

控制与决策

Control and Decision

基于向量映射代理模型的分布式柔性作业车间调度算法

靳思远, 彭程, 王薇, 任磊, 吕金虎, 赖李媛君

引用本文:

靳思远, 彭程, 王薇, 等. 基于向量映射代理模型的分布式柔性作业车间调度算法[J]. *控制与决策*, 2025, 40(5): 1561-1570.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2024.0965>

您可能感兴趣的其他文章

Articles you may be interested in

[基于弱关联的自适应高维多目标进化算法](#)

A weak association-based adaptive evolutionary algorithm for manyobjective optimization
控制与决策. 2021, 36(8): 1804-1814 <https://doi.org/10.13195/j.kzyjc.2019.1723>

[基于最佳相位匹配的减摇鳍低航速控制策略](#)

[Best phase matching based control strategy for ship stabilizer at low speed](#)
控制与决策. 2021, 36(4): 940-946 <https://doi.org/10.13195/j.kzyjc.2019.1028>

[基于预防维护的单机调度问题](#)

[Single-machine scheduling problem with preventative maintenance activities](#)
控制与决策. 2021, 36(2): 395-402 <https://doi.org/10.13195/j.kzyjc.2019.0626>

[求解约束优化问题的改进果蝇优化算法及其工程应用](#)

Improved fruit fly optimization algorithm for solving constrained optimization problems and engineering applications
控制与决策. 2021, 36(2): 314-324 <https://doi.org/10.13195/j.kzyjc.2019.0557>

[基于迁移学习灰支持向量回归机的交互式进化计算](#)

Interactive evolutionary computation based on transfer learning grey support vector regression
控制与决策. 2021, 36(10): 2399-2408 <https://doi.org/10.13195/j.kzyjc.2020.0420>

基于向量映射代理模型的分布式柔性作业车间调度算法

靳思远, 彭程, 王薇, 任磊, 吕金虎, 赖李媛君[†]

(北京航空航天大学 自动化科学与电气工程学院, 北京 100191)

摘要: 随着定制化制造需求的增加, 分布式柔性作业调度问题涉及的调度任务复杂多变, 多产线制造设备动态重构, 人机料法环协同约束倍增. 面对多场景任务调度需求, 单一进化算法难以适应不同场景, 且由于算法对这类复杂约束问题评估时间成本较高, 逐场景调用算法生成调度方案十分耗时. 为解决这一问题, 建立分布式柔性作业车间调度问题数学模型, 并提出基于向量映射代理模型 (VMSM) 的调度算法. VMSM 通过高维特征向量映射进行特征提取, 以增强对历史相似问题的识别能力, 并在进化算法中辅助生成初始解和评估解集, 提高搜索效率. 实验结果表明, 所提出的方法能够将相似调度方案的分类准确率提升 25% ~ 35%, 有效减少进化算法初始解筛选和解集评估次数. 在不同场景复杂度下, VMSM 均能在保障调度方案质量的前提下, 显著提高分布式柔性作业车间调度方案生成速度, 方案生成时间最多缩短 51.26%.

关键词: 调度优化; 代理模型; 进化算法; 柔性作业车间调度问题; 分布式调度; 遗传算法

中图分类号: TH165; TP181 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.0965

引用格式: 靳思远, 彭程, 王薇, 等. 基于向量映射代理模型的分布式柔性作业车间调度算法 [J]. 控制与决策, 2025, 40(5): 1561-1570.

Distributed flexible job shop scheduling algorithm based on a vector mapping surrogate model

JIN Si-yuan, PENG Cheng, WANG Wei, REN Lei, LV Jin-hu, LAI Li-yuan-jun[†]

(School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China)

Abstract: As the demand for customized manufacturing grows, the distributed flexible job shop scheduling problem (DFJSP) involves complex and variable scheduling tasks, dynamic reconfiguration of multi-line manufacturing equipment, and increased collaborative constraints across human, machine, material, method, and environment. Traditional evolutionary algorithms struggle to adapt to multi-scenario scheduling and are inefficient due to high computational costs. To address this, this paper establishes a mathematical model for the DFJSP and subsequently proposes a scheduling algorithm based on the vector mapping surrogate model (VMSM). The VMSM enhances the recognition of historically similar problems through high-dimensional feature vector mapping, and aids in generating initial solutions and evaluating solution sets within evolutionary algorithms, thereby improving search efficiency. Experimental results demonstrate that this method increases classification accuracy for similar scheduling solutions by 25%-35%, significantly reducing the number of initial solution selections and solution set evaluations. Under various scenario complexities, the VMSM effectively enhances DFJSP solution generation speed by up to 51.26%, while maintaining the quality of the scheduling solutions.

Keywords: scheduling optimization; surrogate model; evolutionary algorithm; flexible job shop scheduling problem; distributed scheduling; genetic algorithm

0 引言

在经济全球化和工业互联网的背景下, 制造业领域的企业合作生产产品的现象日益普遍, 分布式制造已经成为一种常见的生产模式. 分布式制造是一种现代制造模式, 它将制造资源分布在多个地理

位置上, 通过网络或其他通信方式协同工作^[1]. 与传统制造模式相比, 分布式制造在减少前期投入、降低生产门槛、提高产品质量、快速响应用户需求等多个层面具有显著优势.

车间调度直接影响到车间生产的效率和成本,

收稿日期: 2024-08-14; 录用日期: 2024-11-14.

基金项目: 国家自然科学基金面上项目 (62173017); 卓越工程师产教联合培养研究基地卓越工程师培养改革战略研究课题项目.

[†]通信作者. E-mail: lailiyuanjun@buaa.edu.cn.

一直是制造系统领域的研究重点. 分布式车间制造模式的出现, 对车间调度提出了更高的要求. 在数学上, 分布式车间调度问题属于 NP-hard 问题, 求解难度很大, 其研究具有重要的学术意义; 在工程上, 实现分布式车间调度问题的最优或次优优化可以大幅减少工期或降低生产成本, 其研究具有重要的应用价值^[2].

分布式柔性作业车间调度问题 (DFJSP) 是分布式车间调度的一种类型, 常见于制造业机械加工领域^[3]. 分布式车间调度比传统的单车间调度难度更大, 除单一车间调度需要考虑的工序调度外, 还需要额外考虑不同车间地理位置不同带来的运输和时间成本. 此外, 从单一车间到多个车间的变化导致调度所涉及的机器数量和工件数量剧增, 解空间的规模远远大于单一车间调度问题. 随着场景和调度流程的复杂化, 问题的求解变得愈加困难. 如果仍然采用传统搜索方法, 如进化算法, 在如此大规模的问题背景下, 全局搜索需要的时间会成倍增加, 而且找到最优解的难度加大.

针对进化算法在求解复杂约束问题时评估代价过高的问题, 目前主流解决方法包括使用基于代理模型的进化算法评估过程^[4]、使用迁移学习方法利用历史经验、使用深度强化学习方法利用神经网络的推理能力快速求解^[5]. 其中, 以基于代理模型的进化算法应用最为典型. 代理模型的使用为进化算法提供了一种有效手段, 能够显著减少适应度函数评估的数量, 降低时间成本. 目前, 相关研究者针对代理模型辅助进化算法搜索的研究主要集中在两个方面, 一方面是如何构建代理模型, 即利用有限的样本量, 在更短的时间内构建出更加准确的代理模型; 另一方面是如何选择代理模型, 即根据代理模型与待解决问题的特征, 在众多代理模型的构建技术中选择出最合适的模型. 针对代理模型构建的研究中, 相关学者针对各种代理模型的构建技术开展了研究, 如神经网络^[6]、随机森林模型^[7]、高斯过程^[8]、径向基函数^[9]、克里金模型^[10-11]、贝叶斯方法^[12]等.

针对分布式柔性作业调度问题的研究主要有两个角度. 一是在原有 DFJSP 的基础上, 继续提出新的问题, 如 Zhu 等^[13] 针对 DFJSP 只分析静态制造环境忽略操作检查的问题, 提出了一种考虑操作检查的动态分布式柔性作业车间调度问题; Du 等^[14] 将生产阶段、装配阶段、交付阶段统筹考虑, 提出了一种分布式装配柔性作业调度问题; Zhang 等^[15] 考虑到工人约束, 提出了一种具备机器约束和工人约束双重资源约束的 DFJSP. 二是使用新的方法解决 DFJSP,

如 Zhang 等^[16] 提出了一种基于 Q-learning 的超启发进化算法求解 DFJSP 问题; Li 等^[17] 基于模因算法 (memetic algorithm) 提出了一种 SPAMA 算法, 并在高效分布式灵活作业车间调度问题上证明了 SPAMA 算法的优越性; Liu 等^[18] 等设计了一种基于 3 层编码机制的改进遗传算法解决 DFJSP 问题中批量协作引起的解空间扩展问题, 并针对不同规模的调度问题进行了测试, 验证了所提出算法的有效性; Du 等^[19] 基于 DQN 求解 DFJSP 并通过数值实验表明了深度强化学习算法具有一定的优势. 基于强化学习的方法在问题求解速度上具有显著优势, 但存在训练时间长、稳定性依赖训练效果的缺点. 基于进化算法的方法在解的可靠性方面有一定优势, 但存在面对复杂场景问题求解速度慢的缺陷. 如何在保证求解质量的前提下提高进化算法的求解效率, 是应用进化算法求解 DFJSP 的关键.

鉴于此, 本文旨在面向分布式柔性作业车间调度问题, 构建覆盖生产订单多个作业阶段的数学模型, 从提高问题的灵活程度和求解效率的角度出发, 设计基于向量映射代理模型的改进算法, 解决进化算法在求解 DFJSP 时因评估成本过高导致的搜索效率较低的问题. 本文剩余内容安排如下: 第 1 节给出问题模型; 第 2 节介绍代理模型的构建和应用过程; 第 3 节对设计的算法进行测试实验, 验证所提出改进算法的有效性; 第 4 节总结本文, 并提出进一步研究方向.

1 问题建模

1.1 问题描述

DFJSP 以提高车间机器使用效率和订单完成度速度为目的, 分配车间和机器, 决定生产订单执行顺序, 同时满足空间 (机器、车间分布在不同地理位置)、时间 (作业前序后继关系及执行时间) 及资源 (机器使用约束) 的限制, 是一个资源约束调度优化问题. 整个生产过程涉及到多个订单、多个车间、多种机器在多个阶段的有效协调.

具体而言, DFJSP 可描述为, 将 n_{product} 个产品订单分配到分布在不同位置的 n_{jobshop} 个车间中处理. 每个车间内有 n_{machine} 台机器, 机器的类型不完全相同, 不同类型的机器执行的工艺工序不同. 每个产品需要原材料工件经过一道或多道加工工序制造得到. 调度策略需要为每个产品订单选择合适的车间和机器, 并确定加工顺序, 以优化整体生产效率并提高订单达成率.

在建立模型前, 首先基于问题实际研究背景, 提

出以下模型假设:

- 1) 初始阶段, 工件和机器均处于就绪状态;
- 2) 同一台机器同一时间仅能处理一个工件;
- 3) 进入机器进料仓的工件不区分加工优先级, 按照进料顺序加工;
- 4) 机器加工作业一旦开始, 在加工过程中不存在故障现象, 各个作业在加工过程中不中断;
- 5) 不单独考虑车间内的运输时间, 运输时间包含在加工时间内;
- 6) 每个产品订单仅能被分配到一个车间, 分配后该产品的所有加工工序均需在该车间内完成;
- 7) 假设产品是由一件原材料经过多道工序加工生产;
- 8) 不同类型的产品需要经过的工序不同, 所有产品加工工序已知;
- 9) 不考虑机器故障、车间配送失败等可能影响加工过程的特殊情况。

1.2 符号定义

为了便于建立 DFJSP 数学模型, 定义相关参数变量如表 1 所示。

1.3 模型建立

1.3.1 问题约束

$$n_{\text{product}} = n_{\text{workpiece}}; \tag{1}$$

$$DS_w \leq T_w \leq DE_w, \forall w \in \{1, n_{\text{workpiece}}\}; \tag{2}$$

$$ET_{wp}^j = ST_{wq}^j + T_{wq} + WT_{wq}^j; \tag{3}$$

$$ET_{wq}^j \leq ST_{w(q+1)}^j, \forall w \in \{1, n_{\text{workpiece}}\}, \forall j \in \{1, n_{\text{jobshop}}\}, \forall q \in \{1, \|O_w\| - 1\}; \tag{4}$$

$$\sum_j A_w^j = 1, j \in \{1, n_{\text{jobshop}}\}, w \in WJ_j; \tag{5}$$

$$\sum_{wq} B_{wq}^{jm} = 1, m \in \{1, n_{\text{machine}}\}; \tag{6}$$

$$C_{mt}^j \leq C_m^j, 0 \leq t \leq T_{\max};$$

$$A_w^j = \begin{cases} 1, & \text{工件 } w \text{ 被分配到车间 } j; \\ 0, & \text{otherwise;} \end{cases}$$

$$B_{wq}^{jm} = \begin{cases} 1, & O_{wq} \text{ 被分配到车间 } j \text{ 机器 } m; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

其中: A_w^j 为判断工件 w 是否被分配到车间 j , 如果是则 $A_w^j = 1$, 否则 $A_w^j = 0$; B_{wq}^{jm} 为判断工件 w 的工序 q 是否正在车间 j 的机器 m 加工, 如果是则 $B_{wq}^{jm} = 1$, 否则 $B_{wq}^{jm} = 0$ 。

问题约束中: 式 (1) 保证根据订单生产的工件数量与订单要求一致; 式 (2) 对工件完成时间进行约束, 其应该在订单要求的最早完成时间与最晚完成

表1 参变量定义

符号	定义
n_{product}	产品数量
$n_{\text{workpiece}}$	工件数量
n_{jobshop}	车间数量
n_{machine}	机器数量
W	工件集合, $W = \{W_i i = 1, 2, \dots, n_{\text{workpiece}}\}$
J	车间集合, $J = \{J_i i = 1, 2, \dots, n_{\text{jobshop}}\}$
M	机器集合, $M = \{M_i i = 1, 2, \dots, n_{\text{machine}}\}$
w	工件索引
p	工序索引
j	车间索引
m	机器索引
t	推演时间
O_w	工件 w 的所有工序
O_{wp}	工件 w 的工序 p
ST_{wp}^j	车间 j 内的工件 w 工序 p 的开始时间
ET_{wp}^j	车间 j 内的工件 w 工序 p 的结束时间
WT_{wp}^j	车间 j 内的工件 w 工序 p 开始前的等待时间
T_{wp}	O_{wp} 完工时间
T_w	工件 w 生产用时时间
C_m^j	工厂 j 机器 m 最大待加工工件数
C_{mt}^j	时间 t 时刻车间 j 机器 m 等待加工工件数
DS_w	订单要求工件 w 交工最早时间
DE_w	订单要求工件 w 交工最晚时间
WT_w	工件 w 的等待时间
WJ_j	分布在车间 j 中的所有工件
T_{\max}	订单完成时间
UT_w	工件 w 总生产等待时间
UT	所有工件生产平均等待时间

时间之间; 式 (3) 定义了工件加工工序的开始时间与完成时间之间的关系; 式 (4) 保证工件的下一个工序加工在上一个工序完成之后; 式 (5) 保证每个工件仅能被分配到一个车间进行加工; 式 (6) 保证工件的每个工序仅能在一个机器上加工; 式 (7) 限制了机器进料仓的工件数量。

1.3.2 目标函数

针对 DFJSP, 以最小化订单完成时间、最大化机器使用率为目标, 定义如下目标函数。

- 1) 最小化订单完成时间。

所生产的工件中最后一个工件完成时间即该批次订单生产作业完成时间, 有

$$T_{\max} = \max\{T_w, w \in \{1, n_{\text{workpiece}}\}\}. \tag{8}$$

订单完成时间越小, 意味着车间整体生产效率越高。因此, 模型的第 1 个优化目标为最小化订单完

成时间,即

$$f_1 = \min(T_{\max}). \quad (9)$$

2) 最大化机器利用率目标.

在生产过程中,工件停留在某一机器的时间并不等于实际在该机器的加工时间,而是加工时间与等待时间的总和.对于任一工件而言,其在机器上的等待时间与加工时间的关系为

$$WT_w = \sum_q WT_{wq}, q \in \{1, \|O_w\|\}; \quad (10)$$

$$UT_w = \frac{WT_w}{T_w} \times 100\%; \quad (11)$$

$$UT = \frac{\sum_w UT_w}{n_{workpiece}}, w \in \{1, n_{workpiece}\}. \quad (12)$$

等待时间一方面影响生产效率,另一方面容易造成部分工件堆积.因此,高效的调度方案应该均衡

利用各个机器生产资源,尽可能缩短工件无意义等待时间,提高机器的利用率.模型第2个优化目标为

$$f_2 = \min(UT). \quad (13)$$

2 代理模型

进化算法在求解约束复杂、昂贵优化问题时,容易出现评估时间过长、评估次数过多的缺陷,严重影响算法求解速度.代理模型可以有效减少进化算法搜索过程中的评估次数,解决评估成本过高的问题^[20].代理模型的应用主要分为解集特征提取、代理模型构建和代理模型配置3个阶段.首先,针对历史调度数据,使用解集特征提取方法进行数据预处理;特征提取完毕后,选择合适的机器学习模型构建代理模型;模型构建完毕后,使用代理模型辅助进化算法完成初始解生成和解集评估,加快搜索速度.VMSM模型配置、应用的流程如图1所示.

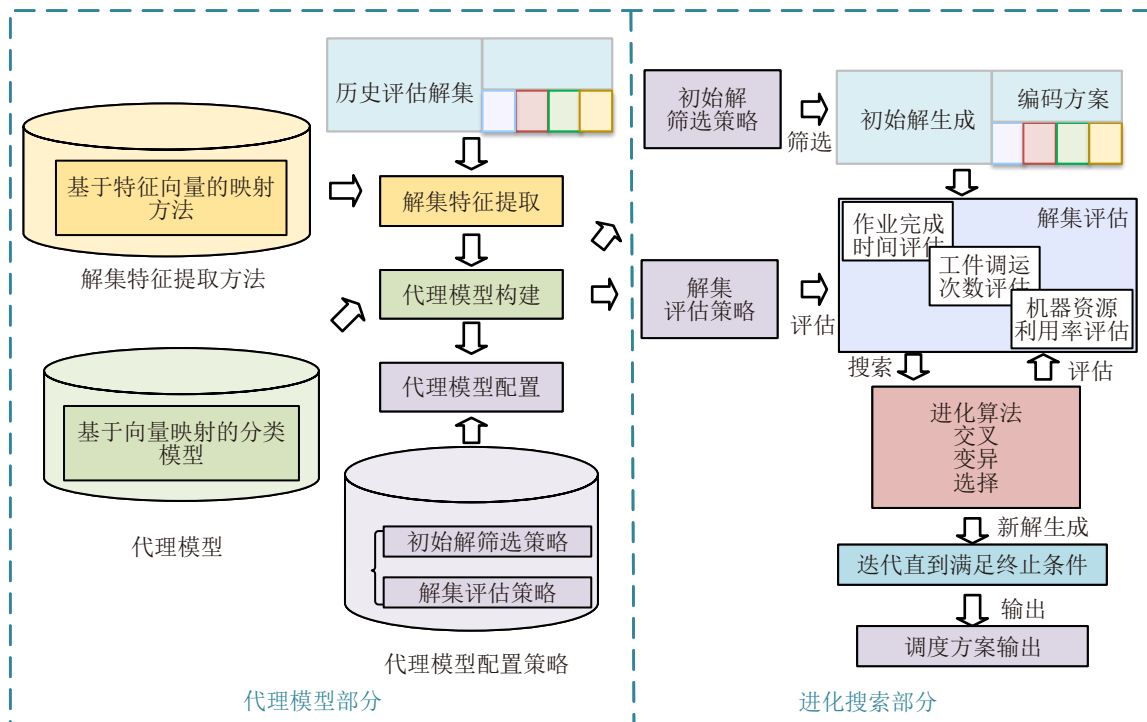


图1 基于代理模型的分布式作业车间调度算法

2.1 基于向量映射的特征提取方法

在代理模型的构建过程中,需要使用进化算法搜索时真实评估过的解,从中提取出与当前求解问题解空间有关的先验知识,最直接的情况是利用问题的解进行代理模型构建,但由于分布式车间调度这类实际的调度问题其解集的空间分布具有与各自问题强相关的无法统一描述的关系,直接使用解时往往效果很差.为此,本文设计一种基于特征向量的映射方法,将原编码解集中的编码映射为连续的特征向量,解决了解集特征提取问题.

在所设计的编码方式中,解的每一位都代表DFJSP中一个实体,如工序、机器、车间等.同一类型的不同实体间既有差异性(如机器的空间位置),也有相似性(如机器的类型).直接将编码作为训练数据时,模型很难区分不同实体之间的相似性,从有限的训练数据中学到的知识极为有限.针对各类实体,本文构建了一套特征变量体系,利用特征变量描述实体的特定属性.将编码中的元素映射为特征向量,从而得到一组连续的高维特征向量.此特征向量蕴含了调度问题解的具体信息,并且具有连续的特

征值, 能够用于后续模型训练.

每个特征的取值由实体属性决定, 根据实体属性的取值不同, 将其映射到范围内的某个区间值. 即对于同一个实体而言, 其映射后的特征向量固定, 两个实体越相似, 其特征向量越接近. 该方法与自然语言处理中 word2vec 类似, 为每个词库中每个单词嵌入一个词向量, 用于描述该单词在整个文本中的语义关系. 例如, 空间位置相似但加工工序不同的两个机器在映射后的特征向量中表示位置的特征值之间的差异很小, 表示工艺的特征值之间的差异很大.

具体步骤如下: 针对样本空间中的每个个体, 取出每个个体解的编码. 编码中的每个元素分别代表车间场景中的一个实体, 根据构建的特征变量体系用每个实体的特征代替编码元素, 得到解的特征向量序列. 方法示意如图 2 所示.

2.2 代理模型设计

在构建分类模型时, 首先需要利用已有的样本评估结果或历史方案的评估结果作为先验知识, 以其作为分类标签构造训练集. 将未评估的解集样本作为测试集, 对可能的类别进行预测, 以达到精优筛选的目的. 为了更准确地利用解集中蕴含的先验知识, 首先利用前文设计的基于特征向量的映射方法对样本解集进行映射. 在制定分类标签时, 对于单目标优化问题, 根据适应度值排名将样本分为好坏两类; 对于多目标问题, 则可根据支配关系将其分为支

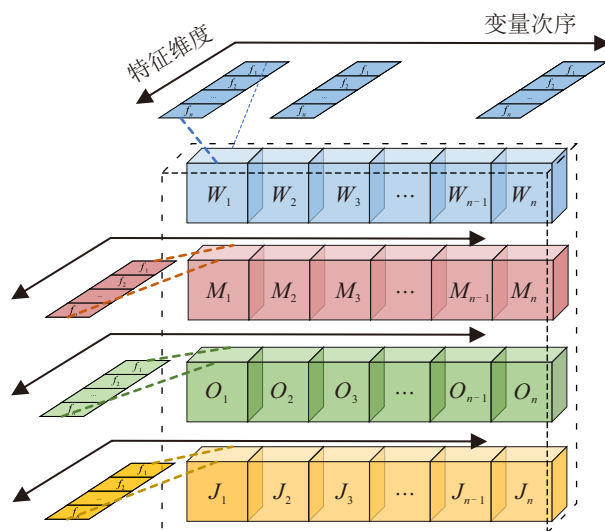


图2 基于特征向量的映射方法

配解和非支配解两类. 在选择分类模型时, 可选用深度神经网络和机器学习模型, 包括支持向量机、决策树、逻辑回归、K-邻近算法、多层感知机等^[21].

与传统分类模型不同, 在本文中, 分类模型的损失函数除了根据预测的类别计算之外, 还包括预测的可信度, 这样除了能够让模型具备预测类别的能力外, 还可以评估预测结果可信度, 并以此作为解集选择的依据. 例如, 在同样预测为好的样本中, 总是选择可信度更高的样本优先加入子代展开搜索可以增强算法搜索能力. VMSM 的模型设计思路如图 3 所示.

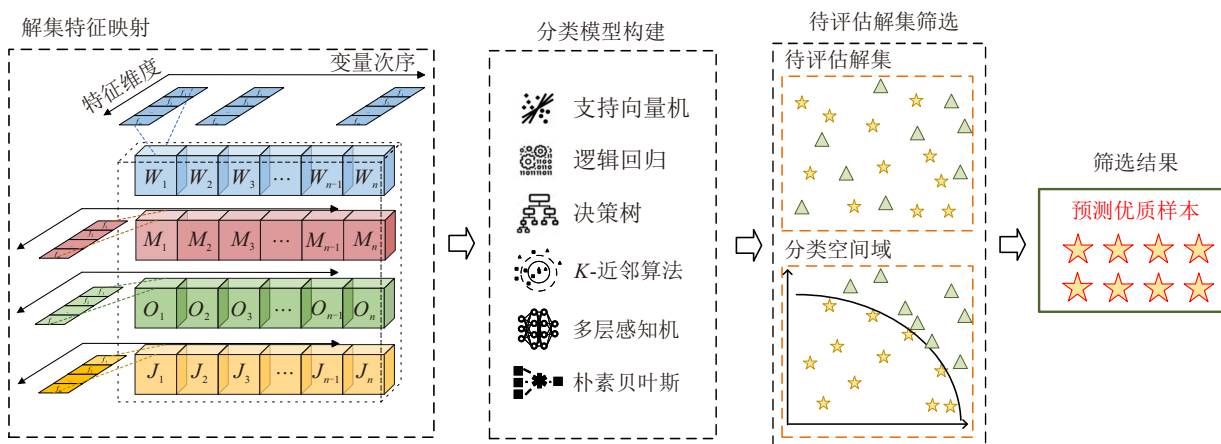


图3 基于向量映射的代理模型设计

2.3 代理模型配置方法

2.3.1 基于代理模型的初始解筛选策略

利用代理模型进行初始解筛选, 可以提升初始解的质量, 减少进化算法的迭代次数, 加快算法的收敛. 通过代理模型选择一组性能预测较好的初始解作为进化算法的起点, 这些解有较高的概率, 具有更好的表现, 从而有助于指导进化算法更有效地搜索

解空间. 一般而言, 利用代理模型进行初始解的筛选包括代理模型选择、代理模型训练、代理模型筛选 3 个步骤, 筛选后的解集可以作为进化搜索的初始解.

本文设计的 VMSM 可以预测解的不同表现, 根据预测结果对解进行聚类和分类, 得到可能表现更好的解, 因此可以用来筛选初始解. 该代理模型在实

际应用前,需要利用相似调度场景下的历史数据或相似调度目标下的历史数据训练模型,利用历史数据进行离线训练无需在求解分布式作业车间调度问题时增加额外的计算开销,但由于此时没有使用当前问题的数据进行训练,所得的模型准确性可能较差.模型预训练完成后,能够实现对解集的好坏进行预测分类.此时使用训练好的分类模型将待求解问题随机生成的初始解进行分类,根据分类结果将预测为优质的解集作为初始解进行后续迭代.

2.3.2 基于代理模型的解集评估策略

在传统进化计算中,进化搜索的迭代过程依赖于目标函数对新个体依次进行评估给出目标值(适应度值).在DFJSP中,目标函数的评估过程是根据解集进行整个作业调度过程的推演从而得出目标值,由于该问题评估过程中涉及的调度实体较多,评估所耗费的时间较长.本文设计了基于代理模型的解集评估策略,使用代理模型来减少耗时的真实目标函数评估次数,提高进化算法的搜索效率.

所设计的基于代理模型的解集评估策略,主要思路是使用训练好的模型区分不同解为优质解的潜在可能性,挑选出更具潜力的解使用真实目标函数评估,进行后续迭代搜索,减少真实评估函数的评估次数,与传统的代理模型通过对真实目标函数进行数学上的拟合相比有所不同.

基于代理模型的解集评估策略包含模型训练、模型启用、模型停用与更新和迭代停止4个阶段策略,如图4所示.模型训练指首先利用历史数据对代理模型进行预训练,同时利用进化搜索的前 N 代真实评估数据进行模型的更新,即在前 N 代中代理模

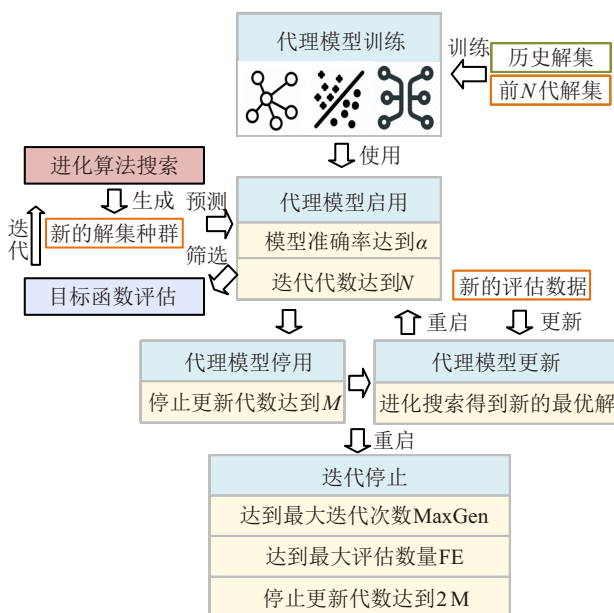


图4 基于代理模型的解集评估策略

型并不参与进化搜索过程.模型启用是指当达到启用条件后,在进化搜索框架中启用代理模型提高搜索的效率.在本文中,代理模型的启用条件设置为模型的准确率达到或超过阈值 α ,或进化迭代的代数到达 N 次.模型停用与更新是为了避免因模型的局限性导致其对搜索空间探索和利用的不平衡,使得整个搜索过程陷入早熟或局部最优的困境,即在一定情况下停止使用代理模型并对其进行更新后再重启.迭代停止是设置相应的迭代终止搜索条件停止搜索.

3 实验验证与分析

3.1 实验设置

3.1.1 实验场景设置

为了验证本文所设计的调度算法在面对不同场景下的调度效果,根据不同订单任务场景下调度需求设计3组不同的调度场景.

1) 场景1: 订单分组间隔到达.

订单分组间隔到达是指订单按照一定的时间间隔分批到达车间.每个订单之间有固定的时间间隔,这样的场景模拟生产过程中,订单并非一次性集中到达,而是分批到达的情况.场景1的具体信息如下所示:

单次订单: A 型号产品 1 个, B 型号产品 1 个;

订单数量: 共 3 组订单.

2) 场景2: 订单连续集中到达.

订单连续集中到达是指订单在一个较短的时间窗口内连续到达车间.该场景常见于促销活动或突发的大量订单需求,具有一定的代表性.场景2的具体信息如下所示:

单次订单: A 型号产品 12 个;

订单数量: 共 2 组订单.

3) 场景3: 订单不规则随机到达.

订单不规则随机到达是指订单到达车间的时间是不规则且随机的,没有明显的规律.这种情况可能出现在多种因素影响下的生产环境中,例如客户订单的随机性、供应链的波动等.场景3的具体信息如下所示:

订单数量: 共 9 组订单;

订单信息: 第 1 组 ~ 第 6 组订单每个订单包含 A 型号产品 5 个, B 型号产品 1 个, 订单 7 包含 C 型号产品 1 个, 订单 8 和订单 9 包含 D 型号产品 1 个.

3.1.2 实验算法设置

本实验选择的进化算法包括粒子群算法(PSO)、遗传算法(GA)、L-SHADE^[22]、MjSO^[23]、GABC^[24],进

化算法参数如表 2 所示.

表2 进化算法参数

优化目标	种群大小	迭代次数	实验次数
f_1, f_2	20	100	10

实验选择作为代理模型分类器的机器学习算法包括支持向量机 (SVM)、朴素贝叶斯算法 (Bayes)、K-近邻算法 (KNN)、逻辑回归 (LR)、多层感知机 (MLP). 所有实验均通过 C++和 Python 实现, 运行在 Intel i9-12900HX 32 cores 16 GB 3080Ti 的主机上, 操作系统为 windows 10. 进化搜索框架使用 C++语言, 在 Visual Studio 平台开发; 模型训练使用 python 语言, 在 Visual Studio Code 平台开发. 在统计迭代时间时, 本文减去了因跨平台调用模型文件所带来的额外开销.

3.2 实验仿真和案例分析

3.2.1 进化搜索框架测试

进化搜索框架测试实验只测试不同进化算法在多个场景下的性能表现, 未添加 VMSM 辅助进化搜索. 该实验一方面可以为后续实验提供基准实验数据, 另一方面可以为 VMSM 训练提供历史数据. 表 3 展示了 3 个场景下, 不同算法更新一代所需要的平均时间 (单位: s).

表3 不同进化算法迭代时间对比

	订单完成时间 f_1			机器资源使用率 f_2		
	场景1	场景2	场景3	场景1	场景2	场景3
PSO	4.67	26.82	61.86	4.78	26.39	75.06
GA	4.61	26.81	66.33	4.52	26.77	75.40
MjSO	4.59	26.08	58.37	4.61	26.76	75.22
L-SHADE	4.69	26.29	57.76	4.63	26.35	72.47
GABC	4.72	26.60	59.03	4.64	26.42	73.71

由表 3 可以看出, 随着场景 1 到场景 3 的复杂度逐渐上升, 进化搜索中每更新一代所需要的时间也大幅增加. 相同问题场景下不同进化算法的迭代时间差异并不显著, 这可能是因为在迭代过程中评估时间占据了更大的比例, 而不同算法的评估时间基本一致.

表 4 和表 5 分别展示从场景 1 到场景 3 不同算法在目标函数 f_1 和目标函数 f_2 下的优化效率. 通过表 4 和表 5 可以看出, 在进化迭代的过程中, 两种目标函数下不同场景均有较好的优化效果, 大多数场景下的搜索优化率都在 30 % 以上, 个别场景甚至能达到 100 % 以上. 在场景 1 中, 两个目标维度下的最

表4 不同进化算法在目标函数为 f_1 时的优化效率对比

	算法	订单完成时间 f_1		
		初始值/s	最优值/s	优化率/%
场景1	PSO	16173	11143	31.10
	GA	16173	10778	33.36
	MjSO	16173	10989	32.05
	L-SHADE	16173	10723	33.70
	GABC	16173	9379	42.01
场景2	PSO	25158	14325	43.06
	GA	25158	14133	43.82
	MjSO	25158	14796	41.19
	L-SHADE	25158	13797	45.16
	GABC	25158	14162	43.71
场景3	PSO	50077	32200	35.70
	GA	50077	30490	39.11
	MjSO	50077	32630	34.84
	L-SHADE	50077	32564	34.97
	GABC	50077	30435	39.22

表5 不同进化算法在目标函数为 f_2 时的优化效率对比

	算法	机器利用率 f_2		
		初始值	最优值	优化率/%
场景1	PSO	0.631	0.762	20.76
	GA	0.631	0.772	22.35
	MjSO	0.631	0.834	32.17
	L-SHADE	0.631	0.775	22.82
	GABC	0.631	0.868	37.56
场景2	PSO	0.325	0.637	96.00
	GA	0.325	0.609	87.38
	MjSO	0.325	0.651	100.31
	L-SHADE	0.325	0.559	72.00
	GABC	0.325	0.626	92.62
场景3	PSO	0.292	0.487	66.78
	GA	0.292	0.507	73.63
	MjSO	0.292	0.511	75.00
	L-SHADE	0.292	0.458	56.85
	GABC	0.292	0.508	73.97

大优化率能够达到 42.01 % 和 37.56 %; 场景 2 能够达到 45.16 % 和 100.31 %, 场景 3 能够达到 39.22 % 和 73.97 %.

比较不同算法迭代效果, GABC 在 6 个优化问题中有 4 个取得了最优值, L-SHADE、MjSO 在其余两个问题中分别取得了最优值. 根据迭代效果确定

最优算法的选择依据为: 场景 1 和场景 3 的问题求解均使用 GABC 算法, 场景 2 的订单完成时间 f_1 问题使用 L-SHADE 算法, 机器利用率 f_2 问题使用 MjSO 算法.

通过实验结果可以看出, 进化算法能够对相应目标进行迭代优化, 验证了本文所设计的基于进化算法的分布式柔性车间调度算法的有效性. 但是, 进化算法的优化存在着运算时间较长的缺点, 对于复杂一些的场景 2 和场景 3, 更新一代需要的平均时间分别在 25 s 以上和 55 s 以上.

3.2.2 代理模型解集特征提取方法测试

首先利用第 3.2.1 节进化搜索过程中生成的样本及其评估值作为原始数据, 根据 $f_1 = \min(T_{\max})$ 目标值将其分为优质解和劣质解两类 (各取前 20%), 并按照 7: 3 的比例划分为训练集和测试集. 构造数据集后, 使用不同的机器学习方法作为分类器进行训练, 以分类的准确率来评价特征提取方法好坏. 不同机器学习模型分类准确率如表 6 和图 5 所示.

表6 特征提取方法效果比较

	SVM	Bayes	KNN	LR	MLP	
未用特征向量	训练集	0.6454	0.6624	0.6907	0.6150	0.5000
	测试集	0.5469	0.5312	0.5535	0.5276	0.5000
基于特征向量	训练集	0.8719	0.8407	0.7266	0.8715	0.9995
	测试集	0.8453	0.8201	0.7088	0.8346	0.4922

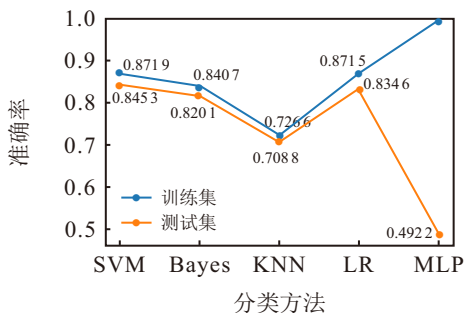
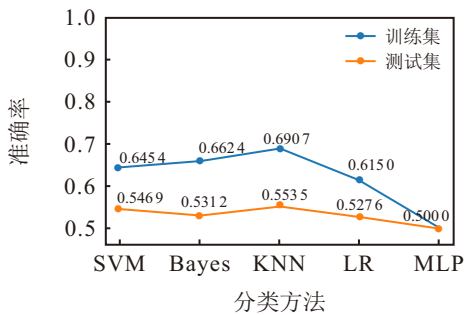


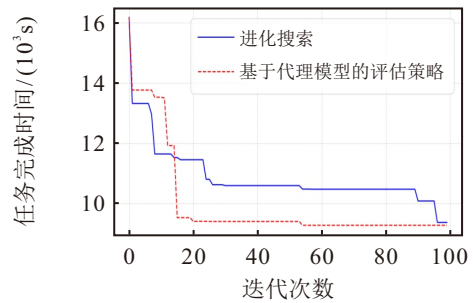
图5 特征提取方法测试结果

根据以上实验结果可以看出, 在不使用特征向量提取方法时, 很难对样本进行分类, 并且训练集与测试集表现差异较大, 而采用所设计的特征向量映射特征提取方法, 模型准确率有了很大提升, 在不同的机器学习模型下, 分类准确率能够提高 25% ~ 35%, 这对于后续构建代理模型具有重要意义. 但是所设计的方法并不能在每种模型上都取得良好的效果, 如在 MLP 模型实验中, 无论是否采取特征提取方法都未能实现样本的正确分类, 这表明所设计方法具有一定的使用局限.

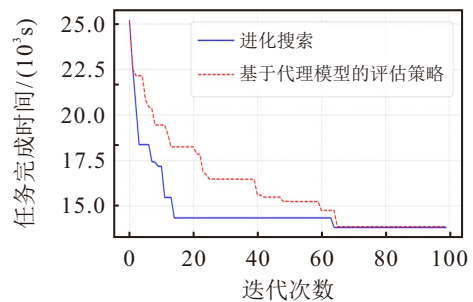
3.2.3 基于代理模型的解集评估策略测试

本节进行对比实验, 以验证基于代理模型的解集筛选策略在进化搜索框架中所发挥的作用. 在进化搜索框架下, 分别设置不使用代理模型策略、使用 VMSM 代理模型策略两种实验条件, 其余实验条件与第 3.2.1 节相同. 本节实验没有进行模型训练, 使用历史数据训练的模型进行测试.

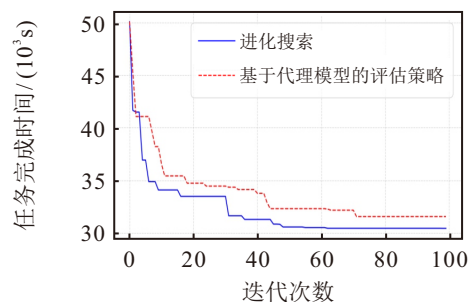
图 6 展示了 3 种场景下, 以最小化订单完成时间为优化目标时的进化搜索情况. 表 7 展示了不同



(a) 场景1-订单分组间隔到达



(b) 场景2-订单连续集中到达



(c) 场景3-订单不规则随机到达

图6 基于代理模型的解集评估策略迭代对比

场景下平均每代搜索所需要的时间(单位: s)以及目标值最优情况(最短加工完成时间)比较。

表7 基于代理模型的解集评估策略效果对比

	场景1		场景2		场景3	
	最优值	平均用时	最优值	平均用时	最优值	平均用时
进化搜索	9379	4.72	13797	26.29	30435	73.71
代理模型	9282	3.06	13836	12.81	31563	38.55
与进化对比	-1.03%	-35.01%	0.28%	-51.26%	3.71%	-47.70%

对图6和表7中数据进行分析可以得出以下结论: 与未采用代理模型的进化算法相比, 添加VMSM代理模型的进化算法迭代效果无明显优劣差异。在场景1中, VMSM辅助的进化搜索算法相较于传统进化搜索算法表现略优, 而在场景2和场景3中略显劣势(差距在5%以内)。然而, 在平均用时方面, 基于VMSM的评估策略显著降低了算法搜索时间, 在3个场景中, 运行时间缩短幅度分别为35.01%、51.26%和47.70%。该实验结果充分表明了VMSM在提升传统进化算法搜索效率、缩短运行时间方面的显著优势, 这对于应用进化算法求解DFJSP、缩短调度方案生成时间具有重要意义。

4 结论

本文从DFJSP入手, 针对进化算法求解速度慢的缺陷, 提出了一种基于向量映射代理模型的调度算法。该模型采用一种基于特征向量映射的解集特征提取方法, 对解集特征有较强的提取能力, 能够应用在进化算法的初始解筛选和解集评估过程中, 减少真实评估次数。通过与进化算法求解结果比较, 表明所提出算法能够在维持迭代效果的同时大幅缩短算法探索时间, 这对于应用进化算法求解DFJSP具有重要参考意义。

后续将在本文的研究基础上从两个方面进行更加深入的探索, 一方面继续扩展和完善本文构建的数学规划模型, 更加贴近最新应用场景; 另一方面, 加大对本文代理模型的适用性测试, 对VMSM在更多调度场景中的性能表现做进一步探索。

参考文献 (References)

[1] Okwudire C E, Madhyastha H V. Distributed manufacturing for and by the masses[J]. *Science*, 2021, 372(6540): 341-342.

[2] 王凌, 邓瑾, 王圣尧. 分布式车间调度优化算法研究综述[J]. *控制与决策*, 2016, 31(1): 1-11. (Wang L, Deng J, Wang S Y. Survey on optimization algorithms for distributed shop scheduling[J]. *Control and Decision*, 2016, 31(1): 1-11.)

[3] 王思涵, 李新宇, 高亮, 等. 分布式车间调度研究综述[J]. *华中科技大学学报: 自然科学版*, 2022, 50(6): 1-10. (Wang S H, Li X Y, Gao L, et al. A review of distributed shop scheduling problems[J]. *Journal of Huazhong University of Science and Technology: Natural Science Edition*, 2022, 50(6): 1-10.)

[4] Tan S L, Wang Y, Sun G Y, et al. A surrogate-assisted evolutionary framework for expensive multitask optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2024(99): 1.

[5] 孙爱红, 雷琦, 宋豫川, 等. 基于深度强化学习求解作业车间机器与AGV联合调度问题[J]. *控制与决策*, 2024, 39(1): 253-262. (Sun A H, Lei Q, Song Y C, et al. Deep reinforcement learning for solving the joint scheduling problem of machines and AGVs in job shop[J]. *Control and Decision*, 2024, 39(1): 253-262.)

[6] Pan L Q, He C, Tian Y, et al. A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(1): 74-88.

[7] Gu Q H, Wang Q, Xiong N N, et al. Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems[J]. *Complex & Intelligent Systems*, 2022, 8(4): 2699-2718.

[8] Zhao M J, Zhang K, Chen G D, et al. A surrogate-assisted multi-objective evolutionary algorithm with dimension-reduction for production optimization[J]. *Journal of Petroleum Science and Engineering*, 2020, 192: 107192.

[9] Wang X J, Wang G G, Song B W, et al. A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 815-827.

[10] Cai X W, Gao L, Li X Y. Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems[J]. *IEEE Transactions on Evolutionary Computation*, 2020, 24(2): 365-379.

[11] Han D, Du W L, Wang X J, et al. A surrogate-assisted evolutionary algorithm for expensive many-objective optimization in the refining process[J]. *Swarm and Evolutionary Computation*, 2022, 69: 100988.

[12] Wang X L, Jin Y C, Schmitt S, et al. An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization[J]. *Information Sciences*, 2020, 519: 317-331.

[13] Zhu K K, Gong G L, Peng N T, et al. Dynamic distributed flexible job-shop scheduling problem considering operation inspection[J]. *Expert Systems with Applications*, 2023, 224: 119840.

[14] Du B G, Han S, Guo J, et al. A hybrid estimation of distribution algorithm for solving assembly flexible job shop scheduling in a distributed environment[J]. *Engineering Applications of Artificial Intelligence*, 2024, 133: 108491.

- [15] Zhang H L, Chen Y, Zhang Y T, et al. Energy-saving distributed flexible job shop scheduling optimization with dual resource constraints based on integrated Q -learning multi-objective grey wolf optimizer[J]. *Computer Modeling in Engineering & Sciences*, 2024, 140(2): 1459-1483.
- [16] Zhang Z Q, Wu F C, Qian B, et al. A Q -learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation[J]. *Expert Systems with Applications*, 2023, 234: 121050.
- [17] Li R, Gong W Y, Wang L, et al. Surprisingly popular-based adaptive memetic algorithm for energy-efficient distributed flexible job shop scheduling[J]. *IEEE Transactions on Cybernetics*, 2023, 53(12): 8013-8023.
- [18] Liu Z F, Zha J M, Yan J, et al. An improved genetic algorithm with an overlapping strategy for solving a combination of order batching and flexible job shop scheduling problem[J]. *Engineering Applications of Artificial Intelligence*, 2024, 127: 107321.
- [19] Du Y, Li J Q. A deep reinforcement learning based algorithm for a distributed precast concrete production scheduling[J]. *International Journal of Production Economics*, 2024, 268: 109102.
- [20] Jin Y C. Surrogate-assisted evolutionary computation: Recent advances and future challenges[J]. *Swarm and Evolutionary Computation*, 2011, 1(2): 61-70.
- [21] 刘书磊. 代理模型辅助的进化组合优化算法研究[D]. 西安: 西安电子科技大学, 2023.
(Liu S L. Research on evolutionary combinatorial optimization algorithm aided by agent model[D]. Xi'an: Xidian University, 2023.)
- [22] Tanabe R, Fukunaga A S. Improving the search performance of SHADE using linear population size reduction[C]. *IEEE Congress on Evolutionary Computation*. Beijing, 2014: 1658-1665.
- [23] Li Y T, Han T, Wang X F, et al. MjSO: A modified differential evolution with a probability selection mechanism and a directed mutation strategy[J]. *Swarm and Evolutionary Computation*, 2023, 78: 101294.
- [24] Mumtaz J, Minhas K A, Rauf M, et al. Solving line balancing and AGV scheduling problems for intelligent decisions using a Genetic-Artificial bee colony algorithm[J]. *Computers & Industrial Engineering*, 2024, 189: 109976.

作者简介

靳思远 (2002-), 男, 博士生, 主要研究方向为智能优化算法、车间调度, E-mail: 20373680@buaa.edu.cn;

彭程 (1999-), 男, 硕士, 主要研究方向为智能优化算法、建模仿真, E-mail: pengc@buaa.edu.cn;

王薇 (1983-), 女, 教授, 博士, 主要研究方向为信息物理系统安全控制、群智能体系统分布式协同控制, E-mail: w.wang@buaa.edu.cn;

任磊 (1979-), 男, 教授, 博士, 主要研究方向为工业互联网、工业智能、智能制造系统, E-mail: renlei@buaa.edu.cn;

吕金虎 (1974-), 男, 教授, 博士, 主要研究方向为智能系统、复杂网络、非线性电路与系统, E-mail: jhlu@iss.ac.cn;

赖李媛君 (1987-), 女, 副教授, 博士, 主要研究方向为智能制造系统、智能优化算法、高性能计算, E-mail: lailiyuanjun@buaa.edu.cn.