

# 基于信息素负反馈的超启发式蚁群优化算法

薛文艳, 赵江, 郝崇清, 刘慧贤

河北科技大学 电气工程学院, 石家庄 050011

**摘要:**针对蚁群算法应用于自动导引小车路径规划收敛速度慢、极易陷入局部最优的缺点,提出一种基于信息素负反馈的超启发式蚁群优化(ACONhh)算法。该算法充分利用历史搜索信息和持续获得错误经验,较快引导蚁群探索最优路径;分层化选择可行节点,加快算法初期收敛速度;设置挥发因子呈类抛物线变化以及调整信息素更新机制,改善路径全局的随机搜索特性。通过严格的数学方式证明了ACONhh算法具有收敛性。仿真和实验结果表明,该算法的收敛速度以及全局搜索性能显著优于目前流行的ACO、ACOhh和ACOi hh算法。

**关键词:**自动导引小车;路径规划;蚁群优化算法;信息素负反馈;分层化选择

**文献标志码:**A **中图分类号:**TP242.6 **doi:**10.3778/j.issn.1002-8331.1711-0301

薛文艳, 赵江, 郝崇清, 等. 基于信息素负反馈的超启发式蚁群优化算法. 计算机工程与应用, 2019, 55(4): 163-172.

XUE Wenyan, ZHAO Jiang, HAO Chongqing, et al. Hyper heuristic ant colony optimization algorithm based on pheromone negative feedback. Computer Engineering and Applications, 2019, 55(4): 163-172.

## Hyper Heuristic Ant Colony Optimization Algorithm Based on Pheromone Negative Feedback

XUE Wenyan, ZHAO Jiang, HAO Chongqing, LIU Huixian

College of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang 050011, China

**Abstract:** The existing ant colony algorithm is applied to the path planning of automated guided vehicle, which is slow in the convergence and easily fails into the local optimum. To solve these problems, this paper proposes a Hyper Heuristic Ant Colony Optimization algorithm based on pheromone negative feedback (ACONhh) for path planning of mobile robots. The algorithm makes full use of historical search information and continues to gain error experience, thus further leads ant colony to explore optimal path. Hierarchical selection of feasible nodes is adopted to accelerate the initial convergence rate of the algorithm. Meanwhile, the volatility factor changes constantly with an analogous parabola, and pheromone update mechanism is adjusted to improve the randomness of global search. The convergence of ACONhh algorithm is strictly proved. Simulation and experimental results show that the convergence speed and global search performance of the proposed algorithm are outperform those of popular ACO, ACOhh and ACOi hh algorithms.

**Key words:** automated guided vehicle; path planning; ant colony optimization; pheromone negative feedback mechanism; hierarchical selection

## 1 引言

自动化搬运设备是先进物流系统的核心组成部分,其能加快物流运输、缩短生产周期以及提高效益等,自动导引小车(Automatic Guided Vehicle, AGV)被广泛应用到仓储物流系统领域<sup>[1-2]</sup>。AGV的运行路径直接影响到其运输效率,路径规划已成为物流运输领域的研究

热点,AGV的路径规划是控制AGV按照一定的性能指标(路径最短、时间最小、损耗最低等)完成作业<sup>[3]</sup>。

许多学者对路径规划进行了大量的研究,例如人工势场法<sup>[4]</sup>、A\*算法<sup>[5]</sup>、拓扑结构<sup>[6]</sup>以及神经网络<sup>[7]</sup>等,然而这些算法存在计算量大、鲁棒性弱以及缺乏适应性等缺点。

**基金项目:**国家自然科学基金(No.51507048);河北省自然科学基金(No.F2014208013)。

**作者简介:**薛文艳(1992—),女,硕士生,主要研究领域为地面机器人路径规划及其控制技术;赵江(1958—),男,教授,硕士生导师,主要研究领域为过程参数检测与智能控制技术;郝崇清(1981—),通讯作者,男,博士,副教授,主要研究领域为智能控制技术, E-mail: haochongqing@hebest.edu.cn;刘慧贤(1983—),女,博士,副教授。

**收稿日期:**2017-11-21 **修回日期:**2018-01-05 **文章编号:**1002-8331(2019)04-0163-10

**CNKI网络出版:**2018-04-18, <http://kns.cnki.net/kcms/detail/11.2127.TP.20180417.1720.014.html>

为克服上述缺点,启发式算法受到各国学者的广泛关注。启发式算法是通过模拟某些自然生物(蚁群、鸟群、蜜蜂群等)的行为而得出的一种优化算法<sup>[8-9]</sup>。蚁群算法(Ant Colony Algorithm, ACA)作为一种基于种群的启发式算法,首次被成功地应用到解决组合优化问题中,例如旅行商问题(Travel Salesman Problem, TSP)<sup>[10]</sup>、错误检测<sup>[11]</sup>、系统识别<sup>[12]</sup>、数据最小化<sup>[13]</sup>、网络数据传输<sup>[14]</sup>、图像处理<sup>[15]</sup>等领域。由于ACA具有正反馈、自组织、分布式计算和较强的鲁棒性等优点<sup>[16]</sup>,其在路径规划方面的研究较多。然而,ACA同时又具有收敛速度慢,极易陷入局部最优等缺点<sup>[17]</sup>。

在早期蚁群系统中,已经有多种解决ACA陷入局部最优的方法,并取得了良好的结果。文献[18]提出差分演化蚁群优化算法,在信息素更新环节,加入混沌因子以增强算法的随机性能,使其有效跳出局部最优点。为了提高蚁群算法的全局搜索能力,文献[19]提出一种自适应调节信息素挥发因子机制,进一步研究发现蚁群优化(Ant Colony Optimization, ACO)算法中最重要的参数为信息素挥发因子。文献[20]为提高算法的随机性能,通过自适应调整相关参数,实现算法的多样性与深度局部搜索,通过仿真说明了所提算法的可行性。文献[21]中,禁忌搜索策略被应用到解决机器人的路径规划问题中,其特点是采用禁忌技术,把已经访问过的局部最优点记录到禁忌表中,继续搜索其他区域,避免陷入局部极小点,结果验证了ACO算法在较小规模的地图模型中的有效性,但算法的收敛速度会受到初始解的影响,较差的初始解会降低算法寻优速度。

同时,针对ACA中收敛速度慢的问题,一系列改进方法被相继提出。一种双并行搜索策略在文献[22-23]中被引入,通过控制两组方向相反的蚂蚁同时进行路径搜索,充分发挥蚁群的合作性,从而提高了算法寻优速度。通过从不同层面改进初始信息素分配可以改善算法初期搜索的盲目性,He等<sup>[24]</sup>应用蚁群算法与遗传算法(ACO-GA)相结合的方法,使算法初期搜索效率得到有效提高;Aziz<sup>[25]</sup>提出了一种超启发式蚁群优化算法(ACOhh),通过缩小更新对象,减少在信息素更新过程中不必要的时间消耗,使算法收敛速度提高。

目前,一些现有的ACO算法<sup>[18-25]</sup>仅局限在对信息素正反馈机制的改进来进一步提高算法性能。尽管这些改进算法优于基本ACA以及传统路径规划方法,但由于仅采纳蚂蚁残留的正确信息,没有吸取错误经验教训,不能充分利用路径信息,仍易陷入局部最优。在一次迭代过程中,蚁群可以搜索出一条最优与最差路径,如果将最差信息充分利用起来,使蚂蚁不断吸取错误经验,则蚁群会不断趋向于最优路径,摒弃较差路径,增加路径选择的多样性。Ye等<sup>[26]</sup>通过引入信息素负反馈矩阵,使后代不断吸取错误经验,同时适时采取成功方

案。该算法成功应用到解决大规模的二元约束问题(Binary Constraint Problem, BCP)中,不仅能够改善寻优速度,而且提高了解决算法陷入局部最优的能力。

从提高物流仓储的经济效益角度出发,应保证AGV以最快速度沿最优路径到达目标点,即在提高算法速度的同时兼顾其全局性能。针对AGV的路径规划问题,ACO在改善收敛速度的同时,提高其跳出局部最优能力的相关文献报道较少。因此,针对ACO算法收敛速度慢,极易陷入局部最优的缺点,本文在文献[26]的基础上提出一种基于信息素负反馈的超启发式蚁群优化算法(ACONhh),用于提高AGV的运输效率。

## 2 环境描述及建模

AGV在仓储系统中执行搬运任务时,利用自身所处位置与目标点位置,通过路径规划算法寻找一条符合规定的路径行驶。AGV实物及其八个行驶方向如图1所示。

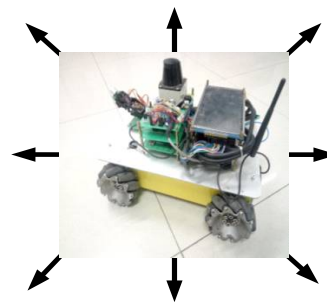


图1 AGV运动方向

栅格地图是表示环境的常用媒介,其中连续空间由类似于位图图像的离散单元集合表示。考虑到栅格表示简单,编程易于实现等优点,本文利用栅格法对AGV的工作空间进行划分<sup>[27]</sup>。

设给定的小车在全局已知且障碍物是静态的二维环境中工作,用栅格划分出AGV的工作区域。用直角坐标以及序号对栅格做出标识,障碍物占据半个或者多于一个栅格,都用黑色阴影表示出来,代表该区域有障碍物,其余自由区域用白色表示,如图2(a)、(b)所示。

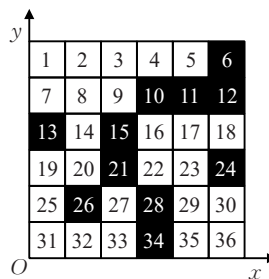


图2(a) 栅格地图

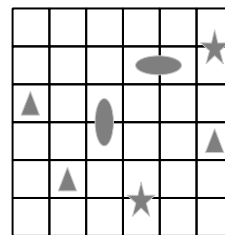


图2(b) 实际地图

设AGV外接圆半径为 $r$ ,其工作环境用长 $\times$ 宽的矩形表示,将矩形分成 $m \times n$ 的小矩形,小车位于白色矩形中心位置。为方便路径规划,做出以下约定:

(1)小车的中心位置设置成质点,对环境中的障碍物做适当膨化处理,使小车方便移动。

(2)假设某一个小矩形被它相邻的水平和竖直方向的三个障碍物矩形包围,则将该矩形视为障碍物,避免无效路径的选择。如图2(a)所示,27号栅格在水平和竖直方向周围有三个障碍物栅格,则将该栅格默认为障碍物栅格。

### 3 基本ACA原理

自然界中的生物蚂蚁能够利用群体智能行为寻找到最短的觅食路径,蚁群算法可以模拟真实蚂蚁的行为,为优化问题提供启发式解决方案。1992年,由Dorigo首次提出。

针对蚂蚁  $k(k=1,2,\dots,m)$  觅食时,每个节点所留的分泌物量的多少来进一步判断可走点,在时刻  $t$ ,蚂蚁  $k$  从节点  $i$  移动到节点  $j$  的概率为:

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

式中,  $\alpha$  与  $\beta$  表示启发因子的重要性;  $\tau_{ij}(t)$  表示路径  $ij$  信息素量的多少;  $\eta_{ij} = 1/d_{ij}$  表示节点  $i$  到节点  $j$  的期望程度,  $d_{ij}$  为蚂蚁  $k$  从节点  $i$  到  $j$  的距离;  $allowed_k$  为蚂蚁  $k$  接下来可以走的所有点。

蚂蚁所经过的每条轨迹残留的信息素量会随着时间的推进而逐渐减小,因此算法中引入挥发因子常数  $\rho(0 < \rho < 1)$ 。

蚂蚁开始运动时,每条节点上信息素量的多少用一个随机给出的常数来表示,由于  $k$  没有记忆,需要把每次  $k$  访问过的路径节点存入禁忌表中,当  $k$  到达目标节点之后,对路径上的信息素进行全局更新,更新公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij} + \Delta\tau_{ij} \quad (2)$$

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

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{若第 } k \text{ 只蚂蚁经过节点 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (4)$$

式中,  $\tau_{ij}(t+1)$  为  $t+1$  时刻路径  $ij$  的信息素量大小;  $\Delta\tau_{ij}^k(t)$  为  $t$  时刻  $k$  在路径  $ij$  留下信息素量的多少;  $\Delta\tau_{ij}(t)$  代表  $t$  时刻在路径  $ij$  留下的信息素总数;  $Q$  代表一个常数,衡量信息素的重要性;  $L_k$  代表  $k$  在当代循环结束后所走总路径长度。

### 4 ACONhh 算法

#### 4.1 节点选择分层化处理

通过对上述基本ACA分析得出,将各条路径的信

息素初始化为一个常数,给蚂蚁的初期寻优带来极大的隐患,导致初期搜索路径过于盲目,寻找范围较大,搜索效率较低。因此,本文采用分层处理方法。第一层蚂蚁具有开发性,用于分配各条路径的信息素,为提高算法初期收敛速度,对较优与较差路径的信息素加以区分;第二层蚂蚁具有探索性,为兼顾算法的全局性,利用第一层蚂蚁分配的信息素,选择可行节点。

第一代蚁群放入第一层,该层蚁群仅利用距离信息寻找可行点,以区分节点信息量,加快算法初期收敛速度。考虑到两点之间线段最短,如果连接起点与终点的直线  $L$  中,没有经过障碍区域,则蚁群沿  $L$  行走,即为最短路径。若在行走途中,遇到不可走点,则利用三角形原理(两边之和大于第三边)避开该点,继续朝  $L$  行走,直到到达最终点。简单流程如下:

步骤1 连接起点与终点,构成直线  $L$ 。

步骤2 判断  $L$  是否为正方形对角线,若是,则  $k$  沿  $L$  行走,否则,沿靠近  $L$  方向行走。

步骤3 启动第一代蚂蚁  $k$ 。

步骤4 每走一步判断有无障碍物,若有,则在該点建立直角三角形,直到避开障碍物,否则,继续行走。

步骤5 判断是否到达目标点,若没有,返回步骤2,否则,输出最短路径,进行全局信息素更新,结束第一层蚂蚁工作。

利用建立三角形原理避开障碍物( $A$ 与 $B$ 在 $L$ 上,蚂蚁 $k$ 在 $A$ 处, $B$ 表示障碍物区域,如图3所示),流程概括如下。

步骤1 连接线段  $AB$ ,作为直角三角形的一个直角边。

步骤2 以  $B$  为直角顶点,过点  $B$  向下做垂线,过  $A$  做水平射线与  $B$  的垂线交于  $D$ ,利用式(5)求出线段  $AB$  与线段  $AD$  的余弦值。

$$\cos \angle BAD = \frac{d_{AD}}{d_{AB}} \quad (5)$$

步骤3 延长线段  $AD$ ,同时过  $B$  向下做斜线交  $AD$  的延长线于  $C$ ,此时线段  $AB$  与线段  $BC$  的夹角为直角,利用式(6)求出线段  $AC$  的长度,  $C$  即为下一步选择的栅格点。

$$d_{AC} = \frac{d_{AB}}{\cos \angle BAD} \quad (6)$$

步骤4 判断  $C$  是否为可行点,若不是,则继续利用直角三角形思想规划,直到  $k$  找到可走点为止。否则,  $k$  选择  $C$  行走。

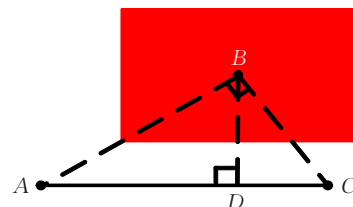


图3 建立直角三角形原理



如图4所示,红色实线段为采用该原理避开障碍物所得到的路径,且栅格14自动视为障碍区域(U形区);第二层即第一代以后蚁群,利用式(1)进行搜索。

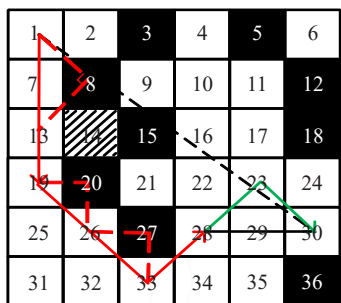


图4 第一代蚂蚁行驶路径

## 4.2 超启发式与负反馈相结合信息素更新策略

### 4.2.1 超启发式ACO算法

在基本的ACA中,信息更新对象为每一代中所有蚂蚁搜索出的可行路径,由于没有必要的路径更新,延长了算法运行时间。由此,一种超启发式(ACO<sub>hh</sub>)更新策略在文献[25]中被提出,并且应用到解决TSP中。超启发式是一种用于提高优化系统性能指标的策略,并广泛应用到多个领域。文献[25]中改进全局信息更新对象,仅对最优方案进行更新,其仿真结果表明,ACO<sub>hh</sub>算法寻优能力改善,运行时间短,收敛速度得到有效提高。

考虑到文献[25]重在提高算法速度,在更新信息素时,没有考虑次优路径的重要性,可能引起局部极小值问题。为提高算法全局性能并扩大更新对象,本文提出一种超启发式ACO改进算法(ACO<sub>i</sub>hh),即仅对较优可行路径进行更新,每次完成迭代后,会产生一条最差和最优路径,根据两条路径的均值选出每次循环后的较优路径,表达式如式(7)所示:

$$L_{\text{better}} \leq \frac{L_{\text{best}} + L_{\text{worst}}}{2} \quad (7)$$

式中,  $L_{\text{best}}$  代表每次搜索完成后产生的最优值;  $L_{\text{worst}}$  代表每次搜索完成后产生的最差值;  $L_{\text{better}}$  代表符合条件的较优路径。

### 4.2.2 带有信息素负反馈的蚁群优化(ACON)算法

算法每次迭代完成后,会产生最差与最好路径信息,大多数现有ACO在改进算法性能时,仅考虑了信息素的正反馈,即仅利用较好的蚂蚁选择出的路径信息,放弃了最差的信息,不能充分利用蚂蚁所留存的历史信息。为了进一步体现蚂蚁间的协作能力,充分利用蚂蚁残留的信息,文献[26]引入带有信息素负反馈的蚁群优化算法(ACON)解决一种随机的二元约束问题。该方法充分利用所留历史信息,持续获得错误经验,不断吸取教训,对当代产生的最差方案进行惩罚,同时利用信息正反馈,对最优方案信息进行奖励。经验证,该方法

不仅能够提高算法的收敛速度,还改善了路径多样性,因此将信息素负反馈思想应用到本研究中。

### 4.2.3 超启发与负反馈相结合的信息素更新策略

本小节充分利用ACO<sub>i</sub>hh与ACON在算法更新时的优点,将文献[26]解决二元约束问题的思想引用到路径规划中,通过超启发式与信息素负反馈机制相结合(ACON<sub>hh</sub>)的方法,进行信息素更新,并且设置最大最小信息值,避免由于信息素挥发因子的原因,出现除较优路径以外的其他路径上信息过度减小的情况,影响算法性能。改进策略概括如下:

(1)每次迭代完成产生的路径集合用  $A_k$  表示。

(2)从  $A_k$  选出  $L_{\text{worst}}$  与  $L_{\text{better}}$ 。

(3)分别针对不同轨迹利用式(8)~(15)进行信息更新,且不在最差与较优范围内的轨迹利用式(16)进行更新,最后根据式(17)将信息素设定在一定范围内。

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (8)$$

$$\tau_{ij}^{\circ}(t+1) = (1-\rho(t)) \times \tau_{ij}(t) + \Delta\tau_{ij} \quad (9)$$

$$\tau_{ij}^{\circ}(t+1) = \frac{\mu \times \tau_{\max}}{t+1} + \Delta\tau_{ij} \quad (10)$$

$$\tau_{ij}(t+1) = \max\{\tau_{ij}^{\circ}(t+1), \tau_{ij}^{\circ}(t+1)\} \quad (11)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \xi \times \frac{Q}{L_{\text{best}}}, & \text{if } ij \in A_k, A_k \in L_{\text{best}} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$$\xi = \frac{L_{\text{best}}}{L_{\text{better}}} \quad (13)$$

$$\Delta\chi_{ij} = \begin{cases} \frac{Q}{L_{\text{worst}}}, & \text{if } ij \in A_k, A_k \in L_{\text{worst}} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

$$\chi_{ij}(t+1) = (1-\rho(t)) \times \chi_{ij}(t) + \Delta\chi_{ij} \quad (15)$$

$$\tau_{ij}(t+1) = (1-\rho(t)) \times (\tau_{ij}(t) + \chi_{ij}(t)) \quad (16)$$

$$\tau_{ij}(t+1) = \begin{cases} \tau_{\min}, \tau_{ij}(t+1) \leq \tau_{\min} \\ \tau_{ij}(t+1), \tau_{\min} < \tau_{ij}(t+1) \leq \tau_{\max} \\ \tau_{\max}, \tau_{ij}(t+1) > \tau_{\max} \end{cases} \quad (17)$$

式中,  $\Delta\tau_{ij}^k(t)$  为  $t$  时刻蚂蚁  $k$  在路径  $ij$  上留下的信息;  $\mu$  为一个常数,使  $t+1$  时刻信息量大小接近最大值;  $\rho(t)$  为  $t$  时刻,衡量残留信息挥发大小的变量;  $\tau(0)$  为  $t=0$  时信息素设定值大小;  $\xi$  值的大小,说明  $L_{\text{better}}$  与  $L_{\text{best}}$  的接近程度,  $L_{\text{better}}$  与  $L_{\text{best}}$  接近程度越大,表示  $L_{\text{better}}$  需要更新的信息素量越多;  $\Delta\chi_{ij}$  代表路径  $ij$  惩罚增量;  $\chi_{ij}(t+1)$  代表  $t+1$  时刻路径  $ij$  的惩罚总量;  $\tau_{\min}$  与  $\tau_{\max}$  分别为信息素最小值与最大值。

### 4.3 动态调整信息素挥发机制

基本ACA中系统会随机生成一个挥发因子常数  $\rho$ ,使蚂蚁在初期搜索速度慢,效率低,增加了搜索时间。因此,本文对  $\rho$  值做了适当调整,其随迭代次数的

变化成类抛物线形曲线。迭代起始和结束阶段  $\rho$  较小,信息素浓度较高,蚁群搜索路径单一,信息素给予蚂蚁较强的导向性,克服了蚁群在路径搜索过程中范围过大问题,由此加快了其搜索速率。迭代中期  $\rho$  较大,信息素量相对较少,其对蚂蚁的导向性较弱,在路径寻优时具有全局性,使蚂蚁跳出局部最优。  $\rho$  的类抛物线图简单描绘如图5所示。

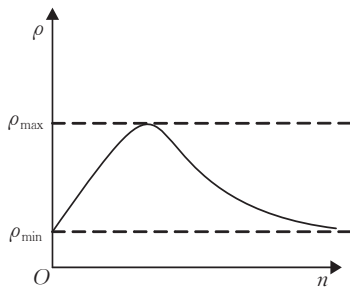


图5  $\rho$  的大致走向曲线

#### 4.4 算法流程图

本文提出的蚁群优化改进算法的流程如图6所示,算法流程概括如下:

步骤1 利用栅格法对小车所处的二维工作环境建立模型。

步骤2 初始化环境信息以及各个参数  $\alpha$ 、 $\beta$ 、 $\rho$ 、 $M$ 、 $N$ 、 $t$ 、 $n$  等,  $M$  表示蚂蚁总数量,  $N$  表示迭代总次数,  $t$  为计时器,  $n$  表示当代迭代次数,将  $k$  置于起点,启动蚂蚁。

步骤3 利用三角形原理以及式(1)选择节点。

步骤4 蚂蚁  $k$  每走一步,需要更新禁忌表,将所走节点存入禁忌表中。

步骤5 所有蚂蚁是否全部搜索,到达最终点,如果没有,返回步骤3,否则,继续下一步工作。

步骤6 到达最终点以后,选出  $L_{\text{worst}}$  与  $L_{\text{better}}$ 。

步骤7 对每代产生的轨迹利用式(8)~(17)进行信息更新。

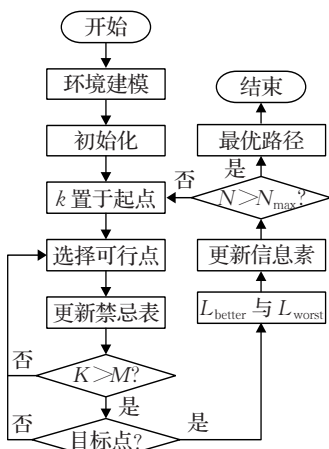


图6 ACONhh 算法流程图

步骤8 判断是否达到最大迭代次数,如果达到,则输出最优路径,否则,转至步骤2,直至符合最大迭代次数。

## 5 算法收敛性证明

### 5.1 符号说明

相关符号说明如表1所示。

表1 相关符号说明

符号	说明
$A_n$	首次找到最优解事件
$P(\bigwedge_{n=1}^N A_n)$	算法在前 $N$ 次循环中找到最优解的概率
$\Delta\tau_{ij}(n)$	第 $n$ 次循环,留在连接节点 $i$ 与 $j$ 路径上信息素的增量
$P(n, k)$	第 $n$ 次循环,蚂蚁 $k$ 找到最优路径的概率
$P(n)$	第 $n$ 次循环,所有的 $M$ 只蚂蚁都找不到最优解的概率

### 5.2 证明

给出假设:起点为  $S$ ,目标点为  $E$ ,每次循环有  $M$  只蚂蚁参加,一共循环  $N$  次,并且认为从  $S$  到  $E$  至少存在一条可走路线,  $n$  表示循环到第几次,则其对应概率为:

$$\lim_{n \rightarrow \infty} P(\bigwedge_{n=1}^N A_n) = 1 \quad (18)$$

证明 由假设及式(12)、(13)、(16)得出,  $\Delta\tau_{ij}(n) \geq 0$ ,进而通过式(8)、(9)、(10)、(11)可以得出下式:

$$\tau_{ij}(n) \geq \max \left( (1 - \rho(n))^n \times \tau(0), \frac{\mu \times \tau(0)}{n+1} \right) \quad (19)$$

假设  $k$  走到第  $i$  步后,位于点  $j$ ,且第  $n$  轮第  $k$  只蚂蚁收敛到最短路线的概率为  $P(n, k)$ 。由式(1)得出:

$$1 \geq P(n, k) = \prod_{i=1}^{t-1} \frac{\tau_{ij}^\alpha \times \eta_{ij}^\beta}{\sum_{s \in allowed_k} \tau_{is}^\alpha \times \eta_{is}^\beta} \quad (20)$$

式(20)可以变形为:

$$1 \geq P(n, k) = \prod_{i=1}^{t-1} \frac{\tau_{ij}^\alpha}{\sum_{s \in allowed_k} \tau_{is}^\alpha \times \left( \frac{\eta_{is}}{\eta_{ij}} \right)^\beta} \quad (21)$$

$$\text{设 } \phi_j = \left( \frac{\eta_{is}}{\eta_{ij}} \right)^\beta \quad (22)$$

$$\psi = \max \{ \phi_j \}, j = 1, 2, \dots, t-1 \quad (23)$$

则式(23)可变为:

$$1 \geq P(n, k) \geq \prod_{i=1}^{t-1} \frac{\tau_{ij}^\alpha}{\sum_{s \in allowed_k} \tau_{is}^\alpha \times \psi} \quad (24)$$

设第  $n$  次循环后,信息更新的最小值为  $\tau_{\min}(n)$ ,从而由式(10)得出:

$$\tau_{\min}(n) \geq \frac{\mu \times \tau_{\max}}{n+1}, \forall n \geq 1 \quad (25)$$

因概率大小与某一点周围的自由栅格数量  $N_c$  有关,  $N_{c \max}$  为节点  $j$  的最多的自由栅格数量,由式(24)、(25)得:

$$\prod_{i=1}^{t-1} \frac{\tau_{ij}^\alpha}{\sum_{s \in allowed_k} \tau_{is}^\alpha \times \psi} > \prod_{i=1}^{t-1} \frac{\left(\frac{\mu \times \tau_{\max}}{n+1}\right)^\alpha}{(\tau_{\max})^\alpha N_{\max} \psi} \quad (26)$$

随之,令

$$\theta = \left( \frac{\mu^\alpha}{N \times \psi} \right)^{t-1} \quad (27)$$

然后式(26)不等式右面简化得:

$$1 \geq P(n, k) > \frac{\theta_j}{(n+1)^{\alpha(t-1)}} \quad (28)$$

由式(28)得出:

$$1 > \frac{\theta_j}{(n+1)^{\alpha(t-1)}} > 0 \quad (29)$$

前  $n$  次循环都没有找到最短路径的概率为:

$$P\left(\bigcap_{n=1}^N A_n\right) = \prod_{n=1}^N P(n) = \prod_{n=1}^N (1 - P(n, k))^M \quad (30)$$

式(30)满足不等式:

$$\prod_{n=1}^N (1 - P(n, k))^M < \prod_{n=1}^N \left(1 - \frac{\theta_j}{(n+1)^{\alpha(t-1)}}\right)^M \quad (31)$$

对式(30)、(31)两边取对数,得:

$$\ln P\left(\bigcap_{n=1}^N A_n\right) < \prod_{n=1}^N \ln \left(1 - \frac{\theta_j}{(n+1)^{\alpha(t-1)}}\right)^M \quad (32)$$

进而整理得:

$$\prod_{n=1}^N \ln \left(1 - \frac{\theta_j}{(n+1)^{\alpha(t-1)}}\right)^M = M \prod_{n=1}^N \ln \left(1 - \frac{\theta_j}{(n+1)^{\alpha(t-1)}}\right) \leq -M \times \theta_j \left( \sum_{n=1}^N \frac{1}{(n+1)^{\alpha(t-1)}} \right) \leq -M \times \theta_j \sum_{n=1}^N \frac{1}{n+1} \quad (33)$$

$$\lim_{n \rightarrow \infty} \ln P\left(\bigcap_{n=1}^N A_n\right) <$$

$$\lim_{n \rightarrow \infty} \prod_{n=1}^N \ln \left(1 - \frac{\theta_j}{(n+1)^{\alpha(t-1)}}\right)^M = -\infty \quad (34)$$

$$\text{即 } P\left(\bigcap_{n=1}^N A_n\right) = 0 \quad (35)$$

最终得:

$$\lim_{n \rightarrow \infty} P\left(\bigcap_{n=1}^N A_n\right) = 1 \quad (36)$$

## 6 仿真及实验结果

### 6.1 仿真结果

为证明算法的可行性,利用 Matlab 仿真将蚁群置于两组不同复杂程度静态环境中,分别针对文献[19]中 ACO 算法(仅对信息素挥发因子的参数进行自适应调节)、文献[25]中 ACOhh 算法、改进超启发式蚁群算法(ACOihh)以及本文所提 ACONhh 算法做对比实验,每组独立重复 20 次实验,然后对其相应结果进行分析。实验数据如表 2 所示。

表2 相关参数列表

参数	数值	参数	数值
$m$	30	$Q$	1
$\alpha$	1	$\tau(0)$	0.7
$\beta$	1	$\tau_{\min}$	0.1
$\rho(0)$	0.2	$\tau_{\max}$	0.9
$\rho_{\min}$	0.2	$\chi(0)$	0.7
$\rho_{\max}$	0.8	$N_{\max}$	50

(1) 为了更好地说明基于信息负反馈的超启发式改进蚁群优化算法(ACONhh)能更高效地解决局部最优问题,分别将前述四种算法应用在  $20 \times 20$  障碍物较集中的栅格环境中进行仿真实验,起点栅格序号为 20,终点栅格序号 394,路径规划结果如图 7 所示。明显看到,运用文献[19]算法找到的轨迹出现严重局部最小值现象;文献[25]算法得到的最优轨迹得到改善,说明缩小更新对象能够提高算法搜索能力;然而改进后的超启发式蚁群优化(ACOihh)算法对较优路径进行信息素更新,相比前两种算法得出了较满意的路径轨迹,搜索能力增强,说明摒弃较差路径的信息素更新不仅减少了算法运行时间,同时也提高了寻优能力。最后,ACOhh 算法对比本文所提的 ACONhh 算法,由于 ACONhh 算法在初期能够分层化选择节点,在搜索过程中不断吸取错误经验,最终本文所提改进算法解决局部最优的能力明显提高,且路径缩短为原来的 13%。四种算法的迭代曲线如图 8 所示,文献[25]中由于更新对象缩小,总的迭代次数小于文献[19],说明收敛速度有所提高;通过观察图 13 与图 14,与 ACOihh 对比,ACONhh 算法收敛速度提高了 56%,且路径缩短,说明本文所提改进算法能够成功解决局部最优问题。

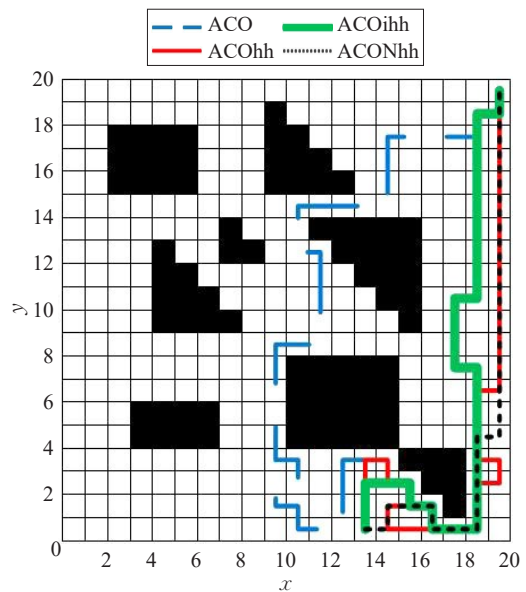


图7 四种算法路径规划结果图

(2) 为了进一步展现 ACONhh 算法的快速性,将 ACO、ACOihh、ACOihh 算法与 ACONhh 算法在  $30 \times 30$  稀



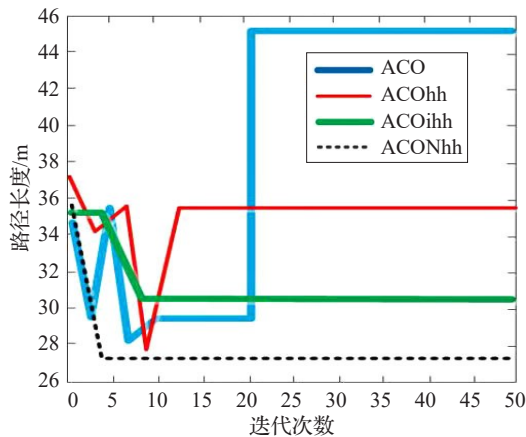


图8 四种算法迭代曲线

疏障碍物栅格环境中进行仿真实验,起点栅格序号为20,终点栅格序号为394,规划结果如图9所示,四种算法最终收敛的最优解分别为41、37、33、29。通过对比迭代曲线图10,由于ACOhh算法对最优路径进行信息素更新,收敛速度明显提高,且最优解缩短为原来的10%,说明摒弃对较差路径的信息素更新提高了收敛速度,且寻优能力有所改善,由于在ACOhh算法的基础上扩大了更新对象,对较优路径进行信息素更新,迭代速度变慢,但最优路径缩短为原来的10%,说明扩大更新范围,提高了解的多样性。通过对比迭代曲线图10,由于ACONhh在ACOihh的基础上引入了信息负反馈机制,分层化选择节点,给予蚂蚁较强的导向性,最优路径不仅缩短,且算法收敛速度显著提高。

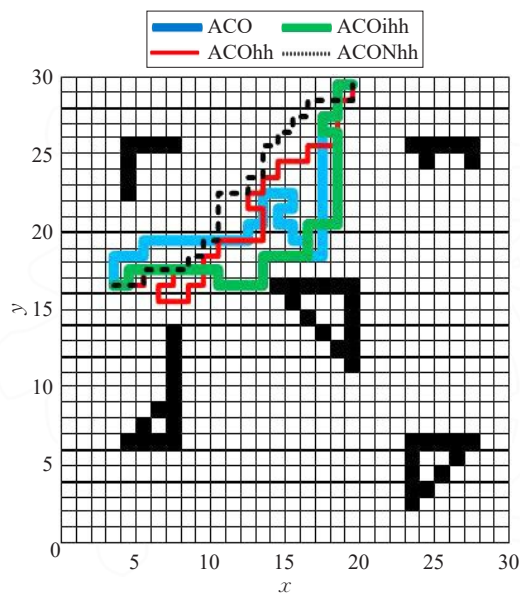


图9 四种算法路径规划结果图

将不同环境中对比算法的运行结果进行统计分析,如表3和表4所示。表3中,虽然ACO与ACOhh算法的迭代时间相对较短,但搜索出的路径轨迹容易落入极小点,使算法性能降低;对比ACOihh算法,由于扩大更新对象,迭代时间较长,但最优路径缩短。通过综合比较,

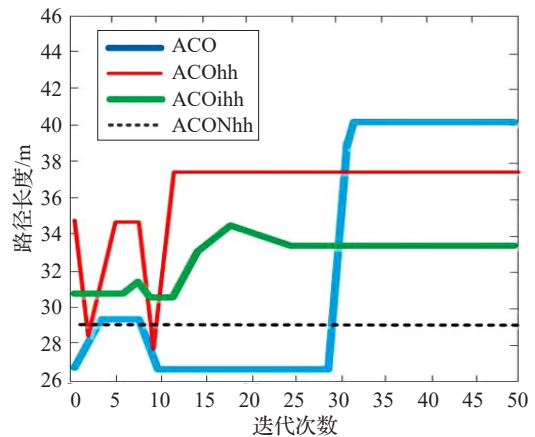


图10 四种算法迭代曲线

本文所提ACONhh算法在ACOihh算法基础上引入信息素负反馈机制以及分层化选择节点,运行时间相对缩短,同时能够使算法跳出局部最优点。表4进一步说明,本文引入的ACONhh算法收敛速度的快速性,与ACOihh相比时间缩短为原来的91.77%。

表3 算法运行结果统计表

环境	算法	最优路径	迭代次数	迭代时间/ms
20 m×20 m	文献[19]ACO	45	21	8 659
	文献[25]ACOhh	35	11	659
	ACOihh	31	9	1 596
	ACONhh	27	4	713

表4 算法运行结果统计表

环境	算法	最优路径	迭代次数	迭代时间/ms
30 m×30 m	文献[19]ACO	41	30	5 352
	文献[25]ACOhh	37	12	734
	ACOihh	33	25	4 462
	ACONhh	29	1	367

(3)本文通过六种典型的测试函数对ACONhh中引入的信息素负反馈机制、分层化选择节点和信息素挥发因子呈类抛物线变化三种机制相结合的优越性进行进一步评价,并将ACONhh与ACO、ACOhh、ACOihh算法进行对比。本文引入的六种典型测试函数的基本信息<sup>[28]</sup>如表5所示,每种测试函数的最大函数迭代次数均为1 000次,每组独立实验30次,四种算法的其他参数相关信息如表2所示。ACO、ACOhh、ACOihh与ACONhh算法在六种不同函数下的迭代优化曲线如图11~图16所示,表6给出了30次实验所得最优解的均值与方差。

表5 测试函数基本信息

测试函数	问题维数	搜索空间
Sphere	10	[-100,100]
Ellipse	10	[-5,5]
Rosenbrock	10	[-2.5,2.5]
Ackley	10	[-32,32]
Griewank	10	[-600,600]
Rastrigin	10	[-5.25,5.25]

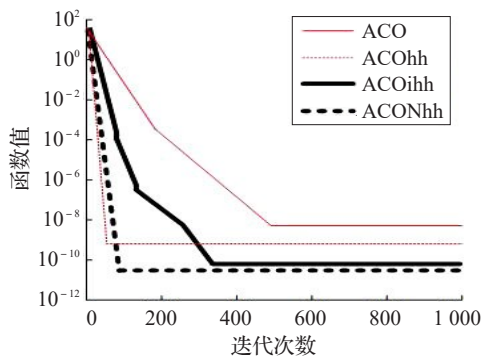


图 11 Sphere 迭代曲线

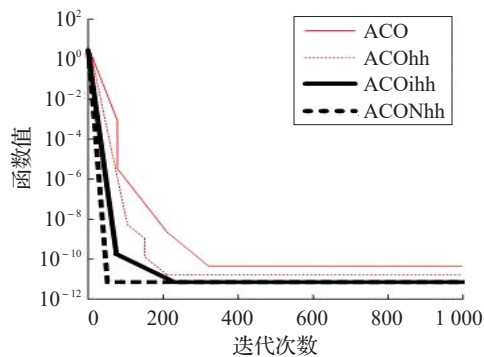


图 12 Ellipse 迭代曲线

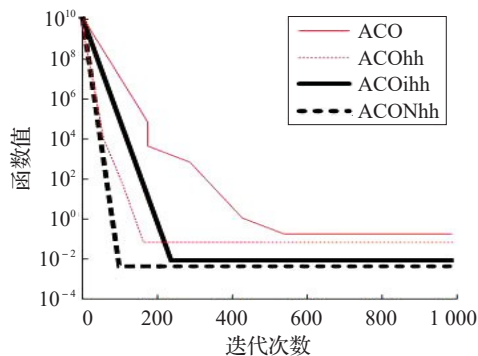


图 13 Rosenbrock 迭代曲线

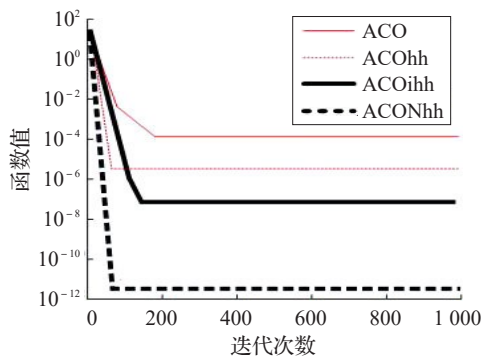


图 14 Ackley 迭代曲线

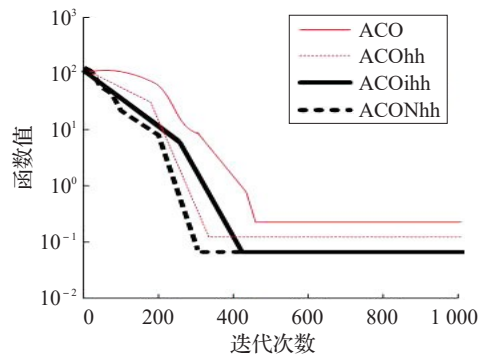


图 15 Griewank 迭代曲线

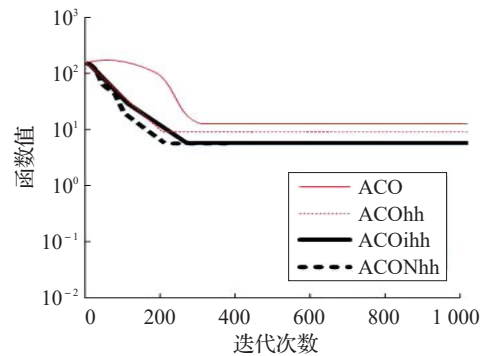


图 16 Rastrigin 迭代曲线

表 6 六种测试函数计算结果

函数	ACO		ACOhh		ACOihh		ACONhh	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
Sphere	$7.734 \times 10^{-8}$	$1.630 \times 10^{-8}$	$2.543 \times 10^{-9}$	$1.348 \times 10^{-9}$	$9.668 \times 10^{-11}$	$1.308 \times 10^{-11}$	$8.234 \times 10^{-11}$	$1.233 \times 10^{-11}$
Ellipse	$9.246 \times 10^{-11}$	0	$8.323 \times 10^{-11}$	0	$1.225 \times 10^{-11}$	0	$1.223 \times 10^{-11}$	0
Rosenbrock	$9.779 \times 10^{-1}$	$7.323 \times 10^{-1}$	$5.247 \times 10^{-1}$	1.043	$9.932 \times 10^{-3}$	$0.978 \times 10^{-3}$	$9.872 \times 10^{-3}$	$0.005 \times 10^{-3}$
Ackley	$8.588 \times 10^{-4}$	$3.543 \times 10^{-4}$	$9.439 \times 10^{-6}$	$6.732 \times 10^{-6}$	$4.726 \times 10^{-7}$	$2.132 \times 10^{-7}$	$8.533 \times 10^{-12}$	0
Griewank	0.573 2	0.041 2	0.279 3	0.172 3	0.079 1	0.039 9	0.078 9	0.005 4
Rastrigin	17.634	5.435	9.973	2.638	8.743	2.577	8.738	2.003

由表6可以看出,对于10维Sphere、Ackley、Rosenbrock三种测试函数,ACONhh算法寻优能力最好,尤其体现在Ackley测试函数中,均值最小,标准差为0;对于Ellipse、Griewank、Rastrigin三种函数,通过观察均值,ACONhh与ACOihh算法收敛精度相近,但在标准差指标中,ACONhh较ACOihh算法稳定。

图11~图16分别为30次实验所得的平均函数值随

迭代次数的变化过程曲线。通过观察图11,对于10维Sphere函数,ACONhh算法迭代到150代后达到精度要求,虽然收敛速度较ACOhh慢,但搜索能力强;图12、图15以及图16,ACONhh与ACOihh算法的收敛精度相近,但在迭代速度方面,ACONhh明显较ACOihh算法速度快;图13所示,对于10维Rosenbrock函数,ACONhh算法的寻优能力最好,收敛速度最快,ACOhh速度次



表7 实验相关结果

环境	算法	经过栅格序号	最优路径	迭代次数	迭代时间/ms
2.4 m×2.8 m	ACO	4-5-12-13-6-7-14-21-20-27-34-33-40-41-42	5.6	40	24 758
	ACOhh	4-5-12-13-20-21-28-27-34-33-40-41-42	4.8	20	2 578
	ACOihh	4-5-12-13-6-7-14-21-28-35-42	4.0	25	13 763
	ACONhh	4-5-6-13-20-27-34-35-42	3.2	10	2 632

之,ACOihh速度比ACOhh略差,而ACO算法速度最差。通过观察图14,对于10维Ackley测试函数,明显发现ACONhh算法以最快的迭代速度达到了精度要求,而其他三种算法的精度较差。

由以上实验可以得出,文献[19]中ACO算法收敛速度最慢,寻优能力最差;文献[25]中ACOhh算法仅对最优路径进行信息素更新,虽然收敛速度显著加快,但易陷入局部最优;ACOihh在ACOhh算法的基础上扩大更新对象,对较优路径进行信息素更新,搜索能力提高,但收敛速度低于ACOhh算法;ACONhh算法通过引入信息素负反馈机制,使蚂蚁不断吸收错误经验,分层化选择节点,最终收敛精度明显提高,收敛速度与ACOhh算法相近。因此本文将三种优化机制结合能够使ACONhh算法表现出良好的性能。

## 6.2 实验结果

为验证ACONhh算法应用于AGV时具有可行性及优越性,在2.4 m×2.8 m的室内环境中进行实验,障碍物在环境中是全局已知且静止的,将该环境划分为6×7的栅格,栅格单位长度为AGV外界圆直径即0.4 m,五角星代表AGV所处起始位置(栅格4),圆点表示AGV需要到达的目的地(栅格42),AGV通过获得所处位置,且通过与上位机通讯,分别按照ACO、ACOhh、ACOihh与ACONhh算法规划出符合一定性能指标的路径。四种算法的收敛曲线如图19所示,其中AGV利用ACOihh与ACONhh算法分别规划出的实际路线如图17和图18所示。通过观察四种算法迭代曲线图19以及表7的实验数据发现,利用ACOhh算法搜索到的最优路径较ACO缩短,迭代时间减少90%,在ACOhh算法的基础上扩大更新对象后,最优路径缩短为原来的17%,说明摒弃较长路径,仅对较优路径进行信息素更新,不但能够提高收敛速度,也能增加解的多样性。利用ACONhh算法规划出的路径较ACOihh缩短20%,迭代时间减少81%,说明该算法能够提高收敛速度,解决局部最优问题。

## 7 结束语

本文提出了一种适合AGV路径寻优的基于信息素负反馈的超启发式蚁群优化(ACONhh)算法,通过分层化选择节点,在加快算法初期收敛速度的同时改善全局性,采用信息素负反馈与改进超启发式相结合的更新策略,以及自适应调整信息素挥发系数,不仅加快了算法的寻优速度,搜索能力也得到了明显提高。从数学角度

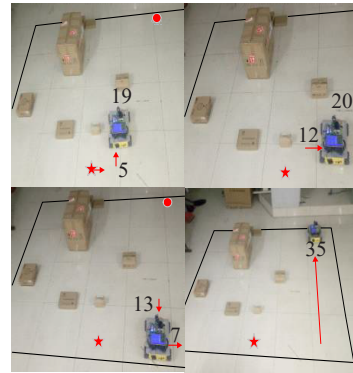


图17 AGV利用ACOihh算法行驶轨迹

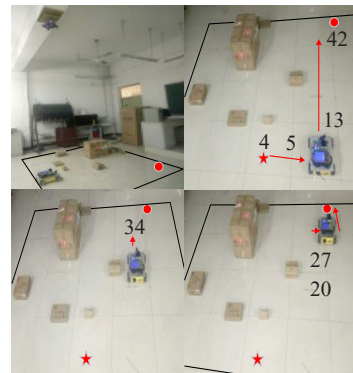


图18 AGV利用ACONhh算法行驶轨迹

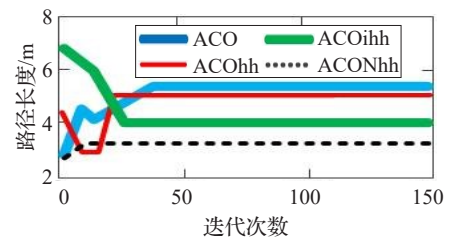


图19 四种算法迭代曲线

证明了只要有可行路径存在,用ACONhh算法就能搜索出一条路径。进一步通过仿真表明,ACONhh算法在不同栅格环境中运行时间减少,路径明显缩短。同时,将ACONhh算法应用到AGV实际环境中,验证了算法在实际应用中的有效性。综上所述,ACONhh算法在提高收敛速度的同时能够保证算法的全局性能,从而提高了AGV的运输效率,为实现立体仓储和AGV的智能化提供了新的思路。

## 参考文献:

- [1] Krug R, Stoyanov T, Tincani V, et al. The next step in robot commissioning: autonomous picking and palletizing[J].

- IEEE Robotics & Automation Letters, 2016, 1(1): 546-553.
- [2] Ignaciuk P. Nonlinear inventory control with discrete sliding modes in systems with uncertain delay[J]. IEEE Transactions on Industrial Informatics, 2013, 10(1): 559-568.
- [3] Kallasi F, Rizzini D L, Oleari F, et al. A novel calibration method for industrial AGVs[J]. Robotics & Autonomous Systems, 2017, 94(1): 75-88.
- [4] Tajti F, Burdelis M. A novel potential field method for path planning of mobile robots by adapting animal motion attributes[J]. Robotics & Autonomous Systems, 2016, 82(C): 24-34.
- [5] Guruji A K, Agarwal H, Parsediya D K. Time-efficient A\* algorithm for robot path planning[J]. Procedia Technology, 2016, 23: 144-149.
- [6] 肖云涛, 欧林林, 俞立. 基于线性时序逻辑的最优巡回路径规划[J]. 自动化学报, 2014, 40(10): 2126-2133.
- [7] Khaksar W, Tang S H, Sahari K S M, et al. Sampling-based online motion planning for mobile robots: utilization of tabu search and adaptive neuro-fuzzy inference system[J]. Neural Computing & Applications, 2017(9/10): 1-15.
- [8] Mac T T, Copot C, Tran D T, et al. Heuristic approaches in robot path planning: a survey[J]. Robotics & Autonomous Systems, 2016, 86: 13-28.
- [9] Byrski A, Świdarska E, Łasiz J, et al. Socio-cognitively inspired ant colony optimization[J]. Journal of Computational Science, 2017, 21: 397-406.
- [10] 万正宜, 彭玉旭. 求解旅行商问题的改进型量子蚁群算法[J]. 计算机工程与应用, 2016, 52(22): 59-63.
- [11] Williams H L. Intellectual property rights and innovation: evidence from health care markets[J]. Innovation Policy and the Economy, 2016, 16(1): 53-87.
- [12] Juang C F, Jeng T L, Chang Y C. An interpretable fuzzy system learned through online rule generation and multi-objective ACO with a mobile robot control application[J]. IEEE Transactions on Cybernetics, 2016, 46(12): 2706-2718.
- [13] Chen Y, Wong M L, Li H. Applying ant colony optimization to configuring stacking ensembles for data mining[J]. Expert Systems with Applications, 2014, 41(6): 2688-2702.
- [14] 侯梦婷, 赵作鹏, 高萌, 等. 采用角度因子的蚁群优化多路径路由算法[J]. 计算机工程与应用, 2017, 53(1): 107-112.
- [15] daSilva J F R, Altafim R A C, Hirakawa A R. Cross-arms identification with adaptive digital image processing[J]. International Journal of Computer Applications, 2015, 121(23): 36-39.
- [16] Wilensky G R. The future of the ACA and health care policy in the United States[J]. JAMA, 2017, 317(1): 21.
- [17] Tang B W, Zhu Z X, Fang Q, et al. Path planning and replanning for intelligent robot based on improved ant colony algorithm[J]. Applied Mechanics and Materials, 2013, 390: 495-499.
- [18] 赵娟平, 高宪文, 符秀辉, 等. 移动机器人路径规划的改进蚁群优化算法[J]. 控制理论与应用, 2011, 28(4): 457-461.
- [19] Wang Z, Zhu X, Han Q. Mobile robot path planning based on parameter optimization ant colony algorithm[J]. Procedia Engineering, 2011, 15(1): 2738-2741.
- [20] Zheng F, Zecchin A C, Newman J P, et al. An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(5): 773-791.
- [21] Châari I, Koubâa A, Bennaceur H, et al. On the adequacy of tabu search for global robot path planning problem in grid environments[J]. Procedia Computer Science, 2014, 32: 604-613.
- [22] Zhao J, Fu X. Improved ant colony optimization algorithm and its application on path planning of mobile robot[J]. Journal of Computers, 2012, 7(8): 1-4.
- [23] Calle J, Rivero J, Cuadra D, et al. Extending ACO for fast path search in huge graphs and social networks[J]. Expert Systems with Applications, 2017, 86: 292-306.
- [24] 何娟, 涂中英, 牛玉刚. 一种遗传蚁群算法的机器人路径规划方法[J]. 计算机仿真, 2010, 27(3): 170-174.
- [25] Aziz Z A. Ant colony hyper-heuristics for travelling salesman problem[J]. Procedia Computer Science, 2015, 76: 534-538.
- [26] Ye K, Zhang C, Ning J, et al. Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems[J]. Information Sciences, 2017, 406/407: 29-41.
- [27] Wang J, Cao J, Sherratt R S, et al. An improved ant colony optimization-based approach with mobile sink for wireless sensor networks[J]. Journal of Supercomputing, 2017(8): 1-13.
- [28] 周袅, 葛洪伟, 苏树智. 基于信息素的自适应连续域混合蚁群算法[J]. 计算机工程与应用, 2017, 53(6): 156-161.