

文章编号: 2095-2163(2020)09-0103-06

中图分类号: TP391.1

文献标志码: A

基于改进粒子群蚁群融合算法的多 AGV 路径规划

熊昕霞, 何利力

(浙江理工大学 信息学院, 杭州 310018)

摘要: 为确定多自动导引小车 (AGV) 在复杂仓库环境下最优路径, 本文提出了一种改进粒子群蚁群融合算法, 用于多 AGV 路径规划。为了最小化所有 AGV 到达仓库中各自目的地的时间并且无碰撞, 本文建立了以每个机器人路径最短、转弯角度最小以及与障碍物保持安全距离为约束的目标函数, 采用所提的混合粒子群蚁群优化算法求解。仿真结果表明, 在优化路径长度、安全性以及到达时间方面都优于改进的粒子群优化算法、蚁群优化算法。

关键词: 自动导引小车; 路径规划; 粒子群优化; 蚁群优化

Multi-AGVs path planning based on improved particle colony and ant colony fusion algorithm

XIONG Xinxia, HE Lili

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

[Abstract] In order to determine the optimal path of multiple automated guided vehicles (AGV) in a complex warehouse environment, this paper proposes a hybridization of an improved particle swarm optimization and ant colony optimization algorithm for path planning of multiple AGVs. In order to minimize the time for all AGVs to reach their respective destinations in the warehouse without collision, this paper establishes an objective function that takes the shortest path of each robot, the smallest turning angle, and the safe distance to obstacles as constraints. The proposed hybrid particle colony ant optimization algorithm is used to solve the problem. The simulation results show that it is better than the improved particle swarm optimization algorithm and ant colony optimization algorithm in terms of optimizing path length, safety and arrival time.

[Key words] automatic guided vehicles; path planning; particle swarm optimization; ant colony optimization

0 引言

在当今的制造业中, 战略已从制造大量单个产品转移到了一系列产品, 并改善了质量和交货时间。许多物流和制造流程都依赖于多 AGV 系统的使用^[1], 系统中的每辆车都执行运输任务, 包括在给定的初始位置提取货物并在所需的目的地进行运输和卸载^[2]。多 AGV 系统运行效率直接影响整个制造车间的高效运转。为了改善多 AGV 系统的运行效率和安全性, 要解决的主要问题是正确规划多 AGV 的路径^[3]。多 AGV 路径最优轨迹是实现基于多 AGV 智能生产车间的关键。

从初始位置到目标位置通过检测并避开障碍物是自动导引小车在无人仓库中的最重要功能。因此, 在简单或复杂的环境中工作时, 正确选择导航技术是机器人路径规划中最重要的步骤。多 AGV 路径规划是当前研究的一个热点和难点^[3]。张丹露等提出一种基于预约表和动态加权地图的改进 A*

算法, 应用在多机器人路径规划、避障和拥堵问题上^[4], 改进的 A* 算法提出预约表与交通规则结合的方式解决了多机器人的碰撞问题, 根据生成的动态加权地图解决了拥堵问题; 袁洋等研究多 AGV 在运行时单个 AGV 的路径规划问题, 提出了使用负载均衡改进的 A* 算法进行路径规划的方法, 解决多 AGV 运行路网的局部拥塞、防止和负载均衡问题, 将路网负载引入到 A* 算法的评价函数中, 避免多 AGV 路径规划时产生局部车流量过大^[5]; 张毅等根据 AGVs 剩余电量分配优先级, 协调 AGVs 路径, 以提高小车的效率, 再使用改进的蚁群算法计算 AGVs 的最优路径^[3]。

在实际应用中, 大多数多 AGV 路径规划研究都是以无障碍、AGV 之间的协调、车间不堵塞等路线调整为研究目标, 很少将路径偏差和能源消耗视作关键因素。本文提出了一种改进粒子群蚁群融合算法, 使每个 AGV 的路径长度到达时间最小化。以解

基金项目: 国家重点研发计划(2018YFB1700702)。

作者简介: 熊昕霞(1996-), 女, 硕士研究生, 主要研究方向: 计算机软件及计算机应用; 何利力(1966-), 男, 博士, 教授, 博士生导师, 主要研究方向: 数据分析、企业智能。

通讯作者: 何利力 Email: 11he@zju.edu.cn

收稿日期: 2020-06-14

决仓库环境中的多 AGV 路径规划问题。

本文的主要工作如下:第一阶段设立目标函数,使每个小车的路径成本、路径偏差和能耗最小化。利用混合算法为每个移动机器人规划出一条最佳路径;第二阶段,障碍物检测,使每个机器人都避免与障碍物和其他机器人发生碰撞,解决充满障碍物的仓库环境中多 AGV 路径规划问题;第三阶段,仿真实验,将获得的结果与改进的粒子群优化(PSO)、ACO 算法进行比较。

1 问题描述与模型构建

1.1 问题描述与假设条件

在无人仓库中,为达到目标位置完成运输任务,要求各个机器人从起点出发,绕过静态障碍物,并且避免与其他机器人相撞,到达目标位置完成运输任务。从仓库中的当前位置计算每个机器人的下一个位置目标。该路径的目标是在不与问题冲突的情况下找到最优或接近最优的路径(最安全、最短和最光滑)。在制定多机器人路径规划问题时要考虑以下约束条件。假设:

- (1) 仓库地图已知;
- (2) 已知机器人的起始位置和目标位置;
- (3) 每个机器人逐步规划动作,直到到达各自的目标位置;
- (4) 机器人到达下一个位置时可能会与仓库中其他机器人/障碍物相撞。

因此,机器人前进方向将以一定角度向左或向右转,从而确定其下一个位置。本文提出的混合算法通过假设机器人的当前位置和速度作为优化给定的多目标函数的参数,来评估每个机器人的下一个位置。其确定了动态和静态环境中从每个状态到目标状态的最佳路径,机器人借助配备的传感器来测量其到障碍物的距离。

1.2 模型建立

假如机器人在时间 t 的初始位置是 (x_i^p, y_i^p) , 目标位置为 (x_i^g, y_i^g) 。该机器人下一个位置公式(1)和(2):

$$x_i^n = x_i^p + v_i^p \cos \theta_i \Delta t, \quad (1)$$

$$y_i^n = y_i^p + v_i^p \sin \theta_i \Delta t. \quad (2)$$

当 $\Delta t = 1$ 时,式(1)、(2)为式(3)、(4):

$$x_i^n = x_i^p + v_i^p \cos \theta_i, \quad (3)$$

$$y_i^n = y_i^p + v_i^p \sin \theta_i. \quad (4)$$

$$f_4(x, y) = \frac{\cos^{-1} [(x_i^p - x_i^g)(x_{g^{best_{i-1}}} - x_i^g) + (y_i^p - y_i^g)(y_{g^{best_{i-1}}} - y_i^g)]}{\sqrt{(x_i^p - x_i^g)^2 + (y_i^p - y_i^g)^2} \times \sqrt{(x_{g^{best_{i-1}}} - x_i^g)^2 + (y_{g^{best_{i-1}}} - y_i^g)^2}}. \quad (9)$$

第一个函数 f_1 使得每个机器人当前位置到目标

其中, v_i^p 是机器人 R_i 的速度, (x_i^p, y_i^p) 为该机器人在 $(t + \Delta t)$ 时的下一个位置。

该多机器人路径规划问题将作为路径和能耗都得到优化的问题提出。优化问题被表述为基于约束的多目标函数。目标函数用于表示最小化每个机器人当前位置到其目的地之间的路径长度,避免与其他机器人和静态障碍物碰撞的同时优化生成路径的平滑度。

1.2.1 路径长度

路径长度影响机器人的运输时间,为更快完成任务,通过形成约束 f_1 使得机器人从当前位置到目标位置总路径长度最小,并且避开障碍物。约束公式为(5):

$$f_1 = \sum_{i=1}^n \left\{ \sqrt{((x_i^p - x_i^n)^2 + (y_i^p - y_i^n)^2)} + \sqrt{((x_i^n - x_i^g)^2 + (y_i^n - y_i^g)^2)} \right\}. \quad (5)$$

将式(3)(4)代入得到式(6):

$$f_1 = \sum_{i=1}^n \left\{ v_i^p + \sqrt{\frac{\alpha}{\beta} x_i^p - v_i^p \cos \theta_i - x_i^g)^2 + \frac{\alpha}{\beta} (y_i^p - v_i^p \sin \theta_i - y_i^g)^2} \right\}. \quad (6)$$

1.2.2 与障碍物的安全距离

为防止机器人与障碍物碰撞,定义了每个障碍物的排斥场,式(7):

$$f_2(X_p) = \begin{cases} \frac{k}{\gamma} \frac{\alpha}{d_{min}(X_p)} - \frac{1}{\eta_0} \frac{\ddot{\alpha}}{\delta}, & d_{min}(X_p) \leq \eta_0 \\ 0, & \text{其他} \end{cases}. \quad (7)$$

其中, η_0 是障碍物的影响范围; k 是常数; $\gamma \geq 2$ 形成电位的径向轮廓; $d_{min}(X_p)$ 为 X_p 到障碍物的最小距离。

1.2.3 预测动态障碍物(其他机器人)位置

动态障碍物将动态出现在机器人的路径轨迹中。因此,在决定下一位置前,机器人需预测动态障碍物的轨迹,式(8):

$$f_3 = \sum_{i=1}^n \sqrt{(x_p - x_i^g)^2 + (y_p - y_i^g)^2}. \quad (8)$$

其中, (x_p, y_p) 是动态障碍物的预计位置。

1.2.4 路径平滑度

最小化直线(目标点到建议点)之间的角度差,使得路线平滑度最大,减少机器磨损,提升机器人运行稳定性。路径平滑度目标函数为式(9):

位置的欧几里得距离最小;第二和第三个惩罚函数

f_2, f_3 用于避免移动机器人与动态障碍物之间的碰撞;第四个惩罚函数 f_4 是平滑路径。总目标(适应度)函数为式(10):

$$fit = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4. \quad (10)$$

其中, $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ 是4个函数的权重,满足条件式(11):

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1. \quad (11)$$

这些权重在仿真实验中做出调整, $\lambda_1 = 0.3, \lambda_2 = 0.3, \lambda_3 = 0.3, \lambda_4 = 0.1$, 通过使用适应度函数最小化来获取最优路径。

2 通过蚁群机制中信息素使用改进的粒子群优化

2.1 粒子群优化

PSO算法是由J. Kennedy和RC Eberhard于1995年提出的基于种群的优化问题启发式策略。与鸟群和鱼类群的行为类似,在PSO中,优化问题的候选解决方案集被定义为可能通过参数(搜索)空间定义轨迹的粒子群,这些空间由它们自己和邻居驱动的最佳表现^[6]。PSO中的每个粒子的特征在于其位置 x_i 和速度 v_i 。位置表示粒子所在搜索区域的位置,而速度是相对于当前位置到下一个位置的变化率。令 N 为粒子群总数,第 i 个例子在 D 维空间的坐标为 $x_i = (x_i^1, x_i^2, \dots, x_i^D)$, 其中 $1 \leq i \leq N$, 速度向量 $v_i = (v_i^1, v_i^2, \dots, v_i^D)$ 。每一代的第 d 维($1 \leq d \leq D$) 根据式(12)和式(13)更新:

粒子的速度更新:

$$v_i^d(t+1) = v_i^d(t) + c_1 r_1 (x_{pbest_i}^d - x_i^d(t)) + c_2 r_2 (x_{gbest}^d - x_i^d(t)). \quad (12)$$

粒子的位更新:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \quad (13)$$

其中, $v_i^d(t)$ 和 $x_i^d(t)$ 分别是在第 d 维粒子 i 第 t 次迭代的速度和位置; c_1 和 c_2 为加速常数; r_1 和 r_2 为两个在 $[0,1]$ 范围内变化的随机数; $x_{pbest_i} = \{x_{pbest_i^1}, x_{pbest_i^2}, \dots, x_{pbest_i^D}\}$ 为第 i 个粒子在第 t 次迭代中的最好的位置; $x_{gbest}^d(t)$ 为所有粒子在第 t 次迭代中经历过的最好的位置。

为了更好地控制开发和探索,Shi和Eberhard提出了有关惯性权重的PSO,其中更新了每个粒子的速度,为全局搜索提供更大的惯性权重值,而较小该值将提供局部搜索。为提高PSO的收敛速度,本文将在自适应权重调整和加速度系数方面改进PSO。根据以式(14)和(15)修改:

$$v_i^d(t+1) = w v_i^d(t) + c_1 r_1 (x_{pbest_i}^d - x_i^d(t)) + c_2 r_2 (x_{gbest}^d - x_i^d(t)). \quad (14)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \quad (15)$$

2.2 改进的粒子群优化

PSO的收敛速度借助多种技术对其参数微调而得到了改善。为了避免群里发散而引入惯性权重 w ,通过调节先前迭代中速度对粒子的定义来控制粒子的速度。因此,当惯性权重值较大时,将实现全局搜索;当惯性权重值较小时,将进行局部搜索。多项研究表明,通过动态改变惯性权重可以提高PSO的搜索能力,而线性递减惯性权重已经显示出更好的结果^[7]。因此,这里采用惯性权重的自适应方法,如式(16)所示。过去已有实验验证,采用组合 $w_{max} = 0.9, w_{min} = 0.4$ 可以达到最佳性能。

$$w_i = w_{min} + (w_{max} - w_{min}) \frac{dist_i}{max_dist}, \quad (16)$$

$$dist_i = \left(\sum_{d=1}^D x_{gbest_d} - x_i^d \right)^{1/2}, \quad (17)$$

$$max_dist = \operatorname{argmax}(dist_i). \quad (18)$$

其中, $dist_i$ 为粒子 i 与全局最佳粒子之间的欧几里得距离,如式(17)所示; x_{gbest_d} 为第 d 次迭代的全局最优位置; max_dist 是一代中粒子与全局最佳粒子之间的最大距离,如式(18)所示。

由于混沌行为类似于随机性,但具有更好的动力学和统计学特性,与随机搜索相比,混沌搜索可以更容易地从局部最优解中逃脱^[8]。因此,将随机参数 r_1, r_2 替换成一维混沌图,使得算法收敛速度更快。已有实验验证,将随机参数 r_1 替换成混沌图获得的结果最佳,并且Singer映射用于该算法的结果最优^[9]。替换后为式(19):

$$v_i^d(t+1) = w_i v_i^d(t) + c_1 r_1 (pbest_i^d(t) - x_i^d(t)) + c_2 \xi (gbest^d(t) - x_i^d(t)). \quad (19)$$

Singer映射定义为式(20):

$$x_{k+1} = \mu (7.86x_k - 23.3x_k^2 + 28.75x_k^3 - 13.3x_k^4). \quad (20)$$

其中, μ 是一个0.9到1.08之间的参数。

2.3 蚁群算法

蚁群算法(ACO)是一个受实际蚁群行为启发的人工系统。在蚂蚁搜索路径过程中,会留下信息素踪迹,该信息素会随时间和距离消散。当蚂蚁经过下一个位置时,或当更多的蚂蚁经过该位置时,该位置上的信息素强度会更高。因此,遵循信息素尾迹的蚂蚁将更具以下情况聚集:信息素“浓度”随着更随尾迹的每增加一个蚂蚁而增加。

在标准ACO中,蚂蚁每一步位置转移按照转盘赌的方式做出概率选择,再将信息素沿其踪迹更新

食物来源。 t 时刻蚂蚁 k 由位置 i 转移到 j 的转移概率为(21):

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t)^\alpha)((\eta_{ij})^\beta)}{\sum_{i \in J_i^k} (\tau_{ij}(t)^\alpha)((\eta_{ij})^\beta)}, & (i,j) \in J_i^k; \\ 0, & (i,j) \notin J_i^k. \end{cases} \quad (21)$$

其中, $\tau_{ij}(t)$ 为当前位置到目标点在 t 时刻的信息素强度; η_{ij} 为启发式因子(能见度); α 和 β 是指定信息素轨迹和启发式因子相对影响的二个因素, J_i^k 是蚂蚁 k 尚未访问的可行域里点的集合。

每只蚂蚁完成一次搜索后,更新信息素迹线。首先以恒定的蒸发速率降低它们,然后依次使每只蚂蚁在其经过的一部分弧上沉积信息素,如式(22):

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^M \Delta\tau_{ij}^k. \quad (22)$$

其中, $\Delta\tau_{ij}^k$ 取决于式(23):

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L^k(t)}, & (i,j) \in T^k(t); \\ 0, & (i,j) \notin T^k(t). \end{cases} \quad (23)$$

其中, $L^k(t)$ 是行程的长度, Q 是固定参数。

2.4 融合改进粒子群优化和蚁群优化算法

由于依赖于外部参数(例如:惯性权重和加速度参数),传统的 PSO 算法难以生成最优解,收敛速度较慢。为了解决优化问题,摆脱局部最小值,最大化探索粒子的搜索范围,通过引入蚁群算法中的信息素协同粒子群优化,如式(24)。

$$v_i^d(t+1) = v_i^d(t+1)_{IPSO} + C_3\tau_{i-1,i}. \quad (24)$$

其中, C_3 为加速度系数,用于调整 IPSO - ACO 上的 PSO 速度。如果 C_3 设置为零,则 IPSO - ACO 成为独立的 IPSO;如果 C_3 设置为 1,则会受到 ACO 的一半的随机影响。

信息素轨迹是转递到构造图边缘的真实值,被视为过程强信息的存储器,这些信息涉及单个解决方案组件在先前的迭代中的效果。 $\tau_{i-1,i}$ 是从点 $i-1$ 到点 i 的信息素,对应于一个由吸引力的线连接的两个点。因此,在混合过程中,IPSO 使用内存来保存迄今为止找到的最佳解决方案,通过信息素确认该粒子将被加速。当所有粒子接近良好的解决方案时,移动都非常缓慢,并且 g_{best} 可以帮助其充分利用全局最佳方法。每个粒子都观察到最佳解并向其移动。

3 仿真实验

3.1 路径求解流程

本文使用混合算法 IPSO-ACO 对充满障碍物的无人仓库中多 AGV 进行路径规划。在该算法中,

从每个 AGV 的现有位置计算每个 AGV 的连续最佳位置,同时满足所有约束并提高收敛速度。使用本文提出的混合算法进行多 AGV 路径规划流程,如图 1 所示。

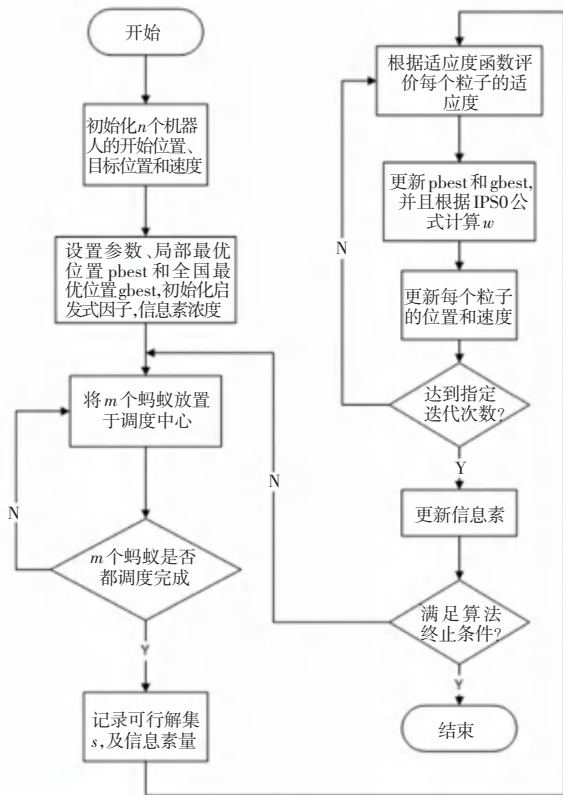


图 1 IPSO-ACO 算法流程图

Fig. 1 Flowchart of IPSO-ACO

3.2 静态障碍物下多机器人路径规划

仿真实验使用 Python 语言实现,使用图 2 作为模拟环境来实现 AGV 路径规划问题。模拟环境中,不同颜色的圆形为每个 AGV,正方形为每个 AGV 对应的目标位置。障碍物随机分布在模拟环境中。

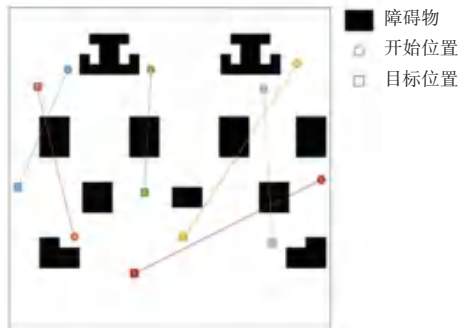


图 2 仿真环境图

Fig. 2 Simulation environment

实验时障碍物和环境是静止的,其他优先级 AGV 可以移动。在仿真实验中设置的算法相关参

数见表 1。每个 GAV 根据定义的适应度值来决定其下一个位置。使用 IPSO-ACO 算法使每个 AGV 到达其相应位置,虚线表示每个 AGV 的路径,如图 3 所示。从图 3 中可知,6 台 AGV 都分别到达了目标位置,第 3 台和第 5 台 AGV 在第 13 步时距离非常接近,在路段上可能发生冲突,第 3 台将第 5 台 AGV 视作动态障碍物,重新计算其路径,避开动态障碍物。

3.3 算法比较

为评估 IPSO-ACO 的性能,运用 IPSO 和 ACO 算法在上述环境中对 6 台 AGV 进行路径规划,算法相关参数设置与 IPSO-ACO 相同。AGV 通过三种不同算法到达其对应目标位置所需的时间进行了比较,其折线图如图 4 所示。由图 4 可知使用本文算法为所有机器人到达目标规划的路径比其他两种算法每步所需时间更短。图 5 显示了三种不同的启发

式算法的每一步的目标距离对比,与 IPSO 和 ACO 相比,使用 IPSO-ACO 算法到达目标位置的距离最短,且最终的路径长度以及所需步数都较小。

表 1 算法相关参数

Tab. 1 Related parameters of the algorithm in this paper		
参数	含义	参数值
α	信息素重要程度	1
β	启发式因子重要程度	7
ρ	信息素蒸发系数	0.3
Q	信息素增加强度系数	1
<i>dimension</i>	维度	10
<i>No_of_particles</i>	粒子数量	100
<i>iter_max</i>	最大迭代次数	500
w_{min}	自适应惯性权重最小值	0.4
w_{max}	自适应惯性权重最大值	0.9
C_1	加速度系数	0.5
C_2	加速度系数	0.5
C_3	加速度系数	0.5

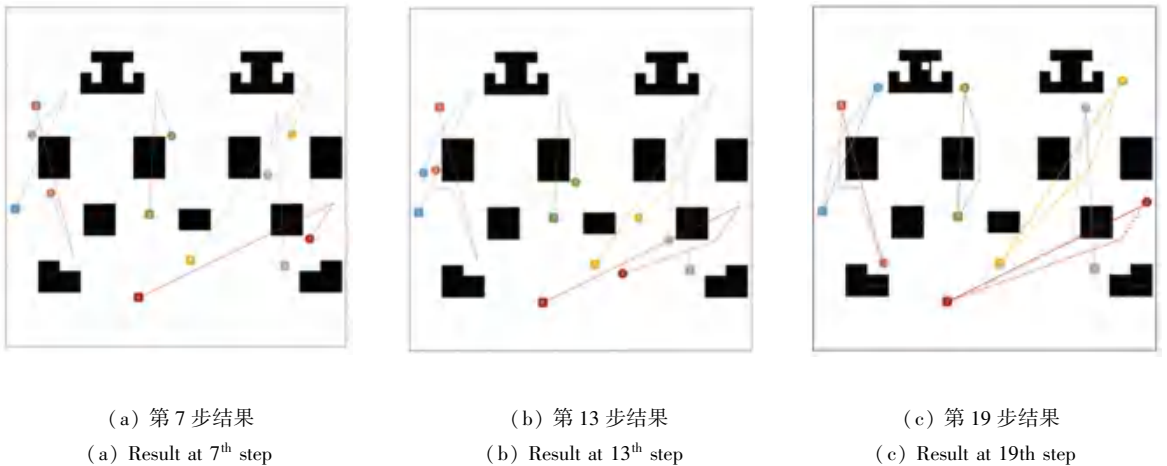


图 3 多 AGV 路径规划结果
Fig. 3 Results of multi-AGV path planning

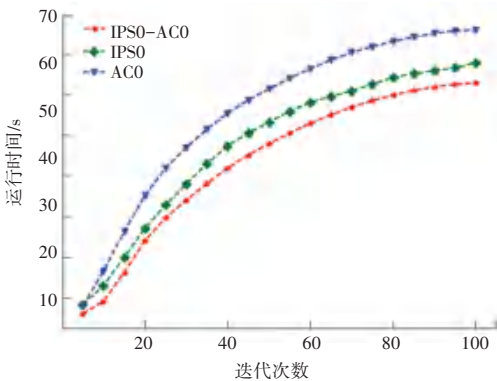


图 4 三种算法迭代次数和运行时间

Fig. 4 Number of iterations and running time of the three algorithms

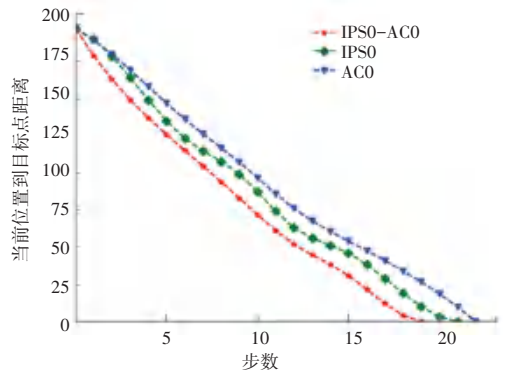


图 5 三种算法的步数和平均目标距离

Fig. 5 The number of steps and average target distance of the three algorithms

4 结束语

本文提出了一种改进粒子群蚁群融合算法,以提高粒子群优化算法的搜索能力,避免陷入局部最小值,实现更好的收敛性。使用该混合算法解决充满障碍物的无人仓库中多 AGV 路径规划。通过软件模拟,并与 IPSO 和 ACO 算法对比,该方法的效率更高,优化效果更好。

参考文献

- [1] DRAGANJAC I, MIKLIC D, KOVACIC Z, et al. Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications[J]. IEEE Transactions on Automation Science and Engineering, 2016, 13(4): 1433-1447.
- [2] 王鑫森. 仓储作业中多搬运机器人动态路径规划研究[D]. 北京交通大学, 2019.
- [3] ZHANG Y, WANG F, FU F, et al. Multi-AGV Path Planning for Indoor Factory by Using Prioritized Planning and Improved Ant Algorithm[J]. Journal of Engineering and Technological Sciences, 2018, 50(4): 534-547.

- [4] 张丹露,孙小勇,傅顺,等. 智能仓库中的多机器人协同路径规划方法[J]. 计算机集成制造系统, 2018, 24(2): 410-418.
- [5] 袁洋,叶峰,赖乙宗,等. 结合负载均衡与 A* 算法的多 AGV 路径规划[J]. 计算机工程与应用, 2020, 56(5): 251-256.
- [6] DAS P K, BEHERA H S, PANIGRAHI B K, et al. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning[J]. Swarm and evolutionary computation, 2016; 14-28. <http://dx.doi.org/10.1016/j.swevo.2015.10.011>. DOI: 10.1016/j.swevo.2015.10.011.
- [7] MARINI F, WALCZAK B. Particle swarm optimization (PSO). A tutorial[J]. Chemometrics and Intelligent Laboratory Systems, 2015; 153-165.
- [8] MIRJALILI S, GANDOMI A H. Chaotic gravitational constants for the gravitational search algorithm [J]. Applied Soft Computing, 2017; 407-419.
- [9] THARWAT A, ELHOSENY M, HASSANIEN A E, et al. Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm [J]. Cluster Computing, 2019, 22(2): 4745-4766.

(上接第 99 页)

参考文献

- [1] 武翔宇. 基于图像的接触网支柱检测与编号识别[D]. 西南交通大学, 2015.
- [2] 徐可佳. 基于深度学习的接触网支柱号在线识别算法研究[J]. 电气化铁道, 2012, 8(1): 140-150.
- [3] 刘华春,侯向宁. 采用视觉显著性和卷积网络的车牌定位算法[J]. 计算机技术与发展, 2019, 29(6): 74-78.

- [4] 陈曦,范敏,熊庆宇. 基于马尔科夫链的显著性区域检测 算法研究[J]. 计算机工程与应用, 2016, 52(7): 171-175.
- [5] 姜博文. 基于马尔科夫链的显著性检测 [D]. 大连理工大学, 2014.
- [6] 李青,张正. 基于局部三进制模式的边缘检测方法[J]. 软件导刊, 2015, 14(1): 129-131.

(上接第 102 页)

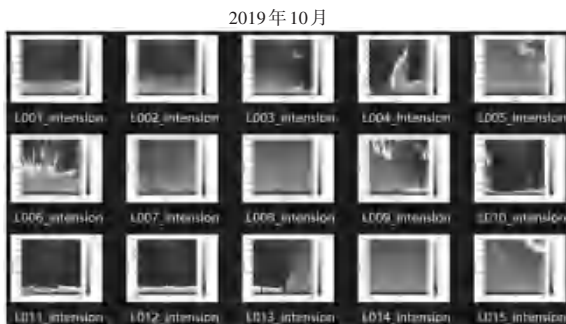


图 8 多日气象伪彩图

Fig. 8 Pseudo-color map of weather over days

3 结束语

本文依据已有的实际陆基激光雷达采集的信号

数据信息,初步筛选后得到有效数据,运用后向散射及 Mie 散射原理编制 MATLAB 算法程序,将有效数据经算法转化得到一系列可视化结果图像,对图中呈现的关键参数信息进行分析。结果表明,当前算法得出的不同类型图像所体现的信息,均与客观实际的气象情况一致,证实了方法的可行性与分析的合理性,可进行下一步实际应用的研究与设计。

参考文献

- [1] 李霖峰. 光后向散射法测量烟尘浓度的理论与实验研究[D]. 山西大学, 2006.
- [2] HULST H C V D. Light scattering by small particles[J]. Physics Today, 1957, 10(12).