



面向人机协同能效车间调度的群智能优化算法

王凌*, 李瑞, 陈靖方

清华大学自动化系, 北京 100084

* E-mail: wangling@mail.tsinghua.edu.cn

收稿日期: 2023-10-27; 接受日期: 2024-01-05; 网络版发表日期: 2024-09-05

国家重点研发计划(编号: 2023YFB3308002)和国家自然科学基金(批准号: 62273193)资助项目

摘要 随着制造企业的智能化,“人-机-物”全面互联融合的新发展模式成为新一代智能制造的方向。同时,工人与机器人协同制造已普遍存在于现代制造车间,相较于传统的人机独立生产模式增加了人机资源配置的问题,使得调度问题更加复杂,对算法设计带来了新的挑战。因此,本文考虑人机协同制造的特点建立了人机协同能效车间调度问题,以最小化最大完工时间和总能耗为优化目标,提出了一种群智能协同优化算法求解。该算法包含如下特点:首先,提出了一种双阶段协同搜索框架,即第一阶段以群智能优化实现快速收敛并构建精英存档,第二阶段以反馈协同搜索在精英解附近进行多方位勘探增加分布性。其次,针对人机资源配置问题提出了多层协同搜索算子实现快速收敛。最后,针对问题特性,提炼问题知识,设计多策略协同初始化方法提供高质量初始种群,同时提出多操作协同局部搜索高效优化两个目标。为了验证提出算法的有效性,本文生成了15个不同规模的案例,并进行了参数、消融和对比仿真实验。结果表明,本文提出了多种的协同方法能有效提升算法性能和搜索质量,通过对比最新相关算法验证了所提算法的有效性,为人机协同制造的调度排产在方法上提供了可靠的指导。

关键词 智能制造, 人机协同制造, 能效调度, 协同优化

1 引言

随着物联网、大数据、云计算、人工智能技术的快速发展,我国提出了“制造强国”的发展目标。在“智能制造发展规划(2016~2020年)”、“中国制造2025”和“互联网+制造”等的指导下,我国制造业加速数字化、信息化、智能化,实现“人-机-物”的全面互联融合^[1~3]。现如今,在装备制造、车辆制造、船舶制造的焊接场景中工人和机器人协同制造已成为普遍现状^[4]。相较

于工人和机器独立生产的传统车间调度场景,人机协同制造中工件可选择工人和机器人两种加工资源,这提升了生产柔性,但同时也增加了问题的复杂性。在该问题中,工人具备多种技能等级,机器人也可被视为具有特殊技能的工人,每种等级的加工资源的加工时间和能耗功率都不同。此外,一个工件的工序可被多个工人或机器人同时加工以缩短完工时间,但使用资源数量越多,能耗也越高。因此,工序需要合理地选择加工资源类型和数量以实现人机高效协同制造,上

引用格式: 王凌, 李瑞, 陈靖方. 面向人机协同能效车间调度的群智能优化算法. 中国科学: 技术科学, 2024, 54: 1676-1692

Wang L, Li R, Chen J F. A swarm intelligence optimization algorithm for human-robot collaborative energy-efficient shop scheduling (in Chinese). Sci Sin Tech, 2024, 54: 1676-1692, doi: 10.1360/SST-2023-0341

述多人多机协同的复杂性对算法设计带来了新的挑战。

目前国内外关于多人多机协同车间调度问题的研究很少, 主要研究集中于人机协同装配线调度^[5-7]。该问题属于单人单机^[5]或单人多机^[8]协同制造, 目标是合理地将工件分配在工人和机器人的产线使完工时间最小, 其中包含一条或多条机器人产线。相关工作如下: Mokhtarzadeh等人^[9]将电路印刷版车间的装配产线抽象为人机协同调度并设计了约束规划(constraint programming, CP)方法进行求解, 并在短时间内获得了比混合整数线性规划(mixed-integer linear programming, MILP)方法更好的解。Faccio等人^[10]认为在产线装配时应该考虑人和机器的安全距离以防止安全事故的发生, 并对此问题建立MILP模型。Ferreira等人^[8]则提出了遗传算法对人机协同装配线调度问题求解, 实验结果表明算法在性能上优于CP和MILP。Ren等人^[11]将启发式规则与粒子群优化算法, 相较于其他进化算法取得了更优的结果。Vieira等人^[6]提出了一种双层优化模型, 上层产生调度序列, 下层通过递归仿真的方式搜索最优解并反馈给上层调整调度序列, 在有限时间内取得了优质解。Yu等人^[7]将装配调度建模为棋盘问题, 并用深度Q网络算法优化棋盘走向最终输出最优调度方案。Casalino等人^[12]构建Petri网络对产线装配序列进行建模, 实时输出工人和机器人的资源分配结果。Alirezazadeh和Alexandre^[13]将人机资源的动态分配构建为有向图, 通过图的结果调整输出排产方案。

多人多机协同制造的问题可分为人机协同流水线调度^[4,14,15]和人机协同柔性车间调度^[16-18]。关于人机协同流水线调度, Wang等人^[15]在人机协同装配线调度上扩展, 并提出了人机协同流水线调度问题, 采用一种近似方法求解。在此基础上Vahedi-Nouri等人^[14]根据车间中的再制造特性提出了新的调度模型, 并采用MILP和CP进行求解。随后, Lu等人^[4]将焊接流水车间的场景抽象为人机协同能效流水车间调度, 并提出了基于Pareto支配的模因算法, 对比相关的多目标优化算法取得了更优的结果。

关于人机协同柔性车间调度, Guo^[16]认为飞机装配场景中工人的飞机装配和机器人的零件配送构成协同制造, 并构建了MILP和CP模型, 实验结果表明CP模型在MILP模型的基础上减少13%的完工时间。Zhang等人^[17]将车间内的工人分为不同的技能等级, 建立了

人机协同柔性车间调度问题, 并提出分层强化学习算法, 对比决策树等机器学习方法以更少的时间取得更优的结果。Cai等人^[18]使用NSGA-II算法求解装配场景下的人机协同柔性车间调度问题, 取得了良好的效果。Maderna等人^[19]提出了决策树算法对调度序列的资源分配进行决策, 实验结果表明所提方法比随机策略减少了15%的生产周期。

目前国内外对人机协同调度问题的研究集中于轻工业场景下的单人单机或单人多机协同调度, 但在装备制造、车辆制造、船舶制造等重工业制造生产场景中主要为多人多机协同调度, 例如焊接车间中的焊工和焊接机器人的协同制造。面对此类复杂的人机协同特性, 目前鲜有相关研究。在方法层面, 尽管针对多人多机协同调度问题已存在初步研究^[6], 但求解性能有待提升。因此, 设计高效的调度算法对指导现实人机协同生产具有重要意义。

此外, 装备制造等重工业车间生产时需要消耗大量能源。但当今世界, 随着全球变暖问题日益严重和极端天气频发, 可持续的绿色制造成为现代制造业的核心主题^[20]。随着不可再生能源的过度开采, 能源价格骤增, 节能成为绿色制造的关键内容^[21]。然而, 仅依靠设备优化和工艺优化实现节能增效的成本高且提升幅度有限。而高效的调度优化方法成本低, 同时能协同优化企业的经济和能耗指标, 实现节能和增效^[22]。因此, 能效车间调度受到国内外学者的广泛关注。

能效车间调度的研究主要面向下列车间类型: 并行机调度^[23]、流水线调度^[24,25]、作业车间调度^[26]和柔性车间调度^[27-30]。针对单机调度问题, Liu等人^[23]提出了一种优化完工时间和能耗的数学模型, 并采用了NSGA-II算法进行求解。针对流水线调度, Ramezani等人^[24]构建了一种带工序相关准备时间的能效流水车间调度问题, 同时提出了一种启发式的贪心迭代算法, 取得了良好的结果。Wang等人^[25]提出一种带异构工厂约束的混合流水能效车间调度问题, 并设计学习驱动的模因算法求解, 取得了更优的结果。针对绿色作业车间, Masmoudi等人^[26]提出了MILP模型并采用遗传算法求解。针对能效柔性作业车间调度问题, Li等人^[30]采用二型三角模糊数表征不确定加工时间, 提出贪心节能策略有效降低能耗, 并提出一种结合Q学习的模因算法, 相较于最新算法取得最优结果。针对分布式同构能效柔性作业车间调度, 李瑞等人^[27,28]设

计了全主动调度节能策略, 提出了知识驱动的模式因算法和基于意外流行指标的自适应模式因算法, 相较于对比算法取得了显著性的优势. Li等人^[29]进一步考虑了分布式异构工厂特性, 将进化算法与深度强化学习结合实现高效优化. 由于在能效车间调度中, 绿色指标和经济指标存在冲突, 因此如何有效平衡两个目标成为关键问题, 这使得能效车间调度比传统车间调度更复杂, 尤其在人机协同的调度问题中, 大量工人和机器人的使用使得生产过程消耗大量电力, 因此, 能耗指标的优化尤为关键. 如何平衡人机协同车间调度中能耗与经济指标, 给算法设计带来了新的挑战.

综上所述, 本文面向重工业制造的焊接车间场景, 结合多人多机协同的特性, 将实际制造场景提炼为人机协同能效车间调度问题(human-robot collaborative energy-efficient shop scheduling, HRCES). 为了求解该复杂问题, 本文提出一种双阶段协同反馈优化方法(two phase cooperative feedback optimization, TPCFO), 通过多阶段协同、多策略协同初始化、多层协同搜索、多操作协同局部搜索, 对HRCES高效求解, 获得综合性能良好的Pareto调度解集, 协同优化最大完工时间和总能耗指标.

2 人机协同能效车间调度问题

2.1 问题表征

在人机协同能效车间调度问题中, 总共有 n 个工件, 所有工件需要以相同的顺序经过 n_s 个工序加工, 每个工件的每道工序在加工时可选择 n_o 个初等工, n_k 个高级工或 n_r 个机器人. 工序分为简单和复杂工序. 初等工和机器人只能加工简单工序, 高级工可以加工所有类型工序. 针对一项简单工序, 三者的原始加工时间关系为 $p_o > p_k > p_r$. 在该问题中, 工件 i 的第 j 个工序 O_{ij} 可同时被多个工人或多个机器人加工, 实际加工时间为所选资源类型的原始加工时间除以资源数量. 使用资源数量越多, 实际加工时间越小, 完工时间越早, 但产生的能耗越高, 目标之间存在明显冲突. 因此, 该问题可以被建模为多目标优化问题.

人机协同能效车间调度问题可以分解为三个子问题: (1) 确定所有工件的加工顺序; (2) 为每个工件的每道工序选择资源类型, 即初等工、高级工或机器人; (3) 为每个工件的每道工序选择合理的资源数量. 使

得车间最大完工时间和总生产能耗同时最小.

该问题中存在如下约束: (1) 所有工件的加工顺序与第一道工序的顺序相同; (2) 每个工件的当前工序需要等待其上一道工序和同工序的前置工件完工; (3) 当前工序选择工人或机器人加工后, 同道工序的其他工件的人机资源类型需要保持一致(如果某道工序选择工人加工, 则该道工序下的不同工件可进一步选择初等工或高级工加工); (4) 每个工件的工序只能被一种资源类型加工, 但一个工件可同时被多个相同类型资源加工; (5) 初等工和机器人只能加工简单工序; (6) 每个工人或机器人同一时刻只能加工一个工件.

该问题还有如下假设: (1) 所有工件的加工时间是确定值; (2) 动态事件(如机器损坏和工件插入)不会发生; (3) 机器人加工时间最少, 但功率最高. 高级工加工时间比初等工少, 使用的设备相同, 所以加工功率相同.

2.2 计算模型

为了进一步描述问题模型, 本节介绍最大完工时间和总能耗的计算公式如下:

$$\min\{f_1, f_2\}, \quad (1)$$

$$f_1 = C_{\max} = \max\{C_{i,j}\}, \forall i \in I, j \in J_i, \quad (2)$$

$$f_2 = \text{TEC} = E_W + E_I, \quad (3)$$

其中, 式(1)表示多目标优化模型; 式(2)表示优化最大完工时间 C_{\max} , $C_{i,j}$ 表示工件 i 的工序 j 的完工时间, $I = \{1, 2, \dots, n\}$ 表示工件集合, $J_i = \{1, 2, \dots, n_s\}$ 表示工件 i 的工序集合; 式(3)表示优化总能耗(total energy consumption, TEC), E_W 和 E_I 分别为总加工能耗和总空闲能耗.

最大完工时间 C_{\max} 的推算公式如下:

$$C_{\pi_1,1} = t_{\pi_1,1}, \quad (4)$$

$$C_{\pi_k,1} = C_{\pi_{k-1},1} + t_{\pi_k,1}, k = 2, \dots, n, \quad (5)$$

$$C_{\pi_1,j} = C_{\pi_1,j-1} + t_{\pi_1,j}, j = 2, \dots, n_s, \quad (6)$$

$$C_{\pi_k,j} = \max\{C_{\pi_{k-1},j}, C_{\pi_k,j-1}\} + t_{\pi_k,j}, \quad (7)$$

$$j = 2, \dots, n_s, k = 2, \dots, n,$$

$$t_{\pi_k,j} = \sum_{h=1}^3 \sum_{q=1}^3 \frac{p_{\pi_k,j,h} \times x_{\pi_k,j,h} \times y_{j,q}}{u_{\pi_k,j}}, \pi_k, k \in I, j \in J_i, \quad (8)$$

其中, 式(4)表示加工顺序 π 中第一个工件的第一道工

序的完工时间为实际加工时间; 式(5)表示第一道工序的后续工件的完工时间为前序工件完工时间与自身加工时间之和; 式(6)为 π 中第一个工件的后续工序的完工时间; 式(7)表示当前工序的完工时间为同道工序的前置工件的完工时间与同工件前置工序的完工时间中最大值与自身加工时间之和; 式(8)表示工序 $O_{\pi_{k,j}}$ 的实际加工时间为各项资源的原始加工时间 $p_{\pi_{k,j},h}$ 除以资源数量 $u_{\pi_{k,j}}$, $x_{\pi_{k,j},h}$ 为0-1决策变量表示工序 $O_{\pi_{k,j}}$ 的资源类型选择, $h=1, 2, 3$ 分别表示选择初等工、机器人、高级工, $y_{j,q}$ 为0-1决策变量表示当前工序 j 中所有工件选择的资源类型, $q=1, 2, 3$ 分别表示简单工序选择工人、简单工序选择机器人、复杂工序选择高级工。

总能耗TEC中 E_W 和 E_I 的推算公式如下:

$$E_W = \sum_{k=1}^n \sum_{j=1}^{n_s} \sum_{h=1}^3 \sum_{q=1}^3 [k_w R_{I,h} + (1 - k_w) R_{W,h}] \times t_{\pi_{k,j}} \cdot x_{\pi_{k,j},h} \cdot y_{j,q} \cdot \gamma_{\pi_{k,j}} \quad (9)$$

$$\gamma_{\pi_{k,j}} = [u_{\pi_{k,j}} + 0.5 \times \ln(u_{\pi_{k,j}})], \quad (10)$$

$$E_I = \sum_{h=1}^3 R_{I,h} \sum_{k=1}^n \sum_{j=1}^{n_s} (C_{\pi_{k,j}} - C_{\pi_{k-1,j}} - t_{\pi_{k,j}}) \times x_{\pi_{k-1,j},h} \quad (11)$$

其中, 式(9)为总加工能耗, $R_{W,h}$ 和 $R_{I,h}$ 为各类资源的加工和空闲功率, k_w 为加工冷却率, 表示机器使用到一定时间需要停工冷却防止使用过度而造成损坏; 式(10)表示能耗与数量的非线性映射 γ ; 式(11)表示总空闲能耗。

2.3 模型举例

为了更直观地介绍该问题的目标函数计算过程, 表1给出了一个简单的HRCES示例, 包含4个工件和3道工序。工序1和2为可被所有资源加工的简单工序,

工序3为只能由高级工加工的复杂工序。在该示例中可分配的加工资源包含4名初等工、3名高级工和2台机器人。工人操作机器的加工功率为 $R_{W,1}=2$, 空闲功率为 $R_{I,1}=1$, 机器人的加工和空闲功率为 $R_{W,2}=4$, $R_{I,2}=2$, 设备冷却率为 $k_w=0.2$ 。

图1展示了一个可行解的甘特图。图中工件的加工顺序为 $I_2-I_3-I_1-I_4$, 每个工序上方的括号表示初等工、高级工、机器人等资源的选择数量。纵轴左侧 y 表示每道工序的资源选择类型, $y=1, 2, 3$ 表示选择初等工、机器人、高级工。该解由工件 I_2 的工序 J_1 开始加工, 由于工序 J_1 选择机器人加工, 因此查询表1中机器人的原始加工时间, 并根据工序的资源数量计算实际加工时间, 再依据式(4)~(8)计算每个工件的完工时间, 最终得到最大完工时间为24。随后, 根据式(9)~(11)计算总能耗TEC。以工件 I_2 的工序 J_2 为例, $E_{W(k=1,j=2)}=(0.8 \times 2 + 0.2 \times 1) \times 3 \times (1 + 0.5 \times \ln 2) = 1.8 \times 3 \times 1.35 = 7.29$ 。该工序完工后的空闲能耗 $E_{I(k=2,j=2)}=1 \times 3 = 3$ 。同样地, 累加所有的加工能耗和空闲能耗可求得TEC。

2.4 问题特性

分析复杂调度问题的特性能提炼求解问题的专家知识, 进而指导设计高效的求解算法。本节对HRCES的两个优化目标的特性进行了如下分析。

特性1: 每个工序使用越多的资源数量和选择越少加工时间的资源类型可使 C_{\max} 越小。在不考虑资源空闲时间的前提下, 可以预估 C_{\max} 的下界。

分析1: 由式(8)可知实际加工时间 t 由原始加工时间 p 、资源数量选择 u 、工序资源类型 y 和工件资源等级 x 协同决定。当所选资源类型不变时, u 越大实际加工时间 t 越小, 使得整体的完工时间越小。当 u 不变时, 选择加工速度越快的资源, 则 t 越小, 使得 C_{\max} 越小。假设所有工件的所有工序选择最多最快的资源, 则解的完

表1 图1示例中所有工件的各类资源的原始加工时间

Table 1 Original processing time of each resource of all jobs in Figure 1

工件	工序 J_1 (简单)			工序 J_2 (简单)			工序 J_3 (复杂)
	初等工	高级工	机器人	初等工	高级工	机器人	高级工
I_1	8	7	6	12	8	6	12
I_2	12	8	6	8	5	4	9
I_3	10	8	6	10	6	4	9
I_4	12	4	4	8	6	4	12

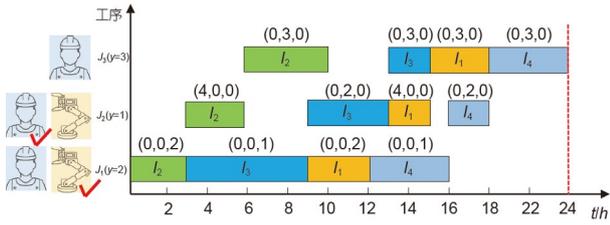


图 1 (网络版彩图)HRCES的甘特图示例
Figure 1 (Color online) Gantt chart example for the HRCES.

工时间可靠近下界, 但能耗也最大.

特性2: 每个工序使用越少的资源数量和选择越低能耗的资源类型可使TEC越小. 在不考虑资源空闲能耗的前提下, 可以预估TEC的下界.

分析2: 该过程与分析1相似. 由式(9)~(11)可知, 总能耗取决于加工和空闲时间, 而工序使用资源数量越多, 则加工能耗越大. 如果工序选择机器人加工, 虽然速度更快, 但能耗更大. 因此, 所有工件的所有工序使用最少工人加工, 则可使TEC靠近下界.

特性3: 增加关键路径上的关键工序的资源数量或替换更快的资源类型, 可有效减少 C_{max} .

分析3: 已知调度序列的最大完工时间只取决于关键路径 π 上的关键工序的完工时间. 而在该问题中

$$C_{max} = \sum_{i \in \pi} t_i^c, t^c \text{ 表示关键工序加工时间. 因此增加关键工序的 } u \text{ 使得实际加工时间 } t \text{ 减少, 因此 } C'_{max} = \sum_{i \in \pi} t'_i < \sum_{i \in \pi} t_i < C_{max}.$$

特性4: 减少非关键路径上工序的资源数量或为非关键工序替换能耗更低的资源, 可在不改变 C_{max} 的前提下有效减少TEC.

分析4: 结合分析2和3可知, 如果不改变关键工件的资源数量和类型, 则 C_{max} 不变. 因此减少非关键工件的实际加工时间, 在关键路径不变的前提下可减少空闲时间, 降低总能耗.

3 协同优化算法

3.1 算法框架

图2展示了所提方法TPCFO的双阶段进化框架. 在第一阶段中, 算法首先通过多策略协同初始化生成初始种群. 其次, 算法通过多层协同全局搜索更新种

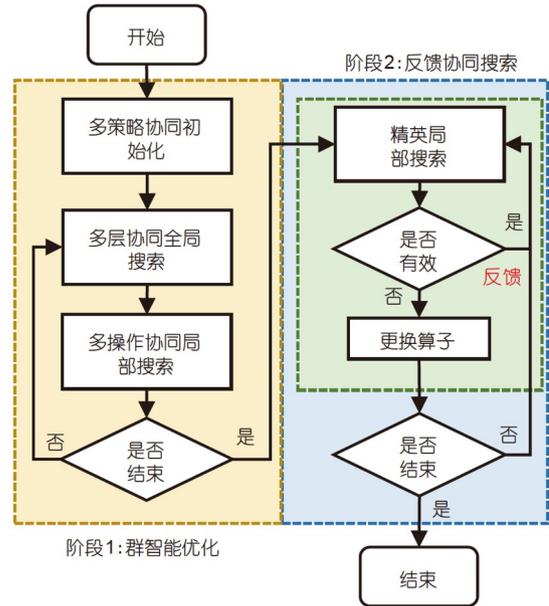


图 2 (网络版彩图)TPCFO算法流程图
Figure 2 (Color online) Flow chart of TPCFO.

群. 随后, 算法对种群的精英个体执行多操作协同局部搜索以提升种群的收敛性. 在第二阶段, 算法在精英存档中执行贪心迭代搜索以增强解的收敛. 最后, 算法在精英存档中输出非支配调度方案集.

3.2 编码与解码

HRCES的排产方案由3个子问题耦合决定: 工件加工顺序、工件资源类型选择和工件资源数量选择. 由于资源类型选择约束复杂, 本文采用4层向量表征HRCES的排产方案, 包括工件顺序(job sequence, JS)、工序资源类型(operation resource type, OT)、工件资源类型(job resource type, JT)和资源数量(resource count, RC). 图3展示了图1中甘特图的编码方案. 通过多向量协同编码保证调度方案的可行性.

解码策略: 首先, 根据JS向量确定工件加工顺序. 其次, 根据OT向量确定每道工序选择工人或机器人. 再次, 根据JT向量细化工件选择初等工或高级工, 并根据以上信息获得每个工件所选资源类型的原始加工时间. 随后, 根据RC向量获取每个工件使用的资源数量, 并根据式(8)计算每个工件的实际加工时间. 重复上述过程可获得所有工序的实际加工时间. 最后, 根据式(4)~(11)计算 C_{max} 和TEC.

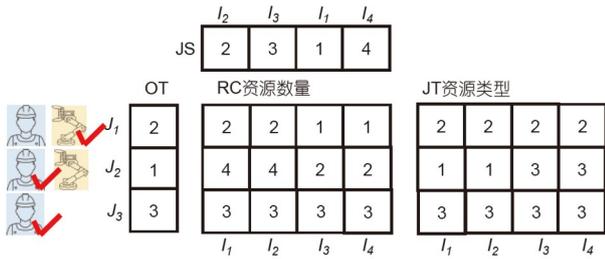


图3 (网络版彩图)图1中示例的编码
Figure 3 (Color online) Encoding of the example in Figure 1.

3.3 多策略协同初始化

初始种群的质量奠定算法后续搜索效率, 设计良好的初始化策略可使算法花费少量计算资源即可获得具有良好收敛性和分布性的种群, 避免算法在前期过多地无效搜索而浪费计算资源, 为算法后续勘探提供更多资源^[31]. 根据特性1和2可知, 影响 C_{max} 和TEC核心的变量为工件的资源类型和资源数量. 基于上述特性, 本文设计并协同三个初始化策略来构造初始化种群实现压缩搜索空间的目标.

策略1(最小加工时间原则): 所有工件使用最多的资源数量和选择最快的资源类型.

策略2(最小能耗原则): 所有工件使用最少的资源数量和选择最小加工能耗的资源类型.

多策略协同初始化策略: 策略3与Nawaz-Enscore-Ham算法^[32]类似, 属于贪心插值法与策略1和2协同使用. 首先, 以策略1构造大小为 $ps/10$ 的子种群, ps 表示初始种群中解的数量. 其次, 计算子种群中每个解的所有工件在所有工序上的累计实际加工时间. 再次, 根据累计时间降序重排工件. 随后, 第一个工件不动, 将剩余工件按序插入到所有可能的位置, 并计算不完整解的最大完工时间. 在所有候选解中, 保留最大完工时间最小的解作为后续待搜索的新的子序列, 直到所有工件都搜索完毕即可获得靠近 C_{max} 真实下界的初始子种群. 同样地, 对使用策略2生成的子种群降序排序, 并在搜索过程中以最小TEC为标准保留子序列, 最终得到靠近TEC下界的初始子种群. 剩余 $8/10 \times ps$ 个体用随机策略生成以确保初始种群良好的分布性.

多策略协同后, 算法获得收敛性和分布性良好的初始种群. 在种群两端分布着靠近优化目标下界的收敛良好的解集, 在种群的中间区域的个体具备良好的

分布性以增加后续全局搜索中种群的分布性.

3.4 多层协同全局搜索

HRCES的目标函数值由4个耦合向量共同决策, 使该问题搜索空间庞大, 即

$$n! \times \left\{ \prod_{s=1:n_s/2} \left[\prod_{i=1:n} (n_k + n_o) + \prod_{i=1:n} n_r \right] + \prod_{s=n_s/2:n_s} \prod_{i=1:n} n_k \right\},$$

其中 $n!$ 表示工件加工顺序的排列, 后半部分为资源类型和数量的组合数. 随着工件和资源数量增加, 问题求解难度呈非线性快速增长, 对算法设计带来了挑战. 因此在设计交叉算子时需要考虑多个子问题的优化特点. 通过不同的信息交换策略, 使得父代之间充分交换基因信息从而产生较大的搜索步长, 让算法在进化前期快速收敛到非支配解集附近.

JS向量交叉: JS向量采用如图4所示的部分匹配交叉(partial match crossover, PMX)算子^[4]. 首先, 随机从1到 n 选择交换区间的上下界. 其次, 交换所选区域中 JS_1 和 JS_2 的基因, 即图4中黄色和蓝色部分. 再次, 将 JS_1 和 JS_2 的交换区域外的基因按照原位置的顺序填入新的解 JS_3 和 JS_4 中. 随后, 以 JS_3 为例, 由于工件2和5已经在蓝色区域出现, 因此根据所选区域数字的映射关系, 2对应7, 则将 JS_3 中2换为7. 同样地, 5与4对应, 但4也在蓝色区域, 所以进一步递归查找, 直到找到一个没出现过的数为止. 如黑色箭头所指, 5最后与1对应, 因此将 JS_3 中1换为5. 经过PMX算子, 父代可以保留所选区域中大部分的优质基因, 并产生小范围的扰动, 在不丢失太多搜索知识的前提下, 使得父代在小范围内探索新的解.

OT-RC-JT协同交叉: HRCES中简单工序的资源选择能影响该工序上所有工件的加工时间和能耗. 但改变OT向量中的资源类型选择后, RC和JT向量的新值可能违反约束. 因此, 在本节中, 根据特性1和2设计了一种协同交叉算子确保父代个体能交换不同工序的资源类型的选择信息. 图5展示了算子执行过程. 算子首先产生图中绿色的0-1决策向量. 其次, 遍历每道工序对应的决策向量, 如果值为1, 则交换父代OT, RC和JT向量中当前工序的资源选择信息. 图5中蓝色和黄色为需要交换的基因, 图5的下方为生成的子代. 通过该算子能在不违反约束的前提下改变整个工序的资源类型选择. 例如 OT_1 的工序1选择工人加工, 在交换后

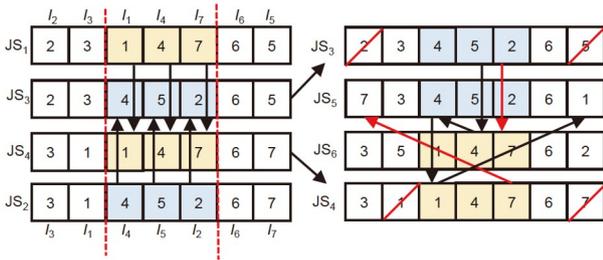


图4 (网络版彩图)PMX算子
Figure 4 (Color online) The PMX operator.

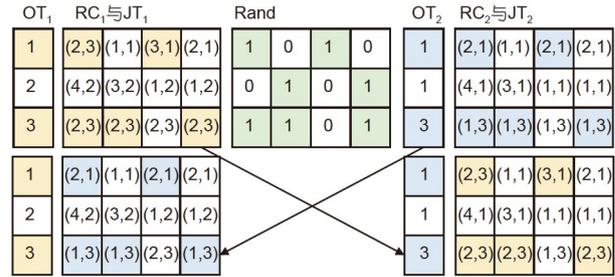


图6 (网络版彩图)JT-RC协同交叉算子
Figure 6 (Color online) JT-RC cooperative operator.

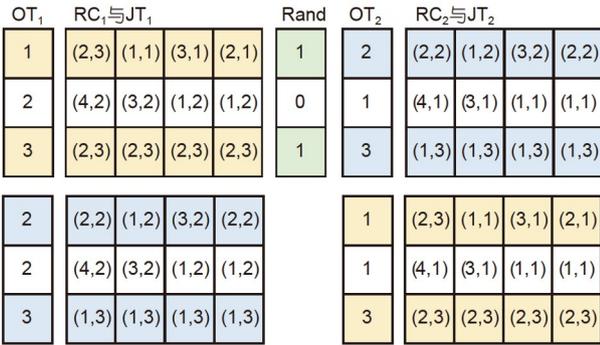


图5 (网络版彩图)OT-RC-JT协同交叉算子
Figure 5 (Color online) OT-RC-JT cooperative operator.

变为了2即机器人加工, 则第一道工序整体的实际加工时间缩短, 使得完工时间减少. 而 OT_2 的工序1变为了1即工人加工, 相应地能耗也会降低.

JT-RC协同交叉: 为了让父代解能交换工件的资源类型和数量选择的优质基因, 本节设计了一种**JT-RC**协同交叉算子. 由于**JT**向量中值的改变可能违反约束, 因此在设计算子时, **RC**与**JT**一同改变. 例如工件1的工序1选择4个初等工加工, 如果在信息交叉后变为高级工, 则违反约束, 因为车间中高级工只有2人, 图6展示了算子的交叉过程. 首先随机产生0-1矩阵. 其次, 扫描**RC**与**JT**矩阵, 如果**Rand**矩阵中同行列为1, 则交换两个解在该位置的**RC**和**JT**的值. 为了防止产生不可行解, 只有两个解在**OT**向量中相同值所在的行才能发生交换. 图6中两个解的第二道工序的资源类型不同, 因此不会交换信息.

本节还设计了两个突变算子以增加全局搜索过程中解的多样性, 其描述如下.

JS突变: 随机交换**JS**向量中两个工件的位置.

RC突变: 随机选择一个工件的工序, 在其约束范

围内重选资源数量.

多层协同全局搜索: 对于两个父代解. 首先, 采用**JS**交叉算子交换工件加工顺序的搜索信息; 其次, 采用**OT-RC-JT**协同交叉算子交换两个解人机类型选择的信息; 再次, 采用**JT-RC**协同交叉算子交换两个解的资源数量与相应资源等级选择的信息; 最后, 对两个子代的**JS**和**RC**向量执行突变算子. 产生两个随机数 r_1 和 r_2 与阈值 P_m 比较, 如果 $r_1 < P_m$ 则执行**JS**突变, 如果 $r_2 < P_m$ 则执行**RC**突变. 通过多层协同全局搜索, 两个解能充分地交换各个子问题层面的搜索信息以加速种群收敛.

3.5 多操作协同局部搜索

分析问题特性有助于设计高效邻域结构, 实现提高局部搜索效率, 进而节省计算资源让算法执行更多次的搜索. 基于特性3和4本文设计了5种邻域结构来优化不同目标函数, 其描述如下.

邻域结构 N_1 : 随机交换**JS**上两个工件的位置使得加工时间大的工件提前加工并减少资源空闲时间以增加种群的分布性.

邻域结构 N_2 : 随机选择两个**JS**上的工件, 提前加工靠后的工件, 将后者插入到前者之前, 尝试减少等待时间.

邻域结构 N_3 : 基于特性3, N_3 随机选择一个关键工件, 增加其使用的资源数量以有效减少 C_{max} . 图7展示了相应案例. 首先找到图1中甘特图的关键路径, 随机选择了 I_3 的工序 J_1 并增加其资源数量, 最终 C_{max} 由24变为19.

邻域结构 N_4 : TEC取决于加工和空闲时间, 两者越少, TEC越小. 而在不增加 C_{max} 的前提下, 无法直接增

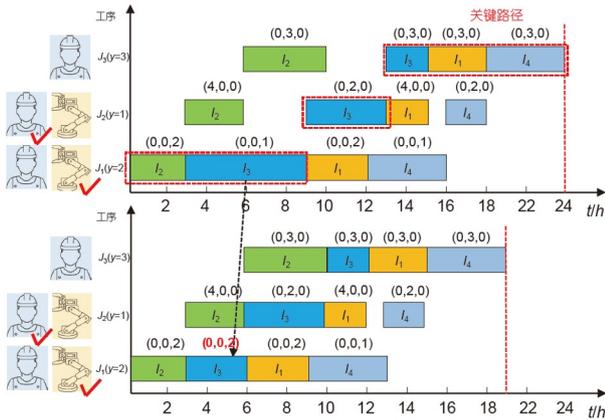


图7 (网络版彩图) N_3 邻域结构
Figure 7 (Color online) Neighborhood structure of N_3 .

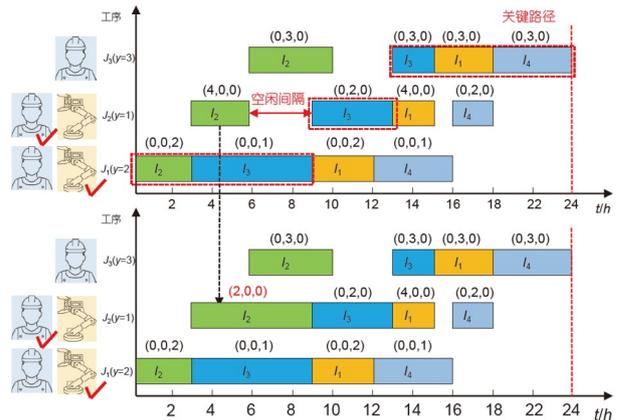


图8 (网络版彩图) N_4 邻域结构
Figure 8 (Color online) Neighborhood structure of N_4 .

加关键路径上工件的加工时间. 因此, N_4 主要关注非关键工件. 通过扫描所有非关键工件, 在不改变关键路径和不违反约束的前提下, 尝试降低工件的资源数量至最少. 图8展示了 N_4 的过程, I_2 的工序 J_2 原本资源数量是4, 该工序后有大量的空闲时间, 因此降低资源数量到2并保证其完工时间不超过 I_3 的开始时间. 最终在不改变 C_{max} 的前提下减小TEC.

邻域结构 N_3 : 随机选择一个工件的工序, 在不改变整道工序资源类型选择的前提下, 尝试改变其资源类型或数量. 如果选择的工件所处的工序都由机器人(简单工序)或高级工加工(复杂工序)完成, 则只改变资源数量. 如果所选工件的所处工序是简单工序并选择初等工加工, 则替换其资源类型, 并在可选范围内重新设置资源数量. 邻域结构 N_3 的步长较小, 可让解在优化后期逐步靠近最优解.

3.6 双阶段协同反馈优化

为了在有限的时间内求得更好的非支配解集, 本文设计了一种双阶段协同反馈优化算法(TPCFO). 在第一阶段, 算法执行协同模因算法让种群快速收敛到非支配解集附近. 在第二阶段, 算法在精英存档附近执行变邻域反馈搜索以广泛地探索不同进化方向进而增加存档的分布性. 具体描述如下.

3.6.1 协同模因算法

协同模因算法的具体步骤如下. 步骤1: 执行多策略协同初始化策略产生高质量的初始种群 P_t . 步骤2:

使用双人锦标赛策略在当前种群 P_t 中选择精英解构建交配池 M_t . 步骤3: 对当前种群的每个解与随机在交配池 M_t 中选择一个精英解执行多层协同搜索全局搜索, 以充分交换优质搜索信息. 步骤4: 对产生的两个新解以 P_m 的概率执行突变策略. 步骤5: 将子种群与原始种群合并, 并采用SPEA2^[33]算法的环境选择策略实现快速更新种群. SPEA2的环境选择策略的密度估计策略仅考虑当前解与最近解的距离, 而在多目标车间调度中常用的NSGA-II^[34]算法的拥挤度距离策略则考虑两侧解的距离. 对于边界点只有一个邻域解的情况, 拥挤度距离策略无法有效计算拥挤度. 因此, 本文选择SPEA2的环境选择策略. 式(12)展示了其过程, 前半部分表示当前个体 i 被父种群 P_t 和子种群 C_t 中所有个体 j 的支配数, 后半部分为个体 i 与最近个体 k 的密度, 其中 σ_i^k 表示两个解之间的欧式距离. 随后选择 $F(i) < 1$ 的个体作为非支配解集进入新的种群. 因为前半部为0表示非支配解, 而后半部分小于1表示密度低. 当新种群中个体不足 p_s 则执行截断操作. 计算剩余个体中密度最高的个体, 直到剩余个体数能等于新种群的空余位置数.

$$F(i) = \sum_{j \in P_t + C_t} \{ \lfloor j \mid j \in P_t + C_t \wedge j \succ i \rfloor \} + \frac{1}{\sigma_i^k + 2}. \quad (12)$$

步骤6: 将得到种群 P_{t+1} 的非支配解集存入精英存档 Ω , 防止精英解个数超过种群容量而造成精英解被丢弃, 进而降低最终非支配解集的收敛性和分布性. 步骤7: 对种群 P_{t+1} 中的非支配解执行多操作协同局部

搜索, 每个解随机地执行邻域动作, 如果产生的子代支配父代, 则替换父代, 如果子代和父代互不支配则存入精英存档 Ω . 重复步骤2~7直到第一阶段的计算资源耗尽.

3.6.2 变邻域反馈搜索

在第一阶段中种群快速靠近真实的非支配前沿面, 但算法会收敛于局部最优解. 步长大的交叉变异算子无法有效地继续向前搜索. 因此, 在第二阶段算法需要使用小步长的局部搜索算子对精英解进行微调. 为了提升局部搜索的效率, 本文设计了一种反馈机制, 在每次执行完局部搜索后, 算法都会获得一个反馈 R_t . 通过反馈值, 算法决定是否继续执行当前算子以持续优化当前解的目标函数. 通过贪心地执行多操作局部搜索提升非支配解集的收敛性. 具体步骤如下: 对精英种群中每个解执行 N_1 , 如果得到的子代更优则 $R_t=1$ 否则 $R_t=0$. 如果当前 $R_t=1$ 则继续执行 N_1 否则执行 N_2 . 同样地, 后续每个邻域结构执行时都会获得反馈, 算法根据反馈决定是否继续执行当前邻域动作, 直到第二阶段的计算资源耗尽时跳出当前局部搜索. 最终算法输出精英存档中的非支配解集作为最终结果.

3.7 算法复杂度分析

在第一阶段中多策略协同初始化采用三种规则, 规则1和2都只与决策变量规模以及种群大小有关, 因此时间复杂度为 $O(ps \times n \times n_s/5)$. 在多策略协同后, 由于NEH规则的使用, 复杂度变为 $O(ps \times n^2 \times n_s/5)$. 剩余个体随机生成, 因而整体的时间复杂度为 $O(ps \times n(n+4) \times n_s/5)$. 交配选择的时间复杂度为 $O(ps)$. 多层协同全局搜索的时间复杂度为 $O(ps \times (n^2 + n_s + n \times n_s))$. 环境选择的复杂度为 $O(ps^3)$. 第二阶段局部搜索中 N_1 复杂度为 $O(1)$, N_2 复杂度为 $O(n)$, N_3 和 N_4 的复杂度为 $O(n^2 \times n_s)$. 因此整体算法的时间复杂度为 $O(ps \times n^2 \times n_s)$.

4 数值仿真实验与分析

4.1 实验设置

本文所有代码均采用MATLAB 2022编写. 硬件仿真环境为Inter(R) Core(TM) i9-13900k CPU浮点计算频率为3.00 GHz, 运行内存为64 GB. 为了验证本文提出算法的有效性, 本文根据文献[4]生成了15个不同

规模的HRCES测试用例. 其中工件总数 $n \in \{20, 40, 60, 80, 100\}$, 工序总数 $n_s \in \{4, 6, 8\}$, 将前 $\lfloor n_s/2 \rfloor$ 工序定义为可被工人和机器人协同加工的简单工序, 将后半部分工序定义为只能由高级工加工的复杂工序. 在实际场景中初等工最多, 机器人和高级工数量相同. 因此, 阶段数4, 6, 8对应的初等工人数、高级工人数、机器人人数 $\{n_o, n_k, n_r\}$ 分别为 $\{2, 1, 2\}$, $\{4, 2, 4\}$, $\{6, 3, 6\}$. 简单工序中工件的初等工原始完工时间 $p_{i,j,1}$ 随机在 $[10, 20]$ 区间取值. 复杂工序中工件的高级工的原始完工时间在 $[20, 40]$ 中取值. 高级工的原始完工时间 $p_{i,j,2}$ 为 $\lfloor p_{i,j,2}/2 \rfloor$. 机器人的原始完工时间 $p_{i,j,3}$ 为 $(0.5 \times \text{rand} + 0.5) \times p_{i,j,2}$. 如果当前工序是复杂工序则 $p_{i,j,1}$ 和 $p_{i,j,3}$ 为0, 表示初等工和机器人无法加工该工序. 每道工序可选的初等工人数和机器人数量相同. 每个测试问题中工人的加工和空闲功率为 $R_{w,1}=R_{w,2}=5$ 和 $R_{l,1}=R_{l,2}=2.5$, 机器人的加工和空闲功率为 $R_{w,3}=10$ 和 $R_{l,3}=5$. 表2展示了不同测试问题的各项参数. 为了公平比较, 所有算法采用相同的最大运行时间为终止准则, $\text{Max}_T=0.05 \times n \times n_s$, Max_T 至少为10 s.

本文采用多目标优化指标超体积(hypervolume, HV)^[35]和世代距离(generation distance, GD)^[34]来评价算法的综合性能和收敛性. HV指标由非支配解集与最大值参考点围成的超立方体的体积决定, HV越大表示非支配解集越接近于真实前沿面, 点与点之间分散得越均匀, 算法综合性能越好. 而GD表示非支配解集上的点与所有参考前沿面上点的最近距离的和. GD越小表示算法求得的非支配解集与参考前沿面越接近, 说明算法的收敛性越好. 参考前沿面为所有运行算法的所有运行次数中得到的非支配解集组成的种群的非支配解集, 表示目前的最好的解集.

4.2 参数实验

算法的参数设置能影响算法的性能, 本文采用田口正交实验^[36]方法分析最优参数配置. 本文所提算法包含三个关键参数: (1) 种群大小(ps); (2) 突变概率(P_m); (3) 第一阶段资源占比 T_r . 每个参数设置3个水平, 采用 $L_9(3^3)$ 的正交设计, 参数组合如表3所示. 每个参数组合在随机生成的测试问题上独立运行20次, 停止条件为最大运行时间, 统计HV和GD指标的均值. 最后对田口实验进行分析. 表4展示了在各指标上各参数

表 2 测试问题参数

Table 2 The parameters of experimental instances

Instance	n	n_s	n_o	n_k	n_r
P01	20	4	2	1	2
P02	20	6	4	2	4
P03	20	8	6	3	6
P04	40	4	2	1	2
P05	40	6	4	2	4
P06	40	8	6	3	6
P07	60	4	2	1	2
P08	60	6	4	2	4
P09	60	8	6	3	6
P10	80	4	2	1	2
P11	80	6	4	2	4
P12	80	8	6	3	6
P13	100	4	2	1	2
P14	100	6	4	2	4
P15	100	8	6	3	6

表 3 正交实验表与各参数组合的指标

Table 3 Orthogonal table and the metric of each parameter combination

ps	P_m	T_r	HV	GD
80	0.1	0.5	0.217185	0.003074
80	0.2	0.6	0.227224	0.001991
80	0.3	0.7	0.232597	0.001331
100	0.1	0.6	0.217311	0.002822
100	0.2	0.7	0.230161	0.001618
100	0.3	0.5	0.216935	0.003232
120	0.1	0.7	0.219096	0.002867
120	0.2	0.5	0.215958	0.003148
120	0.3	0.6	0.240624	0.000610

表 4 各参数水平在所有指标的评价响应值^{a)}

Table 4 Average response values of each parameter level on all metrics

水平	HV			GD		
	ps	P_m	T_r	ps	P_m	T_r
1	0.2257	0.2179	0.2167	0.0021	0.0029	0.0032
2	0.2215	0.2244	0.2284	0.0026	0.0023	0.0018
3	0.2252	0.2301	0.2273	0.0022	0.0017	0.0019
Delta	0.0042	0.0122	0.0117	0.0004	0.0012	0.0013
Rank	3	1	2	3	2	1

a) 加粗数据表示最优指标值

水平的响应值. HV指标越大说明算法性能越好, 而GD指标相反. 由表3和4可知, 最优的参数选择为 $ps=80$, $P_m=0.3$, $T_r=0.6$.

4.3 算法协同策略有效性验证

本文提出的TPCFO算法中包含几个关键策略: 多策略协同初始化、多层协同全局搜索、多操作协同局部搜索、多阶段协同、反馈机制和改进的环境选择策略. 为了分析每个改进的有效性, 本文设置了相应变体算法. (1) A-I: 表示TPCFO删去多策略协同初始化并采用随机初始化. (2) A-L: 表示TPCFO删去多操作协同局部搜索. (3) A-T: 表示TPCFO删去多阶段协同进化, 只采用第一阶段的协同模因算法. (4) A-F: 表示TPCFO在第二阶段的搜索删去反馈机制, 并且每个局部操作执行一次. (5) A-X: 表示TPCFO删去多层协同搜索并只保留JS交叉算子与文献[6]一致. (6) A-E: 表示TPCFO删去SPEA2^[33]的环境选择策略, 而采用NSGA-II^[34]的环境选择策略. (7) A: 表示TPCFO未删去所有改进策略. 每个变体算法以相同的运行时间独立运行20次, 统计HV和GD指标的均值, 其结果在表5和6中展示, 其中加粗表示最优指标值. 由表5和6可知本文提出的算法相较于对比算法在大部分测试问题上

都取得了最好HV和GD指标. 说明了删除了各种改进策略后算法性能明显下降, 表明了本文提出的各类改进点的有效性.

为了进一步分析不同改进点对算法性能的影响, 将得到的指标统计结果进行Friedman秩和检验来分析不同算法之间的显著性. 表7展示了秩和检验的结果. 该方法会根据每个算法的指标统计结果对算法性能做排序, 同时分析每个算法与最优算法之间的显著性差异, 排名的值越小说明算法性能越好. 由表7可知, 本文提出的算法相较于变体算法在HV和GD指标上均取得了最高排名. 通过P值可知, TPCFO算法相较于其他变体算法都取得了显著性的优势, 表明删去改进点后算法性能显著变差, 验证了本文提出的各类改进策略的有效性. 根据表5~7的结果, 通过对比算法A与A-I可知, 算法A在各类指标取得了显著性的优势, 验证了多策略协同初始化的有效性, 因为不同的启发式策略能让种群快速靠近算例下界, 为剩余搜索过程节省了大量的计算资源. 对比算法A与A-L可验证多操作协同局部搜索的有效性. 每个邻域结构对解的有效性不同, 多种操作协同保证每个解执行有效的操作以加速收敛. 对比A与A-T可知, 多阶段协同有效地增加了算法的综合性能, 因为第一阶段的全局搜索使得算法快速靠近

表5 所有变体算法的HV(max)指标的统计结果^{a)}

Table 5 Statistical results of HV(max) metric for all variant algorithms

Instances	A-I	A-L	A-T	A-F	A-X	A-E	A
P01	0.2770	0.2683	0.2686	0.2681	0.2716	0.2679	0.2685
P02	0.4497	0.4619	0.4610	0.4587	0.4583	0.4561	0.4622
P03	0.4839	0.5262	0.5288	0.5256	0.5146	0.5174	0.5326
P04	0.2428	0.2620	0.2628	0.2579	0.2595	0.2552	0.2630
P05	0.4042	0.4264	0.4314	0.4161	0.4256	0.4103	0.4329
P06	0.5636	0.5753	0.5808	0.5481	0.5727	0.5239	0.5856
P07	0.2382	0.2568	0.2515	0.2393	0.2479	0.2255	0.2595
P08	0.4297	0.4618	0.4565	0.4286	0.4531	0.4183	0.4751
P09	0.5436	0.5478	0.5607	0.5351	0.5758	0.5222	0.5877
P10	0.2416	0.2665	0.2599	0.2407	0.2504	0.2343	0.2665
P11	0.4746	0.4784	0.4879	0.4469	0.4867	0.4344	0.5105
P12	0.5937	0.5389	0.5529	0.5289	0.5840	0.5250	0.5722
P13	0.2576	0.2747	0.2641	0.2281	0.2745	0.2051	0.2790
P14	0.5039	0.4824	0.4945	0.4613	0.4956	0.4511	0.5154
P15	0.5421	0.5287	0.5296	0.5180	0.5536	0.5217	0.5414

a) 加粗数据表示最优指标值

表 6 所有变体算法的GD(min)指标的统计结果^{a)}

Table 6 Statistical results of GD(min) metric for all variant algorithms

Instances	A-I	A-L	A-T	A-F	A-X	A-E	A
P01	0.0002	0.0015	0.0014	0.0015	0.0014	0.0016	0.0014
P02	0.0010	0.0003	0.0003	0.0005	0.0009	0.0007	0.0002
P03	0.0017	0.0008	0.0008	0.0008	0.0017	0.0010	0.0005
P04	0.0022	0.0003	0.0002	0.0009	0.0017	0.0012	0.0002
P05	0.0015	0.0008	0.0004	0.0016	0.0017	0.0018	0.0004
P06	0.0011	0.0013	0.0009	0.0025	0.0017	0.0036	0.0005
P07	0.0021	0.0003	0.0007	0.0021	0.0018	0.0034	0.0002
P08	0.0024	0.0018	0.0020	0.0034	0.0030	0.0036	0.0006
P09	0.0016	0.0039	0.0031	0.0038	0.0028	0.0041	0.0008
P10	0.0030	0.0003	0.0009	0.0033	0.0030	0.0038	0.0005
P11	0.0022	0.0028	0.0020	0.0045	0.0032	0.0044	0.0006
P12	0.0008	0.0070	0.0058	0.0052	0.0037	0.0052	0.0027
P13	0.0030	0.0008	0.0018	0.0055	0.0016	0.0070	0.0006
P14	0.0012	0.0044	0.0035	0.0048	0.0040	0.0047	0.0010
P15	0.0007	0.0055	0.0054	0.0045	0.0035	0.0039	0.0028

a) 加粗数据表示最优指标值

表 7 所有变体算法的Friedman秩和检验结果^{a)}

Table 7 Results of Friedman rank and sum test for all variant algorithms

Algorithm	HV			GD		
	Rank	P-value	<0.05?	Rank	P-value	<0.05?
A-I	5.4000	0.0000	Y	4.6000	0.0008	Y
A-L	3.8667	0.0139	Y	4.4667	0.0014	Y
A-T	3.5333	0.0369	Y	3.8667	0.0113	Y
A-F	6.6000	0.0000	Y	6.3333	0.0000	Y
A-X	4.0667	0.0073	Y	5.2667	0.0000	Y
A-E	7.6000	0.0000	Y	7.0667	0.0000	Y
A	1.6667	—	—	1.6000	—	—

a) 加粗数据表示最优指标值

下界, 第二阶段执行精英局部搜索使得种群有足够计算资源探索新的非支配解, 增加了前沿面的多样性. 对比A与A-F可知, 设计的反馈机制有效地增加了算法的综合性能, 因为反馈信息可以使当前解不断地执行当前有效的邻域结构进而持续优化目标函数. 对比A与A-X可知, 针对不同子问题设计的交叉算子能有效地促进解的信息交换增加搜索步长并加速收敛. 对比A与A-E可验证本文采用的环境选择策略的有效性, 因为综合性能指标的快速计算使得算法有更多的时间执

行搜索操作进而增强种群的收敛性.

4.4 与现有算法对比

为了进一步验证所提算法的有效性, 将TPCFO对比最新相关算法PMA^[4]和两种经典求解方法SPEA2^[33]和NSGA-II^[34]. 每个算法的参数设置如下: 根据文献[4], PMA的种群大小 $ps=100$, 变异率 $P_m=0.2$, 局部搜索次数 $LS=30$. 针对SPEA2和NSGA-II的参数与TPCFO保持一致. 种群大小 $ps=80$, 变异率 $P_m=0.3$. 为了保证对比

的公平性, 所有算法采用相同的最大运行时间为终止准则, $\text{Max}_T = 0.05 \times n \times n_s$, Max_T 至少为10 s. SPEA2和NSGA-II使用与TPCFO相同的全局搜索算子. 每个算

法独立运行20次.

表8和9展示了HV和GD指标的统计结果, 表格最后一行统计了算法显著性统计结果. 由表8和9可知,

表 8 所有对比算法的HV(max)指标的统计结果^{a)}

Table 8 Statistical results of HV(max) metric for all compared algorithms

Instances	SPEA2	NSGA-II	PMA	TPCFO
P01	0.3384+	0.3376+	0.2822-	0.3307
P02	0.4576-	0.4476-	0.2852-	0.4719
P03	0.4868-	0.4805-	0.3067-	0.5476
P04	0.3162-	0.3137-	0.1975-	0.3370
P05	0.4254-	0.4114-	0.2298-	0.4766
P06	0.4933-	0.4831-	0.2671-	0.5416
P07	0.3414-	0.3384-	0.1758-	0.3649
P08	0.3906-	0.3883-	0.2021-	0.4735
P09	0.4625-	0.4594-	0.2323-	0.5437
P10	0.3116-	0.3169-	0.1465-	0.3455
P11	0.4100-	0.3973-	0.1817-	0.4951
P12	0.5347=	0.5196=	0.2674-	0.5435
P13	0.3709-	0.3627-	0.1213-	0.3829
P14	0.3896-	0.3739-	0.1633-	0.4401
P15	0.4949-	0.4622-	0.2779-	0.5214
-/=/+	13/1/1	13/1/1	15/0/0	

a) 加粗数据表示最优指标值, 符号“-/=/+”表示对比算法显著性差于、没有显著性、显著性好于TPCFO

表 9 所有对比算法的GD(min)指标的统计结果^{a)}

Table 9 Statistical results of GD(min) metric for all compared algorithms

Instances	SPEA2	NSGA-II	PMA	TPCFO
P01	0.0002+	0.0003+	0.0158-	0.0013
P02	0.0010-	0.0013-	0.0286-	0.0002
P03	0.0021-	0.0027-	0.0350-	0.0004
P04	0.0021-	0.0022-	0.0333-	0.0003
P05	0.0020-	0.0020-	0.0416-	0.0003
P06	0.0021-	0.0024-	0.0361-	0.0004
P07	0.0020-	0.0022-	0.0392-	0.0002
P08	0.0038-	0.0033-	0.0412-	0.0006
P09	0.0028-	0.0031-	0.0439-	0.0009
P10	0.003-	0.0023-	0.0453-	0.0004
P11	0.0045-	0.0040-	0.0564-	0.0006
P12	0.0015=	0.0018=	0.0469-	0.0023
P13	0.0016-	0.0021-	0.0593-	0.0005
P14	0.0025-	0.0034-	0.0466-	0.0012
P15	0.0020+	0.0033=	0.0381-	0.0027
-/=/+	12/1/2	12/2/1	15/0/0	

a) 加粗数据表示最优指标值, 符号“-/=/+”表示对比算法显著性差于、没有显著性、显著性好于TPCFO

TPCFO相较于对比算法几乎在所有测试问题上都取得了最好HV和GD指标, 同时显著性统计说明算法的表现在20次独立运行中具有显著性的优势, 验证了本文提出算法能够有效地求解人机协同的能效车间调度问题.

分析算法能取得成果的原因如下: 人机协同的问题对算法设计带来了新的复杂性即三个耦合的子问题. 首先, 本文分析了问题特性, 对算子设计给出了指导思想. 其次, 采用了多种协同思想, 保证算法在初始化、全局搜索、局部搜索中都有多样化的策略和操作优化目标, 多种协同策略丰富了算法优化的手段, 保证了算法在后期能有效收敛. 再次, 多阶段协同的思想使得算法面对复杂问题时能够有较好的表现. 第一阶段使用大步长的多层协同全局搜索保证算法快速收敛, 同时使用多操作局部搜索强化算法收敛. 第二阶段预留大量计算资源使得算法在精英解的附近展开勘探, 搜索更多非支配解集. 随后, 反馈机制的思想让算法在

第二阶段搜索时能够不断强化有效算子的使用, 避免无效算子浪费计算资源. 最后采用快速环境选择策略节省计算时间, 预留更多计算资源给全局搜索. 虽然SPEA2与NSGA-II都采用了多层协同搜索. 但收敛到后期陷入局部最优, 协同全局搜索无法进一步提升解的质量. 因此TPCFO中的小步长局部搜索使得算法取得了优势. PMA中虽然有相似的局部搜索算子, 但其缺乏对问题性质的有效分析, 没有从不同的子问题设计搜索算子, 导致算法前期收敛速度缓慢.

图9展示了所有对比算法在不同工件规模Pareto面的对比, 每个算法选择20次运行中HV指标最好的解集. 由图可知, 算法TPCFO在收敛性和分布性上都优于所有对比算法验证了本文算法的有效性.

5 总结与展望

本文以实际重工业生产场景为背景将焊接车间中

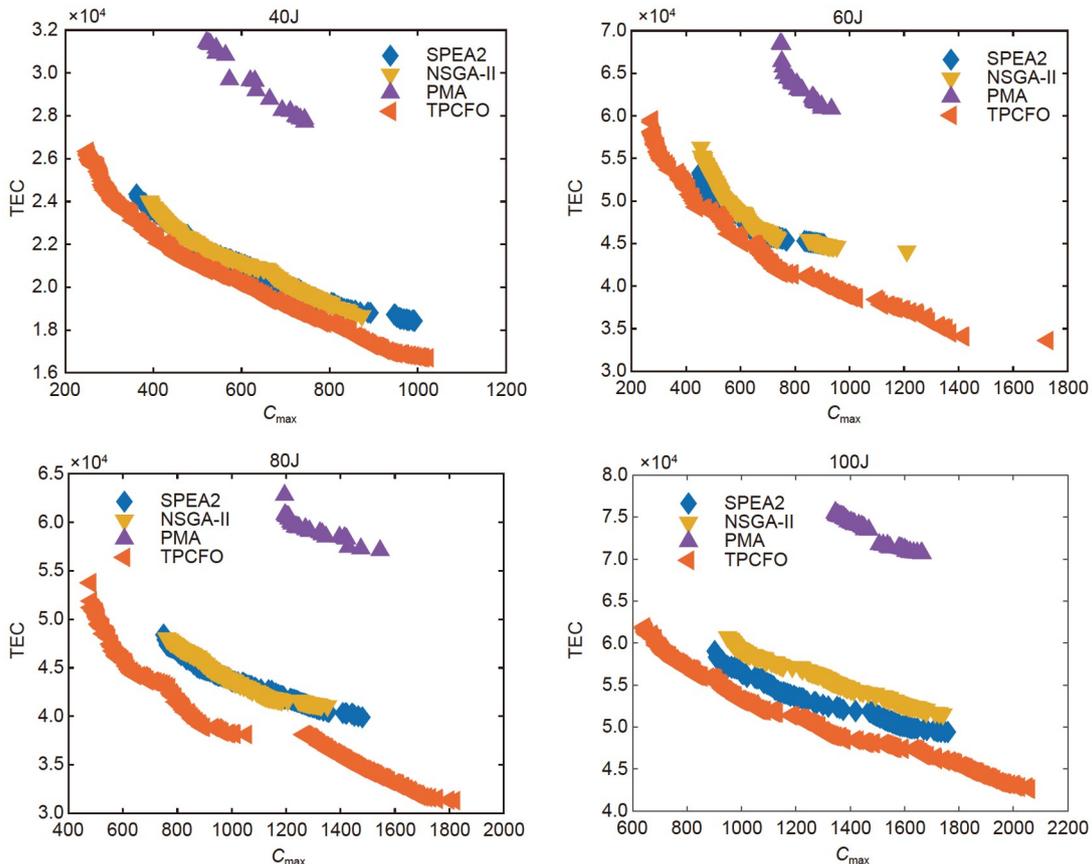


图9 (网络版彩图)对比算法在不同规模问题的Pareto面对比图

Figure 9 (Color online) Pareto front comparison plots for all compared algorithms on different scales of instances.

的人机协同特性提炼为人机协同的能效车间调度问题. 该问题在传统流水线调度的场景下, 考虑了工人和机器资源的类型选择的问题. 另外, 该问题中存在多种技能等级的工人和机器人, 工件的工序需要选择相对应等级的资源进行加工, 这些现实中存在的约束增加了问题求解的复杂性. 面对问题带来新的挑战, 首先, 对问题本身的特性进行了分析, 提炼有效操作指导算子设计. 其次, 算法以协同思想为指导在初始化协同、全局搜索协同、局部搜索协同、算法协同部分设计了多种策略保证算法在搜索步长和方法上具有多样性, 从而能在搜索过程中根据不同的解的现状采取针

对性的有效方法提升算法的收敛性. 最后, 通过仿真实验验证了本文以协同思想为指导的方法设计在问题求解上有显著的优势, 相较于对比算法取得了最好的结果, 为后续关于复杂的人机协同车间调度的研究提供的算法设计的指导思想.

关于人机协同的车间调度问题还有如下方向可以进一步挖掘: (1) 采用数据驱动的机器学习算法加强算法自主求解不同问题的能力; (2) 进一步探索车间调度中, 人机协同制造的新方式, 例如人机存在资源竞争; (3) 考虑人机协同车间调度问题中面临的新的动态事件和应对方法.

参考文献

- 1 Tao F, Ji Q L. Service-oriented smart manufacturing (in Chinese). *J Mech Eng*, 2018, 54: 11–23 [陶飞, 戚庆林. 面向服务的智能制造. 机械工程学报, 2018, 54: 11–23]
- 2 Li X Y, Huang J P, Li J H, et al. Research and development trend of intelligent shop dynamic scheduling (in Chinese). *Sci Sin Tech*, 2023, 53: 1016–1030 [李新宇, 黄江平, 李嘉航, 等. 智能车间动态调度的研究与发展趋势分析. 中国科学: 技术科学, 2023, 53: 1016–1030]
- 3 Zhou J, Li P G, Zhou Y H, et al. Toward new-generation intelligent manufacturing. *Engineering*, 2018, 4: 11–20
- 4 Lu C, Gao R, Yin L J, et al. Human-robot collaborative scheduling in energy-efficient welding shop. *IEEE Trans Ind Inf*, 2024, 20: 963–971
- 5 Zhang M, Li C Q, Shang Y L, et al. A task scheduling model integrating micro-breaks for optimisation of job-cycle time in human-robot collaborative assembly cells. *Int J Prod Res*, 2022, 60: 4766–4777
- 6 Vieira M, Moniz S, Gonçalves B S, et al. A two-level optimisation-simulation method for production planning and scheduling: the industrial case of a human-robot collaborative assembly line. *Int J Product Res*, 2022, 60: 2942–2962
- 7 Yu T, Huang J, Chang Q. Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning. *J Manuf Syst*, 2021, 60: 487–499
- 8 Ferreira C, Figueira G, Amorim P. Scheduling human-robot teams in collaborative working cells. *Int J Prod Econ*, 2021, 235: 108094
- 9 Mokhtarzadeh M, Tavakkoli-Moghaddam R, Vahedi-Nouri B, et al. Scheduling of human-robot collaboration in assembly of printed circuit boards: A constraint programming approach. *Int J Comput Integ M*, 2020, 33: 460–473
- 10 Faccio M, Granata I, Minto R. Task allocation model for human-robot collaboration with variable cobot speed. *J Intell Manuf*, 2024, 35: 793–806
- 11 Ren W B, Yang X N, Yan Y, et al. The decision-making framework for assembly tasks planning in human-robot collaborated manufacturing system. *Int J Comput Integ M*, 2023, 36: 289–307
- 12 Casalino A, Zanchettin A M, Piroddi L, et al. Optimal scheduling of human-robot collaborative assembly operations with time Petri nets. *IEEE Trans Automat Sci Eng*, 2021, 18: 70–84
- 13 Alirezazadeh S, Alexandre L A. Dynamic task scheduling for human-robot collaboration. *IEEE Robot Autom Lett*, 2022, 7: 8699–8704
- 14 Vahedi-Nouri B, Tavakkoli-Moghaddam R, Hanzálek Z, et al. Production scheduling in a reconfigurable manufacturing system benefiting from human-robot collaboration. *Int J Prod Res*, 2024, 62: 767–783
- 15 Wang D, Zhang J L. Flow shop scheduling with human-robot collaboration: A joint chance-constrained programming approach. *Int J Prod Res*, 2024, 62: 1297–1317
- 16 Guo D. Fast scheduling of human-robot teams collaboration on synchronised production-logistics tasks in aircraft assembly. *Robot Cim-Int Manuf*, 2024, 85: 102620
- 17 Zhang R, Lv J, Bao J, et al. A digital twin-driven flexible scheduling method in a human-machine collaborative workshop based on hierarchical reinforcement learning. *Flex Serv Manuf J*, 2023, 35: 1116–1138
- 18 Cai M, Liang R, Luo X, et al. Task allocation strategies considering task matching and ergonomics in the human-robot collaborative hybrid

- assembly cell. *Int J Prod Res*, 2023, 61: 7213–7232
- 19 Maderna R, Pozzi M, Zanchettin A M, et al. Flexible scheduling and tactile communication for human-robot collaboration. *Robot Cim-Int Manuf*, 2022, 73: 102233
- 20 Wang L, Wang J J. A cooperative memetic algorithm for the distributed green flexible job shop with transportation time (in Chinese). *Sci Sin Tech*, 2023, 53: 243–257 [王凌, 王晶晶. 考虑运输时间的分布式绿色柔性作业车间调度协同群智能优化. 中国科学: 技术科学, 2023, 53: 243–257]
- 21 Li M, Wang G G. A review of green shop scheduling problem. *Inf Sci*, 2022, 589: 478–496
- 22 Wang L, Wang J J, Wu C G. Advances in green shop scheduling and optimization (in Chinese). *Control Decis*, 2018, 33: 385–391 [王凌, 王晶晶, 吴楚格. 绿色车间调度优化研究进展. 控制与决策, 2018, 33: 385–391]
- 23 Liu C G, Yang J, Lian J, et al. Sustainable performance oriented operational decision-making of single machine systems with deterministic product arrival time. *J Cleaner Production*, 2014, 85: 318–330
- 24 Ramezani R, Vali-Siar M M, Jalalian M. Green permutation flowshop scheduling problem with sequence-dependent setup times: A case study. *Int J Prod Res*, 2019, 57: 3311–3333
- 25 Wang J J, Wang L. A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-shop scheduling. *IEEE Trans Evol Computat*, 2021, 26: 461–475
- 26 Masmoudi O, Delorme X, Gianessi P. Job-shop scheduling problem with energy consideration. *Int J Prod Econ*, 2019, 216: 12–22
- 27 Li R, Wang L, Gong W Y. Knowledge-driven memetic algorithm for distributed green flexible job shop scheduling problem (in Chinese). *J Huazhong Univ Sci Tech (Nat Sci)*, 2022, 50: 55–60 [李瑞, 王凌, 龚文引. 知识驱动的模式算法求解分布式绿色柔性调度. 华中科技大学学报(自然科学版). 2022, 50: 55–60]
- 28 Li R, Gong W, Wang L, et al. Surprisingly popular-based adaptive memetic algorithm for energy-efficient distributed flexible job shop scheduling. *IEEE Trans Cybern*, 2023, 53: 8013–8023
- 29 Li R, Gong W, Wang L, et al. Co-evolution with deep reinforcement learning for energy-aware distributed heterogeneous flexible job shop scheduling. *IEEE Trans Syst Man Cybern Syst*, 2024, 54: 201–211
- 30 Li R, Gong W, Lu C, et al. A learning-based memetic algorithm for energy-efficient flexible job-shop scheduling with type-2 fuzzy processing time. *IEEE Trans Evol Computat*, 2022, 27: 610–620
- 31 Liao Z W, Gong W Y, Wang L. A hybrid swarm intelligence with improved ring topology for nonlinear equations (in Chinese). *Sci Sin Inf*, 2020, 50: 396–407 [廖作文, 龚文引, 王凌. 基于改进环拓扑混合群体智能算法的非线性方程组多根联解. 中国科学: 信息科学, 2020, 50: 396–407]
- 32 Taillard E. Some efficient heuristic methods for the flow shop sequencing problem. *Eur J Operational Res*, 1990, 47: 65–74
- 33 Zitzler E, Laumanns M, Thiele L, SPEA2: Improving the strength pareto evolutionary algorithm. *TIK Report*, 2001. 103
- 34 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Computat*, 2002, 6: 182–197
- 35 While L, Hingston P, Barone L, et al. A faster algorithm for calculating hypervolume. *IEEE Trans Evol Computat*, 2006, 10: 29–38
- 36 Van Nostrand C. Design of experiments using the Taguchi approach: 16 steps to product and process improvement. *Technometrics*, 2002, 44: 289–289

A swarm intelligence optimization algorithm for human-robot collaborative energy-efficient shop scheduling

WANG Ling, LI Rui & CHEN JingFang

Department of Automation, Tsinghua University, Beijing 100084, China

With the intelligentization of manufacturing enterprises, the new development mode of “human-robot-cyber” comprehensive interconnection and integration has become the direction of the new generation of intelligent manufacturing. Collaborative manufacturing between workers and robots has become a common practice in modern manufacturing workshops, leading to increased challenges in resource allocation due to the departure from traditional, human-machine independent production scheduling mode. This complexity poses new challenges for algorithm design. Therefore, this paper addresses the characteristics of human-robot collaborative energy-efficient shop scheduling problem. The optimization objectives are to minimize the maximum completion time and total energy consumption. To achieve this, a swarm intelligence optimization algorithm is proposed. This algorithm features the following components: Firstly, a two-stage collaborative search framework is present. In the first stage, swarm intelligence optimization is employed to achieve rapid convergence and build an elite archive. In the second stage, a feedback collaborative search is conducted near elite solutions to explore a wide range of solutions. Secondly, multiple collaborative search operators are introduced to address human-robot resource allocation issues for rapid convergence. Lastly, in consideration of the problem’s characteristics, problem-specific knowledge is extracted, and a multi-strategy collaborative initialization method is designed to provide a high-quality initial population. Additionally, multiple collaborative local search operations are proposed for efficient optimization of the two objectives. To validate the effectiveness of the proposed algorithm, 15 cases of varying sizes are generated. The conducts parameter tuning, ablation, and comparative simulation experiments are adopted. The results demonstrate that the various collaborative methods proposed in this paper effectively enhance algorithm performance and search quality. Through comparisons with the latest relevant algorithms, the effectiveness of the proposed algorithm is evaluated, providing reliable guidance for scheduling and production in human-robot collaborative manufacturing.

intelligent manufacturing, human-robot collaborative shop scheduling, energy-efficient scheduling, cooperative optimization

doi: [10.1360/SST-2023-0341](https://doi.org/10.1360/SST-2023-0341)