

基于无冲突路径算法的多目标智能仓储路径规划

官婧, 杨玉发, 郑一帆, 孙知信

引用本文

官婧, 杨玉发, 郑一帆, 孙知信. 基于无冲突路径算法的多目标智能仓储路径规划[J]. 计算机科学, 2026, 53(4): 88-100.

GONG Jing, YANG Yufa, ZHENG Yifan, SUN Zhixin. [Multi-objective Intelligent Warehousing Path Planning Based on Conflict Free Path Algorithm](#) [J]. Computer Science, 2026, 53(4): 88-100.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[知识辅助和强化句法驱动的方面级情感分析](#)

Knowledge-assisted and Reinforced Syntax-driven for Aspect-based Sentiment Analysis
计算机科学, 2026, 53(4): 406-414. <https://doi.org/10.11896/jsjcx.250600117>

[基于多任务强化学习的优先级加权软模块化方法:SM-PHT](#)

SM-PHT:Robust,Scalable,and Efficient Method for Multi-task Reinforcement Learning
计算机科学, 2026, 53(4): 366-376. <https://doi.org/10.11896/jsjcx.250700198>

[基于深度强化学习的长期因果效应估计](#)

Long-term Causal Effect Estimation Based on Deep Reinforcement Learning
计算机科学, 2026, 53(4): 235-244. <https://doi.org/10.11896/jsjcx.250600043>

[基于Maklink图与Boustrophedon路径的工业机器人二维全覆盖路径规划算法](#)

Mobile Robot Two-dimensional Full Coverage Path Planning Algorithm Based on MaklinkDiagram and Boustrophedon Path
计算机科学, 2026, 53(4): 78-87. <https://doi.org/10.11896/jsjcx.250700190>

[基于目标相似性驱动与双端变量引导搜索的大规模多目标进化算法](#)

Large-scale Multi-objective Evolutionary Algorithm Based on Objective Similarity and Dual-EndVariable Guided Search
计算机科学, 2026, 53(3): 351-365. <https://doi.org/10.11896/jsjcx.250200091>

基于无冲突路径算法的多目标智能仓储路径规划

宫婧^{1,2} 杨玉发³ 郑一帆³ 孙知信^{1,2}

1 南京邮电大学江苏省邮政大数据技术与应用工程研究中心 南京 210003

2 南京邮电大学国家邮政局邮政行业技术研发中心(物联网技术) 南京 210003

3 南京邮电大学现代邮政学院 南京 210003

(gongj@njupt.edu.cn)

摘要 研究仓储路径规划对智能仓储具有重要意义,合理的路径规划能够有效避免仓储路径冲突,提升仓库内货物运输效率。针对当前仓储布局较为简单、缺乏针对复杂仓储布局的路径冲突策略问题,提出基于AGV坐标保留表和冲突分类的多目标AGV路径规划算法。首先,构建基于网格法的智能仓储鱼骨布局方案,并根据分区机制,给出存储节点间的距离计算模型,构成单一单向的仓储路径网络有向图。其次,建立AGV坐标保留表方法和路径冲突分类方法,制定路径冲突解决策略和算法。然后,建立以最小化总运输距离、最小化最大运输距离、最小化冲突解决等待时间为目标的多目标智能仓储路径规划模型。最后,结合所提路径冲突解决算法,设计基于进化遗传搜索算法的突变操作,在基于学习的多目标组合优化求解算法P-MOCO的基础上,通过构建偏好条件随机策略,借助多目标降维和强化学习方法,提出改进P-MOCO的无冲突多目标智能仓储路径优化算法CF-MOWVRP,求解无冲突的多目标规划模型的近似帕累托解。实验结果表明,所提算法具备更快的收敛速度和更优的解,能够解决路径冲突,给出无冲突的路径规划方案。

关键词: 路径规划;智能仓储;路径冲突;多目标优化;强化学习;AGV

中图分类号 TP242

Multi-objective Intelligent Warehousing Path Planning Based on Conflict Free Path Algorithm

GONG Jing^{1,2}, YANG Yufa³, ZHENG Yifan³ and SUN Zhixin^{1,2}

1 Engineering Research Center of Post Big Data Technology and Application of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

2 Research and Development Center of Post Industry Technology of the State Posts Bureau(Internet of Things Technology), Nanjing University of Posts and Telecommunications, Nanjing 210003, China

3 Modern Postal College, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract The research on warehouse path planning plays a crucial role in intelligent warehousing, as reasonable path planning can effectively avoid AGV path conflicts and improve in-warehouse transportation efficiency. To address the limitations of simplistic warehouse layouts and the lack of effective path conflict resolution strategies for complex environments, this paper proposes a multi-objective AGV path planning algorithm based on a coordinate reservation table and conflict classification. Firstly, a grid-based fish-bone layout scheme for intelligent warehousing is constructed. A distance calculation model between storage nodes is developed using a partition mechanism, forming a unidirectional directed graph representing the storage path network. Next, an AGV coordinate reservation table and a path conflict classification method are established, followed by the formulation of a hierarchical conflict resolution strategy. Then, a multi-objective intelligent warehouse path planning model is constructed with the goals of minimizing the total transportation distance, minimizing the maximum single transportation distance, and minimizing the waiting time for conflict resolution. Based on the proposed conflict resolution mechanism, a set of mutation operators and crossover operations is designed under an evolutionary genetic search framework. On top of the preference-guided multi-objective combinatorial optimization(P-MOCO) algorithm, an enhanced algorithm named CF-MOWVRP is proposed. This algorithm integrates preference-driven stochastic strategies, multi-objective dimensionality reduction, and reinforcement learning to obtain approximate Pareto-optimal solutions to the conflict-free multi-objective path planning model. Experimental results demonstrate that the proposed algorithm achieves faster convergence and better solution quality, successfully resolves AGV path conflicts, and provides

到稿日期:2025-02-10 返修日期:2025-06-05

基金项目:国家自然科学基金(62272239);江苏省农业科技自主创新项目(CX(22)1007);贵州省科技支撑项目([2023]一般272)

This work was supported by the National Natural Science Foundation of China(62272239), Jiangsu Agriculture Science and Technology Innovation Fund(CX(22)1007) and Guizhou Provincial Key Technology R&D Program([2023]272).

通信作者:孙知信(sunzx@njupt.edu.cn)

feasible conflict-free path planning solutions.

Keywords Path planning, Intelligent warehousing, Path conflict, Multi objective optimization, Reinforcement learning, AGV

1 引言

智能仓储是指利用先进的信息技术和自动化技术来提高仓储管理和物流运作效率的仓储管理系统,目的是将传统仓储系统升级为智能化、自动化的现代化仓储系统,以提高仓库操作的效率、准确性和灵活性,降低仓储成本。在智能仓储领域,仓储的运输效率将直接影响企业的日常运营和长远发展,而影响自动化仓库运输效率的因素主要包括仓储布局、运输车辆路径规划、仓库运行流程等^[1]。因此,如何改进智能仓储的布局 and 实现无冲突的仓储路径规划,以提高仓库内货物的运输效率和准确性,是当前亟待解决的问题。

仓储布局方案是指在一定区域或库区内,对仓库的数量、规模、地理位置等各要素进行科学规划和总体设计的方案,通过优化货架的设置、货物的分类和堆放方式等手段,实现仓库空间的最大化利用。有关智能仓储和多目标车辆路径优化的研究较多,但大多数没有考虑到仓储布局对自动导向车辆(Automated Guided Vehicle, AGV)运行效率的影响和仓储车辆路径优化的动态性和个性化特征。现有仓储布局方案大多是通过栅格化方法来建立的一种单向的行或列布局^[2-5],布局相对单调,AGV的移动路线较为精确,路径冲突概率较低,但该布局存在AGV行车路径较长,吞吐量不高的问题。此外,还有学者建立双向的布局方案^[6],提高了系统的灵活性,但其路线规划十分复杂,导致存在更多类型的冲突,从而使冲突的概率增加。Zhang等^[1]还提出了一种新的仓库斜向布局方案,即鱼骨仓储布局方案。相比传统的行和列布局,该方案的仓储利用率更高,行车路径也更短,但是文章并没有对该布局方案进行详细构建。

AGV路径冲突是指同一时刻至少两台AGV出现在同一位置^[7],造成AGV冲突或死锁问题,严重影响仓储运输效率。由于路径冲突存在严重后果,因此许多学者都对其进行了详细研究。Yna等^[8]提出了基于冲突避让策略的多机器人路径规划方法,解决了狭窄路段环境中部分机器人占据关键通道引发的局部冲突问题。Li等^[9]提出了在A*算法中引入转弯惩罚值和避障等优先级策略,解决了大规模多AGV路径规划中,转弯和避障导致行驶时间增加的问题。Jiang等^[10]提出了一种基于时间窗的改进Dijkstra算法,通过时间窗排布和优先级策略来避免和消除冲突,解决了自动导引车在柔性制造系统中出现的路径规划与冲突问题。Zhang等^[11]提出了基于冲突分类的自动仓库多AGV无冲突路径规划,通过Dijkstra算法确定路径,根据路由的时间窗口来检测潜在冲突,解决了自动化仓库中发生的路径冲突。Zhao等^[12]提出了基于改进深度确定性策略梯度算法的多AGV路径规划算法,通过实时规划路径和动态生成任务序列,以减少AGV间的冲突;并采取动态调整避障策略,解决了传统深度强化学习方法在多AGV系统应用中的效率、协作竞争和动态环境适应性问题。上述对于路径冲突的研究在一定程度上解决了路径冲突问题,但它们都是基于相对简单的行或列布局进行研究的,未能充分考虑智能仓储环境的复杂性、多样性特征。

针对上述问题,本文首先构建了基于网格法的智能仓储鱼骨布局方案,并建立基于鱼骨布局的分区机制,在分区机制条件下,给出存储节点间的距离计算模型,形成单一单向的仓储路径网络有向图;其次,基于仓储鱼骨布局方案,使用AGV坐标保留表方法来检测路径方案中是否存在路径冲突,并根据路径冲突情况将路径冲突分类,制定路径冲突解决策略,提出基于AGV坐标保留表和冲突分类的路径冲突解决算法;然后,考虑到智能仓储环境多样性的特征,建立以最小化总运输距离、最小化最大运输距离、最小化冲突解决等待时间为目标的无冲突多目标智能仓储路径规划模型;最后,在P-MOCO算法^[13]的基础上,通过引入鱼骨布局方案作为仓储路径网络有向图,并设计基于进化遗传搜索算法的4个随机突变算子和两个交叉变异操作,引入所提基于AGV坐标保留表和冲突分类的路径冲突解决算法,提出基于改进P-MOCO的无冲突多目标智能仓储路径优化算法,求解出无冲突的多目标智能仓储路径规划方案。

2 基于网格法的智能仓储鱼骨布局方案

本文基于现有文献中的仓储布局方案,使用网格法构建智能仓储鱼骨布局方案,并基于该方案建立分区机制,在分区机制条件下,制定逆时针方向的AGV单向行驶策略,给出存储节点距离计算模型、服务点到存储节点的距离计算模型,形成单一单向的仓储路径网络有向图。

2.1 鱼骨仓储布局方案的栅格化

鱼骨仓储布局方案使用网格化方式进行二维编码,表示存储节点的位置,最小网格定为边长为单位1的正方形,建立大小为 $m_d \times n_d$ 的存储区域, m_d 为横向网格总数,并规定为奇数, n_d 为纵向网格总数,如图1所示。

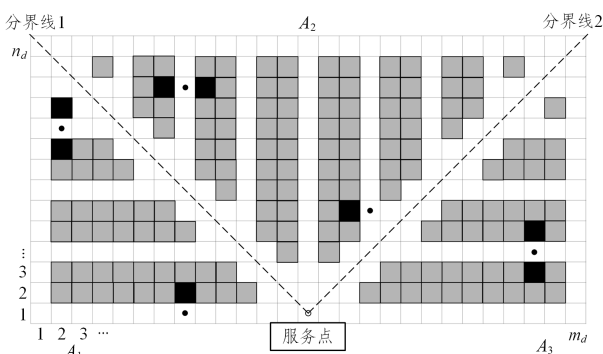


图1 鱼骨仓储布局示意图(电子版为彩图)

Fig. 1 Schematic diagram of fish-bone storage layout

其中,白色网格表示可通行道路,灰色网格表示所有存储节点,黑色网格表示当前需要AGV服务的存储节点。黑色圆点表示对存储节点的服务位置,记为存储节点 i ,根据编码得到 $i = (i_x, i_y)$,表示存储节点 i 的位置坐标, $i_x \in [1, m_d]$, $i_y \in [1, n_d]$,服务点 $D = \left(\frac{m_d+1}{2}, 1\right)$ 位于仓储正下方。

2.2 基于鱼骨仓储布局的分区机制

将仓储布局进行分区,不仅能够简化布局模型的建立,还

能根据存储物品类别和属性进行分区存储,有效应对智能仓储环境的复杂性、多样性特征。将基于网格法构建的智能仓储鱼骨布局方案进行分区,如图1所示。底部中点的服务点和左顶角的连线为分界线1,和右顶角的连线为分界线2。将鱼骨布局方案的区域分为左下 A_1 、正上 A_2 、右下 A_3 这3个存储区域,服务点位于仓储正下方。根据编码,通过构建分界线的函数表达式,能够根据存储节点 i 的坐标 (i_x, i_y) ,判断存储节点所在区域。

1)当存储节点位于分界线1的左下方,存储节点属于 A_1 。

$$i \in A_1, i_x < \frac{m_d+1}{2} \text{ 且 } i_y \leq \frac{(2i-m_d-1)(n_d-1)}{1-m_d} + 1 \quad (1)$$

2)当存储节点位于分界线1的右上方且位于分界线2的左上方时,存储节点属于 A_2 。

$$i \in A_2, i_y > \frac{(2i-m_d-1)(n_d-1)}{1-m_d} + 1 \quad (2)$$

$$\text{且 } i_y > \frac{(2i-m_d-1)(n_d-1)}{m_d-1} + 1$$

$$d_{ij} = \begin{cases} 2n_d - j_y + j_x - i_y + i_x - 2, & i(j) \in A_1, j(i) \in A_2 \\ 2m_d + 2n_d - j_x - j_y - i_x - i_y, & i(j) \in A_2, j(i) \in A_3 \\ (2+2\delta)m_d + (1+2\gamma)i_x + (1+\gamma)i_y + (1+\delta)j_y - (1+2\delta)j_x - 4 - 3\gamma - \delta, & i \in A_1, j \in A_3 \\ 2m_d + 2n_d - i_y - j_y - i_x + j_x - 2, & j \in A_1, i \in A_3 \end{cases} \quad (5)$$

其中,若 $i_y=1$,则 $\gamma=-1$,否则 $\gamma=0$;若 $j_y=1$,则 $\delta=-1$,否则 $\delta=0$ 。

2.4 服务点到存储节点的计算模型

服务点是特殊的存储节点。首存储节点指从服务点出发第一个进行配送的存储节点,尾存储节点指从服务点出发最后一个进行配送的存储节点。首、尾是相对的,因此服务点到首、尾存储节点之间的距离 d_{D_s} 与 d_{D_e} 相同,即 $d_{D_s}=d_{D_e}$ 。考虑到服务点与首、尾存储节点之间距离的特殊性,根据分区规则和编码,计算服务点 $D=(\frac{m_d+1}{2}, 1)$ 到首存储节点 $i=(i_x, i_y)$ 的距离 d_{D_s} 。

1)首存储节点位于 A_1 :

$$d_{D_s} = \frac{(i_y-1)(1-m_d)}{2(n_d-1)} + \frac{m_d+1}{2} - i_x + \sqrt{\frac{(i_y-1)^2(1-m_d)^2}{4(n_d-1)^2} + (i_y-1)^2}, i \in A_1 \quad (6)$$

2)首存储节点位于 A_2 :

$$d_{D_s} = i_y - g_1 - 1 + \sqrt{(i_x - \frac{m_d+1}{2})^2 + g_1^2}, i \in A_2 \quad (7)$$

其中, $g_1 = \frac{(2i_x - m_d - 1)(n_d - 1)}{1 - m_d}$ 。

3)首存储节点位于 A_3 :

$$d_{D_s} = i_x - g_2 - \frac{m_d+1}{2} + \sqrt{g_2^2 + (i_y-1)^2}, i \in A_3 \quad (8)$$

其中, $g_2 = \frac{(i_y-1)(m_d-1)}{2(n_d-1)}$ 。

2.5 智能仓储路径规划解决方案

智能仓储路径规划的实质是车辆路径优化(Vehicle Routing Problems, VRP)。VRP是指在各种限制条件下,对于具有不同货物需求量的需求点,使用车辆从配送中心运输货物到需求点,通过合理地规划车辆配送路线,使得所求目标最优

3)当存储节点位于分界线2的右下方,存储节点属于 A_3 。

$$i \in A_3, i_x > \frac{m_d+1}{2} \text{ 且 } i_y \leq \frac{(2i-m_d-1)(n_d-1)}{m_d-1} + 1 \quad (3)$$

2.3 存储节点距离计算模型

在基于鱼骨仓储布局的分区机制条件下,按照逆时针方向的AGV单向行驶策略,根据节点所属区域,给定两个节点 $i=(i_x, i_y)$ 和 $j=(j_x, j_y)$,建立带参曼哈顿距离的存储节点距离计算模型 d_{ij} 。

1)存储节点都在同一区域:

$$d_{ij} = \begin{cases} |i_x - 1 + \alpha(j_x - 1)| + |i_y - j_y|, & \text{当 } i \in A_1, j \in A_1 \\ |i_x - j_x| + |n_d - i_y + \beta(n_d - j_y)|, & \text{当 } i \in A_2, j \in A_2 \\ |m_d - i_x + \alpha(m_d - j_x)| + |i_y - j_y|, & \text{当 } i \in A_3, j \in A_3 \end{cases} \quad (4)$$

其中,若 $i_y=j_y$,则 $\alpha=-1$,否则 $\alpha=1$;若 $i_x=j_x$,则 $\beta=-1$,否则 $\beta=1$ 。

2)存储节点在不同的区域:

$$d_{ij} = \begin{cases} |i_x - 1 + \alpha(j_x - 1)| + |i_y - j_y|, & \text{当 } i \in A_1, j \in A_1 \\ |i_x - j_x| + |n_d - i_y + \beta(n_d - j_y)|, & \text{当 } i \in A_2, j \in A_2 \\ |m_d - i_x + \alpha(m_d - j_x)| + |i_y - j_y|, & \text{当 } i \in A_3, j \in A_3 \end{cases} \quad (5)$$

化的过程。在形式上,车辆路径优化问题可以使用有向图 $G=(N, A)$ 来表示^[14],其中 N 表示需求点集合 C 和配送中心集合 D 的并集。

在本文构建的鱼骨仓储布局中,只有一个配送中心,即服务点,且节点的距离不再是单纯的欧氏距离,而是需要使用所提出的存储节点距离计算模型进行计算,属于单配送中心的变种VRP问题。在智能仓储中,每个存储节点都有自己的需求量、所需的服务时间,以及允许的最大延迟时间等属性,将AGV路径上的所有路段表示为集合 A ,每条路段代表节点集合 N 中两个顶点之间的连接,有距离、时间等属性;服务点有数量为 m 的AGV集合 K ,每台AGV都有自己的最大载重量、最远行驶距离等属性。智能仓储路径规划的解决方案可以使用一个包括一组路线的集合来表示,如式(9)所示:

$$R = \{R_1, \dots, R_k, \dots, R_m \mid k \in K, K = \{1, 2, \dots, m\}\} \quad (9)$$

其中, R_k 表示在配送中心 D 中的第 k 台AGV的路线。由此,可以将基于网格法的智能仓储鱼骨布局方案建立为单向的仓储路径有向图。图2展示了智能仓储路径规划解决方案的一个解的例子。服务点集合表示为 $D=\{0\}$,存储节点集合表示为 $C=\{1, \dots, 10\}$ 。该例子的一个可行解决方案的路由集合表示为 $R=\{R_1, R_2, R_3, R_4\}$ 。

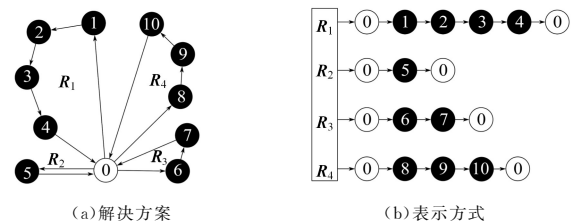


图2 智能仓储路径规划解决方案及其表示方式
Fig. 2 Intelligent warehouse path planning solution and its representation

此外,在鱼骨仓储布局中,规定单个道路网格点只能通行单台 AGV,即 AGV 的通行所需宽度至少是单位长度 1。如图 3 所示,AGV 在任意道路网格点(除边界节点)上,可前进到下一行道路网格点的方向有水平方向、竖直方向和对角线方向共 8 个。当道路网格点属于存储节点或边界节点时,AGV 不可前进到下一行道路网格点,如图 3 中红色虚线路线所示。

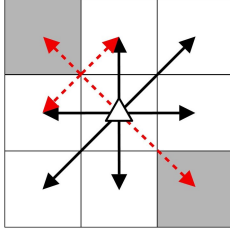


图 3 AGV 前进方向(电子版为彩图)

Fig. 3 AGV forward direction

3 基于 AGV 坐标保留表和冲突分类的路径冲突解算法

3.1 AGV 坐标保留表

AGV 坐标保留表记录了 AGV 的坐标信息。在某一次服务中,根据生成的 AGV 路径规划信息,记录每台 AGV 随时间的路径计划,形成 AGV 坐标保留表。根据 AGV 坐标保留表,如果在同一个时间存在相同坐标的记录,说明多台 AGV 之间发生了冲突,应该重新规划路由。

假设 AGV 通过最小网格的时间为 t ,AGV 数量为 m 台,每台 AGV 的路径坐标为 $\mathbf{P}_{k,t} = (i_x^{k,t}, i_y^{k,t})$,其中, $k \in \{1, 2, \dots, m\}$, $t \in \{1, 2, \dots, T\}$, $i_x^{k,t}$ 和 $i_y^{k,t}$ 分别表示第 k 台 AGV 在时间 t 的横、纵坐标, T 是 AGV 完成一次服务的路径规划的时间长度。若在时间 t 存在多台 AGV 具有相同的坐标,即 $\mathbf{P}_{k,t} =$

$\mathbf{P}_{l,t}$,说明发生了冲突,冲突条件为:

$$\exists k, l \in \{1, 2, \dots, m\}, \text{使得 } (i_x^{k,t} = i_x^{l,t}) \text{ 且 } (i_y^{k,t} = i_y^{l,t}) \quad (10)$$

为了量化冲突问题,本文构建一个冲突检测函数:

$$C(t) = \sum_{k \neq l} \delta(\mathbf{P}_{k,t}, \mathbf{P}_{l,t}) \quad (11)$$

其中, $\delta(\mathbf{P}_{k,t}, \mathbf{P}_{l,t})$ 是一个指示函数,用于判断两台 AGV 是否在时间 t 发生冲突。

$$\delta(\mathbf{P}_{k,t}, \mathbf{P}_{l,t}) = \begin{cases} 1, & \text{冲突条件成立} \\ 0, & \text{否则} \end{cases} \quad (12)$$

为检测全局是否存在冲突,定义全局冲突指标 $C_{\text{total}} = \sum_t C(t)$,若 $C_{\text{total}} = 0$,说明全局路径没有路径冲突,否则说明全局存在路径冲突。

图 4 展示了某次服务的两台 AGV 的某段路径规划,起点时间为 xt ,终点时间为 $(x+2+2\sqrt{2})t$,表 1 记录了该次服务的 AGV 坐标保留表。在 $(x+1+\sqrt{2})t$ 时刻,AGV1 和 AGV2 的坐标相同,表明两个 AGV 在同一时刻处于同一位置,在节点(4, 11)处存在路径冲突,即图中的红色圆圈。设 $\nabla T = (x+2+2\sqrt{2})t - xt$,则 $\delta(\mathbf{P}_{1,(x+2+2\sqrt{2})t}, \mathbf{P}_{2,(x+2+2\sqrt{2})t}) = 1$, ∇T 期间的其他时间的指示函数都为 0, $C_{\text{total}} = 1$,表明全局至少存在一个路径冲突点。

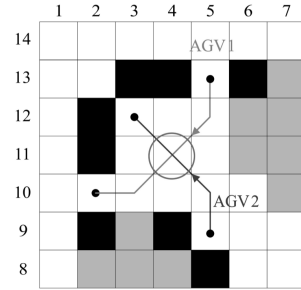


图 4 路径冲突示意图(电子版为彩图)

Fig. 4 Diagram of path conflict

表 1 图 4 对应的 AGV 坐标保留表

Table 1 AGV coordinate retention table corresponding to Fig. 4

| | | 时间 | | | | | | | | |
|----|------|----|-----|--------|----------|-------------------|--------------------|--------------------|-----|-----|
| | | 0 | ... | xt | $(x+1)t$ | $(x+1+\sqrt{2})t$ | $(x+1+2\sqrt{2})t$ | $(x+2+2\sqrt{2})t$ | ... | T |
| 坐标 | AGV1 | D | ... | (5,13) | (5,12) | (4,11) | (3,10) | (2,10) | ... | D |
| | AGV2 | D | ... | (5,9) | (5,10) | (4,11) | (3,12) | (3,12) | ... | D |

3.2 冲突分类

智能仓储系统中存在多种潜在的路径冲突类型,根据路径冲突情况,可以将路径冲突分为 4 类最基本的冲突,如图 5 所示。假设当前为 t 时刻,AGV1 和 AGV2 的位置坐标分别为 i' 和 j' ,到达下一个临近的网格位置所经过时间为 ∇t ,即 $t+\nabla t$ 时刻, $\nabla t = t$ 或 $\sqrt{2}t$,则各冲突类型的冲突条件如下。

1)节点相向冲突:当 AGV1 和 AGV2 同时行驶在一条直线路径上且方向相反,并在某个时间步到达同一位置,则发生节点相向冲突,造成死锁。冲突条件为:

$$\begin{cases} i^{t+\nabla t} = j^{t+\nabla t} \\ (i_y^{t+\nabla t} - i_y^t)(j_x^{t+\nabla t} - j_x^t) = (j_y^{t+\nabla t} - j_y^t)(i_x^{t+\nabla t} - i_x^t) \quad (13) \\ (i_x^{t+\nabla t} - i_x^t)(j_y^{t+\nabla t} - j_y^t) < (i_y^t - i_y^{t+\nabla t})(j_y^{t+\nabla t} - j_y^t) \end{cases}$$

2)边界相向冲突:当 AGV1 和 AGV2 同时行驶在一条直线路径上,方向相反,相互交换位置,即在网格边界相遇,则发

生边界相向冲突,造成死锁。冲突条件为:

$$i' = j^{t+\nabla t} \text{ 且 } i^{t+\nabla t} = j' \quad (14)$$

3)交叉冲突:当 AGV1 和 AGV2 不是在一条直线路径上相交,且同时到达同一个网格点时,发生交叉冲突,造成死锁。冲突条件为:

$$i^{t+\nabla t} = j^{t+\nabla t} \quad (15)$$

4)节点占用冲突:当 AGV1 占据某个网格不动时(如在服务点装卸货、出故障等),AGV2 尝试进入同一位置时,发生节点占用冲突,造成死锁。冲突条件为:

$$i' = j^{t+\nabla t} \text{ 且 } i^t = i^{t+\nabla t} \quad (16)$$

此外,考虑到多 AGV 情况,可能会出现多车交汇、路径回环、环形死锁等更加复杂的冲突情况,但这些冲突都可以分解为上述 4 种基本冲突。多车交汇是指多个 AGV 在同一时刻到达同一个网格点时发生的多 AGV 冲突,可将该冲突分

解为多个基本冲突。例如 3 个 AGV 冲突,可以分解为 AGV1 与 AGV2、AGV1 与 AGV3、AGV2 与 AGV3。路径回环是指在解决路径冲突时,使用相应策略后,可能会造成某台 AGV 循环往复的按照环路路径运行,属于解决路径冲突带来的一个新的问题。环形死锁是指 AGV 形成闭合循环等待链,如 AGV1 等待 AGV2,AGV2 等待 AGV3,AGV3 等待 AGV1。

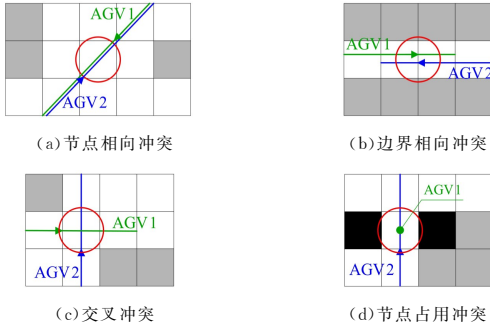


图 5 路径冲突类型

Fig. 5 Types of path conflicts

3.3 路径冲突解决策略

节点相向冲突和边界相向冲突可以统称为相向冲突,在本文建立的鱼骨仓储布局方案下,相向冲突带来的后果是最为严重的。如图 6 所示,鱼骨仓储布局存在较多的横线纵向路径,由于单个网格只能通过一台 AGV,若存在相向冲突,只能让其中一台 AGV 转向进行避让,这将导致过多的成本,如冲突位置 1。如果在其他可通行路径较宽的区域,如冲突位置 2,则可通过简单的避让解决。通过鱼骨仓储布局方案的单向策略,能够较大程度地降低出现相向冲突的概率。此外,冲突位置 1 不会发生交叉冲突,因此在其他位置,只需要其中一台 AGV 进行转向避让或者原地等待即可解决交叉冲突和节点占用冲突。

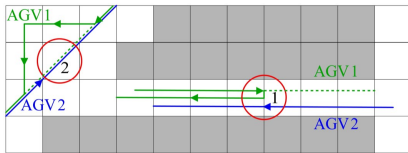


图 6 路径冲突避让示意图(电子版为彩图)

Fig. 6 Diagram of path conflict avoidance

冲突解决策略对智能仓储的安全和效率起着重要的作用,不合理的冲突解决策略会导致 AGV 等待过长时间,甚至导致仓储系统道路死锁,如冲突位置 1 的相向冲突。假设不设置转向避让策略,两台 AGV 将永久等待,导致道路死锁。本文根据服务的存储节点的运输量来判断 AGV 优先级的高低,运输量高的优先级高。根据对冲突分类的分析,提出如下 3 种冲突解决策略。

1)短时间等待策略:优先级高的 AGV 优先获得道路使用权,优先级低的 AGV 在原地等待一定的较短时间 t_w ,直到优先级高的 AGV 释放冲突节点,短时间等待策略需要等待的时间 t_w 大部分都为装卸货时间或单位时间 1,并且设置一个等待时间阈值 T_w ,等待时间小于阈值时使用短时间等待策略,否则认为短时间等待策略会导致长时间等待,不应该使用该策略。短时间等待策略的 AGV 坐标表示为 $(i_x^{k,t} = i_x^{k,t+t_w})$

且 $(i_y^{k,t} = i_y^{k,t+t_w})$ 。假设采取短时间等待策略的冲突点的数量为 N_{cw} ,则所有等待时间为:

$$t_{cw} = \sum_{l=1}^{N_{cw}} t_w^l \quad (17)$$

2)避让策略:优先级高的 AGV 优先获得道路使用权,优先级低的 AGV 选择当前网格的前、后、左、右 4 个方向上可行的道路网格点进行避让,同时增加该 AGV 的行驶距离和行驶时间,如图 7 所示。每执行一次避让策略,重新判断该冲突是否解决,没有解决则继续在新的网格点执行避让策略。假设一个冲突点使用避让策略的最大次数为 N_a ,避让过程 AGV 的位置坐标序列为 $\mathbf{P} = (i_x^k, i_y^k), k \in \{1, 2, \dots, n_a\}, n_a \leq N_a$,采取避让策略的冲突点的数量为 N_{ca} ,则单个冲突点的避让增加距离为 $d_a^l = \sum_{k \in \{1, 2, \dots, n_a\}} \sqrt{(i_x^{k+1} - i_x^k)^2 + (i_y^{k+1} - i_y^k)^2}$,因此所有避让增加的距离为:

$$d_{ca} = \sum_{l=1}^{N_{ca}} d_a^l \quad (18)$$

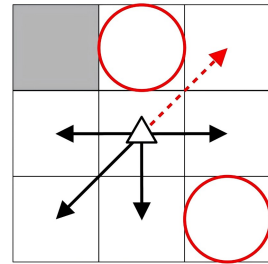


图 7 避让策略可行方向示意图

Fig. 7 Schematic diagram of feasible direction for avoidance strategy

3)CF-MOWVRP 算法更新路由策略:假如短时间等待策略等待时间超过阈值 T_w 、避让策略使用次数超过阈值 N_a ,即都无法解决所有路径冲突,则重新根据第 4 章介绍的 CF-MOWVRP 算法来生成新的路径,该策略需要的计算成本高,不优先考虑。

根据上述 4 种冲突分类和 3 种冲突解决策略的分析,对于交叉冲突和节点占用冲突,优先使用短时间等待策略来解决,其次使用避让策略;对于相向冲突策略,由于其无法使用短时间等待策略来解决,优先使用避让策略。如果短时间等待策略和避让策略都无法解决冲突,或超过设置的阈值,才使用 CF-MOWVRP 更新路由策略解决。

对于多车交汇冲突,将其分解为基本冲突后,再对交汇点依次选取冲突对,按照冲突分类与优先级判定流程调用短等待或避让策略,逐对消解直至所有冲突对得到解决。具体而言,在坐标保留表中识别所有到达同一节点的 AGV 集合,按优先级选取两辆 AGV 进行处理,处理完成后更新坐标保留表,重新扫描,直至无冲突对。

对于路径回环,为避免回环,通过累计避让次数,判断是否超出预设阈值,超出阈值即改用 CF-MOWVRP 全局重规划,以生成无回环的新路径。

对于环形死锁,首先选择环形死锁冲突链中优先级最高的 AGV 先行,其他 AGV 短等待;若最高优先级 AGV 无法前进,则对其执行避让策略,腾出当前节点;若避让也失败,则

触发 CF-MOWVRP 全局重规划。

通过上述短时等待、避让、全局重规划的三阶梯度策略,能够有效解决基本的冲突问题。对于更复杂的多车交汇、路径回环、环形死锁冲突情况,三阶梯度策略也能够有效解决。如果在阈值范围内不能解决冲突,表明当前解不可行,或者代价过大,需要重新进行全局重规划,寻找新的可行解,避免额外的资源浪费。

3.4 路径冲突解决算法

结合所提的 3 种冲突解决策略,提出基于 AGV 坐标保留表和冲突分类的路径冲突解决算法,算法流程如图 8 所示。

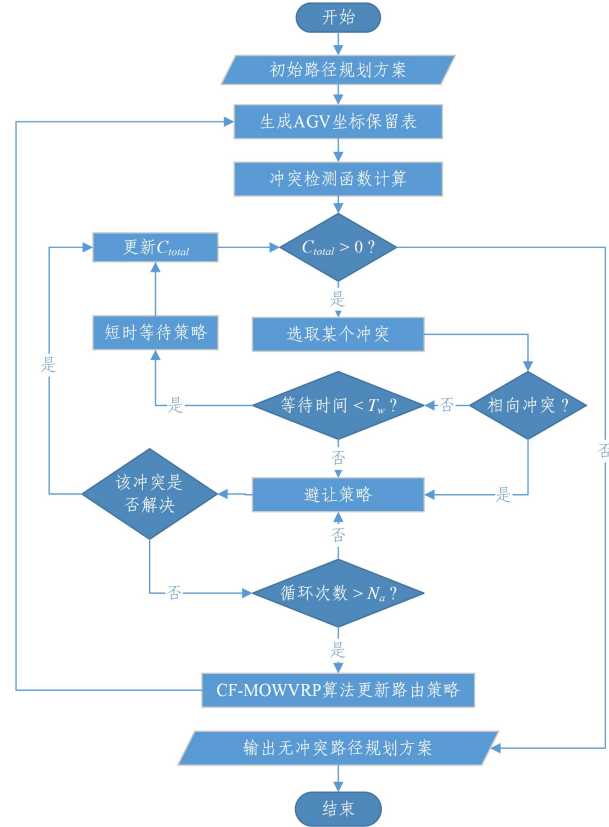


图 8 基于 AGV 坐标保留表和冲突分类的路径冲突解决算法流程图

Fig. 8 Flowchart of path conflict resolution algorithm based on AGV coordinate reservation table and conflict classification

该算法根据生成的路径规划方案,记录 AGV 位置到 AGV 坐标保留表,根据冲突检测函数获取全局冲突指标 C_{total} ,判断是否存在路径冲突:如果不存在,则得到最终的路径规划;若存在,则依次选择冲突点,根据冲突条件将冲突分类,根据对应的冲突类别,选择合适的冲突解决策略,解决路径冲突,并更新得到新的路径规划方案。循环上述过程,直到路径规划方案无冲突为止。

4 基于改进 P-MOCO 的无冲突多目标智能仓储路径优化算法

4.1 无冲突多目标智能仓储路径规划模型

本文根据所提出的鱼骨仓储布局的网络有向图,建立多目标智能仓储无冲突路径规划模型。该模型有一个服务点 D ,为多个存储节点提供运输服务,并要求 AGV 必须满足相

应的约束条件,最终实现最小化总运输距离、最小化最大运输距离、最小化冲突解决等待时间的 3 个目标。为方便开展智能仓储路径优化研究,假设所有 AGV 都是同质的,AGV 配送完后返回配送中心且不再进行配送,不考虑存储节点的时间窗,AGV 速度恒定。

目标函数 1 最小化总运输距离,如式(19)所示:

$$\min f_1 = \sum_i \sum_j \sum_k x_{ijk} d_{ij} + m(d_{D_s} + d_{D_e}) + d_{ca} \quad (19)$$

目标函数 2 最小化最大运输距离,如式(20)所示:

$$\min \{ \max f_2 = x_{ijk} d_{ij} + d_{D_s} + d_{D_e} \} \quad (20)$$

目标函数 3 最小化冲突解决等待时间,如式(21)所示:

$$\min f_3 = t_{cw} + \frac{d_{ca}}{V} \quad (21)$$

其中, x_{ijk} 为 0~1 的决策变量,表示 AGV k 是否由存储节点 i 驶向 j ,是则取 1,否则取 0, $i \neq j$; n 表示存储节点数量; m 表示 AGV 数量; d_{ij} 表示节点 i, j 之间的距离; d_{D_s}, d_{D_e}, d_{ca} 分别表示服务点到路径首和尾存储节点的距离以及避让策略增加的所有距离; t_{cw} 表示短时等待策略等待的时间, V 表示 AGV 的速度。第二个目标函数最小化最大运输距离是指在满足所有节点存储量的情况下,使得所有 AGV 的运输距离中的最大值最小。考虑该目标的原因是如果运输路径过长,会易导致仓库拥堵和路径冲突。

约束条件如式(22)所示:

$$\text{s. t. } \begin{cases} x_{ijk} = \begin{cases} 1, & \text{第 } k \text{ 台 AGV 从节点 } i \text{ 行驶到节点 } j \\ 0, & \text{否则} \end{cases} & i, j \neq j \\ y_{ik} = \begin{cases} 1, & \text{第 } k \text{ 台 AGV 为节点 } i \text{ 进行运输} \\ 0, & \text{否则} \end{cases} \\ \sum_i \sum_j x_{ijk} r_i \leq MG \\ r_i \leq y_{ik} \leq MG \\ \sum_k y_{ik} = 1 \\ \sum_i x_{ijk} \leq y_{ik} \\ \sum_j x_{ijk} \leq y_{ik} \\ C_{total} = N_{cw} + N_{ca} \\ C_{total} = \sum_t C(t) \\ C = \{1, 2, \dots, n\}, K = \{1, 2, \dots, m\}, i, j \in C, k \in K \end{cases} \quad (22)$$

约束条件分别表示决策变量 x_{ijk} ; 决策变量 y_{ik} ; AGV 载重不超过其最大载重量;满足存储节点存储量;每个存储节点必须被 AGV 服务;有且仅有一台 AGV 车到达一个存储节点;有且仅有一台 AGV 离开一个存储节点;采取短时间等待策略和避让策略的冲突点的数量等于全局冲突指标;全局冲突指标通过基于 AGV 坐标保留表的冲突检测函数计算。 i, j 表示每个存储节点的编号; r_i 表示存储节点 i 的存储量; MG 表示 AGV 的最大载重量; m 为 AGV 总数, n 为存储节点总数。

4.2 P-MOCO 算法

P-MOCO 算法是 Lin 等^[13]提出的一种基于学习的方法,

目的是找到一个给定的多目标组合优化(MOCO)问题的近似帕累托(Pareto)集,能够根据一个单一的偏好条件模型来直接生成任何权衡偏好的多目标的近似 Pareto 解。P-MOCO 算法是基于分解的多目标进化算法(MOEA/D)的一个依据学习的扩展。该方法性能较为优越,对于求解本文建立的多目标智能仓储路径规划模型具有很好的效果。

4.2.1 偏好、编码器和节点嵌入

偏好是一组标量化的向量,满足 $\lambda_i \geq 0$ 和 $\sum_{i=1}^d \lambda_i = 1$ 。其中,

d 是目标数目, λ_i 是对第 i 个目标的偏好。通过编码器将实例节点进行嵌入。该算法以一个有向图实例作为输入,通过基于注意力编码器为每个节点输出一组 d 维节点的嵌入 $\{h_1, \dots, h_n\}$:

$$h_i = W\{i_x, i_y, r_i\} + b \quad (23)$$

其中, W 和 b 表示编码器中可学习的参数; $\{i_x, i_y, r_i\}$ 表示存储节点属性,包括节点位置和存储量。

4.2.2 解码器和节点上下文嵌入

P-MOCO 使用一个多层感知机来生成以偏好为条件的解码器参数。多层感知机以 d 维偏好向量 λ 和模型参数 ψ 为输入,首先生成一个隐藏的嵌入 $e(\lambda) = \text{MLP}(\lambda | \psi)$,然后通过线性投影 $\theta_{\text{decoder}} = We(\lambda) + b$ 将隐藏的嵌入映射到解码器参数中。在时间 t 时,解码器先通过初始选择的节点 h_{R_t} 以及最后选择的节点 $h_{R_{t-1}}$ 来构造一个上下文嵌入 $\hat{h}_{(CT)} = [h_{R_t}, h_{R_{t-1}}]W_Q(\lambda)$,其次矩阵 $W_Q(\lambda) \in \mathbb{R}^{2d \times d}$ 将连接的 $2d$ 维向量投影到一个 d 维向量上,然后通过多头注意力模型(Multi-Head Attention, MHA)^[15] 进一步将该上下文嵌入与所有节点的嵌入 $\{h_1, \dots, h_n\}$ 进行聚合。

$$\begin{aligned} h_{(CT)} &= W_{\text{MHA}}(\lambda) \cdot \text{MHA}(Q = \hat{h}_{(CT)}) \\ E &= \{h_1, \dots, h_n\}W_E(\lambda) \\ V &= \{h_1, \dots, h_n\}W_V(\lambda) \end{aligned} \quad (24)$$

其中, Q, E, V 分别为 MHA 的查询、键和值; $W_Q(\lambda), W_E(\lambda), W_V(\lambda)$ 是 3 个投影矩阵。

4.2.3 节点选择概率

P-MOCO 算法定义了一个由 $\theta(\lambda)$ 参数化的偏好条件随机策略,在已经选择的路径 $R_{1:t-1}$ 的基础上,以大小为 $p_{\theta(\lambda)}(R | G) = \prod_{t'=1}^n p_{\theta(\lambda)}(R_t | G, R_{1:t'-1})$ 的概率来选择下一节点 R_t 。该算法根据节点的转置上下文嵌入 $h_{(CT)}^T$ 来计算选择下一节点的概率:

$$\text{logit}_j = \begin{cases} CT \cdot \tanh\left(\frac{h_{(CT)}^T h_j}{\sqrt{d}}\right), & j \neq R_t \text{ 且 } \forall t' < t \\ -\infty, & \text{otherwise} \end{cases} \quad (25)$$

其中, CT 是超参数; R_t 表示已经访问的节点; $\forall t' < t$ 表示已经选择的节点对后续节点选择没有影响。所有访问过的节点都设置为 $-\infty$,不会作为下一个被选择的节点。在时间步长 t 时选择节点 j 的概率通过归一化指数函数(Softmax)计算。

$$p_{\theta(\lambda)}(R_t = j | G, R_{1:t-1}) = \frac{e^{\text{logit}_j}}{\sum_i e^{\text{logit}_i}} \quad (26)$$

4.3 无冲突多目标智能仓储路径优化算法

虽然 P-MOCO 算法对于多目标的处理具有良好的效果,但是其只适用于原始的 VRP 问题,且不具备解决路径冲突的

能力,因此,在本文建立的鱼骨仓储布局的条件下,需要对 P-MOCO 算法进行改进,使其能够解决本文的智能仓储路径规划问题,并得到无冲突的路径解决方案。具体改进如下。

1) 针对 P-MOCO 算法只适用于原始的 VRP 问题,通过改进其计算节点之间距离的方式,使用所提的存储节点距离计算模型和服务点到存储节点的计算模型来计算,最终将得到的单一单向的智能仓储路径有向图作为输入。

2) 针对 P-MOCO 算法不具备解决路径冲突的能力,将所提的基于 AGV 坐标保留表和冲突分类的路径冲突解决算法融入 P-MOCO 算法中,在得到路径解决方案后,通过该冲突解决算法来解决路径冲突。

3) 为进一步加快 P-MOCO 算法的收敛速度,本文基于进化遗传搜索算法的突变操作思想^[16],设计了基于进化遗传搜索算法的突变操作,包括使用一组 4 个随机突变算子和 2 个交叉变异操作实现。4 个随机突变算子能够对个体路径结构进行局部扰动,打破当前解的固有模式,增加种群多样性,有效防止算法陷入局部最优,通过局部范围内路径节点的重组与迁移,拓宽解空间的探索范围。同时,结合启发式继承交叉与随机继承交叉两种机制,在保留父代优良路径特征的同时,适度引入随机性,进一步提高种群的整体质量与多样性。启发式继承交叉优先继承目标函数值较小的局部解,从而加速优质特征在种群中的扩散;而随机继承交叉则在保证基因多样性的基础上,增强算法对全局最优解的探索能力。

4.3.1 4 个随机突变算子

4 个随机突变算子的计算示例如图 9 所示。

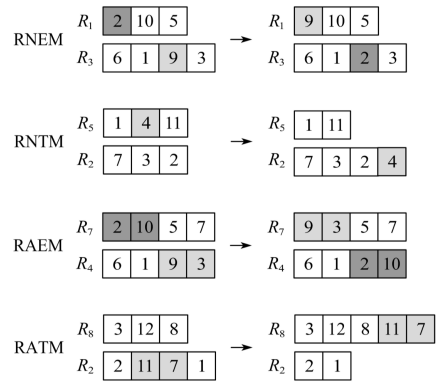


图 9 随机突变算子操作示意图

Fig. 9 Schematic diagram of random mutation operator operation

1) 随机节点交换突变(RNEM):在不违反容量约束的情况下,从已经选择的路由中随机选择两个节点进行交换,生成新路由,并更新路由由容量情况。

$$\begin{cases} R_i = \{\dots, i_1, i_2, \dots\} \\ R_j = \{\dots, j_1, j_2, \dots\} \end{cases} \Rightarrow \begin{cases} R_i = \{\dots, j_2, i_2, \dots\} \\ R_j = \{\dots, j_1, i_1, \dots\} \end{cases} \quad (27)$$

2) 随机节点转移突变(RNTM):将随机选择的节点从一条路由转移到另一条路由,作为末尾节点。

$$\begin{cases} R_i = \{\dots, i_1, i_2, \dots\} \\ R_j = \{\dots, j_1, j_2, \dots\} \end{cases} \Rightarrow \begin{cases} R_i = \{\dots, i_2, \dots\} \\ R_j = \{\dots, j_1, j_2, \dots, i_1\} \end{cases} \quad (28)$$

3) 随机弧形交换突变(RAEM):遵循与随机节点交换突变相同的过程,不同的是随机选择路由内的相邻节点,即弧形节点进行交换。

$$\begin{cases} \mathbf{R}_i = \{\dots, i_1, i_2, \dots\} \\ \mathbf{R}_j = \{\dots, j_1, j_2, \dots\} \end{cases} \Rightarrow \begin{cases} \mathbf{R}_i = \{\dots, j_1, j_2, \dots\} \\ \mathbf{R}_j = \{\dots, i_1, i_2, \dots\} \end{cases} \quad (29)$$

4) 随机弧形转移突变(RATM): 遵循与随机节点转移突变相同的过程, 随机选择路径内相邻节点即弧形节点, 进行转移, 作为末尾节点。

$$\begin{cases} \mathbf{R}_i = \{\dots, i_1, i_2, i_3, i_4, \dots\} \\ \mathbf{R}_j = \{\dots, j_1, j_2, \dots\} \end{cases} \Rightarrow \begin{cases} \mathbf{R}_i = \{\dots, i_1, i_4, \dots\} \\ \mathbf{R}_j = \{\dots, j_1, j_2, \dots, i_2, i_3\} \end{cases} \quad (30)$$

4.3.2 两个交叉变异操作

两个交叉变异操作包括启发式继承交叉和随机继承交叉, 计算示例如图 10 所示, 交叉变异算法如算法 1 所示。

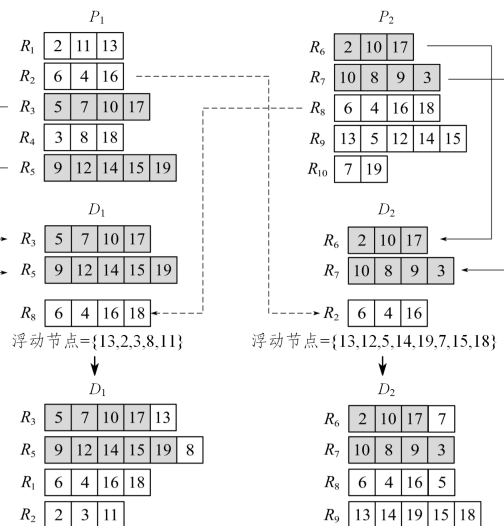


图 10 交叉变异操作示意图

Fig. 10 Schematic diagram of crossover mutation operation

算法 1 交叉变异算法

输入: 两个父节点路由 $\mathbf{P}_1 = \{\dots, \mathbf{R}_i, \dots, \mathbf{R}_{m_1}\}$, $\mathbf{P}_2 = \{\dots, \mathbf{R}_j, \dots, \mathbf{R}_{m_2}\}$,

且满足 $\mathbf{P}_1, \mathbf{P}_2 \subseteq \mathbf{R}$, $\mathbf{P}_1 \cap \mathbf{P}_2 = \emptyset$, $3 \leq m_1 = m_2 \leq \frac{m}{2}$, $i \in \{1, \dots,$

$m_1\}$, $j \in \{1, \dots, m_2\}$

输出: 新路由 $\mathbf{D}_1, \mathbf{D}_2$

- for $i=1$ to m_1 do
- 计算 $\text{MG}_{\mathbf{R}_i} - \text{r}_{\mathbf{R}_i}$, 得到其中的两个最小值 $\text{MG}_{\mathbf{R}_{i_1}} - \text{r}_{\mathbf{R}_{i_1}}$ 和 $\text{MG}_{\mathbf{R}_{i_2}} - \text{r}_{\mathbf{R}_{i_2}}$ 对应的路由 \mathbf{R}_{i_1} 和 \mathbf{R}_{i_2}
- end for
- 随机选择 $\mathbf{R}_{j_1}, \mathbf{R}_{j_2} \in \mathbf{P}_2$
- 找到 $\mathbf{P}_1, \mathbf{P}_2$ 中存在相同节点最多的路由 \mathbf{R}_{i_m} 和 \mathbf{R}_{j_m}
- 执行子路由继承操作: $\mathbf{D}_1 = \{\mathbf{R}_{i_1}, \mathbf{R}_{i_2}, \mathbf{R}_i\}$, $\mathbf{D}_2 = \{\mathbf{R}_{j_1}, \mathbf{R}_{j_2}, \mathbf{R}_{j_m}\}$
- 计算浮动节点: $\text{FN}_1 = \mathbf{P}_1 - \mathbf{D}_1$, $\text{FN}_2 = \mathbf{P}_2 - \mathbf{D}_2$
- 将 FN_1 和 FN_2 随机加入 \mathbf{D}_1 和 \mathbf{D}_2 中路由的末尾, 作为新路由
- 剩下的浮动节点作为新路由加入 \mathbf{D}_1 和 \mathbf{D}_2 , 作为新子路由由 \mathbf{D}_1 和 \mathbf{D}_2

4.3.3 多目标降维和强化学习

还有一项改进在于, P-MOCO 算法使用加权切比雪夫聚合方法(TCH)处理多维目标, 适用于 2 维目标。基于惩罚的边界交叉聚合法(PBI)在超过 2 维的优化问题中优于 TCH^[17], 更加适用于本文建立的无冲突 3 维目标智能仓储路径规划模型。该方法将目标向量分解为沿权重向量方向的分量和正交分量, 并对正交分量施加惩罚, 从而平衡解的多样性与收敛性。对于一个包含 d 个不同目标的向量的路径规划问

题 $f(\mathbf{R}) = [f_1(\mathbf{R}), \dots, f_d(\mathbf{R})]$, 基于偏好 λ 并使用 PBI 进行降维。

$$f(\mathbf{R}|\lambda) = \min_{1 \leq i \leq d} \{g^{\text{pbi}}(\mathbf{R}|\lambda_i, \mathbf{z}_i^*)\} = l_1^i + \rho l_2^i$$

$$l_1^i = \frac{\|(\mathbf{z}_i^* - f_i(\mathbf{R}))^T \lambda_i\|}{\|\lambda_i\|} \quad (31)$$

$$l_2^i = \|f_i(\mathbf{R}) - (\mathbf{z}_i^* - l_1^i \lambda_i)\|$$

其中, $\rho > 0$ 为预设惩罚参数; \mathbf{z}^* 为理想参考点; l_1^i 表示 $f_i(\mathbf{R})$ 在权重向量 λ_i 上的投影到参考点的距离; l_2^i 是 $f_i(\mathbf{R})$ 与权重向量 λ_i 之间的垂直距离。

对于一个给定的实例 \mathbf{G} , 目标是 minimized 各个目标函数的值与目标的理想值之间的距离:

$$F(\theta|\mathbf{G}) = \min \mathbb{E}_{\mathbf{R} \sim \rho_{\theta(\lambda)}(\cdot|\mathbf{R})} f(\mathbf{R}|\lambda) \quad (32)$$

其中, θ 表示模型参数; \mathbb{E} 表示对随机变量进行期望操作, 并从分布 λ 和策略 $\rho_{\theta(\lambda)}$ 中采样得到 λ 和 \mathbf{R} 。为训练模型, 需要在每次迭代中从不同的实例集合 \mathbf{G} 中随机抽样实例 \mathbf{G} , 并将训练损失定义为 $L_{\text{loss}}(\theta) = \mathbb{E} \mathbf{G} \sim \text{GF}(\theta|\mathbf{G})$ 。

对于给定实例 \mathbf{G} 和偏好 λ , 使用强化算法^[13]来估计偏好条件代价的梯度:

$$\nabla F(\theta|\lambda, \mathbf{G}) = \mathbb{E}_{\mathbf{R} \sim \rho_{\theta(\lambda)}(\cdot|\mathbf{R})} [(f(\mathbf{R}|\lambda, \mathbf{G}) - b(\mathbf{R}|\lambda)) \nabla_{\theta(\lambda)} \log \rho_{\theta(\lambda)}(\mathbf{R}|\mathbf{G})] \quad (33)$$

其中, $f(\mathbf{R}|\lambda, \mathbf{G}) - b(\mathbf{R}|\lambda)$ 表示优势函数; $b(\mathbf{R}|\lambda)$ 是减少梯度方差的基线。在每个更新选择存储节点的过程中, 对每个 $\lambda_h - G_i$ 组合使用相同的共享基线估计函数 $b(G_i|\lambda_h) = \frac{1}{m} \sum_{j=1}^m f(\mathbf{R}_{h_j}|\lambda_h, G_i)$, 然后随机抽取 H 个偏好 $\{\lambda_1, \dots, \lambda_H\} \sim \Delta$, B 个实例 $\{G_1, \dots, G_B\} \sim \mathbf{G}$ 和 m 个不同的路由 $\{\mathbf{R}_1^i, \dots, \mathbf{R}_m^i\} \sim \rho_{\theta(\lambda_h)}(\cdot|G_i)$, 给出近似梯度^[18]:

$$\nabla F(\theta) \approx \frac{1}{HBm} \sum_{h=1}^H \sum_{i=1}^B \sum_{j=1}^m [(f(\mathbf{R}_{h_j}^i|\lambda_h, G_i) - b(G_i|\lambda_h)) \cdot \nabla_{\theta(\lambda_h)} \log \rho_{\theta(\lambda_h)}(\mathbf{R}_{h_j}^i|G_i)] \quad (34)$$

最终, 通过随机选择路由 $\mathbf{R}_{h_j}^i$ 的起始节点, 推出不同的路径选择, 经过迭代得到最终的训练模型。根据模型及偏好向量, 可求解出无冲突的多目标仓储路径规划方案。

根据上述对 P-MOCO 算法的改进, 提出基于改进 P-MOCO 的无冲突多目标智能仓储路径优化算法 CF-MOWVRP。CF-MOWVRP 以基于鱼骨布局方案的智能仓储路径优化有向图 $\mathbf{G} = (N, A)$ 为输入, 以无冲突仓储路径解决方案和帕累托前沿为结果输出。

CF-MOWVRP 的训练算法如算法 2 所示, 模型架构如图 11 所示。

算法 2 CF-MOWVRP 训练算法

输入: 偏好分布 Δ , 实例分布 \mathbf{G} , 训练步数 T , 每次迭代的偏好数 H , 批次大小 B , AGV 数量 m

输出: 训练模型和模型参数 θ

- 初始化模型参数 θ
- for $t=1$ to T do
- 偏好采样: $\lambda \sim \Delta, \forall h \in \{1, \dots, H\}$
- 实例采样: $G_i \sim \mathbf{G}, \forall i \in \{1, \dots, B\}$
- 节点选择概率计算: $\mathbf{R}_{h_j}^i \sim \rho_{\theta(\lambda_h)}(\cdot|G_i), \forall j \in \{1, \dots, m\}$
- 突变操作: 4 个随机突变算子 $\begin{cases} \mathbf{R}_{h_j}^i(\dots) \\ \mathbf{R}_{h_j}^i(\dots) \end{cases} \Rightarrow \begin{cases} \mathbf{R}_{h_j}^i(\dots) \\ \mathbf{R}_{h_j}^i(\dots) \end{cases}$, 交叉变异算

法进行突变 $\begin{cases} \mathbf{P}_1 = \mathbf{D}_1 \\ \mathbf{P}_2 = \mathbf{D}_2 \end{cases}$

7. 全局冲突指标计算: $C_{total} = \sum_t \sum_{k=1}^T \delta(\mathbf{P}_{k,t}, \mathbf{P}_{1,t})$

8. for conflict=1 to C_{total} do

9. conflict 冲突分类判断: 相向冲突, 交叉冲突, 节点占用冲突

10. 短时间等待策略, 避让策解决路径冲突

11. if $\delta(\mathbf{P}_{k,t}, \mathbf{P}_{1,t}) = 0$ do $C_{total} = C_{total} - 1$

12. if $\delta(\mathbf{P}_{k,t}, \mathbf{P}_{1,t}) = 1$ do 使用 CF-MOWVRP 算法更新路由策略 跳到步骤 3

13. end for

14. 共享基线估计函数: $b(G_i | \lambda_h) \leftarrow \frac{1}{m} \sum_{j=1}^m f(\mathbf{R}_{ij} | \lambda_h, G_i)$

15. 近似梯度: $\nabla F(\boldsymbol{\theta})$

16. 模型参数: $\boldsymbol{\theta} \leftarrow \text{ADAM}(\boldsymbol{\theta}, \nabla F(\boldsymbol{\theta}))$

17. end for

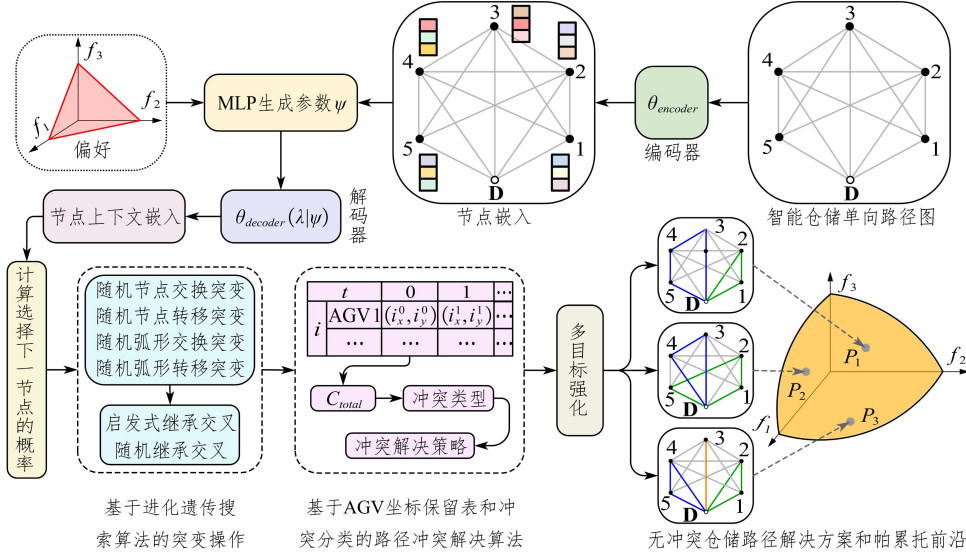


图 11 CF-MOWVRP 算法模型架构

Fig. 11 Architecture of CF-MOWVRP algorithm model

5 仿真结果及分析

仿真环境为 Windows 10, 使用 PyTorch 2.0.0+CPU 和 Python 3.9.6 进行模型训练和测试。智能仓储的鱼骨布局的大小为 $m_d = 27, n_d = 14$ 。采样 λ 个偏好, $\lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ 。存储节点位置的横坐标从区间 $[1, m_d]$ 随机采样, 纵坐标从区间 $[1, n_d]$ 随机采样; 存储节点存储量从集合 $\{1, \dots, 9\}$ 中随机抽

取并除以缩放因子; 最大载重量 $MG = 1$ 。

分别设置存储节点数量 n 和 AGV 数量 m 为 5, 10, 20, 使用原始的 P-MOCO 和改进的 CF-MOWVRP 进行训练, epoch 为 300, episode 为 10000, 得到的训练结果如图 12 所示。由图可知, 在该实验环境下, 3 次训练中, 改进的 CF-MOWVRP 的下降速度更快, 收敛时的损失值都比原始的 P-MOCO 低, 表明 CF-MOWVRP 的训练效果更好。

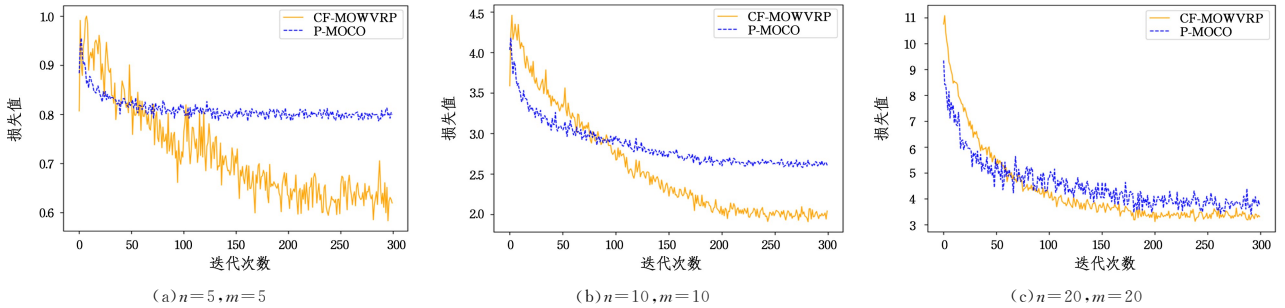


图 12 不同 n 和 m 的训练损失值变化

Fig. 12 Variation of training loss values for different n and m values

在 n 和 m 为 5 和 10 时, CF-MOWVRP 波动性更大, 收敛过程中存在较大振荡, 原因是 CF-MOWVRP 使用了基于进化遗传搜索算法的突变操作, 具有较强的探索性, 寻找的解空间范围更大, 导致波动性较大, 同时也加快了收敛。综上, 相较于 P-MOCO, CF-MOWVRP 在不同的节点和 AGV 下, 最终损失值都更低, 说明其优化效果较好, 适用于更大规模的问题。

episode 为 500, 进行 CF-MOWVRP、P-MOCO 与经典算法 MOEA/D、NSGA-II 的训练对比实验, 结果如图 13 所示。MOEA/D 即基于分解的多目标进化算法, 主要思想是将一个多目标优化问题分解为若干个标量优化子问题, 并同时对它们进行优化。NSGA-II 即非支配排序遗传算法, 该算法是一个基于 Pareto 优势的多目标遗传算法。由图 13 可知, 训练损失值由小到大为 CF-MOWVRP, P-MOCO, MOEA/D, NSGA-II。CF-MOWVRP 最小, 表明 CF-MOWVRP 算法的收敛

节点数量 n 和 AGV 数量 m 为 5 时, 设置 epoch 为 300,

效果较其他算法更优。此外,各算法训练所需时间为:MOEA/D为 6.66 小时,NSGA-II 为 17.77 小时,P-MOCO 为 19.05 分钟,CF-MOWVRP 为 21.1 分钟,表明 CF-MOWVRP 和 P-MOCO 算法训练效率远高于经典的 MOEA/D 和 NSGA-II 算法,训练效率更快。

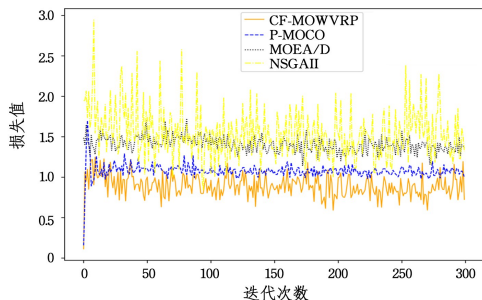


图 13 CF-MOWVRP 与经典算法的训练损失值对比

Fig. 13 Comparison of training loss values between CF-MOWVRP and classical algorithms

采用三维空间均匀划分方法生成的 528 个均匀分布的偏好向量如图 14(a) 所示,使用 n 和 m 为 20,episode 为 10 000 的训练模型进行测试,得到测试结果的近似 Pareto 前沿如图 14(b) 所示,并得到相应的指标(见表 2)。将三维的近似 Pareto 前沿按照二维进行分解,得到各目标间的近似 Pareto 前沿,如图 15 所示。

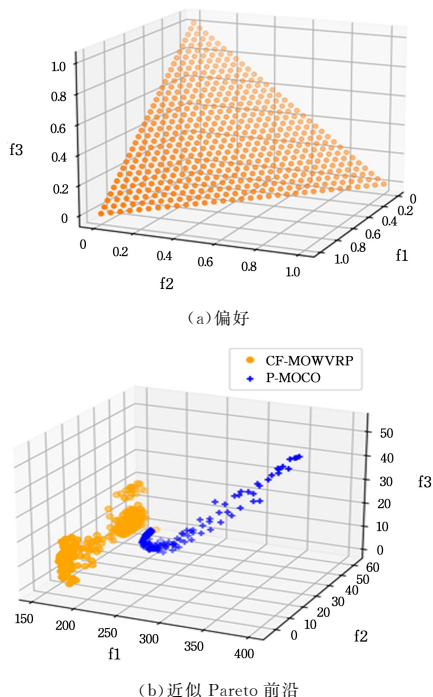


图 14 均匀分布的偏好及其对应的近似 Pareto 前沿

Fig. 14 Preference for uniform distribution and its corresponding approximate Pareto frontier

表 2 近似 Pareto 前沿的指标值

Table 2 Index values of approximate Pareto front

| Methods | HV | IGD | C-metric |
|-----------|--------------|----------|----------|
| CF-MOWVRP | 195 643.4974 | 166.0333 | 0.9943 |
| P-MOCO | 612003.4067 | 201.2132 | 0.0019 |

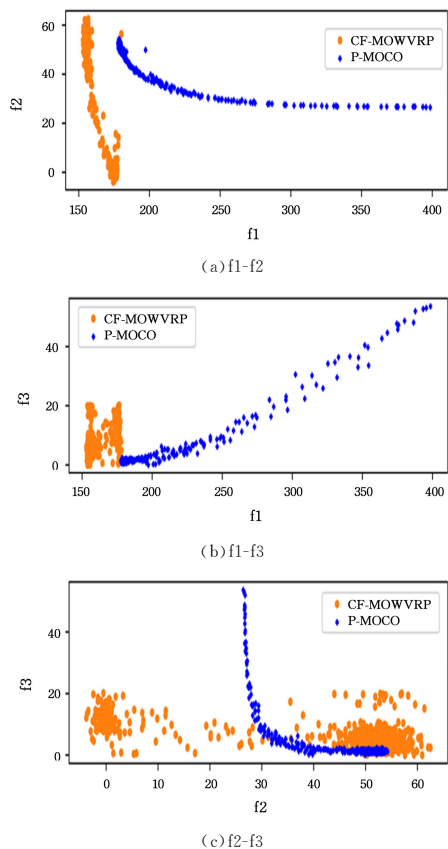


图 15 各目标间的近似 Pareto 前沿

Fig. 15 Approximate Pareto front between objectives

根据图 14 的空间分布可知,改进的 CF-MOWVRP 得到的近似帕累托前沿相较于 P-MOCO 更接近原点,即更接近真实的 Pareto 前沿。由图 15 可知,在 f_1 - f_2 , f_1 - f_3 , f_2 - f_3 的近似 Pareto 前沿图中,CF-MOWVRP 的近似 Pareto 前沿都更靠近左下角的原点,更接近真实的 Pareto 前沿。此外,对 f_3 ,即冲突解决等待时间目标,CF-MOWVRP 得到的值更小,说明 CF-MOWVRP 的冲突解决策略效果更优。

表 2 中,反向生成距离(IGD)表示 Pareto 最优前沿上的点到解集的最近距离的平均值,表征解集对真实 Pareto 前沿的覆盖程度。其值越小,表示解集越接近真实 Pareto 前沿。超体积(HV)表示解集覆盖的目标空间体积,相对于一个参考点,其值越大表示解集质量越好,用于衡量解集在 Pareto 前沿上的分布均匀性和广度。解集间覆盖度(C-metric)衡量一个解集相对于另一个解集的支配情况。

由表 2 可知,CF-MOWVRP 的 HV 小于 P-MOCO,表明 CF-MOWVRP 得到的解集分布均匀性和广度不及 P-MOCO。但 CF-MOWVRP 的 IGD 小于 P-MOCO,表明 CF-MOWVRP 得到的解集越接近真实 Pareto 前沿,且 CF-MOWVRP 的 C-metric 大于 P-MOCO,说明 CF-MOWVRP 的帕累托前沿基本覆盖 P-MOCO,CF-MOWVRP 在大部分情况下都找到了比 P-MOCO 更优的解。因此,改进的 CF-MOWVRP 相较 P-MOCO 更适用于 3 个目标的情况。

此外,使用 n 和 m 为 5,episode 为 500 的训练模型进行测试,得到 CF-MOWVRP 与经典算法 MOEA/D、NSGA-II 的测试结果的近似 Pareto 前沿,如图 16 所示,各目标间的近似

Pareto 前沿对比如图 17 所示,并得到相应的指标(见表 3)。

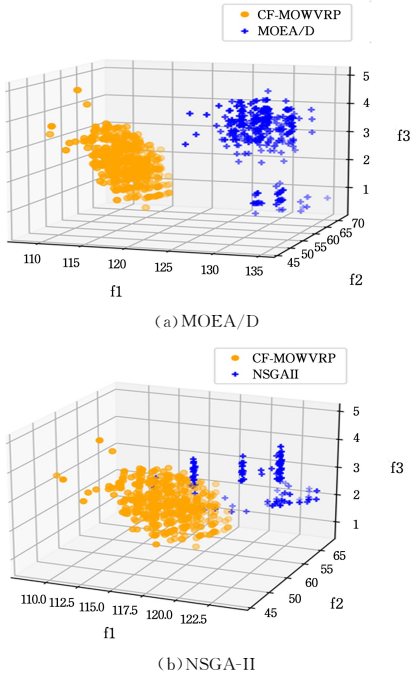


图 16 CF-MOWVRP 与经典算法的近似 Pareto 前沿

Fig. 16 CF-MOWVRP and the approximate Pareto frontier of classical algorithms

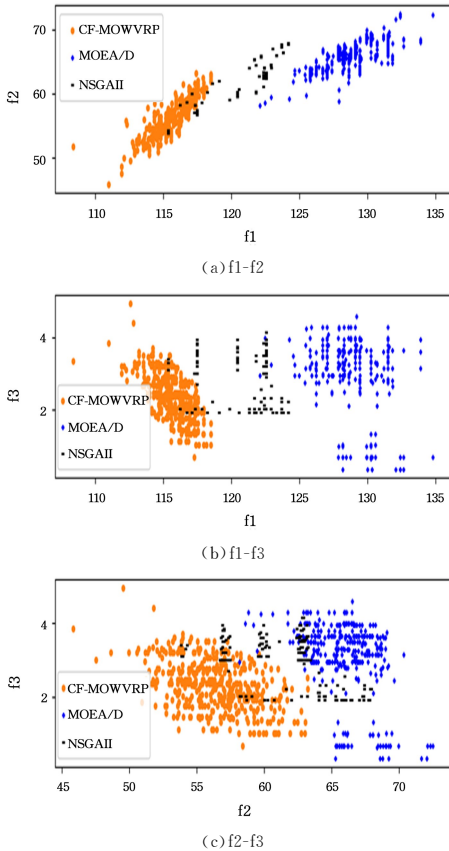


图 17 各目标间的近似 Pareto 前沿

Fig. 17 Approximate Pareto frontiers between objectives

根据图 16 的空间分布,相较于经典算法 MOEA/D 和 NSGA-II,CF-MOWVRP 得到的近似帕累托前沿更接近原点,即真实 Pareto 前沿。由图 17 可知,在各个目标对应的近

似 Pareto 前沿对比中,CF-MOWVRP 的近似 Pareto 前沿都更靠近左下角的原点,更接近真实的 Pareto 前沿,其次是 NSGA-II,最后是 MOEA/D。由表 3 可知,CF-MOWVRP 除了 HV 小于 MOEA/D,即 CF-MOWVRP 的解集分布均匀性和广度不及 MOEA/D 外,其 IGD 和 C-metric 都优于 MOEA/D 和 NSGA-II,表明 CF-MOWVRP 得到的解集比 MOEA/D 和 NSGA-II 的更接近真实 Pareto 前沿。CF-MOWVRP 的帕累托前沿基本覆盖 MOEA/D 和 NSGA-II,在大部分情况下都找到了比 MOEA/D 和 NSGA-II 更优的解。综上,CF-MOWVRP 相较于经典算法 MOEA/D 和 NSGA-II,得到的解更优。

表 3 CF-MOWVRP 与经典算法的近似 Pareto 前沿的指标值

Table 3 Index values of approximate Pareto front between CF-MOWVRP and classical algorithms

| Methods | HV \uparrow | IGD \downarrow | C-metric \uparrow |
|-----------|---------------|------------------|---------------------|
| CF-MOWVRP | 34 001.7341 | 127.9468 | 0.9830 |
| MOEA/D | 42 853.7730 | 143.7438 | 0.0001 |
| CF-MOWVRP | 33 945.8783 | 127.9468 | 0.9999 |
| NSGA-II | 33 794.7331 | 135.8309 | 0.1496 |

给定随机的节点和需求量,如表 4 所列,节点和 AGV 数量为 20,测试后得到的路径规划方案结果如表 5 所列。

表 4 测试节点信息

Table 4 Test node information

| 节点序号 | 坐标 | 需求量 | 节点序号 | 坐标 | 需求量 |
|------|--------|--------|------|---------|--------|
| 0 | (14,0) | 0 | 11 | (14,11) | 0.2333 |
| 1 | (6,1) | 0.3000 | 12 | (17,5) | 0.2333 |
| 2 | (11,1) | 0.3000 | 13 | (17,13) | 0.1333 |
| 3 | (2,4) | 0.2667 | 14 | (20,11) | 0.2000 |
| 4 | (6,7) | 0.2333 | 15 | (23,13) | 0.2333 |
| 5 | (2,10) | 0.2000 | 16 | (20,1) | 0.2333 |
| 6 | (5,13) | 0.1667 | 17 | (19,4) | 0.1000 |
| 7 | (8,12) | 0.2667 | 18 | (25,4) | 0.1667 |
| 8 | (11,6) | 0.0667 | 19 | (23,7) | 0.1667 |
| 9 | (11,9) | 0.2333 | 20 | (26,7) | 0.1667 |
| 10 | (14,4) | 0.3000 | | | |

表 5 路径规划方案信息

Table 5 Path planning scheme information

| 路径规划方案 | 距离 | 载重 | 冲突数量 |
|--|----------|--------|------------------|
| $R_1 = \{0, 5, 4, 3, 1, 0\}$ | 50.7279 | 1.0000 | |
| $R_2 = \{0, 8, 9, 7, 6, 0\}$ | 40.9706 | 0.7334 | |
| $R_3 = \{0, 2, 0\}$ | 6.0000 | 0.3000 | |
| CF-MOWVRP $R_4 = \{0, 15, 13, 11, 10, 0\}$ | 40.7279 | 0.8999 | $C_{total} = 18$ |
| $R_5 = \{0, 14, 12, 0\}$ | 32.7279 | 0.4333 | |
| $R_6 = \{0, 17, 18, 20, 19, 0\}$ | 32.7279 | 0.6001 | |
| $R_7 = \{0, 16, 0\}$ | 12.0000 | 0.2333 | |
| P-MOCO $R_1 = \{0, 19, 12, 5, 1, 0\}$ | 94.4853 | 0.9000 | |
| $R_2 = \{0, 16, 14, 10, 9, 0\}$ | 82.2426 | 0.9666 | |
| $R_3 = \{0, 17, 15, 6, 2, 0\}$ | 80.2426 | 0.8000 | |
| $R_4 = \{0, 18, 20, 8, 4, 3, 0\}$ | 102.4853 | 0.9001 | $C_{total} = 44$ |
| $R_5 = \{0, 7, 0\}$ | 26.9706 | 0.2667 | |
| $R_6 = \{0, 13, 0\}$ | 26.4853 | 0.1333 | |
| $R_7 = \{0, 11, 0\}$ | 20.0000 | 0.2333 | |

由表 5 可知,两种方法都只使用了 7 台 AGV 即可完全服务 20 个节点。CF-MOWVRP 得到的每台 AGV 的距离都比 P-MOCO 的小,且 CF-MOWVRP 的冲突数量为 18,而 P-MOCO 的冲突数量为 44,表明 CF-MOWVRP 比 P-MO-

CO 得到的路径规划方案的路径更短,冲突数量也更少。CF-MOWVRP 得到的路径规划方案存在的 18 个冲突点的信息如表 6 所列,18 个冲突点都属于节点占用冲突,优先级低的 AGV 使用短时间等待策略进行避让,以解决路径冲突。

表 6 冲突信息及解决策略

Table 6 Conflict information and resolution strategies

| 冲突时刻 | 冲突点 | 冲突 AGV | 冲突类型 | 解决策略 |
|------|--------|--------|--------|------------|
| 1 | (13,2) | 1,2 | | AGV2 短时间等待 |
| 1 | (15,2) | 4,5 | | AGV5 短时间等待 |
| 1 | (15,2) | 4,6 | | AGV6 短时间等待 |
| 1 | (15,2) | 5,6 | | AGV5 短时间等待 |
| 2 | (12,3) | 1,2 | | AGV2 短时间等待 |
| 2 | (16,3) | 4,5 | | AGV5 短时间等待 |
| 2 | (16,3) | 4,6 | | AGV6 短时间等待 |
| 2 | (16,3) | 5,6 | | AGV5 短时间等待 |
| 3 | (11,4) | 1,2 | 节点占用冲突 | AGV2 短时间等待 |
| 3 | (17,4) | 4,5 | | AGV5 短时间等待 |
| 3 | (17,4) | 4,6 | | AGV6 短时间等待 |
| 3 | (17,4) | 5,6 | | AGV5 短时间等待 |
| 4 | (18,5) | 4,5 | | AGV5 短时间等待 |
| 5 | (19,6) | 4,5 | | AGV5 短时间等待 |
| 6 | (20,7) | 4,5 | | AGV5 短时间等待 |
| 26 | (17,4) | 5,6 | | AGV5 短时间等待 |
| 27 | (16,3) | 5,6 | | AGV5 短时间等待 |
| 28 | (15,2) | 5,6 | | AGV5 短时间等待 |

CF-MOWVRP 得到的该测试实例的无冲突路径规划方案如图 18 所示。

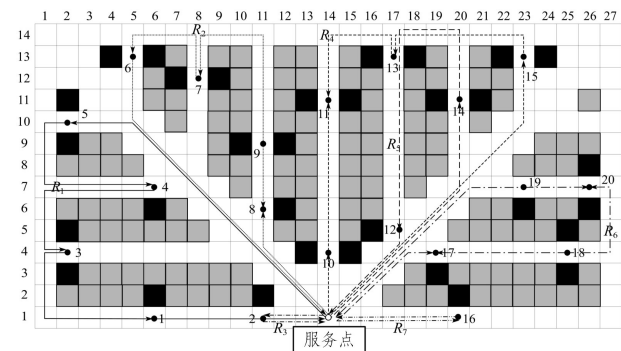


图 18 路径规划方案

Fig. 18 Path planning scheme

结束语 智能仓储路径优化研究对物流仓储领域具有重要意义,本文针对鱼骨仓储布局方案的智能仓储路径优化中的路径冲突问题,给出了改进措施。首先,通过构建基于网格法的智能仓储鱼骨布局方案及分区机制,提出了节点距离计算模型,提供了新的布局模型构建方法。通过设计 AGV 坐标保留表方法,实现了路径冲突的检测,并根据冲突类型制定了相应的解决策略,确保了路径规划的可行性和安全性。其次,本文建立了一个多目标路径规划模型,综合考虑了总运输距离、最大运输距离以及冲突解决等待时间,进一步提高了路径规划的效率和灵活性。最后,在 P-MOCO 算法的基础上,结合鱼骨布局方案与进化遗传搜索算法,通过创新的随机突变算子和交叉变异操作,提出了改进的无冲突多目标智能仓储路径优化算法 CF-MOWVRP,能够有效解决路径冲突问题,给出可权衡

的解决方案。最终仿真实验表明,所提算法能够有效解决路径冲突,路径优化算法更加有效。

未来的研究将更加偏重于以下方向:仓储布局越大,理论上存在的冲突越多,可以进一步针对不同大小的仓储布局进行研究;建立多服务点、带有时间窗等更加符合实际情况的无冲突多目标智能仓储路径规划数学模型;研究偏好输入情况对多目标处理的影响。

参 考 文 献

- [1] ZHANG Z, CHEN J, GUO Q. Application of Automated Guided Vehicles in Smart Automated Warehouse Systems: A Survey [J]. Computer Modeling in Engineering & Sciences, 2023, 134(3):1529-1563.
- [2] WAN Y, WANG S, HU Y, et al. Multiobjective Optimization of the Storage Location Allocation of a Retail E-commerce Picking Zone in a Picker-to-parts Warehouse [J]. Engineering Letters, 2023, 31(2):481-493.
- [3] CHEN Y, WU J, HE C, et al. Intelligent warehouse robot path planning based on improved ant colony algorithm [J]. IEEE Access, 2023, 11:12360-12367.
- [4] WU Z S, CHANG D F, GAI Y H. Optimization of Storage Location Allocation in Four Directional Shuttle Dense Warehousing System Based on Two Stage Hybrid Algorithm [J]. Journal of System Simulation, 2025, 37(5):1234-1245.
- [5] WANG Y F, CAO X H, GUO X. Warehouse AGV path planning method based on improved A* algorithm and short-term system state prediction [J]. Computer Integrated Manufacturing Systems, 2023, 29(11):3897-3908.
- [6] KAWABE T, NISHI T, LIU Z. Flexible Route Planning for Multiple Mobile Robots by Combining Q-Learning and Graph Search Algorithm [J]. Applied Sciences, 2023, 13(3):1879.
- [7] CAO X H, ZHU M. Optimization of Collision Avoidance Decision for Multi Automatic Guided Vehicles Based on Conflict Prediction [J]. Computer Integrated Manufacturing System, 2020, 26(8):2092-2098.
- [8] YAN X Y, MAO J L, WANG N, et al. CBS Multi Robot Path Planning Based on Conflict Avoidance Strategy [J]. Small Microcomputer System, 2025, 46(4):841-846.
- [9] LI T, DING P P, LIU J F. Multi stage and Multi AGV Path Planning for Goods to Person Picking System [J]. Journal of System Simulation, 2022, 34(7):1512-1523.
- [10] JIANG C K, LI Z, PAN S B, et al. AGVs collision free path planning based on improved Dijkstra algorithm [J]. Computer Science, 2020, 47(8):272-277.
- [11] ZHANG Z, GUO Q, CHEN J, et al. Collision-Free Route Planning for Multiple AGVs in an Automated Warehouse Based on Collision Classification [J]. IEEE Access, 2018, 6:26022-26035.
- [12] ZHAO X J, YE H, LI H, et al. Multi AGV path planning algorithm based on improved DDPG [J]. Computer Science, 2025, 52(6):306-315.
- [13] LIN X, YANG Z, ZHANG Q. Pareto Set Learning for Neural

Multi-Objective Combinatorial Optimization[C]//10th International Conference on Learning Representations (ICLR 2022). 2022.

- [14] WANG J, WENG T, ZHANG Q. A Two-Stage Multiobjective Evolutionary Algorithm for Multiobjective Multidepot Vehicle Routing Problem With Time Windows[J]. IEEE Transactions on Cybernetics, 2019, 49(7): 2467-2478.
- [15] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.
- [16] ELGHARABLY N, EASA S, NASSEF A, et al. Stochastic multi-objective vehicle routing model in green environment with customer satisfaction [J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(1): 1337-1355.
- [17] WANG L L, ZHANG M Z, WU F, et al. High dimensional objective evolutionary algorithm under dynamic penalty decomposition strategy [J]. Small and Micro Computer Systems, 2018, 39(10): 2154-2161.

- [18] KWON Y D, CHOO J, KIM B, et al. POMO: Policy Optimization with Multiple Optima for Reinforcement Learning[C]//Proceedings of the 34th Conference on Neural Information Processing Systems. 2020: 21188-21198.



GONG Jing, born in 1977, Ph.D, associate professor. Her main research interests include data network security technology, data mining technology and modern logistics technology.



SUN Zhixin, born in 1964, Ph.D, professor, doctoral supervisor. His main research interests include the theory and technology of network communication, computer network and security.

(责任编辑:何杨)