

式中　ρ——信息素挥发速度因子

$1-\rho$——信息素残留因子

$\Delta\tau_{ij}(t)$——本轮循环结束后信息素的增量

通常使用蚁周模型进行更新，其计算方法为

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{L_k} & (tour(t,j)\in tourk)\\ 0 & (tour(t,j)\notin tourk)\end{cases}\tag{5}$$

式中　Q——信息素强度因子

L_k——蚂蚁在此循环中走过路径的总长度

2　DENACO 算法

2.1　动态扩展邻域搜索方式

移动步长是蚁群算法中最重要的参数，在传统蚁群算法中，通常邻域搜索范围为 7 个栅格，如图 2a 所示，图中红色圆点为蚂蚁起始点，蓝色圆点为当前位置，黄色栅格为下一步可选择移动的邻域栅格。使用该移动策略，移动步长为 1，而每一步最多需要计算附近 7 个栅格的转移概率，同时固定步长存在灵活性不高的缺点[26]。

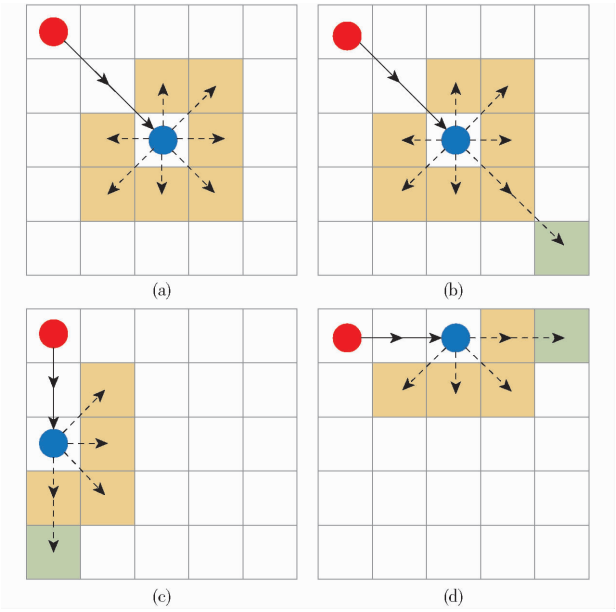


图 2　动态扩展邻域选择方法图

Fig. 2　Methods of selection of dynamic extended neighbourhoods

以图 2a 中搜索方式为基础，在蚂蚁行进的方向上扩展一个邻域，邻域位置与当前位置间隔 1 个栅格。如图 2b ~ 2d 所示，图中绿色栅格为扩展搜索邻域位置。在利用轮盘赌进行选择时，蚂蚁可将搜索邻域扩展为 2 步长。

单次搜索邻域数量的增加能够使算法总的搜索次数减少。在图 3 所示栅格场景中，红色栅格表示起点 S，绿色栅格表示终点 E，绿色路径为传统蚁群搜索方式所规划路径，蓝色路径为动态扩展邻域搜索方式所规划路径，两条路径长度相等。其中，绿色路径搜索次数为 12 次，而蓝色路径搜索次数仅为 9 次，因此，动态扩展邻域搜索方式产生的路径节点更

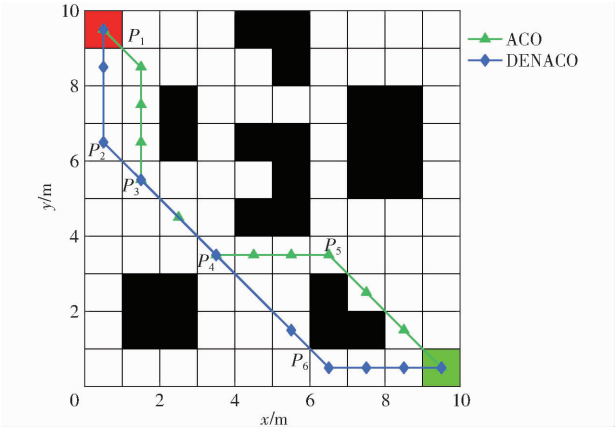


图 3　动态扩展邻域优势图

Fig. 3　Advantages of dynamic extended neighbourhoods

少。在实际的机器人移动过程中,更少的路径节点数量可以减少向机器人下达运动命令的次数,确保机器人移动的平滑性[27]。

基本蚁群开始选择向 P_1 方向行走,而到达 P_3 后不得不选择继续转向,相比蓝色路径多一个转折点。P_4 为两条路径的分叉点,蓝色路径在该点时通过扩展的邻域增加了沿原方向行走的概率,最终选择直线行走,而绿色路径选择转向而行,产生新的路径转折点,最终使蓝色路径相比绿色路径减少 2 个转折点 P_3 及 P_4。在实际机器人移动过程中,转向会导致更多的时间及能量消耗,而本文寻路方法减少了转折点数量,使机器人更快达到目标点,且转折点的减少也会使路径的安全性得到提升。

当蚂蚁以固定步长移动时,若移动方向根据概率选择偏离了目标点,则步长越长将导致蚂蚁偏离目标点的距离越远,使算法容易陷入局部最优。而在动态扩展邻域搜索方式中,蚂蚁按照转移概率函数进行邻域选择,以一定概率选择扩展邻域为移动目标,避免了由于步长的增加所带来的向偏离目标点方向移动的问题,防止陷入局部最优。

2.2 改进转移概率函数

2.2.1 自适应启发式信息

启发式信息是转移概率函数中的一部分,在传统蚁群算法中,启发式信息使蚂蚁走向距离当前位置更近的栅格,忽略了目标点的所在方向,不利于引导蚂蚁向目标点前进。并且在搜索后期,启发式信息对路径的选择有影响,若保持不变会导致收敛速度过慢。因此,本文在启发式信息中加入邻域栅格到目标点的距离因素,增强搜索全局性,并引入自适应调整因子,随着迭代次数的增加,削弱启发式信息值,增强算法的全局搜索能力,防止算法陷入局部最优,其计算方法为

$$\eta_{ij}(t)=\mu\left(\omega_1\frac{1}{d_{ij}}+\omega_2\frac{1}{d_{jg}}\right) \tag{6}$$

其中

$$\mu=1-\gamma\frac{N_c}{N_{\max}} \tag{7}$$

$$\omega_1+\omega_2=1$$

式中 d_{jg}——周围邻域栅格到目标点的欧氏距离

ω_1、ω_2——距离权重

μ——启发式信息自适应调整因子

N_c——当前迭代次数

$N_{\max}$——最大迭代次数

γ——自适应调整因子的影响常量

信息素浓度因子 α 和启发式信息因子 β 对算法的收敛速度影响至关重要。在一定范围内,α 越大,蚂蚁选择已探索路径的概率越大,算法加速收敛。β 越大,蚂蚁更偏向选择局部最短路径,算法收敛速度加快,但易陷入局部最优[28]。

为进一步平衡路径长度与迭代次数的关系,设置迭代阈值 T,当迭代次数小于 T 时,μ 按照式(7)进行更新,当迭代次数大于或等于 T 时,μ 停止更新,同时对 α 和 β 进行更新,将其设为 K_1 和 K_2。在算法前期取得最优搜索结果后,提高 α 和 β,以加快算法收敛速度。

2.2.2 动态扩展邻域信息素计算方式

传统蚁群算法只需存储每个栅格到周围栅格路径上的信息素浓度,其空间复杂度为 $O(M\times M)$,而增加搜索邻域需要扩充信息素矩阵,若以无限步长为搜索方式,信息素矩阵的空间复杂度甚至接近 $O((M\times M)^2)$,带来过高的空间开销,栅格规模越大,劣势越为突出。

根据动态扩展邻域的搜索方式提出了一种新的信息素计算方式,即

$$\tau_{ij}(t)=\lambda_1\tau_{ir}(t)+\lambda_2\tau_{rj}(t) \tag{8}$$

其中 $\lambda_1+\lambda_2=1$

式中 $\tau_{ir}(t)$——当前位置到临近邻域路径上的信息素浓度

$\tau_{rj}(t)$——临近邻域到扩展邻域路径上的信息素浓度

λ_1、λ_2——信息素浓度权重

在使用该信息素计算方式时,信息素矩阵仍只需存储每个栅格到周围栅格路径上的信息素,不需要扩充额外空间,其空间复杂度可保持为 $O(M\times M)$ 不变。选择权重可以避免由步长增加导致可能存在的路径偏离目标点的问题,防止陷入局部最优。

2.3 动态扩展邻域信息素增量规则

基于动态扩展邻域的信息素增量规则为

$$\Delta\tau_{ir}(t)=\lambda_1\frac{Q}{L_k} \tag{9}$$

$$\Delta\tau_{rj}(t)=\lambda_2\frac{Q}{L_k} \tag{10}$$

式中 $\Delta\tau_{ir}(t)$——当前位置到临近邻域路径上的信息素增量

$\Delta\tau_{rj}(t)$——临近邻域到扩展邻域路径上的信息素增量

当蚂蚁向扩展邻域移动时采用该信息素增量规则进行信息素更新。

路径上信息素的积累会影响后续蚂蚁对于路径的选择。在某条路径上,若信息素浓度过高,会使大量蚂蚁集中在该条路径上,若信息素浓度过低,会使蚂蚁完全放弃该条路径,导致整个蚁群的路径探索

范围缩小，本文对路径上的信息素设置上下限。对信息素进行最大值最小值限制将提升算法的路径多样性，避免算法过早停止，陷入局部最优，即

$$\tau_{ij}(t)=\begin{cases}\tau_{\max} & (\tau_{ij}(t)\geqslant\tau_{\max})\\ \tau_{ij}(t) & (\tau_{\min}<\tau_{ij}(t)<\tau_{\max})\\ \tau_{\min} & (\tau_{ij}(t)\leqslant\tau_{\min})\end{cases}\quad(11)$$

式中　$\tau_{\max}$——信息素浓度上限

$\tau_{\min}$——信息素浓度下限

2.4　路径节点双优化策略

蚁群算法所规划的路径由栅格中心点集合构成，无效路径节点和路径转折次数较多，因此，本文提出一种路径节点双优化策略，优化过程如图 4 所示。假设规划的路径节点依次为 S、A、B、C、E，S 为起点，E 为终点。优化方法 1：对于路径（S，A，B），经过障碍物检测算法判断，如果路径节点 A 和 B 之间没有障碍物，则将路径节点 S 与 B 相连，删去路径节点 A。对路径（S，B，C）用同样的方法，循环操作，最终得到路径（S，E）。优化方法 2：将起点 S 依次与路径倒序（E，C，B，A）进行连接，若路径（S，E）之间没有障碍物则直接删除路径节点 C、B、A，得到路径（S，E）。

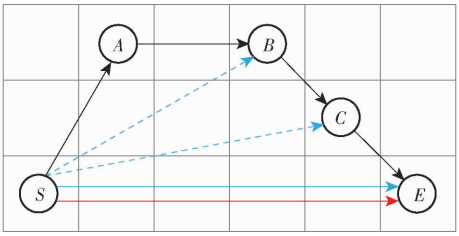


图 4　路径节点双优化策略图

Fig. 4　Double optimal strategy of nodes of path

通过以上两种方法对路径优化结果如图 5 所示。图中红色栅格为起点，绿色栅格为终点，绿色路径为 DENACO 算法规划未经优化的路径，红色和蓝色路径为优化方法 1 和 2 所生成的路径。在红色栅格与黄色栅格之间，蓝色路径和红色路径长度分别为 6.513 m 和 6.606 m，蓝色路径比红色路径短 0.093 m。在紫色栅格与绿色栅格之间，蓝色和红色路径长度分别为 6.831 m 和 6.434 m，蓝色路径相比红色路径长 0.397 m。可见在不同路径中两种优化方法各有优点，因此同时使用两种方法进行路径优化，取其最优路径为最终路径，以适应更复杂的栅格环境图。

2.5　算法流程

动态扩展邻域蚁群算法流程图如图 6 所示，具体步骤如下：

（1）确定起点坐标和终点坐标，建立信息素矩阵和启发式因子矩阵，设置参数 $N_{\max}$、m、α、β、ρ、Q、λ_1、λ_2、ω_1、ω_2、γ、T、K_1、K_2、$\tau_{\max}$、$\tau_{\min}$。

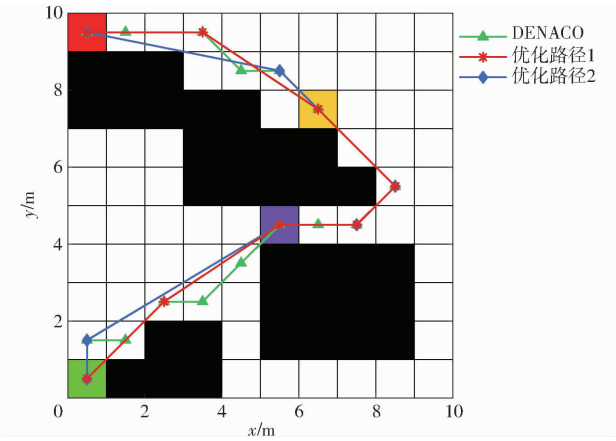


图 5　优化路径对比

Fig. 5　Comparison of optimal path

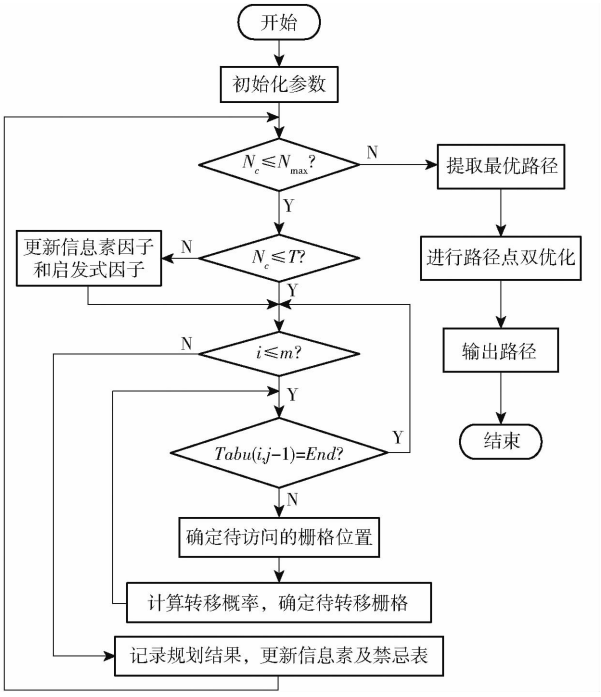


图 6　DENACO 算法流程图

Fig. 6　Flow of DENACO

（2）判断迭代次数是否大于迭代阈值 T，若大于阈值则调整 α 和 β，否则将所有蚂蚁放到起点开始迭代。

（3）使用动态扩展邻域移动规则确定下一步可移动栅格位置，根据式（2）和式（6）~（8）计算转移概率，利用轮盘赌选择下一个栅格位置，重复此步骤直到终点置于禁忌表中。

（4）记录迭代路径和转折点，根据式（4）、（5）和式（9）~（11）更新信息素矩阵，清空禁忌表，迭代次数加 1。

（5）循环步骤（2）~（4），当前迭代次数 N_c 大于 $N_{\max}$ 时结束迭代，使用路径节点双优化策略对路径进行优化，输出最终路径和转折点。

3 实验仿真

从以下方面验证 DENACO 算法可行性与适用性:①主要算法影响参数 α、β、ρ 的选择。②在不同随机栅格环境中对动态扩展邻域的有效性进行对比。③在不同复杂度的栅格环境下与对比算法进行比较分析。算法运行环境为:Windows 10(64bit);Matlab R2018a;处理器 Intel CORE i5 7th Gen;内存容量 16 GB。

3.1 主要参数优化

蚁群算法参数的选取直接影响算法的整体性能。信息素浓度因子 α 体现蚂蚁在寻路过程中路径上信息素的浓度对其寻路决策的相对重要程度。启发式信息因子 β 体现蚂蚁在寻路过程中先验性因素和确定性因素对其寻路决策的相对重要程度。信息素挥发因子 ρ 为蚂蚁在走过的路径上留下的信息素所挥发的速度,关系算法的全局最优搜索能力和收敛速度。

本文对影响算法性能的主要参数 α、β、ρ 进行优化选择,设置不同的参数组进行实验。同时为了分析参数对算法的影响,将参数默认值设置为 $\alpha=0.8$、$\beta=2$、$\rho=0.1$。每次只改变一个参数值,另外两个设为默认值。其它设为固定值,如表 1 所示。

表 1 DENACO 算法参数

Tab.1 Parameters of DENACO algorithm

参数	N_{max}	M	λ_1	λ_2	ω_1	ω_2	γ	Q	T	K_1	K_2	τ_{max}	τ_{min}
数值	50	50	0.5	0.5	0.1	0.9	1	100	10	2	10	40	5

对比在不同 α、β、ρ 取值下路径长度的变化,以平均路径长度较短时参数取值为实验参数的推荐值,每组参数值下进行 10 次实验取平均值,结果如表 2 所示。可以看出,α 取 0.7 ~ 0.9 时,算法所规划的平均路径长度分别为 30.64、30.22、30.28 m,当 α 大于 0.9 时,随着 α 的递增,平均路径长度指标逐渐变差。β 取 1 和 2 时,算法陷入局部最优,平均路径长度大于 30 m。β 取 6 时,算法所规划的路径平均长度为 29.42 m,平均路径长度最短。ρ 取 0.1 ~ 0.4 时,算法所规划的平均路径长度均在 30 m 以下。ρ 大于 0.4 时,平均路径长度指标逐渐变差,陷入局部最优,所规划路径长度平均值皆在 30 m 以上。

表 2 主要参数优化实验结果

Tab.2 Optimized experimental results of main parameters

α	平均路径长度/m	β	平均路径长度/m	ρ	平均路径长度/m
0.4	33.02	1	30.92	0.1	29.48
0.6	31.06	2	30.54	0.2	29.54
0.7	30.64	3	29.84	0.3	29.78
0.8	30.22	4	29.78	0.4	29.66
0.9	30.28	5	29.60	0.5	30.20
1.0	31.36	6	29.42	0.6	30.14
1.5	33.02	7	29.72	0.7	30.04
2.0	35.74	8	29.60	0.8	30.38
4.0	41.76	9	29.60	0.9	30.58

因此,当 $\alpha\in\{0.7,0.8,0.9\}$、$\beta\in\{5,6,8,9\}$、$\rho\in\{0.1,0.2,0.3,0.4\}$ 时,改进算法的性能较好。故后续实验均采用 $\alpha=0.8$、$\beta=6$、$\rho=0.1$ 作为算法参数值,其他参数值同表 1。

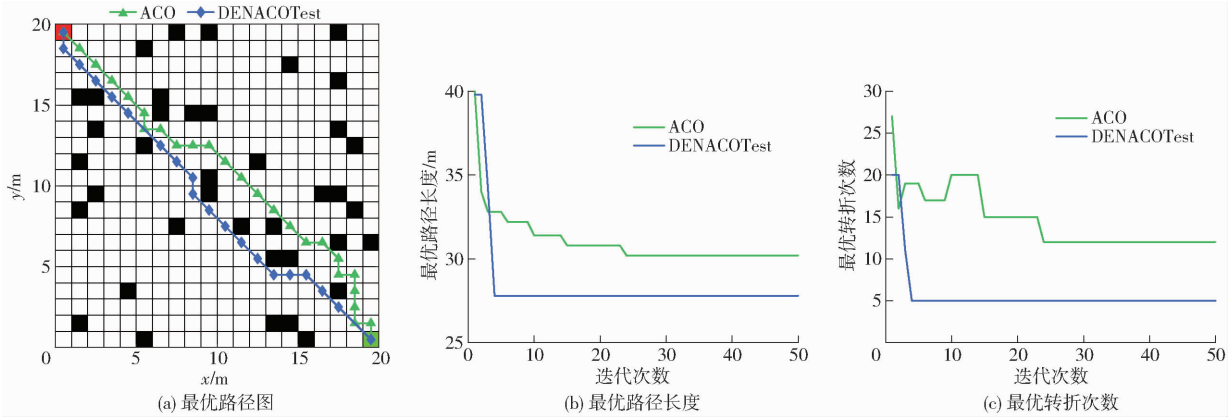
3.2 DENACO 算法有效性分析

为了验证 DENACO 算法改进蚁群算法部分的有效性,将使用路径节点双优化策略之前的改进算法记 DENACOTest,在 2 种不同的 20×20 随机栅格环境下,对 DENACOTest 进行仿真实验验证。2 种环境障碍栅格位置随机分布,简单栅格环境中障碍物占栅格总数的 10%,复杂栅格环境中障碍物占栅格总数的 20%。同时,为了验证 DENACOTest 算法在不同规模栅格环境中的适用性,分别在 15×15、30×30、50×50 复杂栅格环境中进行仿真实验。取 10 次实验结果的平均值及最优值与 ACO 对比,评价指标包括通过路径长度、迭代次数、路径转折点数、路径节点数。

3.2.1 不同复杂度栅格地图中的仿真实验

在 20×20 简单随机栅格环境图中最优路径长度的路径规划结果如图 7 所示。由图 7b 得出,DENACOTest 相比 ACO,收敛速度更快,仅用 4 次迭代达到最优,而 ACO 在 24 次迭代时才达到最优,为 DENACOTest 的 6 倍。ACO 陷入局部最优,最优收敛路径长度为 30.2 m,而 DENACOTest 最优收敛路径长度仅为 27.8 m。

更少的路径转折点和路径节点数更有利于移动机器人在行走过程中的稳定性,保证移动机器人能安全快速的到达目标点,同时也反映了路径长度。由图 7a 得出,ACO 的路径节点数和转折点数分别为 26 和 12,DENACOTest 因动态扩展邻域的存在位置与上一步移动方向相同,因此蚁群有更高的概率选择以 2 步长直行,路径节点数和转折点数分别为 20 和 5,相比 ACO 减少 6 和 7。由



(a) 最优路径图　(b) 最优路径长度　(c) 最优转折次数

图 7　20 × 20 简单随机栅格环境图

Fig. 7　Map in 20 × 20 simple random grid

图 7c 得出，DENACOTest 转折点数量在达到收敛的过程中始终处于减少的趋势，而 ACO 转折点数量收敛趋势相对不稳定。表 3 为 20 × 20 随机栅格实验结果，可以看出 DENACOTest 各项评价指标明显优于 ACO。

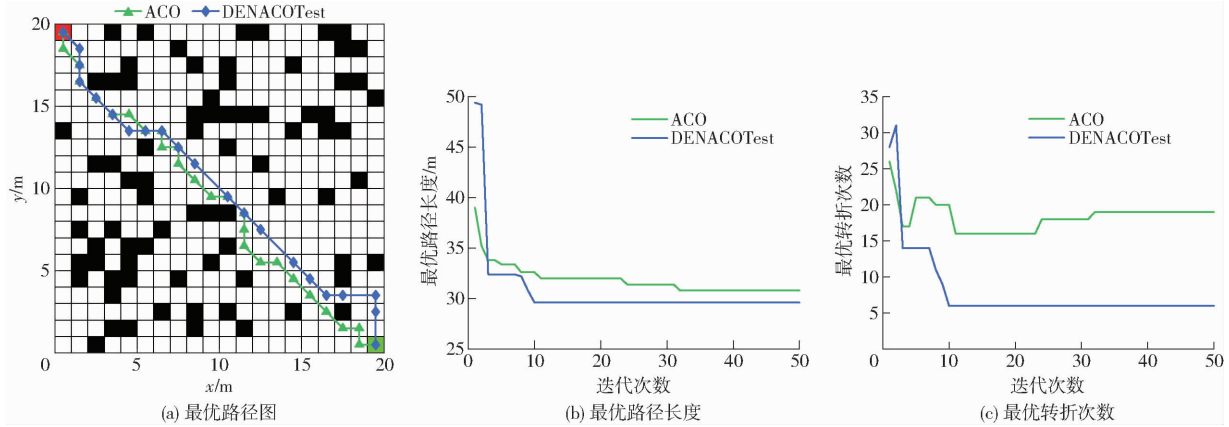
如图 8b 所示，在 20 × 20 复杂随机栅格环境所规划的最优路径中，DENACOTest 在第 11 次迭代时收敛，而 ACO 需要 32 次迭代才能找到最优路径，且 DENACOTest 的最优路径长度为 29.6 m，而 ACO 陷入局部最优，在路径长度为 30.8 m 时收敛。由图 8a 得出，DENACOTest 路径节点数和转折点数分别为 21 和 6，ACO 的路径节点数和转折点数分别为 27 和 19，两者在转折点数上相差更大，证明 DENACOTest 具有更好的适应性。由图 8c 得出，

表 3　20 × 20 随机栅格实验结果

Tab. 3　Experimental results in 20 × 20 random grid

地图规模	评价指标	算法	
		ACO	DENACOTest
简单随机栅格	平均路径长度/m	30.74	27.98
	平均迭代次数	20.20	5.70
	平均转折点数	14.20	6.60
	平均路径节点数	26.90	18.60
复杂随机栅格	平均路径长度/m	31.30	30.26
	平均迭代次数	28.00	9.00
	平均转折点数	16.80	8.40
	平均路径节点数	27.30	21.90

DENACOTest 的转折点数量仍然能始终保持下降趋势，而 ACO 出现了回升。由表 3 可见，DENACOTest 各项评价指标也优于 ACO。



(a) 最优路径图　(b) 最优路径长度　(c) 最优转折次数

图 8　20 × 20 复杂随机栅格环境图

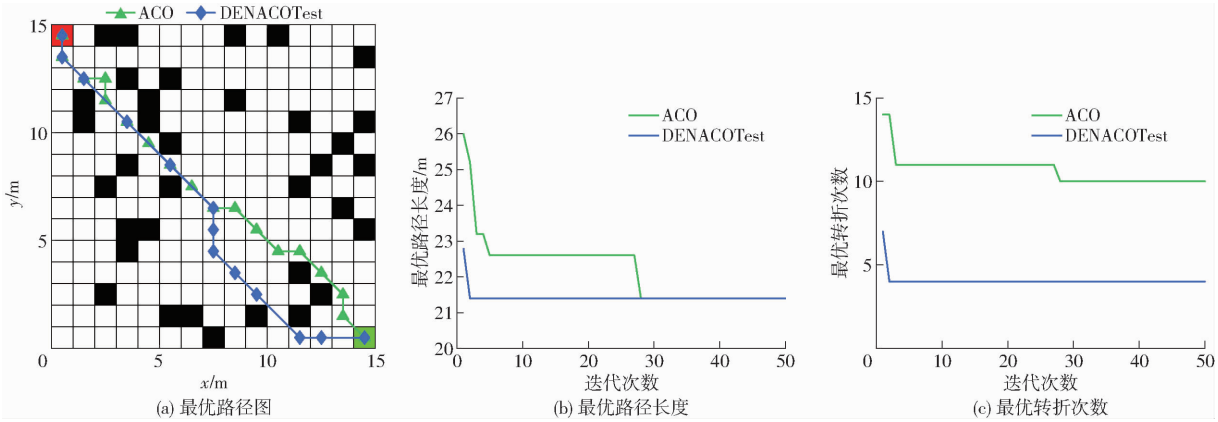
Fig. 8　Map in 20 × 20 complicated random grid

3.2.2　不同规模复杂栅格地图中的仿真实验

不同规模复杂栅格地图的最优路径长度的路径规划仿真实验结果如图 9 ~ 11 所示，10 次仿真实验平均结果如表 4 所示。

由图 9 可得，ACO 与 DENACOTest 均能找到最优路径，最优路径长度同为 21.4 m，但在平均值上 DENACOTest 略高于 ACO，DENACOTest 在迭代收敛次数、转折点数量、路径路径点数量上有很大提升，ACO 的迭代收敛次数为 DENACOTest 的 14 倍，转折点数量为 2.5 倍，路径节点数量相差 5。

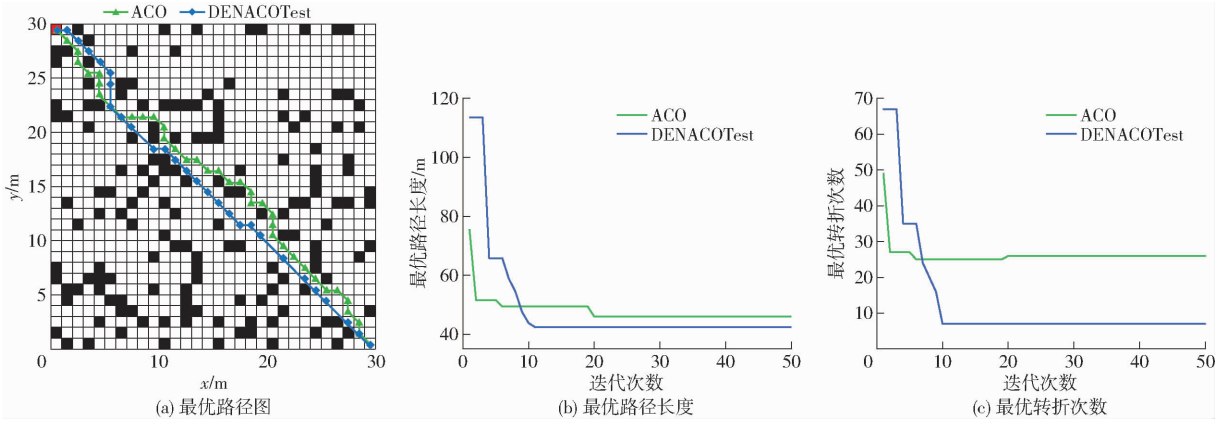
由图 10 可得，DENACOTest 的 4 个评价指标最优值及平均值与 ACO 相比均为更优，ACO 转折点数量为 26，DENACOTest 为 7，相较减少 19。ACO 路径节点数量为 39，DENACOTest 为 27，相较减少 12，



(a) 最优路径图 (b) 最优路径长度 (c) 最优转折次数

图 9 15×15 复杂随机栅格环境图

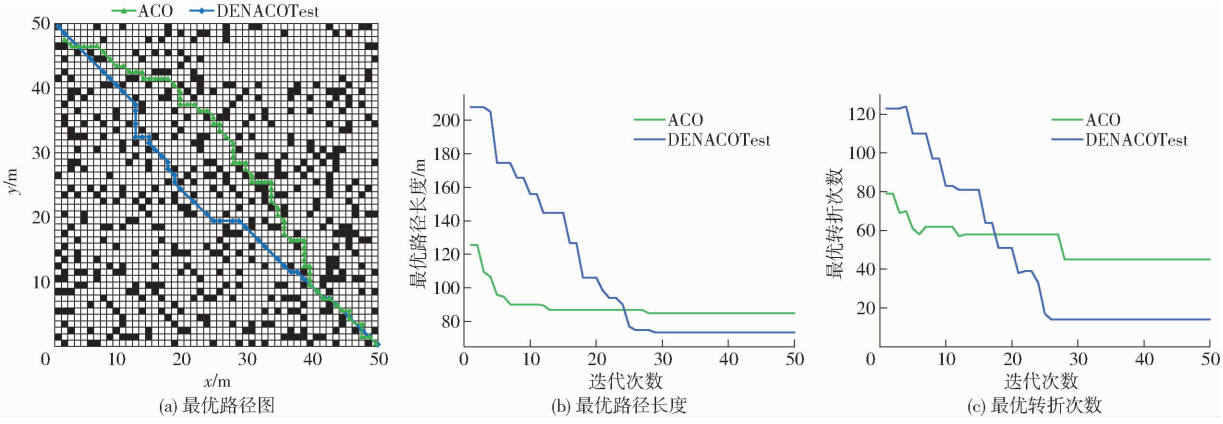
Fig. 9 Map in 15×15 complicated random grid



(a) 最优路径图 (b) 最优路径长度 (c) 最优转折次数

图 10 30×30 复杂随机栅格环境图

Fig. 10 Map in 30×30 complicated random grid



(a) 最优路径图 (b) 最优路径长度 (c) 最优转折次数

图 11 50×50 复杂随机栅格环境图

Fig. 11 Map in 50×50 complicated random grid

且 ACO 的路径转折点数量出现回升。

由图 11 可得,随着栅格规模进一步扩大,DENACOTest 为搜索到更优路径在最优收敛迭代次数上与 ACO 相近,但 DENACOTest 在其他 3 个评价指标上均为最优,路径转折点数量相差 31,在路径节点数量上的差距进一步扩大,相差达 34,因此 DENACOTest 规划路径更为优秀。

3.3 算法仿真对比实验

为进一步验证 DENACO 算法的优越性,采用文献[20]提供的栅格地图进行仿真实验,取 10 次实验结果的平均值及最优值,与文献中的 ACO、改进蚁群算法(Improved ant colony algorithm,IACO)、改进自适应蚁群算法(An improvement adaptive ant colony algorithm,IAACO)进行对比,各算法参数值如表 5 所示。

3.3.1 20×20 简单栅格环境

在 20×20 简单环境中 4 种算法路径规划结果如图 12 及表 6 所示。ACO 陷入局部最优,收敛速

表 4　不同规模栅格地图中的实验结果

Tab. 4　Experimental results in grid maps of different scales

地图规模	评价指标	算法	
		ACO	DENACOTest
15×15	平均路径长度/m	22.18	21.62
	平均迭代次数	13.20	2.40
	平均转折点数	10.90	4.80
	平均路径节点数	19.30	14.50
30×30	平均路径长度/m	47.42	42.86
	平均迭代次数	31.60	14.10
	平均转折点数	28.30	9.40
	平均路径节点数	41.40	28.10
50×50	平均路径长度/m	85.60	76.6
	平均迭代次数	34.80	28.2
	平均转折点数	48.20	17.0
	平均路径节点数	76.60	49.4

表 5　文献[19]算法参数

Tab. 5　Parameters of reference [19]

参数	σ_1	σ_2	R_s	δ_0	Q	k_L	k_S
数值	0.1	0.9	0.5	0.15	2.5	0.7	0.1
参数	K_E	m	N_{max}	α	β	λ	κ
数值	0.2	50	100	1	7	7	0.9

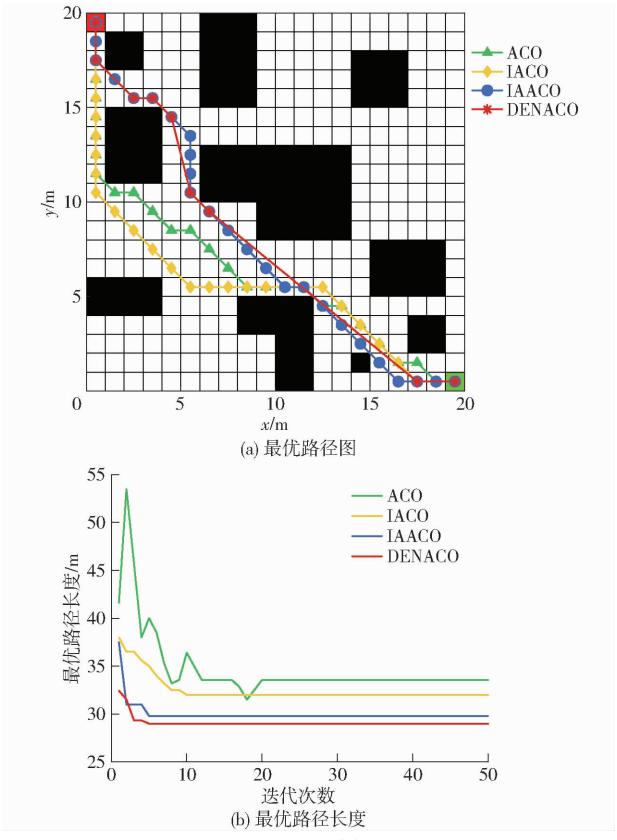


图 12　20×20 简单栅格环境图

Fig. 12　Map in 20×20 complicated grid

度较慢，并且所规划的路径转折点较多，迭代收敛曲线不够平滑。IACO 和 IAACO 在各项指标上均有提升，最优路径分别仅需 6 次和 5 次迭代就收敛于最

表 6　20×20 简单栅格实验结果

Tab. 6　Experimental results in 20×20 simple grid maps

评价指标	算法			
	ACO	IACO	IAACO	DENACO
最优路径长度/m	33.560	32.000	29.800	28.992
平均路径长度/m	36.609	33.220	30.122	29.415
最优迭代次数	14	6	5	5
平均迭代次数	23.1	9.8	6.4	4.7
最优转折点数	12	4	8	7
最优路径节点数	28	29	25	9

优解，并且所规划的路径更平滑，路径长度更优。DENACO 相比 IAACO 在最优路径长度、转折点数上有一定提升，相比 IACO 略有不足但相差不大。但在路径节点数上，IACO 和 IAACO 与 DENACO 相比差距很大，DENACO 最优路径节点数为 9，IACO 和 IAACO 最优路径节点数分别为 29 和 25，分别为 DENACO 的 3.2 倍和 2.8 倍。且 DENACO 路径长度和迭代次数平均值也优于 IACO 和 IAACO。因此，本文所提出的 DENACO 算法在简单栅格环境中具有更好的性能。

3.3.2　20×20 复杂栅格环境

在 20×20 复杂环境中 3 种算法路径规划结果如图 13 和表 7 所示。ACO 所规划的路径平滑性更差，转折点更多，且陷入局部最优。IACO 相比 ACO 收敛速度更快，但收敛路径长度较差。

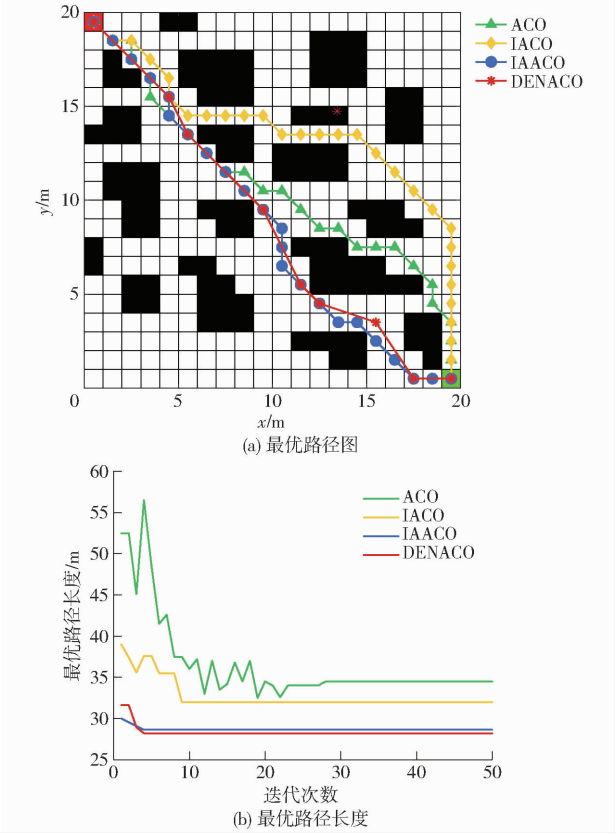


图 13　20×20 复杂栅格环境图

Fig. 13　Map in 20×20 complicated grid

IAACO 在复杂栅格环境中仍可以保持较快的收敛速度和较短的收敛路径长度。由表 7 可知,无论是平均值还是最优值,IAACO 与 ACO 的路径长度、迭代次数、转折点数差距进一步扩大,IAACO 表现出更好的性能。DENACO 相比 IAACO 在最优路径长度和最优迭代次数上有了提升。而 DENACO 路径节点数仅为 9,与 IAACO 的 24 个路径节点相差多达 15,且路径长度和迭代次数的平均值皆有提升。

表 7 20×20 复杂栅格实验结果

Tab. 7 Experimental results in 20×20 complicated grid maps

评价指标	算法			
	ACO	IACO	IAACO	DENACO
最优路径长度/m	31.800	32.000	28.630	28.203
平均路径长度/m	33.764	33.260	29.505	28.687
最优迭代次数	17	6	3	2
平均迭代次数	27.4	9.3	4.0	3.6
最优转折点数	17	9	7	7
最优路径节点数	26	29	23	9

综上所述,本文所提出的 DENACO 算法无论是在简单栅格环境中还是复杂栅格环境中,与对比算法相比,均展现出了更好的性能。

4 结束语

提出基于动态扩展邻域蚁群算法,解决了传统蚁群算法易陷入局部最优、路径转折点多、收敛速度慢的问题。将蚂蚁的搜索方式由 8 搜索邻域在其移动方向上向外扩展一个搜索邻域,并设计新的信息素计算方式和增量规则,以减少转折点数及路径节点数并且得到更优的收敛路径长度。在启发函数加入邻域栅格到目标点的距离因素,并引入自适应调整因子,设计迭代阈值,在加强算法寻路效果的同时,提升算法的收敛速度。对规划路径进行路径节点双优化,提高路径平滑度,得到更适合于移动机器人行进的路径。通过在不同复杂度及不同规模下的对比实验表明,DENACO 算法与基本蚁群算法及对比算法相比有了一定提升,验证了 DENACO 算法的可行性和适用性。

参 考 文 献

[1] ZHU S N, ZHU W Y, ZHANG X Q, et al. Path planning of lunar robot based on dynamic adaptive ant colony algorithm and obstacle avoidance[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 172988141989897.

[2] ZHONG X Y, TIAN J, HU H S, et al. Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment[J]. Journal of Intelligent & Robotic Systems, 2020, 99(1): 65 – 77.

[3] 刘慧,张世义,段云鹏,等. 基于改进双向 RRT* 的果园机器人运动规划算法[J]. 农业机械学报,2022,53(11):31 – 39.
LIU Hui, ZHANG Shiyi, DUAN Yunpeng, et al. Orchard robot motion planning algorithm based on improved bidirectional RRT*[J]. Transactions of the Chinese Society for Agricultural Machinery, 2022, 53(11): 31 – 39. (in Chinese)

[4] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269 – 271.

[5] 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 Xiao, WANG Zheng, HUANG Chengkan, et al. Mobile robot path planning based on an improved A* algorithm[J]. Robot, 2018, 40(6): 903 – 910. (in Chinese)

[6] HUANG Y J, DING H T, ZHANG Y B, et al. A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach[J]. IEEE Transactions on Industrial Electronics, 2020, 67(2): 1376 – 1386.

[7] FOX D, BURGARD W, THRUN S. The dynamic window approach to collision avoidance[J]. IEEE Robotics & Automation Magazine, 1997, 4(1): 23 – 33.

[8] CHANG L, SHAN L, JIANG C, et al. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment[J]. Autonomous Robots, 2021, 45(1): 51 – 76.

[9] WEN J B, YANG J C, WANG T Y. Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm[J]. IEEE Transactions on Vehicular Technology, 2021, 70(9): 8529 – 8544.

[10] KIM J J, LEE J J. Trajectory optimization with particle swarm optimization for manipulator motion planning[J]. IEEE Transactions on Industrial Informatics, 2015, 11(3): 620 – 631.

[11] DORIGO M, BIRATTARI M, STUTZLE T. Ant colony optimization-artificial ants as a computational intelligence technique[J]. IEEE Computational Intelligence Magazine, 2006, 1(4): 28 – 39.

[12] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29 – 41.

[13] LIU J H, YANG J G, LIU H P, et al. An improved ant colony algorithm for robot path planning[J]. Soft Computing, 2017, 21(19): 5829 – 5839.

(下转第 449 页)

Journal of Mechanical Engineering,2016,62(6):373 – 380.
[22] 张慧宇,张惟斌,衡亚光,等. 基于浸入实体法的低压外啮合齿轮泵不稳定流动特性[J]. 船舶工程,2021,43(12):120 – 125.
ZHANG Huiyu, ZHANG Weibin, HENG Yaguang, et al. Unsteady flow characteristics of low pressure external gear pump based on submerged entity method[J]. Marine Engineering,2021,43(12):120 – 125. (in Chinese)
[23] 王乐. 基于 CFD 和 ANSYS 技术对 CBN 齿轮泵仿真实验分析[J]. 煤矿机械,2019,40(6):185 – 187.
WANG Le. Simulation experiment analysis of CBN gear pump based on CFD and ANSYS technology[J]. Coal Mine Machinery,2019,40(6):185 – 187. (in Chinese)
[24] 杨永敏,卢前顺. 基于 CFD 的商用航空发动机齿轮泵浮动侧板设计技术研究[J]. 制造业自动化,2016,38(4):49 – 53,62.
YANG Yongmin, LU Qianshun. Research on floating side plate design of commercial aero-engine gear pump based on CFD [J]. Manufacturing Automation,2016,38(4):49 – 53,62. (in Chinese)
[25] SUJAN D, ANDREA V. A novel CFD-axial motion coupled model for the axial balance of lateral bushings in external gear machines[J]. Simulation Modelling Practice and Theory,2012,26:60 – 76.

(上接第 432 页)

[14] WU L, HUANG X D, CUI J G, et al. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot[J]. Expert Systems with Applications, 2023, 215: 119410.
[15] GAO W X, TANG Q, YE B F, et al. An enhanced heuristic ant colony optimization for mobile robot path planning[J]. Soft Computing, 2020, 24(8): 6139 – 6150.
[16] TAO Y, GAO H, REN F, et al. A mobile service robot global path planning method based on ant colony optimization and fuzzy control[J]. Applied Sciences, 2021, 11(8): 3605.
[17] 马小陆,梅宏,龚瑞,等. 基于改进 ACS 算法的移动机器人路径规划研究[J]. 湖南大学学报(自然科学版),2021,48(12):79 – 88.
MA Xiaolu, MEI Hong, GONG Rui, et al. Research on path planning of mobile robot based on improved ACS algorithm[J]. Journal of Hunan University (Natural Sciences), 2021, 48(12): 79 – 88. (in Chinese)
[18] XUE T, LI L, SHUANG L, et al. Path planning of mobile robot based on improved ant colony algorithm for logistics[J]. Mathematical Biosciences and Engineering, 2021, 18(4): 3034 – 3045.
[19] 徐菱,付文浩,江文辉,等. 基于 16 方向 24 邻域改进蚁群算法的移动机器人路径规划[J]. 控制与决策,2021,36(5):1137 – 1146.
XU Ling, FU Wenhao, JIANG Wenhui, et al. Mobile robots path planning based on 16-directions 24-neighborhoods improved ant colony algorithm[J]. Control and Decision, 2021, 36(5): 1137 – 1146. (in Chinese)
[20] MIAO C W, CHEN G Z, YAN C L, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm[J]. Computers & Industrial Engineering, 2021, 156: 107230.
[21] LUO Q, WANG H B, ZHENG Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm [J]. Neural Computing and Applications, 2020, 32(6): 1555 – 1566.
[22] 张恒,何丽,袁亮,等. 基于改进双层蚁群算法的移动机器人路径规划[J]. 控制与决策,2022,37(2):303 – 313.
ZHANG Heng, HE Li, YUAN Liang, et al. Mobile robot path planning using improved double-layer ant colony algorithm[J]. Control and Decision, 2022, 37(2): 303 – 313. (in Chinese)
[23] DAI X L, LONG S, ZHANG Z W, et al. Mobile robot path planning based on ant colony algorithm with A* heuristic method [J]. Frontiers in Neurorobotics, 2019, 13: 15.
[24] YANG H, QI J, MIAO Y C, et al. A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization[J]. IEEE Transactions on Industrial Electronics, 2019, 66(11): 8557 – 8566.
[25] 敖邦乾,杨莎,叶振环. 改进蚁群算法水面无人艇平滑路径规划[J]. 控制理论与应用,2021,38(7):1006 – 1014.
AO Bangqian, YANG Sha, YE Zhenhuan. Improved ant colony algorithm for unmanned surface vehicle smooth path planning [J]. Control Theory & Applications, 2021, 38(7): 1006 – 1014. (in Chinese)
[26] 王洪斌,尹鹏衡,郑维,等. 基于改进的 A* 算法与动态窗口法的移动机器人路径规划[J]. 机器人,2020,42(3):346 – 353.
WANG Hongbin, YIN Pengheng, ZHENG Wei, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method[J]. Robot, 2020, 42(3): 346 – 353. (in Chinese)
[27] 劳彩莲,李鹏,冯宇. 基于改进 A* 与 DWA 算法融合的温室机器人路径规划[J]. 农业机械学报,2021,52(1):14 – 22.
LAO Cailian, LI Peng, FENG Yu. Path planning of greenhouse robot based on fusion of improved A* algorithm and dynamic window approach[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(1): 14 – 22. (in Chinese)
[28] 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 Jianhua, YANG Jianguo, LIU Huaping, 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. (in Chinese)