

•智能交叉科学与工程•

DOI:10.12454/j.jsuese.202300971



本刊网刊

基于改进多目标粒子群算法的钢板入库垛位分配研究

钟传捷^{1,2},程文明^{1,2*},杜润^{1,2},高旭超³,张琳³

(1.西南交通大学机械工程学院,四川成都 610031;2.轨道交通运维技术与装备四川省重点实验室,四川成都 610031;
3.四川省钢构智造有限公司,四川眉山 620033)

摘要:由于钢板类货物在自动化仓储系统中常采用重叠式堆码,针对自动化钢板料库中钢板入库的垛位分配问题,对料库的总体布局和入库作业流程进行分析,根据实际需求提出出库效率、板垛差异度和库存均衡度 3 项指标,用以评价垛位分配方案的优劣程度,并将其作为目标函数,以建立钢板入库垛位分配问题的多目标决策优化模型。为了求解该模型,基于 Pareto 占优概念,设计一种基于种群聚类度的多目标粒子群算法(PCDMOPSO),该算法以平衡求解的收敛性和多样性为需求,采用种群聚类度机制实时监控并调整算法的各认知参数和粒子的进化状态,并引入局部搜索策略以提高其外部存档中 Pareto 解集分布的多样性。在实际工况下的仿真结果表明:在不同的入库规模下,与经典的多目标算法 NSGA-II、MOEA/D-DE、MOPSO 相比,PCDMOPSO 在各目标下的寻优能力均呈现出明显优势;同时,分析各评价指标可知,PCDMOPSO 求解出的 Pareto 解集相较于其他算法具有更好的收敛性、分布性、多样性和占优性。为验证所提方法的可行性,设计算法改进策略的对比试验,结果表明所提出的改进措施对求解的改善效果较为明显。最后,从输出优化前后的堆码情况可以观察到,优化后的垛位分配方案相较于优化前的传统堆码方式,其优化效果较为显著,可以有效满足钢板入库问题中对各评价指标的实际需求,为钢板类货物在自动化仓储系统中的垛位分配和出入库管理提供了有力的决策支持。

关键词:钢板料库;垛位分配;粒子群优化;多目标优化;Pareto 最优解

中图分类号:TP273

文献标志码:A

文章编号:2096-3246(2025)04-0150-15

传统的钢结构加工制造企业在其以往的管理运营中,对原料库的管理重视程度并不高,一般仅保持正常运转即可,由此导致原料库的管理存在诸如生产作业任务杂乱、设备基础条件差异大、作业人员来源复杂、主动型作业导致效率较低等问题^[1]。随着自动化立体仓库的相关技术和研究逐渐发展,制造产业的自动化仓储管理取得长足进步。自动化钢板料库作为制造企业进行生产活动的起点,既是生产线存放钢板原材料的主要场地,也是重要的货物中转场地,承担着钢材的原料分类、入库、堆码、保存、拣选以及根据生产计划供应所需原材料的功能。针对钢板的入库作业,垛位分配根据待入库钢板属性、生产计划以及库存钢板的分布情况为当前批次的待入库钢板指派合理的垛位,以便对在库钢板进行盘点

管理和出入库运送。在制定钢板入库的垛位分配决策时,往往存在多个须重点满足的目标。一方面,由于料库优先服务于生产,故在制定钢板入库策略时,须优先考虑可以降低各批次钢板出库时间的方案;另一方面,须考虑钢板尺寸、项目批次等属性,在满足堆码安全性要求的前提下尽可能增加存储率;同时,使用分区分散存放的策略,使料库中桥吊、RGV 轨道车的作业均衡性一致。因此,针对钢板类货物,合理的垛位分配决策可以在降低翻垛频率、提升料库吞吐效率的同时节省管理成本,对企业生产活动的精益化管理具有重要意义。

目前,国内外学者对自动化仓储管理系统货位分配的研究,主要通过建立适当的数学模型和约束条件,设计不同的优化算法对其进行求解。Zhang 等^[2]借

收稿日期:2023-11-30 修回日期:2024-05-24 网络出版日期:2024-05-30

基金项目:四川省重点研发项目(2022YFG0241;2022YFG0245);中车“十四五”科技重大专项科研课题(2021CHZ010-3)

作者简介:钟传捷(1999—),男,硕士生。研究方向:机械设计与仿真;工业工程。E-mail:271143516@qq.com

*通信作者:程文明,教授,E-mail:wmcheng@home.swjtu.edu.cn

鉴非传统布局仓库和储位分配的优势,开发一种基于烟花算法的储位分配优化算法,以处理非传统布局仓库的储位分配问题。Nastasi等^[3]设计了基于遗传算法的3种不同解决方案,并将其应用于炼钢行业自动化仓储策略的多目标优化,测试评估了不同优化方法所实现的性能差异。Park等^[4]分析了不同的周转率下货物堆码布局与生产计划之间的关系。Guerrero等^[5]将多层仓库中的产品分类问题描述为混合整数线性规划模型,设计了基于Rollout的启发式算法,并对该问题进行求解。杨玮等^[6]设计了一种双层遗传算法,将多载具协同作业下立体仓库的货位分配与作业调度问题进行集成优化,取得了良好的优化效果。王娟等^[7]针对湿度敏感类货物,引入了湿度变化系数构造立体仓库的货位湿度计算函数,并提出了一种基于自适应多种群遗传算法的湿度敏感类物品立体仓库的货位分配方法。何李等^[8]将货位分配和作业调度进行集成优化,并设计两阶段狼群算法对其进行求解,其动态关联反馈模式相比于其他优化方式更易得到满意解。陈雷等^[9]从任务提前调度、货位分配和任务序列优化3个方面设计针对纺织企业的调度模式,提出工人操作时间的预测方法和改进的启发式算法以提升立体仓库的调度效率。

虽然AS/RS(automated storage and retrieval system)的相关研究随着现代化建设中产业链结构和相关技术的不断迭代优化取得了极大成果,然而针对一些具有特殊类别或需求的货物,其在AS/RS基本布局、硬件配置和存储策略的设计中均需考虑到不同的影响。例如,集装箱或一些大型板材类货物,因大多采用重叠式堆码,所以储位分配会有不同的需求。Saleh等^[10]通过将矩阵整合成多维结构建立集装箱堆垛模型,以此处理港口集装箱存储的动态问题。Zhang等^[11]在出库集装箱堆放问题中制定了一套最优堆垛策略用以减少额外倒箱,以此设计动态规划模型,并集成优化算法和机器学习技术用以求解该问题。Kim等^[12]提出了一种能够适应随时间变化的船舶操作工作量的动态策略的推导方法,与传统的静态策略相比,该策略显著地减少了集装箱的整体装卸时间。张志英等^[13]针对船舶制造中钢板分散时间入库引起的大量倒垛问题,建立了多时间段的倒垛作业优化模型,提出了一种动态规划的启发式算法和变邻域搜索算法组成的两阶段求解方法,大大降低了大规模倒垛问题的运行成本。赵天毅等^[14]针对船舶装配件的入库垛位分配问题,考虑了船舶出库的不确定性特点,设计了基于遗传算法的热启动分支定界算法,克服了启发式算法求解不稳定的问题,且相比于传统优化

算法,其求解时间也大大降低。侯俊等^[15]针对造船厂提出的分段管理原则,优化了出入库倒垛方案,设计了基于模拟退火的双层遗传算法进行优化求解,降低了成本预算和倒垛时间。Xu等^[16]建立了双层组合优化模型和配送操作的多阶段决策过程研究钢板翻垛问题,并将贪婪算法和遗传算法结合起来对其进行求解,证明该方法可以在较少运行时间内得到满意解。

此外,在以往的自动化立体仓库储位分配研究中,许多研究者考虑的影响因素较为单一,或将多目标的复杂因素以一定方式转化为单目标加以优化。张天星^[17]针对立体仓库的作业效率、稳定性和离散程度需求建立了储位分配模型,并设计了一种多种群映射的遗传算法,有效解决了储位的动态可分问题。叶康^[18]设计了一种混合蛙跳算法,该算法相较传统算法性能优越,在用其求解双向式立体仓库中的货位分配问题时取得了良好效果。钢板入库的垛位分配问题可以归类为带约束的A型装箱问题,其作为已被证明的NP难问题^[19],往往存在多个相互冲突的优化目标,很难通过构建单一数学模型的方式采用传统解析式算法求解。基于Pareto最优理论的多目标研究方法相比于将多个目标进行加权求和而言,由于其输出的是一个解集,能够排除权重系数取值这一主观性的先验因素,所以研究者们基于Pareto占优思想设计了大量多目标优化算法。Zhou等^[20]通过引入基于收敛性的标准,设计了基于遗传的混合多目标进化算法。Guan等^[21]为了解决多目标粒子群算法易陷入局部最优的缺点,引入了levy飞行和双存档机制,增加了算法迭代时的全局搜索能力。Zhou等^[22]综合考量了车间机械加工和起重机运输过程,提出了基于粒子滤波的levy飞行策略以杂化粒子群,使得在柔性作业车间绿化调度问题中算法可以求得更具多样性的解集。李浩君等^[23]采用种群曼哈顿距离实时监测算法进化状态并调整认知参数和更新方式,根据进化状态平衡收敛性和多样性,设计的基于曼哈顿距离的自适应多目标粒子群算法相比于其他算法具备更好的收敛性和多样性。Li等^[24]针对MOPSO算法提出了一种领导者选择策略,其通过动态判断个体之间相似度的方式,结合多样性的增强学习策略,实现了种群的多样性监测,并有效避免了个体退化的问题。

综上,国内外学者对多目标问题进行了大量研究,针对自动化立体仓库的货位分配研究模式也已较为成熟,但使用Pareto最优理论研究钢板类货物的入库垛位分配问题的研究尚且较少。因此,本文从企业进行生产活动与物流管理的实际需求出发,以某钢结

构加工制造基地的自动化钢板料库为研究对象,针对其钢板入库的垛位分配问题,以钢板出库效率、板垛差异度和库存均衡度为优化指标,构建钢板入库的多目标垛位分配模型,设计高度匹配问题特殊性的基于种群聚类度的多目标粒子群算法(PCDMOPSO)求解 Pareto 最优解集,并通过仿真试验,从解集的收敛性、均匀性、多样性和占优性 4 个维度评估解的质量,以选择合理的钢板入库垛位分配方案。

1 问题描述

图 1 为某企业的自动化钢板料库总体布局示意图。设 Z 为料库库区 z 的集合, $z \in Z$; Y 为料库所设垛位行数集合, $y \in Y$; X 为所设垛位列数集合, $x \in X$; 设 J 为垛位集合。则有第 y 行、第 x 列,位于 z 库区的垛位 $j(j \in J)$ 的坐标可表示为 $b_j^z = (x_j, y_j)$ 。钢板料库的基本布局可以描述为一个由 y 行、 x 列垛位组成且被均匀划分为 z 等块区域的矩形场地。相邻垛位的几何中心距离在 x 方向上为 l_s , 在 y 方向上为 w_s , 垛位限高为 H 。设待入库钢板集合为 I , 则第 i 块钢板的长度、宽度、厚度和项目号可表示为集合 (l_i, w_i, t_i, p_i) , $i \in I$ 。针对在库钢板, 设垛位 j 上的钢板的平均长度为 l_j , 平均宽度为 w_j , 初始堆垛高度为 h_j 。

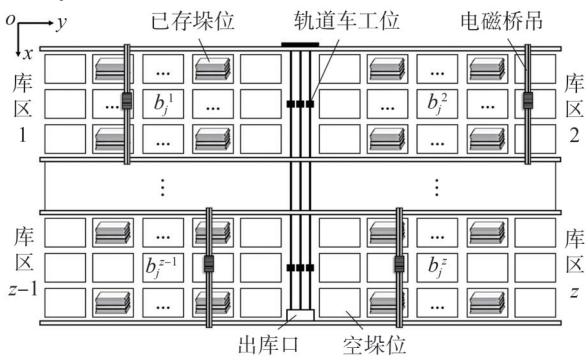


图 1 自动化钢板料库总体布局

Fig. 1 Overall layout of automatic steel plate storage

为便于研究,对该问题提出如下假设:每个库区均配备一台电磁桥吊负责钢板的搬运,在作业过程中一次只能吊起一块钢板,且不可进行跨区作业。电磁桥吊和自动引导运输车(automated guided vehicle, AGV)轨道车运行速度恒定,由于其运行速度较慢,忽略其启动和制动时间。桥吊在 x 方向上的运行速度为 V_x , y 方向上的运行速度为 V_y , 起升机构运行速度为 V_{up} , 起升高度为 h_{up} 。在料库中间的 x 方向设置 AGV 轨道,其上每经过一个库区设置一轨道车工位,记为 b_{agv}^z 。在进行原料出库作业时,钢板经桥吊从垛位搬运至 AGV 轨道车上后,其沿轨道方向运行至出口口 b_0 进行出库,运行速度为 V_{agv} 。具体钢板入库流程见图 2。

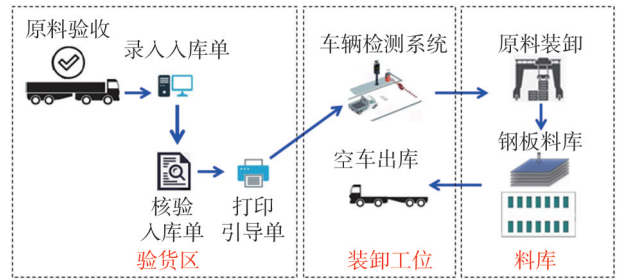


图 2 钢板入库流程图

Fig. 2 Flow chart of steel plate import and export

来料卡车通过原料验收后即需要制定垛位分配决策,而在此之前,基于分类堆码和分散存取原则,需要先对待入库钢板进行入库预分配。一方面,由于待入库集合中各钢板尺寸差异较大,若采取混合堆放的模式会对后续的堆码顺序和翻垛作业等造成较大难度,不便于管理,因此,在预分配阶段需针对钢板的尺寸属性进行提前预分类;另一方面,为了提升钢板的出入库存取效率,同类型的钢板需分散堆码至不同库区。例如,若某一批次钢板在预分配中按长度区间 $[l_{min}, l_s]$ 、 $[l_s, l_s]$ 被划分为了两类,基于分类堆码原则,两类钢板应各自分配不同的区域进行堆码;且为了满足分散存取原则,每类钢板应在不同库区为其分配垛位,不可全部存放于同一库区。图 3 为入库预分配示意图。图 3 中,点网格区域和横线区域分别代表了不同尺寸区间的钢板,且每类分散在不同库区。

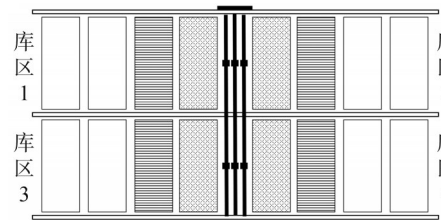


图 3 入库预分配示意图

Fig. 3 Schematic diagram of pre-allocation in storage

2 模型建立

通过对钢板料库实际的场景布局和使用需求调研分析,发现在入库作业中垛位分配需考虑的主要因素大致分为 3 点。首先,由于料库服务于生产,所以对入库垛位的选择应尽量保证靠近出口口,以尽可能提高生产出库的作业效率;其次,基于堆码安全性和减少倒垛量原则,一方面需要保证堆垛的安全因素,除了须合理设置堆垛高度外,若大小钢板随意混堆极易出现安全问题,且提升了后续倒垛作业难度,所以需要不同尺寸的钢板进行分类管理;最后,钢板分配的均衡程度也是评价垛位分配规划效果的重要指标,各库区分配的入库均衡度越高则越能保证库区设备使用率接近,在方便管理和充分调动全库区资源的同

时也降低了相应的维护成本。因此,针对上述因素,设计出库效率、板垛差异度和库存均衡度3个评价指标作为垛位分配的优化目标。

2.1 出库效率指标

出库效率是衡量垛位分配方案的重要指标之一,由于料库服务于生产,基于效率优先原则,应尽量减少钢板的出库时间,其影响因素主要由设备运行方式和垛位位置决定。在桥吊和轨道车运行速度固定的情况下,目标垛位设置得离轨道车工位越近、库区选择离出库口越近,其出库效率越高。根据此规则,出库效率指标 f_1 的数学模型为:

$$f_1 = \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \sum_{k=\frac{h_j}{dh}}^{\frac{H}{dh}} T_{ij} X_{ijk} \quad (1)$$

$$T_{ij} = \max \left(\frac{d_{jx}}{V_x}, \frac{d_{jy}}{V_y} \right) + \frac{2h_{up} - h_j}{V_{up}} + \frac{d_z}{V_{agv}} \alpha \quad (2)$$

$$\begin{cases} d_{jx} = |b_j - b_{agv}^z|_x l_s, \\ d_{jy} = |b_j - b_{agv}^z|_y w_s, \\ d_z = |b_{agv}^z - b_0|_y w_s \end{cases} \quad (3)$$

式(1)~(3)中: T_{ij} 为 j 垛上单块钢板 i 出库所花时间,其由桥吊搬运时间、起升机构升降时间和AGV轨道车运行时间组成, T_{ij} 值越小代表出库效率越高; dh 为在库钢板单位厚度; d_{jx} 和 d_{jy} 分别为 x 和 y 方向上 j 垛到轨道车工位的距离; d_z 为库区 z 的轨道车工位到出库口 b_0 的距离; α 为AGV集中载运比系数; X_{ijk} 为决策变量,其值为:

$$X_{ijk} = \begin{cases} 1, & \text{钢板}i\text{位于垛位}j\text{的第}k\text{层;} \\ 0, & \text{其他} \end{cases} \quad (4)$$

式中, $k=1,2,\dots,\frac{h_j}{dh},\dots,\frac{H}{dh}$ 。

2.2 板垛差异度指标

为了保证钢板堆垛的稳定性和减少倒垛频率,为钢板和垛位均赋予各自的特征指数,以此对钢板和垛位类型进行归类。设 $L_{ij} = |l_i - l_j|$ 表示钢板 i 和垛位 j 长度匹配指数; $W_{ij} = |w_i - w_j|$ 表示钢板 i 和垛位 j 宽度匹配指数; P_{ij} 为钢板 i 和垛位 j 项目相关度指数,其取值为:

$$P_{ij} = \begin{cases} 0, & \text{板垛项目号相同;} \\ 0.3, & \text{垛中存在同项目板;} \\ 0.5, & \text{板垛项目号不同;} \\ 1, & \text{目标垛位为空} \end{cases}$$

则板垛差异度指标 f_2 的数学模型为:

$$f_2 = \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \sum_{k=\frac{h_j}{dh}}^{\frac{H}{dh}} \beta_{ij} X_{ijk} \quad (5)$$

式中: β_{ij} 为板垛特征差系数, $\beta_{ij} = \omega_1 L_{ij} + \omega_2 W_{ij} + \omega_3 P_{ij}$,用以反映钢板 i 和垛位 j 的特征差异度,其值越小代表板垛之间特征越接近;权重系数 $\omega_1, \omega_2, \omega_3$ 的取值采用层次分析法确定。

2.3 库存均衡度指标

根据料库的基本布局,一方面,由于料库分为多个库区,分散入库策略可以使来料卡车前往不同的入库口进行卸货,以充分调动料库设备执行入库作业,有利于缓解卡车入库的排队阻塞现象,提高入库效率;另一方面,钢板分散入库可提高料库的垛位利用率,使得各库区设备的使用率和磨损率平衡,以降低检修成本;并且,当某一库区设备出现异常或出库任务较多时,可以先安排其他库区的同类型钢板出库,便于出库作业管理。根据以上的分类存储原则,库存均衡度指标 f_3 的数学模型为:

$$f_3 = \sqrt{\frac{1}{|Z|} \sum_{z=1}^{|Z|} (|I_z| - \bar{I})^2} \quad (6)$$

式中, I_z 为分配给 z 库区的钢板集合, \bar{I} 为各库区钢板数量的均值。

综上,自动化钢板料库垛位分配多目标优化模型形式如下:

$$\begin{cases} \min f_1 = \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \sum_{k=\frac{h_j}{dh}}^{\frac{H}{dh}} T_{ij} X_{ijk}, \\ \min f_2 = \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \sum_{k=\frac{h_j}{dh}}^{\frac{H}{dh}} \beta_{ij} X_{ijk}, \\ \min f_3 = \sqrt{\frac{1}{|Z|} \sum_{z=1}^{|Z|} (|I_z| - \bar{I})^2} \end{cases} \quad (7)$$

优化模型的约束条件如下。

1) 垛位限高约束为:

$$\sum_{i=1}^{|I|} \sum_{k=1}^{\frac{H}{dh}} t_i X_{ijk} \leq H, j \in J \quad (8)$$

2) 入库钢板的尺寸约束为:

$$0 < l_i X_{ijk} < l_s, i \in I, j \in J, k = 1, 2, \dots, \frac{h_j}{dh}, \dots, \frac{H}{dh} \quad (9)$$

$$0 < w_i X_{ijk} < w_s, i \in I, j \in J, k = 1, 2, \dots, \frac{h_j}{dh}, \dots, \frac{H}{dh} \quad (10)$$

3) 预分配后钢板 i 能够被堆放至垛位 j 的尺寸条件为:

$$\left| l_i - l_j \right| X_{ijk} \leq (\delta l_{j\max} - l_{j\min}),$$

$$i \in I, j \in J, k = 1, 2, \dots, \frac{h_j}{dh}, \dots, \frac{H}{dh} \quad (11)$$

4) 入库分散性约束为, 即所有钢板不能置于于同一个库区, 可表示为:

$$\sum_{i=1}^{|I|} z_i X_{ijk} < |I|, j \in J, k = 1, 2, \dots, \frac{h_j}{dh}, \dots, \frac{H}{dh} \quad (12)$$

5) 每块钢板能且仅能堆放于某个垛位的某一层上, 即不允许钢板重复堆放, 可表示为:

$$\sum_{j=1}^{|J|} \sum_{k=1}^{\frac{H}{dh}} X_{ijk} = 1, i \in I \quad (13)$$

6) 每个垛位的任何一层均能且仅能堆放一块钢板, 即不允许出现堆码冲突, 可表示为:

$$\sum_{i=1}^{|I|} X_{ijk} = 1, j \in J, k = 1, 2, \dots, \frac{h_j}{dh}, \dots, \frac{H}{dh} \quad (14)$$

式(7)~(14)中, t_i 为钢板 i 的厚度, δ 为尺寸偏差系数, $l_{j\max}$ 和 $l_{j\min}$ 分别代表垛位 j 中钢板长度的最大值和最小值, z_i 为钢板 i 所分配至的库区。

3 算法设计

3.1 基本多目标粒子群算法

传统的粒子群算法思想是模仿鸟类觅食行为, 从随机解出发, 各个粒子通过不断追踪并刷新个体历史最优位置 p_{best} 和种群全局最优位置 g_{best} 来不断更新自身的状态。为了解决复杂的多目标优化问题, 基于 Pareto 占优理论, 多目标粒子群算法应运而生, 其通过设置一个外部存档的方式来筛选并储存每代的 Pareto 最优解, 并在外部存档中选择出每代的 g_{best} 。粒子 r 在 $t+1$ 代的速度及位置由第 t 代的速度和位置更新而来, 更新方程如下:

$$v_{rd}^{t+1} = \omega v_{rd}^t + c_1 \text{rand}_1 (p_{\text{best}rd}^t - x_{rd}^t) + c_2 \text{rand}_2 (g_{\text{best}rd}^t - x_{rd}^t) \quad (15)$$

$$x_{rd}^{t+1} = x_{rd}^t + v_{rd}^{t+1} \quad (16)$$

式(15)~(16)中, t 为更新代数, ω 为惯性权重, c_1 、 c_2 为学习因子, rand_1 、 rand_2 为 (0,1) 之间相互独立的随机数, v_{rd}^t 为粒子 r 在迭代至第 t 代时第 d 维的速度, x_{rd}^t 为粒子 r 在迭代至第 t 代时第 d 维的位置。

多目标粒子群算法由于其独特的粒子更新方式, 具有全局寻优能力强、易于实现、收敛速度较快且适用于多种类型问题等的优点。与此同时, 该算法也存在对参数设置较为敏感、容易陷入局部最优等的缺陷, 为了克服这些缺陷, 需要对算法进行改进, 制定合适的更新策略加以辅助, 以获得更好的求解效果。

3.2 编码方式

编码方式会直接影响垛位分配优化算法的求解效率, 所以须慎重选择合适的编码方式。针对钢板类货物堆码储存特性和料库垛位分配原则, 结合本文建立的数学模型, 采用整数编码策略, 设整数向量 $\mathbf{B} = [b_1, b_2, \dots, b_i, \dots, b_{|I|}]$ 表示种群中的一个个体, 即一种垛位分配方案, b_i 表示当前垛位分配方案中钢板 i 所分配的目标垛位坐标, 即当决策变量 $X_{ijk}=1$ 时, 有 $b_i=j$, 堆码层数 k 在钢板 i 被分配至垛位 j 后通过料库的堆码状态变更而确定。粒子编码如图 4 所示。所有钢板的目标垛位坐标按顺序组合在一起形成一组整数编码串, 构成种群中的一个个体。

5	11	7	6	...	1	9	...	10	3
钢板1		钢板2		...	钢板 <i>i</i>		...	钢板 <i>I</i>	

图 4 粒子编码示意图

Fig. 4 Particle coding schematic diagram

3.3 基于种群聚类度的多目标粒子群算法(PCDMO-PSO)

针对本文提出的钢板入库垛位分配多目标优化模型, 基于粒子群算法和 Pareto 占优概念, 提出了一种 PCDMOPSO 算法, 通过引入种群聚类度概念监测算法的迭代状态, 实时调整算法的速度更新模式和相关参数, 以平衡种群的多样性和收敛性; 同时设置外部存档用以存储每代种群选出的 Pareto 最优解, 并设计局部搜索策略帮助算法跳出局部最优。

3.3.1 定义种群聚类度和 ζ 指数监测迭代状态

由传统的多目标粒子群算法的粒子速度更新方程 (15) 可知, 粒子会受个体认知和社会认知的影响不断寻优并收敛靠拢, 即随着迭代次数的增加, 粒子之间的距离会越来越近。文献[25]提出了粒子间相似度的概念:

$$s(r_1, r_2) = 1 - \frac{d(r_1, r_2)}{d_{\max}},$$

$$d_{\min} \leq d(r_1, r_2) \leq d_{\max} \quad (17)$$

式中, $d(r_1, r_2)$ 为粒子 r_1 和 r_2 之间的欧氏距离, d_{\min} 和 d_{\max} 分别为所有种群中所有粒子之间的最大距离和最小距离。由式(17)可知, 粒子相似度 $s(r_1, r_2)$ 的取值范围为 [0,1], 两个粒子距离越近则相似度越大。据此定义种群中各粒子与当代全局最优粒子 r_g 的平均相似度为种群聚类度, 即:

$$c(t) = \frac{1}{R} \sum_{r=1}^R s(r, r_g) \quad (18)$$

式中, R 代表种群规模, $s(r, r_g)$ 为粒子 r 与全局最优粒子之间的相似度, $c(t)$ 表示迭代至 t 代时的种群聚类度指标。设算法的最大迭代次数为 T , 则定义前 t 代的历史最小种群聚类度为 $c_{\min}(t)$, 前 t 代的历史最大种群聚类

度为 $c_{\max}(t)$ 。

由式(15)、(16)可知,基本的多目标粒子群算法速度更新方程由上一代粒子自身的速度惯性、个体认知和社会认知部分组成,其固定的速度更新方式导致求解的盲目性较高,很容易陷入局部最优。PCDMOPSO算法通过引入种群聚类度定义指数 ζ 对粒子的飞行状态进行实时监控:

$$\zeta = \frac{T-t}{T} s(r, r_g) c(t) \quad (19)$$

由式(19)可知, ζ 的取值范围为 $[0, 1]$, 为了利于算法收敛,设计 ζ 的值随着迭代次数的增加呈递减趋势。

3.3.2 PCDMOPSO 算法的粒子速度更新方式

在迭代初期,算法采用的是基本粒子群速度更新方式,即式(15)。随后,种群将进入搜索寻优阶段,每经过一次迭代,算法都将更新一次种群聚类度。当 t 代的种群聚类度小于历史最小种群聚类度时,代表当代的种群多样性较好,收敛性较差,需要提升局部开采精度,采用式(20)所示的速度更新方式帮助种群收敛:

$$v_{rd}^{t+1} = \omega v_{rd}^t + c_2 \text{rand}_2(g_{\text{best } rd}^t - x_{rd}^t) \quad (20)$$

当 t 代的种群聚类度大于历史最小种群聚类度时,代表当代的进化状态趋于收敛,算法由 ζ 指数作为概率,采用 levy 飞行策略提高全局开采能力,以跳出局部最优。基于 levy 飞行的粒子速度更新方程为:

$$v_{rd}^{t+1} = \omega v_{rd}^t + c_1 \text{rand}_1(p_{\text{best } rd}^t - x_{rd}^t) + c_2 \text{rand}_2(g_{\text{best } rd}^t - x_{rd}^t) + c_3 \text{levy}(g_{\text{best } rd}^t - \text{rand}_3) \quad (21)$$

式(20)~(21)中, ω 为惯性权重, c_1, c_2, c_3 为学习因子, $\text{rand}_1, \text{rand}_2, \text{rand}_3$ 均为相互独立的随机数。Levy 飞行的步长符合 levy 分布,常使用 mantegna 算法模拟,其计算方程为^[26]:

$$\text{levy}(\lambda) = \frac{\mu}{|v|^{1/\beta}} \quad (22)$$

$$\mu \sim N(0, \sigma_\mu^2) \quad (23)$$

$$v \sim N(0, \sigma_v^2) \quad (24)$$

$$\sigma_\mu = \left[\frac{\Gamma(1+\beta) \times \sin \frac{\pi\beta}{2}}{2^{\frac{\beta-1}{2}} \times \beta \times \Gamma\left(\frac{1+\beta}{2}\right)} \right]^{\frac{1}{\beta}} \quad (25)$$

式(22)~(25)中, σ_v 通常取值为 1, Γ 为伽马函数, $\beta \in [1, 1.5]$, $\text{levy}(\lambda)$ 为随机步长, λ 为幂次数。

综上,设算法初期采用的式(15)为 v_1 ; 当 t 代的种群聚类度小于历史最小种群聚类度时,采用的式(20)为 v_2 ; 当 t 代的种群聚类度大于历史最小种群聚类度

时,将会有 ζ 的概率选择式(21)作为 v_3 , 以提高全局开采能力,则 PCDMOPSO 算法的粒子速度更新方程为:

$$v = \begin{cases} v_1, t \leq 2 \cup [c(t) > c_{\min}(t) \cap \text{rand} \geq \zeta]; \\ v_2, c(t) \leq c_{\min}(t); \\ v_3, c(t) > c_{\min}(t) \cap \text{rand} < \zeta \end{cases} \quad (26)$$

式中, rand 为一个随机数,位置更新模式采用式(16)。PCDMOPSO 算法通过采用不同的粒子速度更新策略,可以实时监控并干预算法的寻优模式,平衡算法全局搜索和局部开采的能力,使得求解在具备良好的多样性的同时还能够提升寻得最优解的精度,满足了算法收敛性的需求。

3.3.3 认知参数的自适应调整策略

由式(26)可知,各速度更新方程中的认知参数包括惯性权重 ω 、个体认知参数 c_1 、社会认知参数 c_2 和 levy 飞行认知参数 c_3 。结合种群聚类度概念,设计 PCDMOPSO 算法的自适应参数如下:

$$\omega = \begin{cases} 0.4, t \leq 2; \\ (e_\omega + 1) \left[\frac{T-t}{T} (\omega_{\max} - \omega_{\min}) + \omega_{\min} \right], t > 2 \end{cases} \quad (27)$$

$$c_1 = \begin{cases} 1, t \leq 2; \\ e_\omega \left[\frac{T-t}{T} (c_{1\max} - c_{1\min}) + c_{1\min} \right], t > 2 \end{cases} \quad (28)$$

$$c_2 = \begin{cases} 2, t \leq 2; \\ e_\omega + 1, t > 2 \end{cases} \quad (29)$$

$$c_3 = \begin{cases} 0.01, t \leq 2; \\ e_\omega, t > 2 \end{cases} \quad (30)$$

其中,

$$e_\omega = \exp\left(\frac{-1}{|c_{\max}(t) - c(t)| + 1}\right) \quad (31)$$

式(27)~(30)中, $c_{1\max}, c_{1\min}$ 分别表示个体认知系数的最大值和最小值, $\omega_{\max}, \omega_{\min}$ 分别表示惯性权重的最大值和最小值。

从式(27)~(31)可以看出,各认知参数的值均与粒子的种群聚类度和当前迭代次数有关。为满足算法收敛性随着迭代次数的增加而不断增强的需要,设计 ω 和 c_1 整体呈递减趋势;同时,由于算法在迭代过程中会不断趋于收敛,则需在迭代后期增加其全局寻优的能力以防止陷入局部最优,故设计 c_2 整体呈递增趋势; c_3 可使算法在迭代后期拥有更多跳出局部最优的可能。

3.3.4 局部搜索策略与外部存档维护

外部存档用于筛选并储存种群在迭代进化中所产生的 Pareto 最优解。不同于基本的多目标粒子群算法,PCDMOPSO 算法在每次迭代至更新外部存档时,

都会对种群进行一次局部搜索策略,随后将种群、外部存档和局部搜索后的解集进行合并去重,再根据 Pareto 支配关系选择出新的非支配解集,若支配解个数没有超出外部存档的设定大小,则采用拥挤距离策略对各 Pareto 解进行选拔以组建新的外部存档,以此实现对外部存档的更新和维护。

1) 局部搜索策略

设 archive 为外部存档,其容量为 U_a ,参考遗传算法选择、交叉、变异操作,视每个个体为一条染色体并对种群进行遗传操作。先将种群 popm 和上一代外部存档 archive^{t-1} 组合,构成新的临时种群 popm',并将 popm' 的各染色体随机两两组合;随后,对每组的两条染色体随机截取一段,执行交叉操作;最后,根据设定的变异概率选取个体执行变异操作,即随机标记目标染色体的一段基因,并重新对之编码。如此便完成一次局部搜索操作,其输出的解集 popm'' 有更多跳出局部最优的可能,可帮助算法搜索到新的潜在解空间。

2) 拥挤距离筛选策略

将上一代的外部存档 archive^{t-1} 与当代更新完成的 popm 和经过局部搜索后输出的 popm'' 进行合并去重,经过非支配解判断筛选出临时候选集 E 后,若其中的 Pareto 解的数量超过了外部存档的设定值 U_a ,则使用拥挤距离筛选策略对之进行裁剪。设 $f_g(B_e)$ 为 Pareto 最优解 B_e 的第 g 个目标函数值($g=1,2,3$; e 为临时候选集中的任一解, $e \in [1, |E|]$), 分别在 3 个目标空间中对临时候选集进行排序,排序的最大值和最小值解的距离均赋值无穷,其余解的距离由其序列的相邻两位计算得出,再对 3 个维度的距离求和,即为各 Pareto 解的拥挤距离,具体计算方式如下:

$$D_{eg} = \sum_{g=1}^3 \frac{f_g(B_{e+1}) - f_g(B_{e-1})}{f_g^{\max} - f_g^{\min}} \quad (32)$$

式中, D_{eg} 为 B_e 在前 g 个目标空间的拥挤距离, f_g^{\max} 和 f_g^{\min} 分别代表在解空间中目标函数所能达到的最大值和最小值。在求出 E 中各 Pareto 最优解的拥挤距离后,对其进行降序排列,前 U_a 个拥挤距离最大的解即组成新一代的外部存档 archive^t。

3.3.5 PCDMOPSO 算法流程

PCDMOPSO 流程如下。

输入: 垛位和待入钢板信息, 种群规模 M , 目标函数 $f_g(g=1,2,3)$, 迭代次数 T 。

输出: 非支配解集外部存档 archive。

1) 初始化种群 popm、各初始参数和外部存档 archive 等。

2) 由式(7)计算各粒子目标函数值作为适应度值, 初始化各粒子的 p_{best} 和初始种群的 g_{best} 。

3) 计算当代的各粒子相似度和种群聚类度, 由式(27)~(31)更新算法中各认知参数, 再由式(19)计算指数 ξ 。

4) 进行种群更新, 即根据式(26)的不同策略更新粒子速度, 并利用式(16)更新粒子的位置。

5) 计算更新后种群的适应度值, 并将 popm^t 与 popm^{t-1} 进行适应度比较: 若新解支配旧解, 则将 p_{best} 替换之; 若新解被旧解支配, 则沿用上一代的 p_{best} ; 若新旧解互不支配, 则随机选择是否替代。

6) 合并 popm 和 archive^{t-1}, 进行局部搜索策略, 将得到的 popm'' 再次与当代种群与上一代外部存档合并, 在去除重复解后, 根据式(32)对 E 进行拥挤距离判断, 排序并裁剪出新一代的外部存档 archive^t。

7) 采用自适应网格策略更新 g_{best} 。

8) 判断算法当前的迭代次数 t 是否达到最大迭代次数 T , 若满足则输出解集 archive, 算法终止; 否则, 跳转至步骤 3)。

4 仿真实验与数值分析

4.1 实验设计与参数选择

为了验证本文提出的 PCDMOPSO 算法在钢板入库垛位分配问题上的实用性, 以某钢结构智能加工制造基地的自动化钢板料库为研究对象, 对料库实际垛位分布、桥吊和 RGV 轨道车参数与配置、初始堆码情况以及入库钢板的规格进行模拟, 分别生成小批量 ($I=80$)、中批量 ($I=120$) 和大批量 ($I=150$) 的 3 组试验算例对 PCDMOPSO 算法进行验证。自动化钢板料库总体布局情况如图 1 所示, 表 1 为各实例数据基本参数。

表 1 钢板料库垛位与设备相关参数

Tab. 1 Palletizing location and equipment related parameters of steel plate warehouse

参数	设定值
料库垛位总行数 y	18
料库垛位总列数 x	10
最大垛位数量	180
在库钢板数量	1 800
相邻垛位形心间隔 $l_s \times w_s / (m \times m)$	19.0 × 4.5
垛位限高 H/m	3
桥吊大车运行速度 $V_x / (m \cdot \min^{-1})$	71.18
桥吊小车运行速度 $V_y / (m \cdot \min^{-1})$	43.12
桥吊起升机构运行速度 $V_{up} / (m \cdot \min^{-1})$	15.28
起升高度 h_{up}/m	9
AGV 轨道车运行速度 $V_{agv} / (m \cdot \min^{-1})$	48.96

在Windows10操作系统下基于MATLAB2017a编写算法,硬件环境为Intel Core i7-11700,2.50 GHz主频,内存为4 GB。为验证本文的PCDM-OPSO算法针对钢板入库垛位分配问题的优化性能,将其与基于分解的多目标差分进化算法MOEA/D-DE、多目标遗传算法NSGA-II和基本多目标粒子群算法MOPSO进行对比,观察求得解集中各目标函数值的情况,并设计不同的指标评估Pareto解集的收敛性、均匀性、多样性和互相占优性情况,通过以上几个维度的优化效果评价PCDMOPSO算法的适用性和优越性^[27]。各算法的最大迭代次数 T 均设置为3 000,种群规模 M 设置为90,外部存档容量 U_a 设置为50。PCDMOPSO算法中,个体认知系数 c_1 初始值为1, $c_{1max}=2,c_{1min}=1$;社会认知系数 c_2 初始值为2, $c_{2max}=2,c_{2min}=1$;levy飞行认知系数 c_3 初始值为0.001, $c_{3max}=2,c_{3min}=1$;惯性权重 $\omega_{max}=0.4,\omega_{min}=0.2$ 。

4.2 试验结果分析

4.2.1 优化结果

试验中各算法在不同入库规模下均使用相同的

输入数据,将每种算法单独运行20次后取平均值,输出各算法求解的Pareto最优解集下3个优化目标的最优值和平均值。不同的入库批量数据下各算法的优化结果如表2所示。由表2可知,相比其他算法,PCDMOPSO算法在3个优化目标上的寻优能力具备明显优势,在各批量任务中都获得了最优均值,在 f_1 和 f_3 目标上其最优值也均优于其他算法。虽然在 f_2 目标上PCDMOPSO算法未取得最优值,但仅次于NSGA-II算法,且相差较小,同时其均值也明显优于NSGA-II算法。NSGA-II算法求解表现较好的原因是其使用的迭代方式使其在处理离散型问题时具有天然优势。值得注意的是,在 f_3 目标上各算法均能搜索到同样的最优值,而PCD-MOPSO在该目标上的平均值明显更优,这也反映了PCDMOPSO算法在具备较强搜索能力的同时还具备更好的稳定性。图5为各算法在不同的批量数据下求得Pareto解集的收敛情况和在解空间中的分布状态。从图5可以看出,PCDMOPSO算法的解集其收敛性、分布性等均表现良好,能够搜索到其他算法难以触及的Pareto前沿。

表2 各算法的运行结果对比

Tab. 2 Comparison of the running results of each algorithm

入库规模	目标函数	PCDMOPSO		MOEA/D-DE		NSGA-II		MOPSO	
		最优值	平均值	最优值	平均值	最优值	平均值	最优值	平均值
小批量	f_1	204.774	210.025	228.443	241.639	206.918	212.340	220.921	227.204
	f_2	1.467×10^{-2}	2.907×10^{-1}	2.757×10^{-2}	4.590×10^{-1}	6.915×10^{-3}	4.293×10^{-1}	6.366×10^{-2}	4.024×10^{-1}
	f_3	4.714×10^{-1}	2.736	4.714×10^{-1}	4.389	4.714×10^{-1}	4.270	4.714×10^{-1}	3.099
中批量	f_1	307.391	313.217	346.810	364.422	310.977	317.616	334.602	342.601
	f_2	1.118×10^{-2}	3.159×10^{-1}	2.044×10^{-2}	4.774×10^{-1}	1.158×10^{-3}	4.288×10^{-1}	3.321×10^{-2}	4.273×10^{-1}
	f_3	0	2.859	0	7.602	0	4.448	0	3.364
大批量	f_1	391.015	395.153	439.874	458.029	391.193	399.839	422.944	433.823
	f_2	2.231×10^{-2}	3.072×10^{-1}	4.410×10^{-2}	4.837×10^{-1}	0	4.502×10^{-1}	1.706×10^{-2}	3.953×10^{-1}
	f_3	0	2.759	0	8.614	0	5.099	0	4.420

4.2.2 均匀性S指标测试

平均空间距离指标 S 用以评价算法求得解集在解空间分布的均匀性,其数学含义本质是求得解集的离散系数,用以观测解集中样本数据分布的离散程度,其值越小代表分布的均匀性越好。其计算方式为:

$$S = \frac{1}{\bar{d}} \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2} \quad (33)$$

式中, $\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$, d_i 为第 i 个Pareto解与解集中其他解的最小欧式距离, n 为Pareto解集的规模。

表3为各算法在不同入库规模下独立运行20次后的平均空间距离 S 指标的均值与方差。

由表3可知:MOEA/D-DE算法和MOPSO算法获得的 S 值的均值和方差均表现较差;在小批量的情况下,PCDMOPSO算法和NSGA-II算法结果较为接近,甚至NSGA-II算法获得的值会更好,但随着入库规模的增加,中批量和大批量情况下,PCDMOPSO算法获得的 S 值的均值和方差明显优于其他算法,证明了PCDMOPSO算法求得的Pareto最优解集分布最为均匀,且在不同的输入数据下算法的适应性和鲁棒性都较高。

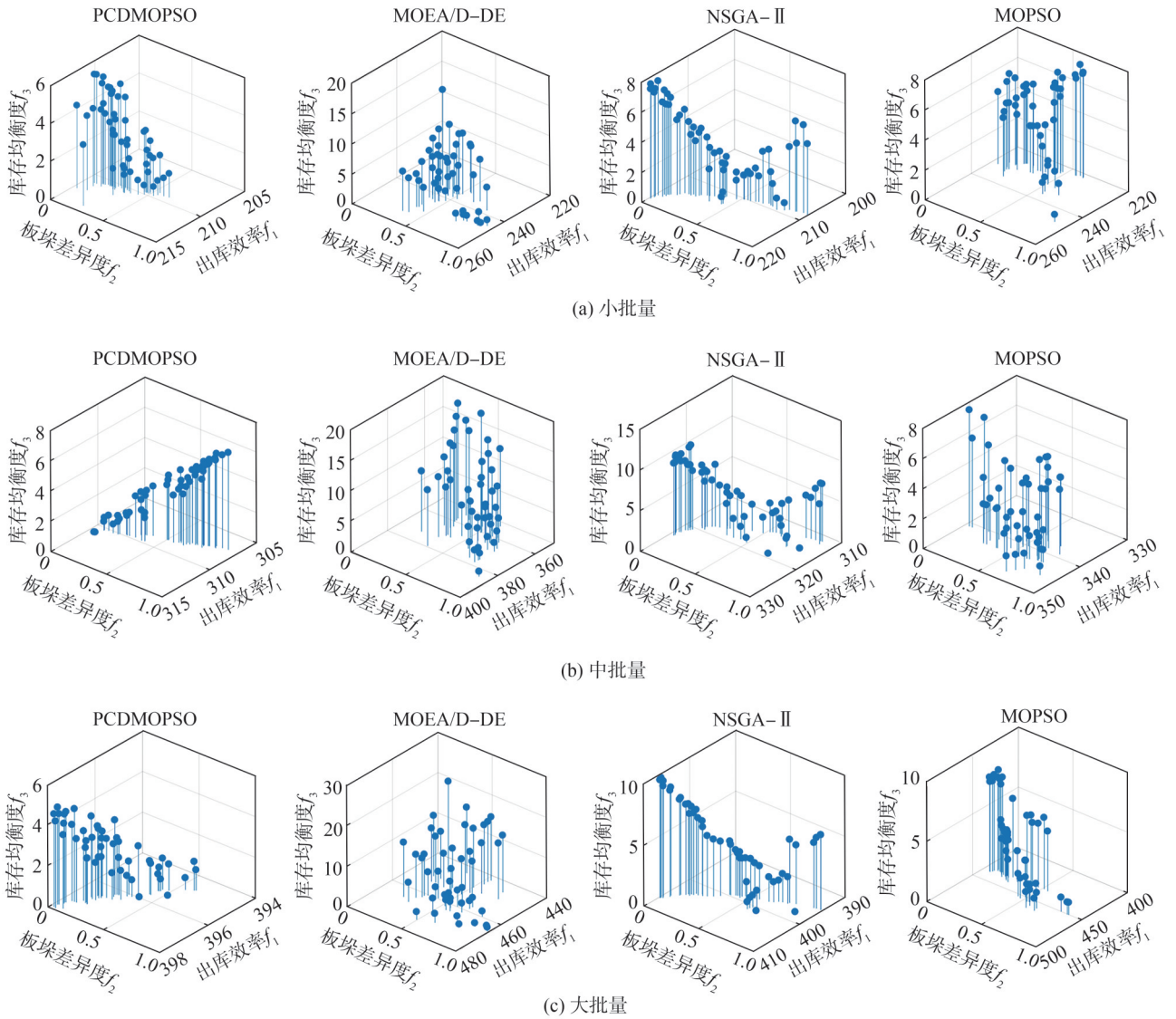


图 5 各算法的 Pareto 解集在目标空间中的分布比较

Fig. 5 Comparison of Pareto solution sets of different algorithms under different database size

表 3 各算法的 S 指标运行结果

Tab. 3 Running result of S index of each algorithm

算法	小批量		中批量		大批量	
	均值	方差	均值	方差	均值	方差
PCDMOPSO	6.766×10^{-1}	1.201×10^{-2}	6.856×10^{-1}	4.685×10^{-3}	6.727×10^{-1}	9.681×10^{-3}
MOEA/D-DE	7.546×10^{-1}	4.402×10^{-2}	7.351×10^{-1}	4.172×10^{-2}	7.411×10^{-1}	1.952×10^{-2}
NSGA-II	6.448×10^{-1}	1.115×10^{-2}	7.175×10^{-1}	1.330×10^{-2}	7.031×10^{-1}	1.771×10^{-2}
MOPSO	8.909×10^{-1}	6.300×10^{-2}	1.402	4.820×10^{-1}	9.061×10^{-1}	1.354×10^{-1}

4.2.3 收敛性 IGD 指标测试

反转世代距离 (inverted generational distance, IGD) 可用于评价算法求得的 Pareto 最优解集在求解问题上的收敛性能, 其反映的是算法求得的解集与真实 Pareto 前沿面的贴近程度, 作为一个综合指标, 在一定程度上也能反映解集的多样性。其计算方式为:

$$I_{IGD} = \frac{1}{|P^*|} \sqrt{\sum_{i=1}^{|P^*|} d(i, A)} \quad (34)$$

式中: $d(i, A)$ 为真实 Pareto 解集中的解 i 与所求解集 A 中解的最小距离; P^* 为真实 Pareto 前沿面的最优解集, 通过将各种算法反复多次求解, 再取其并集后经过非支配

解判断的方式获取。为保证实验的可靠性,取 $|P^*|=500$,设置较大的值可以更好地满足 IGD 指标评价解集收敛性的可靠性。IGD 值越小,代表解集的收敛性越好。

表 4 为各算法在不同入库规模下独立运行 20 次后的反转世代距离 IGD 指标的均值与方差。由于引入了种群聚类度和局部搜索等策略,PCDMOPSO 算法

具备自适应机制,其收敛能力明显强于其他算法,能够在解空间中搜索到其他算法难以触及的 Pareto 前沿。在方差上,PCDMOPSO 算法也明显优于其他算法,虽然在大批量的规模下略低于 NSGA-II,但差距也极小,展现出了 PCDMOPSO 算法优良的收敛性和求解的稳定性。

表 4 各算法的 IGD 指标运行结果

Tab. 4 Running result of IGD index of each algorithm

算法	小批量		中批量		大批量	
	均值	方差	均值	方差	均值	方差
PCDMOPSO	3.085	3.626×10^{-1}	3.288	2.485×10^{-1}	5.255	1.497
MOEA/D-DE	26.665	2.337	43.014	2.485	54.430	1.805
NSGA-II	3.138	1.253	4.241	4.749×10^{-1}	5.908	1.456
MOPSO	16.643	1.637	28.119	2.183	37.492	1.533

4.2.4 多样性指标 Δ^* 测试

多样性指标可以衡量所求解集的分布性和延展性,Deb 等^[28]曾提出了多样性指标 Δ :

$$\Delta = \frac{d_f + d_1 + \sum_{i=1}^{|A|-1} |d'_i - \bar{d}|}{d_f + d_1 + (|A| - 1) \times \bar{d}} \quad (35)$$

式中: d'_i 为解集 A 中两连续解间的欧氏距离; \bar{d} 为所有 d_i 的均值; d_f 和 d_1 代表解集 A 与真实 Pareto 前沿面边界点间的最小欧式距离,可以衡量解集的延展性。由于其仅适用于 2 维目标优化问题,无法应用于本文的 3

个优化目标,故参考 Zhou 等^[29]提出的多样性指标 Δ^* :

$$\Delta^* = \frac{\sum_{g=1}^G d_g(P^*, A) + \sum_{i=1}^{|A|} |d_i - \bar{d}|}{\sum_{g=1}^G d(P^*, A) + |A| \bar{d}} \quad (36)$$

式中, $d_g(P^*, A)$ 表示解集 A 与真实 Pareto 前沿面 P^* 边界点间在第 g 个目标上的最小欧式距离, G 为目标数。 Δ^* 值越小,代表所求 Pareto 解集的多样性越好。

表 5 为各算法在不同入库规模下独立运行 20 次后的多样性指标 Δ^* 的均值与方差。

表 5 各算法的 Δ^* 指标运行结果

Tab. 5 Running result of Δ^* index of each algorithm

算法	小批量		中批量		大批量	
	均值	方差	均值	方差	均值	方差
PCDMOPSO	1.169×10^{-1}	3.222×10^{-5}	4.199×10^{-2}	2.057×10^{-7}	7.007×10^{-2}	7.927×10^{-6}
MOEA/D-DE	5.191×10^{-1}	2.208×10^{-2}	2.432×10^{-1}	9.884×10^{-3}	4.061×10^{-1}	7.812×10^{-3}
NSGA-II	3.113×10^{-1}	1.095×10^{-3}	1.261×10^{-1}	2.112×10^{-4}	2.087×10^{-1}	1.525×10^{-3}
MOPSO	1.702×10^{-1}	8.489×10^{-5}	5.015×10^{-2}	4.838×10^{-6}	8.393×10^{-2}	8.088×10^{-5}

由表 5 可以看出:PCDMOPSO 算法在各入库规模下的均值和方差数据都明显优于其他算法,这是因为在外部存档的维护策略中采用拥挤距离策略,并使用自适应网格技术干涉全局最优解 g_{best} 的选择,使解集在具备优越的分布性和延展性的同时给算法的求解带来了良好的稳定性。

4.2.5 占优性 DPO 指标测试

Pareto 最优度指标 (degree of Pareto optimality, DPO) 可用以评价各算法所求的非支配解集相较于其他算法的被支配程度,其计算方式为:

$$D_{DPO} = \frac{|A_Q^*|}{n} \quad (37)$$

式中, A_Q^* 代表算法 Q 所求得的解集中不受其他算法的任何解所支配的解集, n 为找到的非支配解的数量。DPO 的值越大代表求解质量越高。

表 6 为各算法在不同入库规模下独立运行 20 次后的 Pareto 最优度指标 DPO 的均值与方差。

由表 6 可以看出,在 3 种不同入库规模下,PCDMOPSO 算法求得解集的不受支配率均明显高于其他算法,在小批量规模下获得的解集甚至完全不受其他算法的支配,而且在中批量和大批量算例中也处于明显的支配地位,进一步证明了 PCDMOPSO 算法具备更好的寻优能力,所求解集能够更加接近真实的 Pareto 前沿。

表 6 各算法的 DPO 指标运行结果

Tab. 6 Running results of DPO index of each algorithm

算法	小批量		中批量		大批量	
	均值	方差	均值	方差	均值	方差
PCDMOPSO	1	0	9.867×10^{-1}	8.890×10^{-4}	9.667×10^{-1}	1.289×10^{-3}
MOEA/D-DE	3.333×10^{-3}	5.556×10^{-5}	1.333×10^{-2}	8.889×10^{-4}	7.900×10^{-2}	4.055×10^{-3}
NSGA-II	2.200×10^{-1}	1.693×10^{-2}	8.667×10^{-2}	5.689×10^{-3}	1.633×10^{-1}	4.779×10^{-3}
MOPSO	6.403×10^{-2}	2.782×10^{-3}	6.453×10^{-2}	1.247×10^{-2}	2.997×10^{-2}	1.738×10^{-3}

综上所述,与其他算法对比,PCDMOPSO算法在展现出良好的优化效果的同时,其输出的 Pareto 最优解集相比其他算法还具备更优秀的多样性、分布性和收敛性,能够搜索出其他算法所触及不到的潜在解空间,且不易被它们的解所支配。

PCDMOPSO 算法通过采用种群聚类来实时监测算法的迭代情况,并合理地对其迭代参数和更新方式进行调整,在迭代中不断修正引导粒子寻优的收敛性和多样性求解趋势,在此基础上设计局部搜索策略,进一步帮助粒子跳出局部最优,最后制定合理的外部存档维护策略,保证 PCDMOPSO 算法在求解钢板入库垛位分配问题下表现出优越的性能。

4.3 算法改进有效性分析

算法的改进策略可大致分为两点:一是以种群聚类度为基础,以平衡收敛性和多样性为需求,设计自适应的粒子速度更新模式,其中的 levy 飞行阶段及其采用的参数 c_3 可保证算法在迭代后期拥有更多跳出局部最优的可能;二是针对外部存档的归档阶段额外进行一次局部搜索策略,该环节在不影响种群进化的前提下额外进行了一次全局搜索,有助于提升求得解集的多样性。

定义 PCDMOPSO-no-levy 表示算法在粒子速度更新阶段不受参数 c_3 及其对应的 levy 飞行阶段的影响,

PCDMOPSO-no-ls 表示算法未使用局部搜索策略。则在最贴合钢板料库现有订单量的小批量 ($I=80$) 的入库规模下,将引入上述改进策略后的 PCDMOPSO 与改进前的算法进行运算对比。表 7 为反复求解 20 次后获取的各评价指标的均值对比。由表 7 可以看出:未添加 levy 飞行速度更新策略的算法,其求解的收敛性显著降低;未添加局部搜索策略的算法在多样性上显著降低;改进后算法求解的各评价指标均有了不同程度的优化。采用 levy 飞行的全局搜索策略可以避免种群收敛于解空间的某一局部,而基于遗传的局部搜索策略在改善求解多样性上具有重要意义。

表 7 小批量下 PCDMOPSO 得到的 Pareto 最优解集

Tab. 7 Pareto optimal solution set obtained by the PCDMOPSO in medium batch

算法	S	I_{IGD}	Δ^*	D_{DPO}
PCDMOPSO	5.867×10^{-1}	3.055	1.407×10^{-1}	8.667×10^{-1}
PCDMOPSO-no-levy	6.481×10^{-1}	12.201	2.229×10^{-1}	1.633×10^{-1}
PCDMOPSO-no-ls	7.500×10^{-1}	7.830	4.337×10^{-1}	2.400×10^{-1}

图 6 为 PCDMOPSO 在引入两项改进策略前后求得解集的分布对比。

由图 6 可以比较直观地看出,添加了 levy 飞行速度更新策略和局部搜索策略后,求解的分布性、多样性和收敛性均得到了明显改善,证明了改进策略的有效性。

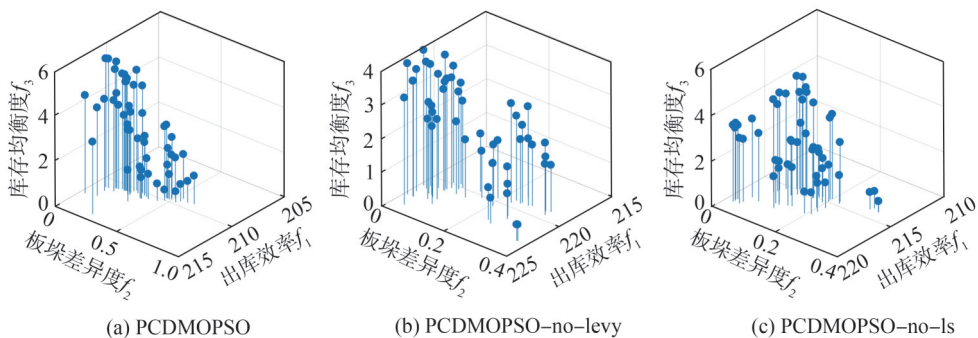


图 6 PCDMOPSO 在引入两项改进策略前后求得解集的分布比较

Fig. 6 PCDMOPSO algorithm obtains the distribution comparison of solution set before and after introducing two improved strategies

4.4 应用实例

在小批量 ($I=80$) 的入库规模下,PCDMOPSO 算法输出的 Pareto 最优解集的目标函数值如表 8 所示。由表 8 可以看出,各个解之间互不占优,管理人员可以根

据其偏好任选其一作为入库垛位分配决策。本文采用文献[30]提出的一种基于熵权法的优劣距离法选择 Pareto 最优解集中的综合最优解,即第 15 号的目标函数值,为(209.685,0.497,0.471)。

表8 小批量下PCDMOPSO得到的Pareto最优解集
Tab. 8 Pareto optimal solution set obtained by the PCDMOPSO in medium batch

序号	出库效率	板垛差异度	库存均衡度
1	211.270	0.139	5.528
2	211.839	0.050	6.289
3	211.553	0.104	5.437
4	211.242	0.150	5.121
5	211.475	0.117	4.955
6	211.147	0.197	3.986
7	209.827	0.401	1.795
...
15	209.685	0.497	0.471
...
49	209.768	0.400	2.134
50	211.738	0.089	4.422

为更直观地观察堆码情况,输出优化前的堆码方案(260.008,0.523,1.247)和优化后选择的第15号方案下的入库堆码情况示意图,如图7所示。

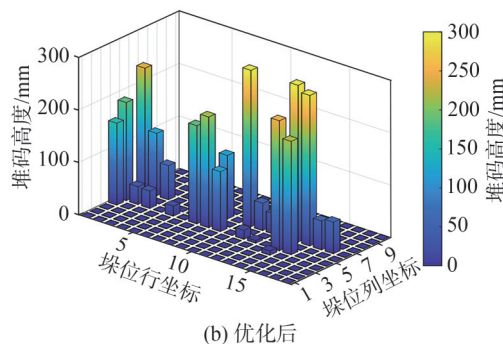
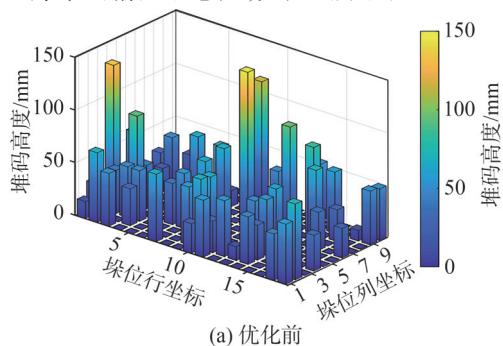


图7 优化前后的垛位分配方案堆码示意图

Fig. 7 Stacking diagram of distribution scheme before and after optimization

图7中,对钢板料库垛位的分布和堆码情况进行了仿真,其中的每个柱状图代表一个垛位,反映其堆叠高度,列坐标5和6之间为轨道车工位,不设置堆垛。

由图7(a)可知,优化前依赖传统的单一堆码原则所制定的入库策略,更多的是采用随机存储方式,入库垛位杂乱无序,不便于后续的出库作业和在库管理;图7(b)优化后的堆码方案明显分配更加合理,既实现了分类集中存储的要求,同时各库区分布也相对

较为均匀,靠近出库方向堆码也大大提升了后续的出库效率。

5 结论

针对企业对自动化钢板料库入库决策优化的实际需求,根据其基本布局和作业流程进行分析,研究钢板入库的垛位分配问题。提出了出库效率、板垛差异度和库存均衡度3个指标对钢板入库方案进行评价,并据此设计了钢板入库的多目标优化模型。基于该模型,设计了PCDMOPSO算法对之进行求解;算法引入种群聚类策略实时调控不同阶段种群对收敛性和多样性的不同需求,并设计局部搜索策略进一步使算法有更多机会跳出局部最优。实验结果表明:PCDMOPSO算法在面对不同规模的入库任务时均能求得满意解;求得的Pareto最优解集的质量明显优于其他算法,其分布的均匀性、多样性、收敛性和占优性指标也更好,可以有效解决钢板入库垛位分配问题。在进一步的研究中,将完善钢板重量、起重机能耗及料库总体空间利用率,以使问题更贴合实际;并且,由于本文提出的算法输出的是一组Pareto最优解集,而非确切的一组解,所以后续需要设置额外的手段进一步挑选出适宜实际需求的垛位分配方案。

参考文献:

- [1] 王黎明,孙博,王国连,等.论中国钢铁企业板坯库和钢卷库智能化控制的发展[C]/2019年第二届钢铁工业智能制造发展论坛会议论文集.石家庄:首钢京唐钢铁联合有限责任公司钢轧作业部,2019:5-8.
- [2] Zhang Xuan, Mo Tiantian, Zhang Yougong. Optimization of storage location assignment for non-traditional layout warehouses based on the firework algorithm[J]. Sustainability, 2023, 15(13): 10242.
- [3] Nastasi G, Colla V, Cateni S, et al. Implementation and comparison of algorithms for multi-objective optimization based on genetic algorithms applied to the management of an automated warehouse[J]. Journal of Intelligent Manufacturing, 2018, 29(7): 1545-1557.
- [4] Park C, Seo J. Comparing heuristic algorithms of the planar storage location assignment problem[J]. Transportation Research Part E: Logistics and Transportation Review, 2010, 46(1): 171-185.
- [5] Guerriero F, Pisacane O, Rende F. Comparing heuristics for the product allocation problem in multi-level warehouses under compatibility constraints[J]. Applied Mathematical Modelling, 2015, 39(23/24): 7375-7389.
- [6] Yang Wei, Liu Jiang, Yue Ting, et al. Integrated optimization

- of location assignment and job scheduling in multi-carrier automated storage and retrieval system[J]. *Computer Integrated Manufacturing Systems*, 2019, 25(1): 247–255. [杨玮, 刘江, 岳婷, 等. 多载具自动化立体仓库货位分配与作业调度集成优化[J]. *计算机集成制造系统*, 2019, 25(1): 247–255.]
- [7] Wang Juan, Yi Duhan, Cheng Yuli, et al. Location assignment problem for humidity sensitive cargoes in stereoscopic warehouses based on adaptive MPGA[J]. *Computer Integrated Manufacturing Systems*, 2025, 31(5): 1905–1914. [王娟, 易杜涵, 程玉丽, 等. 基于自适应 MPGA 的湿度敏感类物品立体仓库货位分配方法[J]. *计算机集成制造系统*, 2025, 31(5): 1905–1914.]
- [8] He Li, Tao Yifei, Luo Junbin, et al. Job integrated optimization of automated storage/retrieval systems based on two-stage wolf pack algorithm[J]. *China Mechanical Engineering*, 2022, 33(21): 2538–2546. [何李, 陶翼飞, 罗俊斌, 等. 基于两阶段狼群算法的自动化立体仓库作业集成优化[J]. *中国机械工程*, 2022, 33(21): 2538–2546.]
- [9] Chen Lei, Cheng Jiangfeng, Zhu Yonghuai, et al. Multi-task textile stereoscopic warehouse location allocation and task sequence optimization method[J]. *Computer Integrated Manufacturing Systems*, 2023, 29(4): 1371–1385. [陈雷, 程江峰, 朱永怀, 等. 多任务纺织立库货位分配与任务序列优化方法[J]. *计算机集成制造系统*, 2023, 29(4): 1371–1385.]
- [10] Saleh M A M, Hicham A, Ben Maâti M L, et al. Development of a sustainable strategy model for predicting optimal container stacking locations in container yards using artificial intelligence and cubic data[J]. *Kuwait Journal of Science*, 2024, 51(2): 100174.
- [11] Zhang Canrong, Wang Qi, Yuan Guoping. Novel models and algorithms for location assignment for outbound containers in container terminals[J]. *European Journal of Operational Research*, 2023, 308(2): 722–737.
- [12] Kim T, Ryu K R. Deriving situation-adaptive policy for container stacking in an automated container terminal[J]. *Applied Sciences*, 2022, 12(8): 3892.
- [13] Zhang Zhiying, Wang Weize, Hou Jun. Optimization of multistage operation scheduling for a steel plate stockyard in shipbuilding[J]. *Journal of Harbin Engineering University*, 2015, 36(5): 638–643. [张志英, 王维泽, 侯俊. 船厂钢板堆场多时段作业计划优化[J]. *哈尔滨工程大学学报*, 2015, 36(5): 638–643.]
- [14] Zhao Tianyi, Li Na, Li Minghao, et al. Warehousing optimization of ship outfitting after considering retrieval uncertainty[J]. *Journal of Harbin Engineering University*, 2024, 45(1): 1–8. [赵天毅, 李娜, 李明昊, 等. 考虑出库不确定性的船舶舾装件立库入库优化[J]. *哈尔滨工程大学学报*, 2024, 45(1): 1–8.]
- [15] Hou Jun, Zhang Zhiying. Location allocation for inbound and outbound scheduling of mixed storage steel plate in shipyard[J]. *Journal of Harbin Engineering University*, 2017, 38(11): 1786–1793. [侯俊, 张志英. 船厂钢板堆场混合存储分配及出入库调度研究[J]. *哈尔滨工程大学学报*, 2017, 38(11): 1786–1793.]
- [16] Xu Liyun, Shu Zhongyu, Yang Liansheng. Steel plate scheduling optimization in shipbuilding based on storage area partition[J]. *Procedia CIRP*, 2020, 93: 1001–1006.
- [17] Zhang Tianxing. Research on location optimization and picking efficiency of intelligent warehouse system[D]. Changchun: Jilin University, 2021. [张天星. 智能仓储系统储位优化与拣选效率研究[D]. 长春: 吉林大学, 2021.]
- [18] Ye Kang. Research on location allocation and scheduling optimization of two-way automatic stereo warehouse[D]. Xi'an: Xi'an University of Architecture and Technology, 2021. [叶康. 双向式自动化立体仓库货位分配及调度优化研究[D]. 西安: 西安建筑科技大学, 2021.]
- [19] Zhang Qiqi, Zhang Tao, Liu Peng. Slab location decision optimization based on multi-objective population cooperative algorithm[J]. *Computer Integrated Manufacturing Systems*, 2015, 21(7): 1820–1828. [张琦琪, 张涛, 刘鹏. 基于多目标种群协同算法的板坯入库优化[J]. *计算机集成制造系统*, 2015, 21(7): 1820–1828.]
- [20] Zhou Aimin, Jin Yaochu, Zhang Qingfu, et al. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion [C]// *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*. Vancouver: IEEE, 2006: 892–899.
- [21] Guan Tianhua, Han Fei, Han H. A modified multi-objective particle swarm optimization based on levy flight and double-archive mechanism[J]. *IEEE Access*, 2019, 7: 183444–183467.
- [22] Zhou Binghai, Liao Xiumei. Particle filter and Levy flight-based decomposed multi-objective evolution hybridized particle swarm for flexible job shop greening scheduling with crane transportation[J]. *Applied Soft Computing*, 2020, 91: 106217.
- [23] Li Haojun, Zhang Pengwei, Guo Haidong. Adaptive multi-objective particle swarm optimization algorithm based on population Manhattan distance[J]. *Computer Integrated Manufacturing Systems*, 2020, 26(4): 1019–1032. [李浩君, 张鹏威, 郭海东. 基于种群曼哈顿距离的自适应多目标粒子群优化算法[J]. *计算机集成制造系统*, 2020, 26(4): 1019–1032.]
- [24] Li Wei, Meng Xiang, Huang Ying, et al. Knowledge-guided

- multiobjective particle swarm optimization with fusion learning strategies[J]. *Complex & Intelligent Systems*, 2021,7(3):1223–1239.
- [25] Liu Jianhua, Fan Xiaoping, Qu Zhihua. A new particle swarm optimization algorithm based on similarity[J]. *Control and Decision*, 2007,22(10):1155–1159. [刘建华,樊晓平,瞿志华.一种基于相似度的新型粒子群算法[J].控制与决策,2007,22(10):1155–1159.]
- [26] Liang Tian, Cao Dexin. Improved and simplified particle swarm optimization algorithm based on levy flight[J]. *Computer Engineering and Applications*, 2021,57(20):188–196. [梁田,曹德欣.基于莱维飞行的改进简化粒子群算法[J].计算机工程与应用,2021,57(20):188–196.]
- [27] Wang Liping, Ren Yu, Qiu Qicang, et al. Survey on performance indicators for multi-objective evolutionary algorithms [J]. *Chinese Journal of Computers*, 2021, 44(8): 1590–1619. [王丽萍,任宇,邱启仓,等.多目标进化算法性能评价指标研究综述[J].计算机学报,2021,44(8):1590–1619.]
- [28] Deb K, Agrawal S, Pratap A, et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II [C]// *Parallel Problem Solving from Nature PPSNVI*. Berlin, Heidelberg: Springer, 2000: 849–858.
- [29] Wang Xin, Chen Ni, Zhao Yawen. Value evaluation of innovative technical talents in enterprises based on entropy weight TOPSIS[J]. *Journal of Northeastern University (Natural Science)*, 2020, 41(12): 1788–1793. [王馨,陈妮,赵雅雯.基于熵权TOPSIS法的企业创新型技术人才价值评价[J].东北大学学报(自然科学版),2020,41(12):1788–1793.]

Research on Stacking Distribution of Steel Plates Input Based on Improved Multi-objective Particle Swarm Optimization

ZHONG Chuanjie^{1,2}, CHENG Wenming^{1,2*}, DU Run^{1,2}, GAO Xuchao³, ZHANG Lin³

(1. College of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China;

2. Key Laboratory of Rail Transit Operation Technology and Equipment of Sichuan Province, Chengdu 610031, China;

3. Sichuan Steel Structure Wisdom Manufacturing Company Limited, Meishan 620033, China)

Abstract:

Objective Different from the basic layout of traditional automated three-dimensional warehouses, steel plate goods are often stacked in the automated storage and retrieval system (AS/RS) rather than stored on high-rise shelves. This difference renders the classic AS/RS storage allocation model and job scheduling strategy inapplicable in steel plate warehouses. This study analyzes the overall warehouse layout and operational flow, proposes a multi-objective optimization model, and designs an efficient multi-objective algorithm to address the problem. Methods Based on the actual demand of the principle of priority in warehouse delivery, the principle of stacking safety, the principle of minimum stacking amount, and the principle of inventory balance, three indices were proposed: warehouse delivery efficiency, plate and stack difference, and inventory balance. First, warehouse delivery efficiency was one of the important indicators utilized to measure the palletizing plan. Since the warehouse served production purposes and was based on the principle of efficiency priority, the steel plate delivery time was minimized as much as possible. The influencing factors were primarily determined by the equipment operation mode and the palletizing position. Second, in order to ensure the stability of steel plate stacking and reduce the frequency of stacking, the steel plate and stacking position were assigned characteristic indices based on length, width, and item number to classify the steel plate and stacking type and to establish the difference degree index of plate stacking. The inventory balance index was established based on the standard deviation of the number of steel plates allocated to each reservoir to fully mobilize and balance material storage resources during operation. These three indices were utilized to evaluate the degree of stack allocation and served as the objective function to establish a multi-objective decision optimization model for the stack allocation problem of steel plates. This problem was classified as a Type A packing problem with constraints. In addition, when multiple conflicting optimization objectives were present, it was difficult to construct a single mathematical model and apply traditional analytic algorithms to solve it. A multi-objective particle swarm optimization algorithm (PCDMOPSO) based on species clustering degree was designed to solve this model using the concept of Pareto dominance and the classical particle swarm optimization algorithm. The algorithm adopted convergence and diversity of the solution as basic requirements and used the species clustering degree mechanism to monitor and adjust the algorithm's cognitive parameters and the evolution state of particles in real time. Convergence and diversity of solutions were adaptively adjusted, and a local search strategy was introduced to improve the diversity of the Pareto solution set distribution in the external archive after population updates. Then, the crowding distance strategy was utilized to maintain the external archive. The improved algorithm addressed problems such as high dependence on parameter setting, unstable solving efficiency, and a tendency to fall into local optimality.

Results and Discussions The automatic steel plate warehouse of a steel structure intelligent processing and manufacturing base was taken as the research object to verify the practical performance of PCDMOPSO in solving the steel plate loading and palletizing problem. Parameters and data under actual working conditions were used for simulation. The simulation results showed that compared to the classical multi-objective algo-

rithms NSGA-II, MOEA/D-DE, and MOPSO, PCDMOPSO demonstrated clear advantages in optimization ability for each target under different storage scales. Although NSGA-II achieved a better minimum value than PCDMOPSO in the index of the difference degree of plate and stack in 20 tests, PCDMOPSO showed stronger overall optimization ability. However, the difference was slight. Since the output of the multi-objective algorithm was a Pareto solution set, four indices, uniformity, convergence, diversity, and dominance, were selected to evaluate the distribution in the solution space and to compare the solution results of each algorithm. Among the convergence indices, NSGA-II yielded slightly better results than PCDMOPSO with small batch data scales. However, as batch size increased, the mean and variance of the S value obtained by PCDMOPSO significantly outperformed the other algorithms. PCDMOPSO showed clear advantages in the remaining three indicators, demonstrating high solving efficiency under varying input data as well as strong adaptability and robustness. The distribution of Pareto solution sets from different algorithms further illustrated these conclusions. Then, to verify the feasibility of the improved strategy in the proposed multi-objective particle swarm optimization algorithm, a comparative test of the algorithm improvement strategy was conducted. The Levy flight speed update mode, which served as the core of the improvement, and the local search strategy of the external archive were removed separately. The simulation was conducted using a small-batch data scale that best fit the actual production requirements of the steel sheet stock warehouse. The algorithm without the Levy flight speed update strategy exhibited significantly reduced convergence, while the algorithm without the local search strategy showed a marked reduction in diversity. The evaluation indices solved by the improved algorithm were optimized to varying degrees. Finally, the stacking situations before and after optimization were compared. Compared to the traditional stacking method used before optimization, the optimized stacking distribution scheme improved by 19.35%, 4.97%, and 62.23% under the three objectives, respectively, indicating a more significant optimization effect.

Conclusions Based on the actual demand of enterprises for optimizing automatic steel plate warehouse loading decisions, the PCDMOPSO algorithm has demonstrated good performance in the simulation test of solving the stack allocation model. The results indicate that the levy flight update strategy and the local search strategy are significant for maintaining population diversity and assisting in escaping local optima, respectively. The proposed improvement measures have an apparent positive effect on the quality of the solution. In addition, satisfactory solutions can be obtained for warehousing tasks of different scales, and the quality of the output Pareto solution set is obviously superior to that of the traditional algorithm. This effectively meets the practical requirements of various evaluation indicators in the steel plate warehousing problem and provides a valuable reference for research in the same field. It also provides strong decision support for pallet distribution and warehouse management of steel plate goods in the AS/RS.

Key words: steel plates storage warehouse; stacking distribution; particle swarm optimization; multi-objective optimization; Pareto-optimal solution

(编辑 张琼)

引用格式: Zhong Chuanjie, Cheng Wenming, Du Run, et al. Research on stacking distribution of steel plates input based on improved multi-objective particle swarm optimization[J]. *Advanced Engineering Sciences*, 2025, 57(4): 150–164. [钟传捷, 程文明, 杜润, 等. 基于改进多目标粒子群算法的钢板入库垛位分配研究[J]. *工程科学与技术*, 2025, 57(4): 150–164.]