



计算机科学

COMPUTER SCIENCE

基于离散变邻域蜉蝣优化的装配作业车间调度算法

陈雅莉, 潘友林, 刘耿耿

引用本文

陈雅莉, 潘友林, 刘耿耿. 基于离散变邻域蜉蝣优化的装配作业车间调度算法[J]. 计算机科学, 2024, 51(9): 283-289.

CHEN Yali, PAN Youlin, LIU Genggeng. [Assembly Job Shop Scheduling Algorithm Based on Discrete Variable Neighborhood Mayfly Optimization](#) [J]. Computer Science, 2024, 51(9): 283-289.

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)

[一种时延能耗感知的在轨边缘计算任务卸载调度方法](#)

Delay and Energy-aware Task Offloading Approach for Orbit Edge Computing

计算机科学, 2024, 51(6A): 240100188-9. <https://doi.org/10.11896/jsjx.240100188>

[基于改进型蛇算法的RFID网络规划部署](#)

Enhanced Snake Optimizer Based RFID Network Planning

计算机科学, 2024, 51(6): 375-383. <https://doi.org/10.11896/jsjx.230300130>

[一种基于修正机制和强化学习的作业车间调度问题的优化算法](#)

Optimization Algorithms for Job Shop Scheduling Problems Based on Correction Mechanisms and Reinforcement Learning

计算机科学, 2023, 50(6): 274-282. <https://doi.org/10.11896/jsjx.220900112>

[海上风电场通用运维路径规划模型优化及仿真](#)

Optimization and Simulation of General Operation and Maintenance Path Planning Model for Offshore Wind Farms

计算机科学, 2022, 49(6A): 795-801. <https://doi.org/10.11896/jsjx.210400300>

[多目标优化算法求解多柔性作业车间调度问题](#)

Solving Multi-flexible Job-shop Scheduling by Multi-objective Algorithm

计算机科学, 2020, 47(12): 239-244. <https://doi.org/10.11896/jsjx.191100042>

基于离散变邻域蜉蝣优化的装配作业车间调度算法

陈雅莉 潘友林 刘耿耿

福州大学计算机与大数据学院 福州 350116

福建省网络计算与智能信息处理重点实验室 福州 350116

(chen_yali@126.com)

摘要 由于受到疫情影响,企业迫切地需要通过升级改造自动化柔性生产线来实现降本增效。在这一背景下,装配作业车间调度问题(Assembly Job Shop Scheduling Problem,AJSSP)再一次成为学术界和企业界的研究热点。AJSSP比普通作业车间调度问题多了一道装配阶段,故其存在前后工序相互制约和多机并行现象,问题求解也更加复杂。针对该问题,提出了一种基于离散变邻域蜉蝣优化算法(Discrete Variable Neighborhood Mayfly Algorithm,D-VNMA)的调度方法,主要工作如下:1)采用符合 Lamarckian 特性的编码解码机制,实现个体有效信息的迭代继承;2)使用 Circle 映射融合常见启发式算法初始化蜉蝣种群,保证种群的多样性;3)加入新的邻域探索策略,采用多种不同的邻域结构和搜索策略的差异组合,增加搜索方案的多样性,提高寻找局部最优解的搜索效率;4)提出改进的雌雄蜉蝣交配策略,提高算法全局探索能力,加快算法整体收敛速度。在实验过程中,通过试验设计(Design of Experiment,DOE)方法获得 D-VNMA 的最佳参数设置,并在不同规格 AJSSP 算例数据上将 D-VNMA 和其他算法进行比较。实验结果表明,D-VNMA 得到最优解的概率提升了 30%,且收敛效率最高可提升 62.15%。

关键词: 装配作业车间;车间调度;蜉蝣优化算法;Circle 映射;邻域搜索

中图分类号 TP301

Assembly Job Shop Scheduling Algorithm Based on Discrete Variable Neighborhood Mayfly Optimization

CHEN Yali,PAN Youlin and LIU Gengeng

College of Computer and Data Science,Fuzhou University,Fuzhou 350116,China

Fujian Key Laboratory of Network Computing and Intelligent Information Processing,Fuzhou 350116,China

Abstract Due to the impact of the epidemic,it is more urgent for enterprises to reduce costs and increase efficiency by upgrading automated flexible production lines. In this context,the assembly job shop scheduling problem(AJSSP) has once again become a research hotspot in academia and business circles. AJSSP has one more assembly stage than ordinary job-shop scheduling problems,so it has the phenomenon of mutual restriction and multi-machine parallel,and the problem solving is also more complicated. To solve this problem,a scheduling method based on a discrete variable neighborhood mayfly algorithm(D-VNMA) is proposed. The main work is as follows:1) Adopt the encoding and decoding mechanism conforming to Lamarckian characteristics to realize the iterative inheritance of individual effective information. 2) Circle mapping and common heuristic algorithm are used to initialize the ephemera population to ensure the diversity of the population. 3) A novel strategy for exploring neighborhoods,incorporating a variety of distinct neighborhood structures and search strategies,is employed to enhance the diversity of search schemes and optimize the efficiency of finding local optimal solutions. 4) An improved mating strategy of male and female mayflies is proposed to accelerate the global exploration ability of the algorithm and improve the overall convergence speed of the algorithm. During the experiment,the optimal parameter setting of D-VNMA is obtained by the design of experiment(DOE) method,and D-VNMA is compared with other algorithms in AJSSP example data of different specifications. Experimental results show that the probability of obtaining the optimal solution of D-VNMA is increased by 30%,and the convergence efficiency is increased by 62.15%.

Keywords Assembly job shop,Job shop scheduling,Mayfly optimization algorithm,Circle mapping,Neighborhood search

到稿日期:2023-09-15 返修日期:2024-03-29

基金项目:福建省杰出青年科学基金(2023J06017)

This work was supported by the Science Funds for Distinguished Young Scholars of Fujian Province,China(2023J06017).

通信作者:刘耿耿(liugengeng@fzu.edu.cn)

1 引言

疫情期间,企业在面对激烈市场竞争的同时还面临着极大的用工缺口,因此不得不通过更加灵活的生产线实现最短的生产时长。在制造体系中,最终产品都是由加工阶段生产出来的组件组装而成。因此,研究结合了生产和装配阶段的装配作业车间调度问题(Assembly Job Shop Scheduling Problem, AJSSP)更加符合市场需求,可为企业提供强有力的竞争优势。

AJSSP 相比普通的作业车间调度问题(Job Shop Scheduling Problem, JSSP)^[1]多了一个装配阶段,涉及的物料结构清单(Bill of Materials, BOM)产品多样、零件繁多、层次嵌套,导致工艺路径复杂多变,增加了调度难度。AJSSP 属于柔性作业车间调度问题(Flexible Job Shop Scheduling Problem, FJSSP)^[2]的一个特例,求解中既要考虑前后工序制约,又需考虑并行机调度场景,因装配阶段的复杂性,故被独立成一个单独的 NP 难题。从 1999 年至今,针对 JSSP 和 FJSSP 问题已有学者进行了大量研究,而 AJSSP 问题的研究则相对较少^[3]。

文献[4]提出了一种知识驱动的构造启发式算法,该方法虽然在结果搜索上优于其他启发式算法,但却只适用于求解单目标问题。文献[5]使用一种改进的象群优化算法最小化总耗能,同时提出一种左移解码方式,虽然极大地提高了解的质量,但也增加了计算时间,故而只适合求解单目标的静态调度问题。针对多目标问题,文献[6]则将问题进一步表述为一种混合整数规划模型,并提出结合粒子群算法和遗传算法的启发式算法来求解。文献[7]以完工时间和库存最小为求解目标,采用一种具有杂交染色体编码机制的带精英策略的快速非支配排序遗传算法(Nondominated Sorting Genetic Algorithm II, NSGA-II)进行求解。由于 AJSSP 中前后工序存在制约,因此上述两种方法在问题求解过程中会产生大量的无效解。文献[8]则在求解最小滞后成本、运输成本和完工时间时,利用 ϵ -约束算法和拉格朗日松弛算法降低了其线性模型在多目标问题上的求解复杂度,然而其无法求解具有更加复杂 BOM 结构产品的 AJSSP 问题。在动态调度问题的求解上,文献[9]抛弃了传统的启发式和元启发式算法,提出一种多智能体强化学习方法。该方法中的训练样本由基于规则的启发式方法生成,这也使得该算法在实际生成过程中的执行率很低。文献[10]设计了一种基于图神经网络和近端策略优化算法的强化学习算法求解作业车间调度问题,算法的训练数据来源于实际生产甘特图,计算出来的结果更贴合实际生产,但是该算法在较大规模实例上表现一般。为了实现算法在实际生产环境中的落地,文献[11]将真实检测数据构建到产品质量转换模型中,并将该产品质量转换模型和一种新提出的基因型融入到 NSGA-II 中,最终实现以产品质量、产品数量和交货期为目标的问题求解。该方法在性能和可用性指标方面都优于随机搜索和爬坡,且具有较高的实际应用价值。文献[12]首先按照装配机器进行批量分配,然后再进行生产的调度分配,并利用单机的逆向分配公式,反向进行批量调度分配。文献[13]则直接将问题转换成了批分割和批调度,

提出了一种改进的采用四层染色体的人工蜂群算法。然而,分批调度更适用于多品种、小批量的生产场景。

在现实具有半自动化的装配作业车间内,APS, ERP 和 MES 等上游系统大部分还处于彼此独立阶段,调度结果和生产数据无法联动,真实生产过程中,还离不开人工通过 Excel 和邮件对进展和计划进行沟通协调。随着订单量的不断增加,单纯依靠排产员的手工调度已无法满足生产要求,且会带来大量的成本亏损,故现实需求对提高车间调度精准度和计划的可执行率具有极大的迫切性。而在上述研究成果中,每个求解的问题均存在一定的假设前提,或者偏离真实生产数据,或者将实际调度问题理想化处理,或者只考虑部分关键场景,或者将问题约束提前、滞后处理,导致问题求解的约束条件被部分裁剪,算法得到的结果存在优化瓶颈或者计划执行率差,故而很多研究成果更偏向理论化,缺乏一定的实际应用场景。

为了弥补这一空白,本文将基于某个真实环境下的装配作业车间场景,提出解决真实 AJSSP 问题场景的求解方法。本文的主要贡献如下:

1) 构建包括工序顺序、工序节拍、加工能力、加工时长、释放时长、准备时长等约束,且以最小化最大完工时间(Makespan)为目标的新型 AJSSP 数学模型,通过该模型更易于 MES 系统中实时采集的生产数据在问题求解过程中实现价值转换;

2) 首次提出将蜉蝣优化(Mayfly Algorithm, MA)算法运用到 AJSSP 问题求解中,并从编码方式、种群生成、邻域搜索方面对算法进行改进,提出一种新的 D-VNMA 算法以提高算法求解效率和求解精度,同时实现在装配作业车间中的快速应用;

3) 实验结果表明, AJSSP 问题求解过程中, D-VNMA 算法与其他算法相比,在收敛速度和求解效率上效果更佳。

2 AJSSP 数学模型

2.1 问题描述

本文的 AJSSP 可描述为加工工件按照其工序拆分成 N 个加工单元,所有加工单元可选的工序工位共有 M 个,每个加工单元间存在一定的加工能力和顺序约束,求解加工单元在满足约束条件下的工序工位选择和加工顺序。问题中的具体约束条件描述如下:

1) 工序工位的加工能力有限,不是每个加工单元都可在任意工序工位上加工;

2) 加工单元之间存在顺序依赖关系,即紧前紧后的顺序强约束;

3) 加工单元加工前需要一个准备时长,该值与其顺序加工的前置加工单元有关,不同加工顺序会产生不同的准备时长;

4) 每个加工单元只能在一个工序工位上加工,每个工序工位同时也只能加工一个加工单元,且加工单元在加工过程中不能中断;

5) 每个工位的释放时长各不相同,仅当已排加工单元全部加工完成且共享资源准备就绪才可安排新的加工单元;

6)存在一个流约束的加工单元选择的工序工位必须成组配套出现,当某个加工单元确定了其工序工位为 M_i ,则同一个流约束的其他加工单元只能选择与之配套的其他工序工位。

2.2 变量定义

M :所有工序可选工位数量之和;

N :加工单元数量;

N_i :加工单元 i 的批量个数;

$\tilde{C}T_{ik}$:加工单元 i 在工位 k 上的工序节拍;

\tilde{P}_{ik} :加工单元 i 在工位 k 的实际加工时长;

ST_{ijk} :工位 k ,加工单元 j 紧跟在加工单元 i 后加工的准备时长, i 为 0 时表示工位第一个加工单元;

R_k :工位 k 的初始释放时间,即该工位允许排工的最早时间,该时间为工位加工完已排单元且共享资源已准备就绪;

$A_{ik} = \begin{cases} 1, & \text{允许} \\ 0, & \text{不允许} \end{cases}$:加工单元 i 是否允许在工位 k 上加工的能力;

紧前紧后关系;

$D_{ij} = \begin{cases} 1, & \text{存在} \\ 0, & \text{不存在} \end{cases}$:加工单元 i 和加工单元 j 是否存在紧前紧后关系;

选择

$a_{ik} = \begin{cases} 1, & \text{选择} \\ 0, & \text{不选则} \end{cases}$:加工单元 i 是否选择工位 k 进行加工;

$b_{ij} = \begin{cases} 1, & \text{是} \\ 0, & \text{否} \end{cases}$:加工单元 j 是否紧跟在加工单元 i 后顺序生产, i 为 0 时表示工位第一个加工单元;

S_i :加工单元 i 开始的时间;

C_i :加工单元 i 完工的时间。

2.3 数学模型

本文求解的 AJSSP 优化目标为最小化最大完工时间:

$$\min(\max_{1 \leq i \leq N} C_i) \quad (1)$$

其约束条件如下:

1)加工单元至多只能紧跟在一个加工单元后顺序生产。

$$\sum_{i=1}^N b_{ij} \leq 1, \forall j \in [1, N] \quad (2)$$

$$\sum_{j=1}^N b_{ij} \leq 1, \forall i \in [1, N] \quad (3)$$

2)加工单元只能在一个工位上生产。

$$\sum_{k=1}^M a_{ik} = 1 \quad (4)$$

3)至少有一个工位允许生产加工单元。

$$\sum_{k=1}^M A_{ik} \geq 1, \forall i \in [1, N] \quad (5)$$

4)加工单元只能选择有加工能力的工位,没有加工能力的工位不能选择。

$$\sum_{k=1}^M a_{ik} A_{ik} = 1, \forall i \in [1, N] \quad (6)$$

5)如果同一个工位上加工单元 i 在加工单元 j 之前加工,则加工单元 i 的完工时间一定在加工单元 j 的开工时间之前。

$$C_i > S_j, (\forall b_{ij} = 1) \quad (7)$$

6)加工单元的开始时间必须在前置加工单元的结束时间之后。

$$S_j \geq \max_{1 \leq i \leq N} C_i D_{ij}, \forall j \in [1, N] \quad (8)$$

7)加工单元的开始时间必须晚于工位释放时间,且加工开始前需要有一个加工准备时长。

$$S_i \geq a_{ik} R_k, \forall i \in [1, N] \quad (9)$$

$$S_j = \sum_{i=1}^N b_{ij} C_i + \sum_{k=1}^M \sum_{i=1}^N a_{ik} b_{ij} ST_{ijk} + b_{0j} \sum_{k=1}^M a_{jk} (R_k + ST_{0jk}) \quad (10)$$

8)加工单元的加工时长为工序节拍与加工批量的乘积。

$$\tilde{P}_{ik} = \tilde{C}T_{ik} N_i \quad (11)$$

9)加工单元 i 的完工时间等于开始时间加上加工时长。

$$C_i = S_i + \sum_{k=1}^M \tilde{P}_{ik} a_{ik} \quad (12)$$

3 D-VNMA 算法设计

MA 算法^[14]是 2020 年 7 月提出的一种结合粒子群算法^[15](Particle Swarm Optimization, PSO)、遗传算法^[16](Genetic Algorithm, GA)和萤火虫算法^[17](Firefly Algorithm, FA)的新型算法。MA 算法受到蜉蝣飞行行为和交配过程的启发,具有更强的空间探索能力和更易跳出局部最优的能力,增强了解空间的探索性,避免算法陷入局域最优。本文针对 AJSSP,首次基于 MA 算法提出一种新的 D-VNMA 求解算法。具体算法流程如图 1 所示。

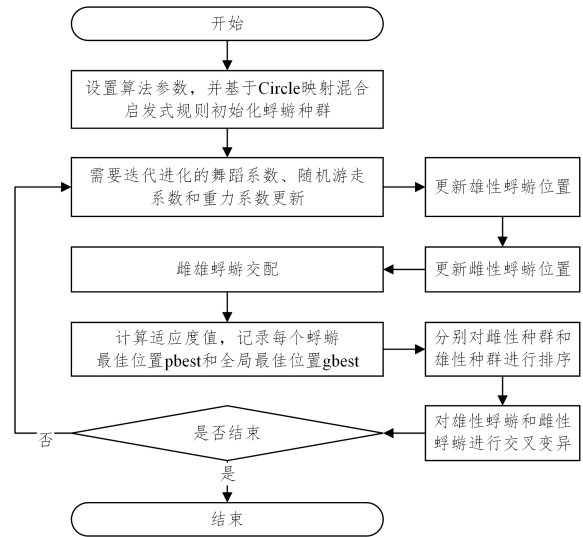


图 1 D-VNMA 算法流程图

Fig. 1 Flow chart of D-VNMA algorithm

3.1 编码与解码

考虑到 AJSSP 的问题模型,本文提出了一种具备 Lamarckian^[18]特性的编码方式。该编码在随机键实数编码上进行改善,具体可描述为:对于加工单元数为 N 且工序工位数为 M 的 AJSSP 问题,其编码用长度为 N 的一维数组 \mathbf{X} 表示。每个编码位的值为浮点数,可表示为“[工位号. 顺序号]”,其整数部分对应工位号,小数部分对应工位上的加工顺序,小数部分的值越小,则其加工越靠前。小数部分的取值为 $0.1 \sim 0.N$ 。每个编码位的取值范围为 $[0, M)$ 。

本文求解的 AJSSP 问题,不仅需要考虑式(6)中的加工能力约束,还需要考虑式(7)中的顺序依赖关系,故其解码规则按照编码位值大小重排后的顺序逐一向后解码,具体解码

方案如算法 1 所示。该解码方案中,通过将解码过程中的工位调整次数 dm 和顺序调整次数 ds 加入其适应度评价指标中,使得算法迭代中能继承上一次迭代中解码的效率以及加工能力约束和顺序依赖关系约束。在不断收敛迭代后,最终工位调整次数 dm 和顺序调整次数 ds 将收敛到 0。

算法 1 D-VNMA 解码算法

输入:加工单元数量 N , 螭蟾个体编码 \mathbf{X}

输出:调度分配结果

1. $n \leftarrow N; ds \leftarrow 0, dm \leftarrow 0$ (initialization);
2. \mathbf{X} 排序;
3. while($n > 0$) do
4. for $m \leftarrow 1$ to M do
5. 选择 \mathbf{X} 中整数位为 m 且值最小的 X_i ;
6. f 找到可分配的加工单元 i then
7. 找到加工单元 i 可加工的工位 m' ;
8. $n \leftarrow n - 1; X_i \leftarrow 0$;
9. $dm \leftarrow dm + m' - m$;
10. else $ds \leftarrow ds + 1$
11. 将 dm 和 ds 更新到适应度评价指标中
12. 输出调度分配结果

3.2 螭蟾个体更新

D-VNMA 通过雄性螭蟾和雌性螭蟾的运动实现螭蟾个体更新。假设 x_i^t 为时间步长 t 下搜索空间中雄性螭蟾 i 的位置,螭蟾的位置通过速度 v_i^t 进行移动, x_{ij}^t 和 v_{ij}^t 分别对应于维度 j 下的位置和速度,则雄性螭蟾的运动计算如下:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (13)$$

$$v_{ij}^{t+1} = g * v_{ij}^t + a_1 e^{-\beta r_p^2} (pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2} (gbest_{ij} - x_{ij}^t) \quad (14)$$

$$v_{ij}^{t+1} = v_{ij}^t + d * r \quad (15)$$

其中, a_1, a_2 为正向吸引常数;重力系数 g 近似于粒子群算法的惯性权值,可平衡探查和开发之间的冲突,其取值在迭代过程中会逐渐减小,保证算法可以挖掘出一些特定的区域; $pbest_i$ 是螭蟾 i 探索过的最好的位置; $gbest$ 是全局最佳位置; r_p, r_g 表示螭蟾 i 和 $pbest_i$ 及 $gbest$ 的笛卡尔积; β 为固定系数; d 表示婚礼舞蹈系数; r 为 $[-1, 1]$ 区间内的随机数。普通雄性螭蟾通过式(14)进行聚集;而群体中最优秀的雄性螭蟾则通过式(15)执行它们特有的上下婚礼舞蹈,以此来不断改变其运动速度。

与雄性不同,雌性螭蟾为了繁殖,会飞向雄性。设 y_i^t 为在时间步长 t 下搜索空间中雌性螭蟾 i 的位置,其位置通过速度 v_i^t 来改变,对应的雌性螭蟾维度 j 的位置和速度用 y_{ij}^t 和 v_{ij}^t 表示。雌性螭蟾的运动计算可表示如下:

$$y_i^{t+1} = y_i^t + v_i^{t+1} \quad (16)$$

$$v_{ij}^{t+1} = \begin{cases} g * v_{ij}^t + a_3 e^{-\beta r_{mf}} (x_{ij}^t - y_{ij}^t), & f(y_i^t) > f(x_i^t) \\ g * v_{ij}^t + fl * r, & f(y_i^t) \leq f(x_i^t) \end{cases} \quad (17)$$

其中, a_3 表示雌性螭蟾和雄性螭蟾相互吸引的常数; r_{mf} 表示雌雄螭蟾的笛卡尔距离; fl 是随机游走系数,当雌性不被雄性吸引时,雌性随机游走; r 是随机值。

3.3 螭蟾初始化

螭蟾优化算法中,随机生成的初始解分布无规律,可能导致初始解过于密集,搜索空间覆盖率低,搜索效率低,或者容易

陷入局部最优解。为解决该问题,本文使用一种比较稳定且混沌值的覆盖率较高的 Circle 映射,以增强种群的多样性,提升算法的寻优能力。使用 Circle 映射初始化雌性螭蟾和雄性螭蟾个体,用 ω_j 表示螭蟾在位置 j 的混沌映射值,其中 ω_0 为 $[0, 1]$ 范围内的随机数。以雄性螭蟾 i 的初始位置 x_{ij}^0 的计算为例:

$$x_{ij}^0 = (\omega_j - \lfloor \omega_j \rfloor) * M \quad (18)$$

$$\omega_{j+1} = \text{mod} \left(3.85\omega_j + 0.4 - \left(\frac{0.7}{3.85\pi} \right) \sin(3.85\pi * \omega_j), 1 \right) \quad (19)$$

鉴于使用启发式规则,在部分场景下可以得到最优解或者局部最优解,本文在算法中加入最短加工时间优先规则、最小负荷最早完成规则等常见的启发式规则^[19]作为初始解,以减少算法迭代中对局部最优的探索耗时。

3.4 邻域搜索策略

由于不同的邻域结构对求解问题的收敛效果差异显著, Diana^[20] 提出在使用邻域搜索时可设计多种不同的邻域结构和搜索策略,然后通过不同的组合方式找到最佳的收敛方案。组合的搜索策略可选择最佳改进(Best Improvement, BI)和首次改进(First Improvement, FI)两种。组合的邻域结构^[21] 可选择下述 5 种将准备时长考虑在内的结构。

$TSWS(i, j, k)$: 同工位位置交换操作, 交换工位 k 上第 i 个加工单元和第 j 个加工单元的位置 ($j > i$);

$TISS(i, j, k)$: 同工位位置移动操作, 将工位 k 上第 i 个加工单元放到第 j 个加工单元之后 ($j > i + 1$);

$TIVS(i, j, k)$: 同工位逆序操作, 对工位 k 上从第 i 个加工单元到第 j 个加工单元进行逆序操作 ($j > i + 1$);

$TSWD(i, j, k_1, k_2)$: 不同工位位置交换操作, 交换工位 k_1 上第 i 个加工单元和工位 k_2 上第 j 个加工单元的位置;

$TISD(i, j, k_1, k_2)$: 不同工位位置移动操作, 将工位 k_1 上第 i 个加工单元放在工位 k_2 上第 j 个加工单元之后。

假设有 N 个加工单元在 M 个互不相关的工位上加工, 且满足下述条件:

- 1) 每个调整的加工单元彼此不存在紧前紧后关系;
- 2) 每个加工单元的准备时长和其前面的加工单元有关, 也和其所在加工工位有关;
- 3) 每个工位上只能同时加工一个加工单元, 而且每个加工单元也只能选择一个工位加工, 加工单元在不同工位上的加工时长也不尽相同。

以最小化 C_{\max} 为指标判断 5 种邻域搜索有效性的充要条件如下(符号说明详见表 1):

1) 若 $j = i + 1$, 当且仅当 $R_k > R_k' (\forall k' \neq k)$ 时, 邻域搜索 $TSWS(i, j, k)$ 是有效的。

$$ST_{i-1, j, k} + ST_{j, i, k} + ST_{i, j+1, k} < ST_{i-1, i, k} + ST_{i, j, k} + ST_{j, j+1, k} \quad (20)$$

2) 若 $j > i + 1$, 当且仅当 $R_k > R_k' (\forall k' \neq k)$, 时, 邻域搜索 $TSWS(i, j, k)$ 是有效的。

$$ST_{i-1, j, k} + ST_{j, i+1, k} + ST_{j-1, i, k} + ST_{i, j+1, k} < ST_{i-1, i, k} + ST_{i, i+1, k} + ST_{j-1, j, k} + ST_{j, j+1, k} \quad (21)$$

3) 当且仅当 $R_k > R_k' (\forall k' \neq k)$ 时, 邻域搜索 $TISS(i, j, k)$ 是有效的。

$$ST_{i-1,i+1,k} + ST_{j,i,k} + ST_{i,j+1,k} < ST_{i-1,i,k} + ST_{i,i+1,k} + ST_{j,j+1,k} \quad (22)$$

4) 当且仅当 $R_k > R_k' (\forall k' \neq k)$ 时, 邻域搜索 $TIVS(i, j, k)$ 是有效的。

$$ST_{i-1,j,k} + ST_{i,j+1,k} + \sum_{a=i}^{j-1} ST_{a+1,a,k} < ST_{i-1,i,k} + ST_{j,j+1,k} + \sum_{a=i}^{j-1} ST_{a,a+1,k} \quad (23)$$

5) 当且仅当 $R_{k_1} = C_{\max}, R_k' < C_{\max} (\forall k' \neq k_1, k_2)$ 时, 邻域搜索 $TSWD(i, j, k_1, k_2)$ 是有效的。

$$AP_{i-1,i,j_1} + ST_{j,i+1,i_1} < AP_{i-1,i,i_1} + ST_{i,i+1,i_1} \quad (24)$$

$$R_{k_2} + AP_{j-1,i,k_2} + ST_{i,j+1,k_2} < R_1 + AP_{j-1,j,k_2} + ST_{j,j+1,k_2} \quad (25)$$

6) 当且仅当 $R_1 = C_{\max}, R_k' < C_{\max} (\forall k' \neq k_1, k_2)$ 时, 邻域搜索 $TISD(i, j, k_1, k_2)$ 是有效的。

$$ST_{i-1,i+1,i_1} < ST_{i,i+1,i_1} + AP_{i-1,i,i_1} \quad (26)$$

$$R_{k_2} + AP_{j,i,k_2} + ST_{i,j+1,k_2} < R_1 + ST_{j,j+1,k_2} \quad (27)$$

表 1 符号及定义

Table 1 Symbol definition

符号	定义
$ST_{i,j,k}$	工位 k 上加工单元 j 紧跟在加工单元 i 后加工的准备时长
$t_{i,k}$	工位 k 上第 i 个工件的加工时长
$AP_{i,j,k}$	工位 k 上第 j 个工件紧邻第 i 个工件之后加工的总时间
k^*	调度解的关键工位
R_k	工位 k 的释放时间, 即工位 k 上加工单元的完工时间
C_{\max}	最大完工时间
g_{il}	工件 i 的第 l 道工序
$S_{i,l}$	工件 i 的第 l 道工序的开工时间
$E_{i,l}$	工件 i 的第 l 道工序的完工时间
$E'_{i,l}$	工件 i 的第 l 道工序顺序调整后的完工时间
$A_{i,l}$	工件 i 的第 l 道工序不影响其紧后工序的可活动空间
$SE_{i,l}$	工件 i 的第 l 道工序允许的最早开工时间

当考虑加工单元存在多个顺序依赖关系时, 上述 6 个充要条件将不一定成立。本文提出一种前置约束条件, 将上述公式转为必要条件, 以提高邻域搜索效率。假设调整加工序列顺序时, 工位 k 上工件 i 的第 l 道工序完工时间延后。

如果工件 i 第 l 道工序后不存在紧后工序, 则式(20)一式(27)的 6 个充要条件仍然成立。

如果工件 i 第 l 道工序后存在紧后工序, 由于工件 i 的紧后工序开工不得早于其紧前工序, 即:

$$S_{i,l_1} < S_{i,l_2} (\forall l_1 < l_2) \quad (28)$$

故其紧后工序的完工时间均将延后, 即:

$$E_{i,l'} < E'_{i,l'} (\forall l' \geq l) \quad (29)$$

同样地, 工位 k 上加工顺序在工件 i 之后的其他工件的完工时间也将跟着滞后, 因此邻域搜索策略需要重新调整。

工件 i 第 l 道工序允许往前移动而不影响其他工件完工时间的可调整活动空间 $A_{i,l}$ 的计算如下:

$$A_{i,l} = S_{i,l} - ST_{i,i+1,k} - E_{i,l-1} \quad (30)$$

工位 k 上, 紧跟在工件 $i-1$ 第 l_{i-1} 道工序后加工的工件 i 第 l 道工序允许的最早开工时间 $SE_{i,l}$ 的计算如下:

$$SE_{i,l} = \max\{E_{i-1,l'} + ST_{i-1,i,k} E_{i,l-1}\} \quad (31)$$

1) 工件 i 第 l 道工序可尝试执行 $TSWS(i, j, k)$ 操作, 在满足式(20)或式(21)的前提下, 还需要满足如下条件:

$$E_{i-1,t_{i-1}} + ST_{i-1,i,k} < SE_{i,l} \quad (32)$$

$$SE_{j,t_j} < SE_{i,l} \quad (33)$$

2) 工件 i 第 l 道工序可尝试执行 $TIVS(i, j, k)$ 操作, 在满足式(22)的前提下, 还需满足如下条件 ($\forall j > i$):

$$A_{i,l} > 0 \quad (34)$$

$$SE_{j,t_j} < S_{i+1,t_{i+1}} \quad (35)$$

$$E_{i,l} + ST_{i,j,k} < S_{i+1,t_{i+1}} \quad (36)$$

3) 工件 i 第 l 道工序可尝试执行 $TSWD(i, j, k_1, k_2)$ 操作, 在同时满足式(24)和式(25)的前提下, 还需要满足如下条件:

$$E_{i-1,t_{i-1}} + ST_{i-1,i_1} < SE_{i,l} \quad (37)$$

$$SE_{j,t_j} < SE_{i,l} \quad (38)$$

4) 工件 i 第 l 道工序可尝试执行 $TISD(i, j, k_1, k_2)$, 在同时满足式(26)和式(27)的前提下, 还需要满足如下条件 ($\forall j > i$)。

$$A_{i,l} > 0 \quad (39)$$

$$S_{i,l} < S_{j,t_j} \quad (40)$$

$$SE_{j,t_j} < S_{i,l} \quad (41)$$

$$E_{i,l} + ST_{i,j_1} < S_{i+1,t_{i+1}} \quad (42)$$

4 仿真实验与分析

为验证 D-VNMA 算法的性能, 将对算法参数和仿真测试进行充分讨论与分析。所涉及的算法均采用 Java 语言编写, 机器运行环境为 Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz, 16 GB 内存。测试的数据来源于 Monaldo 测试集, 从中选取 Mk01—Mk10^[22] 算例对算法进行验证测试。

4.1 参数设置

本节探讨 D-VNMA 中参数对其性能的影响时, 依托 Mk01 算例, 以最小化最大完工时间为目标, 采用 DOE 方法^[23]。选取 4 个算法重要参数作为因子, 每个参数均取得 4 个水平, 各个水平的取值详见表 2。根据参数和水平的数量选择规模 $L_{16}(4^4)$, 共 16 组实验。算法在每种参数组合下各自独立运行 20 次, 将所得到的最大完工时间取平均值作为正交实验结果分析的响应值 (Response Value, RV)。根据 RV 统计各个参数的极差和其重要程度 (见表 3), 进而绘制参数对算法性能影响的趋势图 (见图 2)。

表 2 各参数水平

Table 2 Parameter level

Parameter	Level			
	1	2	3	4
α_1	1.20	1.30	1.50	1.60
α_2	0.50	0.80	1.00	1.20
α_3	0.50	1.00	2.00	2.50
β	0.05	0.10	0.50	1.00

表 3 正交实验结果

Table 3 Orthogonal experimental results

Level	α_1	α_2	α_3	β
1	45.900	44.625	45.650	48.325
2	45.525	45.725	45.600	47.725
3	45.975	46.125	45.850	44.325
4	44.975	45.900	45.275	42.000
worst	1.000	1.500	0.575	6.325
grade	3	2	4	1

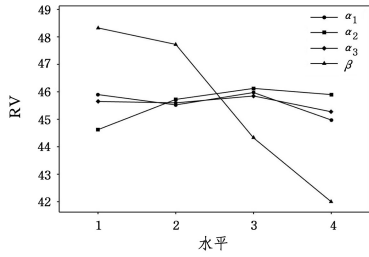


图2 参数正交结果分析图

Fig. 2 Analysis diagram of parameter orthogonal results

经分析可知, β 对算法收敛的影响程度最大, 不宜过小; a_2 在水平 1 时效果最佳, 在水平 2、水平 3、水平 4 的效果相近; 参数 a_1 和 a_3 对算法收敛的影响程度最低, 算法中直接取水平 4。最终算法参数设置为: $a_1 = 1.6$, $a_2 = 0.5$, $a_3 = 2.5$, $\beta = 1$ 。

4.2 算法对比与分析

本节将所提算法与其他算法在 Mk01—Mk10 实例上的运行结果进行对比分析。选择这些实例主要是因为它们代表了不同规模和难易程度的调度问题场景, 有助于评估算法的通用性和鲁棒性。为了保证算法结果的可靠性, 将算法种群规模和迭代次数均设为 200 和 1000。这些参数的选择是基于先前的实验研究和经验, 也是为了确保算法能够在合理的时间内收敛到最优解。同时, 通过进行多次实验并计算平均值和标准差来评估算法的稳定性和一致性, 以进一步验证算法的有效性。

为全面评估 D-VNMA 算法的性能, 选取文献[24]的 DP-SO 算法、文献[25]的 GA-PSO 算法、文献[26]的 FA 算法分别进行 20 次运算, 最终选择最优解进行比较。如表 4 所列, 在 10 个标准算例中, “*” 表示最优解。

表4 D-VNMA 和其他算法运行结果的对比

Table 4 Running results comparison between D-VNMA and other algorithms

算例	$M \times N$	DPSO	GA-PSO	FA	D-VNMA
MK01	10×6	42*	42*	42*	42*
MK02	10×6	26*	26*	32	26*
MK03	15×8	207	204	211	202*
MK04	15×8	65	60*	81	60*
MK05	15×4	171*	173	186	180
MK06	10×15	61*	63	86	61*
MK07	20×5	173	139*	157	139*
MK08	20×10	523*	523*	523*	523*
MK09	20×10	307*	311	369	307*
MK10	20×15	312	212*	296	212*

从结果可以看出, D-VNMA 在 MK03 算例中的表现显著优于 DPSO, GA-PSO 和 FA; 且在 MK06—MK10 这几个规模较大的算例中均能稳定获得最优解, 在 MK05 算例上的表现虽然明显劣于 DPSO, 但总体上有 90% 的概率能得到最优解, 其结果准确率高。在 MK01 算例中, 4 个算法的结果没有优劣之差, 但根据对图 3 的收敛曲线进行分析, D-VNMA 的收敛效率较 DPSO, FA, GA-PSO 分别提升了 47.44%, 62.15%, 58.72%。因此, D-VNMA 较其他 3 个算法具有更快的收敛速度和更高的求解精度。研究结果表明, D-VNMA 在解决 AJSSP 方面表现出色, 并具有良好的性能和效果。该

算法可以为实际生产调度问题提供有力的支持和帮助。

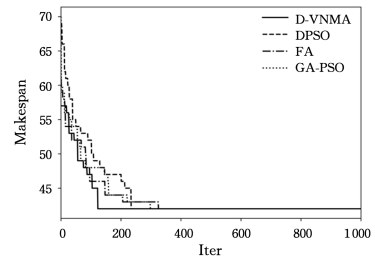


图3 Mk01 算例的 1000 次迭代收敛过程的对比

Fig. 3 Comparison of 1000 iterations convergence process of Mk01 example

4.3 车间应用

图 4 给出了算法在某一个装配车间的应用示例。图中加工时间为 13 的刻度线上有一条长实线, 该实线表示当前时刻, 实线左侧为来自 MES 系统的工位上已过站的实时条码数据, 实线右侧为已开工待完成的加工单元剩余加工时长和已排待加工的加工单元加工时长。图中左侧两条相邻的已过站条码数据之间的空白时段体现了当前所在工位的实时生产节拍, 如图中第 1 个工位, 其中间有个节拍时长为 3, 该节拍数据和前面的节拍数据分布差异极大, 故可以判定当前工位可能存在异常, 当前状态可以直接实时直观地反馈给车间人员, 若车间人员确认反馈工位存在异常, 则后台算法也能及时根据产线情况重新调整调度方案。

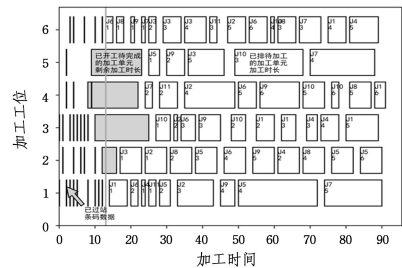


图4 与生产数据联动的实时甘特图

Fig. 4 Real-time Gantt chart linked to production data

结束语 本文针对 AJSSP 构建了一个融入了工序节拍和工位释放时间等符合真实场景约束的数学模型, 该模型更易于实现 MES 系统中实时采集的生产数据在生产调度中的价值转换; 同时还对 MA 算法进行改进, 提出一种可用于高效求解 AJSSP 的 D-VNMA 算法。该算法的编码解码策略在随机数编码基础上进行改进, 使其编码能在进化过程中将上一代的优点通过迭代传承给下一代种群, 不但能保证群体的多样性, 而且提高了算法性能, 使得下一代能更加快速地逼近最优解。在种群初始过程中, 将 Circle 映射和部分常见启发式算法进行融合, 既保证了初始种群的多样性, 又减少了在局部最优空间的探索消耗。针对问题特性进行设计的邻域搜索策略, 很大程度上提高了邻域搜索效率。最后, 基于标准算例库的算法验证结果表明, 使用同样的种群规模, D-VNMA 较其他算法能更快地收敛到最优解, 且在不同规模的测试集下, 其结果稳定, 鲁棒性高。未来的工作将在工序节拍数据预测和算法多场景适应等方面展开, 以推动 AJSSP 问题研究进一步向前发展。

参 考 文 献

- [1] HUANG X, YANG H, WEI J. Shifting Bottleneck Algorithm Based on Filtered Beam Search for JSSP[J]. Computer Science, 2009, 36(4): 254-256, 284.
- [2] GUO P, ZHAO W C, LEI K. Dual-resource Constrained Flexible Job Shop Optimal Scheduling Based on An Improved Jaya Algorithm[J]. Journal of Jilin University(Engineering and Technology Edition), 2023, 53(2): 480-487.
- [3] KOMAKI G M, SHEIKH S, MALAKOOTTI B. Flow Shop Scheduling Problems with Assembly Operations: A Review and New Trends[J]. International Journal of Production Research, 2019, 57(10): 2926-2955.
- [4] YANG Y, LI X. A Knowledge-Driven Constructive Heuristic Algorithm for the Distributed Assembly Blocking Flow Shop Scheduling Problem [J]. Expert Systems with Applications, 2022, 202: 117269.
- [5] JIANG T, LIU L, ZHU H, et al. An Improved Elephant Herding Optimization for Energy-Saving Assembly Job Shop Scheduling Problem with Transportation Times[J]. Axioms, 2022, 11(10): 561.
- [6] REN W, WEN J, YAN Y, et al. Multi-Objective Optimisation for Energy-aware Flexible Job-shop Scheduling Problem with Assembly Operations[J]. International Journal of Production Research, 2021, 59(23): 7216-7231.
- [7] WANG Z Y, LU C. An Integrated Job Shop Scheduling and Assembly Sequence Planning Approach for Discrete Manufacturing [J]. Journal of Manufacturing Systems, 2021, 61: 27-44.
- [8] HAJIBABAEI M, BEHNAMIAN J. Fuzzy Cleaner Production in Assembly Flexible Job-Shop Scheduling with Machine Breakdown and Batch Transportation: Lagrangian Relaxation [J]. Journal of Combinatorial Optimization, 2023, 45(5): 112.
- [9] JOHNSON D, CHEN G, LU Y. Multi-Agent Reinforcement Learning for Real-Time Dynamic Production Scheduling in A Robot Assembly Cell[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 7684-7691.
- [10] MIAO K, LI C. Optimization Algorithms for Job Shop Scheduling Problems Based on Correction Mechanisms and Reinforcement Learning[J]. Computer Science, 2023, 50(6): 274-282.
- [11] SUKKERD W, WUTTIPORN PUN T, LATTHAWANICH-PHAN J, et al. A New Improvement of the NEH Heuristic to Either Minimise Total Tardiness or Makespan for a Hybrid Flow Shop with Assembly Operations[C]//Proceedings of the International Conference on Industrial Engineering and Applications. Chengdu: IEEE, 2021: 315-320.
- [12] HALIM A H, YUSRISKI R. Batch Scheduling for the Two-stage Assembly Model to Minimize Total Actual Flow Time [C]//Proceedings of the Asia Pacific Industrial Engineering and Management Systems. Kitakyushu: APIEM, 2022: 1742-1748.
- [13] LI X, LU J, YANG C, et al. Research of Flexible Assembly Job-Shop Batch-Scheduling Problem Based on Improved Artificial Bee Colony[J]. Frontiers in Bioengineering and Biotechnology, 2022, 10: 909548.
- [14] ZERVOUDAKIS K, TSAFARAKIS S. A Mayfly Optimization-Algorithm[J]. Computers & Industrial Engineering, 2020, 145: 106559.
- [15] EBERHART R, KENNEDY J. A New Optimizer Using Particle Swarm Theory[C]//Proceedings of the International Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995: 39-43.
- [16] GREFENSTETTE J J. Genetic Algorithms and Machine Learning[J]. Machine Learning, 1988, 3(2): 95-99.
- [17] YANG X S. Firefly Algorithms for Multimodal Optimization [C]//Proceedings of the International Symposium on Stochastic Algorithms. Berlin: Springer Berlin Heidelberg, 2009: 169-178.
- [18] WHITLEY D, GORDON V S, MATHIAS K. Lamarckian Evolution, the Baldwin Effect and Function Optimization[C]//Proceedings of the International Conference on Parallel Problem Solving from Nature. Israel: Springer Berlin Heidelberg, 1994: 5-15.
- [19] OW P S, MORTON T E. The Single Machine Early / Tardy Problem[J]. Management Science, 1989, 35(2): 177-191.
- [20] DIANA R, SOUZA S. Analysis of Variable Neighborhood Descent as a Local Search Operator for Total Weighted Tardiness Problem on Unrelated Parallel Machines[J]. Computers & Operations Research, 2020, 117: 104886.
- [21] WANG L, WANG S, ZHENG X. A Hybrid Estimation of Distribution Algorithm for Unrelated Parallel Machine Scheduling with Sequence-dependent Setup Times[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(3): 235-246.
- [22] YI Z, GONG M, ZENG J, et al. Hybrid Multi-objective Algorithm for Solving Flexible Job Shop Scheduling Problem[J]. Computer Science, 2015, 42(9): 220-225.
- [23] JANKOVIC A, CHAUDHARY G, GOIA F. Designing the Design of Experiments(DOE) — An Investigation on the Influence of Different Factorial Designs on the Characterization of Complex Systems[J]. Energy and Buildings, 2021, 250: 111298.
- [24] NOURI M, BEKRAR A, JEMAI A, et al. An Effective and Distributed Particle Swarm Optimization Algorithm for Flexible Job-shop Scheduling Problem[J]. Journal of Intelligent Manufacturing, 2018, 29: 603-615.
- [25] KHRAIBET T J, GHAFIL W K. Using Hybrid GA-PSO Algorithm to Solve Problem in Machine Scheduling[J]. Journal of Discrete Mathematical Sciences and Cryptography, 2021, 24(7): 2027-2035.
- [26] RASHID M, OSMAN M. Optimisation of Energy Efficient Hybrid Flowshop Scheduling Problem Using Firefly Algorithm [C]//Proceedings of the Symposium on Computer Applications & Industrial Electronics. Malaysia: IEEE, 2020: 36-41.



CHEN Yali, born in 1989, postgraduate, is a member of CCF(No. P7071G). Her main research interests include job shop scheduling algorithm and so on.



LIU Genggen, born in 1988, Ph.D, professor, Ph. D supervisor, is a senior member of CCF(No. 75198S). His main research interests include EDA algorithm, and computational intelligence and its application.