

控制与决策

Control and Decision

基于新型蛙跳算法的低碳混合流水车间调度

雷德明, 杨冬婧

引用本文:

雷德明, 杨冬婧. 基于新型蛙跳算法的低碳混合流水车间调度[J]. *控制与决策*, 2020, 35(6): 1329–1337.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1162>

您可能感兴趣的其他文章

Articles you may be interested in

基于总加权完成时间的可重入混合流水车间调度问题

Reentrant hybrid flowshop scheduling problem based on total weighted completion time

控制与决策. 2018, 33(12): 2218–2226 <https://doi.org/10.13195/j.kzyjc.2017.0907>

基于新型教学优化算法的低碳柔性作业车间调度

Novel teaching–learning–based optimization algorithm for low carbon scheduling of flexible job shop

控制与决策. 2017, 32(9): 1621–1627 <https://doi.org/10.13195/j.kzyjc.2016.1008>

求解PFSP的双种群协同学习算法

Double population co–learning algorithm for permutation flow–shop scheduling problems

控制与决策. 2017, 32(1): 12–20 <https://doi.org/10.13195/j.kzyjc.2015.1568>

考虑多约束的混合流水车间MOJ调度

Scheduling multiple orders per job with various constraints for hybrid flow shop

控制与决策. 2016, 31(5): 776–782 <https://doi.org/10.13195/j.kzyjc.2015.0399>

基于GMOGSO的多目标流水车间调度问题

Multi–objective flow shop scheduling problem based on GMOGSO

控制与决策. 2016, 31(10): 1772–1778 <https://doi.org/10.13195/j.kzyjc.2015.1309>

基于新型蛙跳算法的低碳混合流水车间调度

雷德明[†], 杨冬婧

(武汉理工大学 自动化学院, 武汉 430070)

摘要: 针对低碳混合流水车间调度问题(HFSP),提出一种新型蛙跳算法(SFLA)以同时最小化总能耗和总延迟时间. 该算法将种群内最差的部分解排斥在模因组之外,运用模因组构建和模因组搜索新策略产生新解,同时利用搜索过程产生的优化数据替换种群中未进入模因组的差解并更新外部档案以提高求解质量. 通过算例验证新型蛙跳算法解决该问题的有效性,实验结果分析表明,新型SFLA对于所研究的低碳HFSP具有较强的搜索能力和优势.

关键词: 低碳混合流水车间调度; 蛙跳算法; 保留解集; 模因组

中图分类号: TP18

文献标志码: A

A novel shuffled frog-leaping algorithm for low carbon hybrid flow shop scheduling

LEI De-ming[†], YANG Dong-jing

(School of Automation, Wuhan University of Technology, Wuhan 430070, China)

Abstract: For the low carbon hybrid flow shop scheduling problem(HFSP), a novel shuffled frog-leaping algorithm (SFLA) is proposed to minimize simultaneously total energy consumption and total tardiness. Some worst solutions of population are excluded out of memplexes. New solutions are generated by using new strategies for memplex construction and memplex search. Optimization data of search process are utilized to substitute for the worst solutions out of memplexes and update archive to improve solution quality. A series of examples are given to demonstrate the effectiveness of the new SFLA. The analysis of computational results show that the new SFLA has strong search ability and advantages in solving the low carbon HFSP.

Keywords: low carbon hybrid flow shop scheduling; shuffled frog-leaping algorithm; the set of the saved solutions; memplex

0 引 言

低碳调度作为一类综合考虑低碳要求和制造活动目的且子问题较多的复杂优化问题,其早期研究主要集中于单机环境. 近年来,尤其是最近两三年,低碳并行机调度、低碳流水车间调度和低碳作业车间调度受到研究者的广泛关注,涌现出大量的研究成果^[1-26]. 由于问题的高度复杂性,研究者大多采用智能优化算法对问题求解. 目前,各种智能算法,如遗传算法(GA^[9-12])、蛙跳算法(shuffled frog-leaping algorithm, SFLA^[13-15])、果蝇算法^[16]、粒子群算法^[17]、教学优化算法^[18-19]和蚁群优化算法^[20]等已成功地应用于低碳调度.

混合流水车间调度问题(HFSP)是一类离散工业和流程工业广泛存在的问题. 低碳HFSP通常包括3

个子问题: 1) 调度子问题; 2) 机器分配子问题,为工件分配合适的机器; 3) 速度选择子问题,确定工件加工时相应机器的加工速度. 3个子问题相互影响、相互制约,比一般的HFSP更复杂. 目前,低碳HFSP研究重点关注于能耗与总延迟时间等指标之间的冲突关系. Tang等^[17]应用一种改进的粒子群算法解决柔性流水车间节能调度问题. Lei等^[18]提出了一种新型教学优化算法,在最小化总延迟时间的同时兼顾总能耗的优化. Lin等^[19]建立了加工参数优化和流水车间调度的集成模型,并设计了一种教学优化算法以最小化最大完成时间和碳排放. Luo等^[20]给出了一种求解考虑电力消耗成本HFSP的蚁群优化算法. Mansouri等^[21]考虑以最大完成时间和总能耗为目标的两机流水线调度问题,通过机器调速改变能耗,提出了混合

收稿日期: 2018-08-30; 修回日期: 2018-10-25.

基金项目: 国家自然科学基金项目(61573264).

责任编辑: 王凌.

[†]通讯作者. E-mail: deminglei11@163.com.

整数线性多目标优化模型. 针对电耗和材料损耗的问题, Zeng等^[22]提出了一种基于非劣排序遗传算法-II(NSGA-II)^[23]的HFSP. Dai等^[24]在柔性流水车间环境下提出了一种遗传-模拟退火算法, 探讨了总能耗与最大完成时间之间的冲突关系.

SFLA于2003年由Eusuff等^[27]提出, 作为一种模拟青蛙活动的群体智能算法, 它具有概念简单、参数少、计算速度快且全局寻优能力强等显著特点, 已成功地应用于装配序列规划^[28]、车间布局^[29]、旅行商问题^[30]和制造系统调度^[13-15, 31-33]等问题求解, 在低碳调度领域也出现了较多的应用成果^[13-15]. 艾子义等^[13]运用一种新型SFLA来最小化柔性作业车间调度的总碳排放, 该算法运用记忆保留搜索所得一定数量的最优解, 采取新的种群划分方法, 应用新的搜索策略, 同时取消种群重组以简化算法. Lei等^[14]提出了一种新型SFLA以最小化机器负荷和总能耗. 张晓星等^[15]针对柔性作业车间调度问题改进了SFLA以同时优化最大完成时间和总能耗. 这些应用显示了SFLA在求解低碳调度方面较强的搜索优势和能力, 只是SFLA较少用来解决低碳HFSP, 而且现有SFLA对搜索过程的优化数据利用不够充分, 丢弃了大量的数据.

现有研究很少利用优化数据, 文献[24]和文献[34]提出了一种利用优化数据的方法, 即将子代与父代合并然后从中选出最优解作为新种群. 本文提出一种新的利用优化数据的方法, 即将原有种群中的一部分较差解与优化数据合并, 然后从中选出一定数量的较好解作为种群当中的解.

针对具有总能耗和总延迟时间的低碳HFSP, 本文提出一种合理利用算法搜索过程优化数据的有效策略, 进而提出一种新型SFLA. 该算法应用模因组构建新策略、模因组搜索新过程以及种群重构的新过程对问题进行求解. 通过大量的计算实验, 对所得结果分析表明, 新型SFLA在低碳HFSP求解方面具有较强的搜索优势.

1 问题描述

低碳HFSP描述如下: n 个独立工件 J_1, J_2, \dots, J_n 按照阶段1, 阶段2, \dots , 阶段 m 的顺序依次加工, $m \geq 2$. 每个阶段 k 的变速机集合为 S_k , 至少有一个阶段存在一台以上的变速机, 机器 $M_{kj} \in S_k$ 表示阶段 k 的第 j 台机器, $j = 1, 2, \dots, |S_k|$. 假设每台变速机具有 d 种速度, 速度集合为 $V = \{v_1, v_2, \dots, v_d\}$, 机器的加工速度在工件的加工过程中不能改变. 生产过程中, 机器存在两种不同状态: 加工状态和空闲状态. 机器

$M_{kj} \in S_k$ 不加工工件时, 处于空闲状态, 单位时间机器瞬时能耗为 SP_{kj} ; 当机器 M_{kj} 以速度 v_l 加工时, 单位时间能耗为 PP_{kjl} .

每个工件 J_i 在机器 $M_{kj} \in S_k$ 上有一个给定的标准加工时间 p_{ijk} , 当工件 J_i 在机器 $M_{kj} \in S_k$ 上以速度 v_l 加工时, 相应的实际加工时间 $p_{ijkl} = p_{ijk}/v_l$. 一个工件至少在一个阶段上加工, 根据加工路径, 它可能跳过其中一个或几个阶段, 当工件 J_i 跳过阶段 k 时, 它在阶段 k 的加工时间 $p_{ijkl} = 0$.

关于 p_{ijkl} 与 PP_{kjl} 的关系, Ding等^[4]给出了一种假设, 其描述如下:

如果工件 J_i 在机器上以更高速度加工, 它的加工时间将缩短但能耗上升, 即 $\forall v_l > v_g, l, g \in \{1, 2, \dots, d\}$, 则 $p_{ijkl} < p_{ijk}, PP_{kjl} \times p_{ijkl} > PP_{kjg} \times p_{ijk}$.

同时还应满足如下约束: 所有机器和工作从零时刻起可用; 每个工件同一时刻只能在一台机器上加工; 每台机器同一时刻只能加工一个工件; 准备时间包含在加工时间内; 工件在机器上的加工一旦开始, 不容许中断; 缓冲区大小没有限制; 同一工件在两个阶段之间的加工不能重叠等.

问题的目标函数为

$$f_1 = \text{TEC} = \int_0^{C_{\max}} \left(\sum_{k=1}^m \sum_{j=1}^{|S_k|} \sum_{l=1}^d PP_{kjl} y_{kjl}(t) + \sum_{k=1}^m \sum_{j=1}^{|S_k|} SP_{kj} z_{kj}(t) \right) dt, \quad (1)$$

$$f_2 = \text{TT} = \sum_{i=1}^n \max\{C_i - D_i, 0\}. \quad (2)$$

其中: $y_{kjl}(t)$ 和 $z_{kj}(t)$ 均为二进制变量. 如果机器 $M_{kj} \in S_k$ 在时间 t 以速度 v_l 加工, 则 $y_{kjl}(t) = 1$, 否则 $y_{kjl}(t) = 0$; 如果机器 $M_{kj} \in S_k$ 在时间 t 处于空闲状态, 则 $z_{kj}(t) = 1$, 否则 $z_{kj}(t) = 0$. C_i 、 D_i 表示工件 J_i 的完成时间和交货期, 目标函数 f_1 为总能耗, f_2 为总延迟时间.

Pareto支配是多目标优化的基本概念, 假设优化问题的目标总数为 G 且都是最小化目标, 如果解 x 和 y 满足 $\forall i \in \{1, 2, \dots, G\}, f_i(x) \leq f_i(y)$ 且 $\exists i \in \{1, 2, \dots, G\}, f_i(x) < f_i(y)$, 则解 x 支配 y 或者 y 支配 x ; 如果解 x 不受某集合内的任何解支配, 则解 x 关于该集合是非劣的.

2 基于新型蛙跳算法的低碳HFSP

在SFLA中, 问题的解 x 用青蛙的位置表示, 虚拟青蛙的集合为可能解的集合. SFLA具体过程描述如下: 产生初始种群 P 之后, 种群中的解按适应度值降

序排序, 将整个种群划分为 s 个模因组; 然后执行模因组搜索和种群重构, 重复初始化之后的各个步骤, 直到终止条件得到满足。

SFLA 在多目标搜索过程中产生了大量的优化数据, 除了部分精英个体或者非劣解保留在外部档案外, 其他数据常常被丢弃, 如何利用这些数据来改善算法的求解质量, 是一个值得探讨的课题。本文给出一种常被丢弃的数据的利用策略, 在此基础上, 提出一种新型 SFLA 以解决低碳 HFSP。

2.1 编码和解码

现有研究很少单独对机器分配子问题编码, 通常根据工件排列中工件的顺序^[35]或者贪婪策略^[36]进行; 同时由于每台机器都有多种不同的速度可供选择, 需要考虑速度选择子问题, 对该子问题进行编码。考虑到 3 个子问题之间的相互影响, 采用 3 个编码串表示所有子问题的解。

针对有 n 个工件、 m 个加工阶段和每台机器具有 d 种加工速度的 HFSP, 问题的解由调度串 $[\pi_1, \pi_2, \dots, \pi_n]$ 、机器分配串 $[q_{11}, \dots, q_{1m}, \dots, q_{n1}, \dots, q_{nm}]$ 和速度选择串 $[v_{11}, v_{12}, \dots, v_{1n}, \dots, v_{n1}, \dots, v_{nm}]$ 描述。其中: $\pi \in \{1, 2, \dots, n\}$, $q_{ik} \in S_k$ 为阶段 k 分配给工件 J_i 的机器, $v_{ik} \in V$ 为机器 $q_{ik} \in S_k$ 对应的速度。

解码过程如下: 从工件排列 $[\pi_1, \pi_2, \dots, \pi_n]$ 的第 1 个工件 π_1 开始, 依次安排各工件的加工, 对于工件 π_i , 根据机器分配串和速度选择串确定它在每个阶段的加工机器和相应的加工速度, 然后安排各阶段的加工, 直到所有阶段的加工完毕为止。

2.2 模因组构建

通常, 在种群内的解按适应度值降序排序后, 再按如下过程构建 s 个模因组: 将第 1 只青蛙分到第 1 个模因组 Me_1 , 第 2 只青蛙进入第 2 个模因组 Me_2 , 第 s 只青蛙分配到第 s 个模因组 Me_s , 第 $s+1$ 只青蛙进入第 1 个模因组 Me_1 , 依此进行, 直到所有个体分配到模因组中。

本文提出一种模因组构建的新过程, 其具体描述如下:

1) 对种群内所有解进行 Pareto 支配排序, 将 rank 值位于后 10% 的解保存在集合 Δ 中。

2) 运用锦标赛方法将 rank 值排在前 90% 的解分配到 s 个模因组中, 即随机选择两个解 x_1 、 x_2 进行 Pareto 比较: 如果 x_1 支配 x_2 , 则让 x_1 进入模因组 Me_1 中, 同时将 x_2 放回到种群中; 如果两个解彼此非受支配, 则随机选择一个解进入模因组 Me_1 中, 未选中的解放回到种群中。再随机选择两个解, 重复上述过程,

直到所有能分配的解都分配完毕。

与现有模因组构建方法^[13,31]不同, 上述过程未将种群内的所有解都分配到模因组中, 而是将质量最差的部分解排斥在模因组之外, 这样提高了模因组的整体质量, 有助于各模因组在搜索过程中产生高质量的解。

2.3 模因组搜索

SFLA 主要通过模因组搜索产生新解, 通常以组内最差解 x_w 为优化对象。模因组搜索的具体过程如下: 首先确定模因组内最好解 x_b 、最差解 x_w 以及全局最好解 $x_g \in P$ 。然后根据下式利用 x_b 和 x_w 产生一个新位置 x_w^{new} :

$$x_w^{\text{new}} = x_w + \text{rand} \times (x_b - x_w). \quad (3)$$

如果 x_w^{new} 优于 x_w , 则替换最差解 x_w ; 否则, 利用下式产生新解 x_w^{new} :

$$x_w^{\text{new}} = x_w + \text{rand} \times (x_g - x_w). \quad (4)$$

如果 x_w^{new} 优于 x_w , 则替换最差解 x_w ; 否则, 随机产生一个解直接替换 x_w 。重复上述步骤, 直到达到设定的迭代次数。

式(3)和式(4)中 rand 是区间 $[0, 1]$ 上服从均匀分布的随机数。

也有文献以组内最好解 x_b 为优化对象^[31], 而本文则选择一种新的优化对象。

模因组搜索的具体过程描述如下:

1) 令 $i = 1$ 。

2) 令 $t = 1$ 。

3) 对模因组内所有解进行非劣排序, 随机选择一个非劣解作为模因组内的最好解 x_b 。从模因组内除 x_b 以外的解中随机选择 x_1 和 x_2 , 如果 x_1 受 x_2 支配, 则选择 x_2 作为优化对象 x ; 否则, 选择 x_1 作为优化对象 x 。假设优化对象为种群的第 w 个个体。

4) 对 x_b 和 x 执行全局搜索产生新解 x^{new} , 如果 x^{new} 符合替换条件, 则更新外部档案 Ω , 同时用 x 更新保留解集 Ψ 并转到步骤 7); 否则, 运用 x^{new} 更新保留解集 Ψ 。

5) 如果能从外部档案 Ω 中随机选择一个解 $y \neq x$, 则对 $y \in \Omega$ 和 x 执行全局搜索; 否则, 直接对 x 执行 swap。产生新解 x^{new} , 如果 x^{new} 达到替换要求, 则 $x \leftarrow x^{\text{new}}$, 更新外部档案 Ω , 用 x 更新保留解集 Ψ 并转到步骤 7); 否则, 根据 x^{new} 更新保留解集 Ψ 。

6) 产生新解 $x^{\text{new}} \in \mathcal{N}_{\rho_i}(x)$, 若 x^{new} 达到替换条件, 则 $x \leftarrow x^{\text{new}}$, 更新外部档案 Ω , 同时用 x 更新保留解集 Ψ ; 否则 $\rho_i = \rho_i + 1$ 。当 $\rho_i = 5$ 时, 令 $\rho_i = 1$, 并运

用 x^{new} 更新保留解集 Ψ .

7) $t = t + 1$, 如果 $t \leq \mu$, 则转到步骤3).

8) $i = i + 1$, 如果 $i \leq s$, 则转到步骤2).

上述搜索过程中, $x \leftarrow x^{\text{new}}$ 表示用 x^{new} 替换 x ; μ 为整数; ρ_i 为整数, 其初始值在初始化时确定.

当 x^{new} 支配 x 或者两者彼此不受支配时, x^{new} 满足替换条件.

步骤4) 中的全局搜索过程如下: 产生随机数 rand , 如果 $\text{rand} < 0.7$, 则执行两个解的调度串交叉以产生新解; 如果 $0.7 \leq \text{rand} < 0.85$, 则执行机器分配串的交叉操作; 如果 $\text{rand} \geq 0.85$, 则通过速度选择串交叉获得新解.

步骤5) 中的全局搜索过程如下: 产生随机数 rand , 如果 $\text{rand} < 0.8$, 则对 y 与 x 的调度串进行交叉; 否则, 对两个解的机器分配串进行交叉. 由于速度选择相对简单, 步骤5) 中的全局搜索未对速度选择串进行全局搜索.

步骤4) 和步骤5) 中, 0.7、0.8、0.85 根据实验确定, 交叉作用在两个解 x 与 $x_b(y)$ 之间, 3种交叉操作的具体描述见文献[18]. 与文献[18]一样, 每次交叉只产生一个子代, 且在交叉过程中 $x_b(y)$ 保持不变.

采用4种邻域结构产生新解(由于调度子问题具有更高的复杂性的特点, 它对应于其中的两种邻域结构): 1) 邻域结构 insert 用于改变 x 的调度串 $[\pi_1, \pi_2, \dots, \pi_n]$, 随机确定一个工件 π_j 和一个位置 $k \neq j$, 再将 π_j 插入到新位置. 2) 邻域结构 swap 通过互换一对或多对工件序来产生新解. 3) 邻域结构 change 用于改变机器分配串, 首先构建一个集合 $\Theta = \{q_{ik} | |S_k| > 1, 1 \leq i \leq n, 1 \leq k \leq m\}$; 然后从集合中随机选择一些元素, 假设选择了 q_{ik} , 则从 S_k 中随机选择一台机器替代 q_{ik} . 4) 邻域结构 speed 用于改变部分被选中机器的加工速度, 它通过随机选择速度串中的一些元素, 并为这些元素重新随机确定一个值. 令 \mathcal{N}_1 、 \mathcal{N}_2 、 \mathcal{N}_3 和 \mathcal{N}_4 表示 insert 、 swap 、 change 和 speed , $\mathcal{N}_i(x)$ 为执行 \mathcal{N}_i 所产生的 x 的邻域解集.

与现有选择优化对象的方式^[13]不同, 本文运用锦标赛的方法选择一个除 x_b 以外的解 x 作为优化对象, 这样选择的优化对象质量较高, 有利于产生更高质量的新解, 同时避免优化对象单一化. 由于部分最差解未分配到模因组, 参与模因组搜索的解的质量都较高, 获得新的最好解和新的外部档案成员的可能性更大.

外部档案 Ω 的更新过程如下: 首先判断新解是否与集合 Ω 中的解相同, 若不相同, 则将新解加入外部档案 Ω 中, 然后对集合内的所有解根据目标 f_1 和

f_2 进行 Pareto 比较, 保留非劣解, 剔除受支配解.

保留解集 Ψ 用来保存在模因组搜索过程中未能加入外部档案 Ω 中的解.

利用解 x 和 x^{new} 更新保留解集的具体过程如下:

1) 确定保留解集的最大规模 λ , λ 为集合 Δ 中解的个数.

2) 判断这些解是否与集合 Ψ 中的解相同, 若不相同, 则将解 x 或 x^{new} 加入保留解集 Ψ 中, 对集合内所有解根据目标 f_1 和 f_2 进行 Pareto 比较, 保留非劣解, 剔除受支配解; 若保留解集的规模超过其最大规模, 则计算保留解集中解的拥挤距离, 剔除拥挤距离最小的解, 直到保留解集中解的个数为 λ .

外部档案和保留解集都用来保留 SFLA 搜索过程产生的优化数据, 其中外部档案为精英个体数据, 而保留解则为除精英个体之外的最好解. 由于部分最差解无法参加任何模因组搜索, 这部分解的更新主要利用保留解集来进行.

2.4 算法描述

新型 SFLA 的详细步骤描述如下:

1) 初始化参数: 模因组个数 s , 种群规模 N , 终止条件等.

2) 产生初始种群 P , $\rho_i = 1, i = 1, 2, \dots, N$.

3) 对种群内的所有解进行非劣排序, 确定 rank 值排在后 10% 的 λ 个解.

4) 将剩余的 $N - \lambda$ 个解均分到 s 个模因组中.

5) 执行每一个模因组的搜索过程.

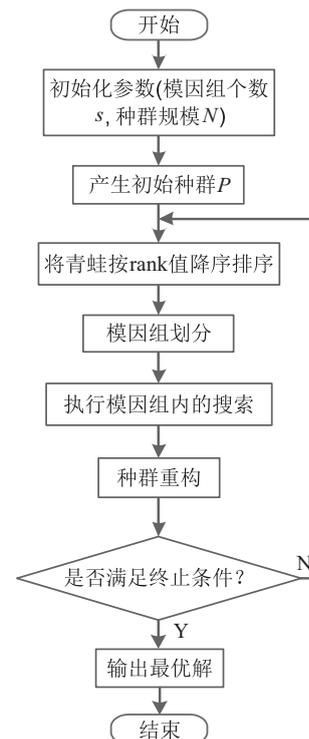


图1 新型 SFLA 流程

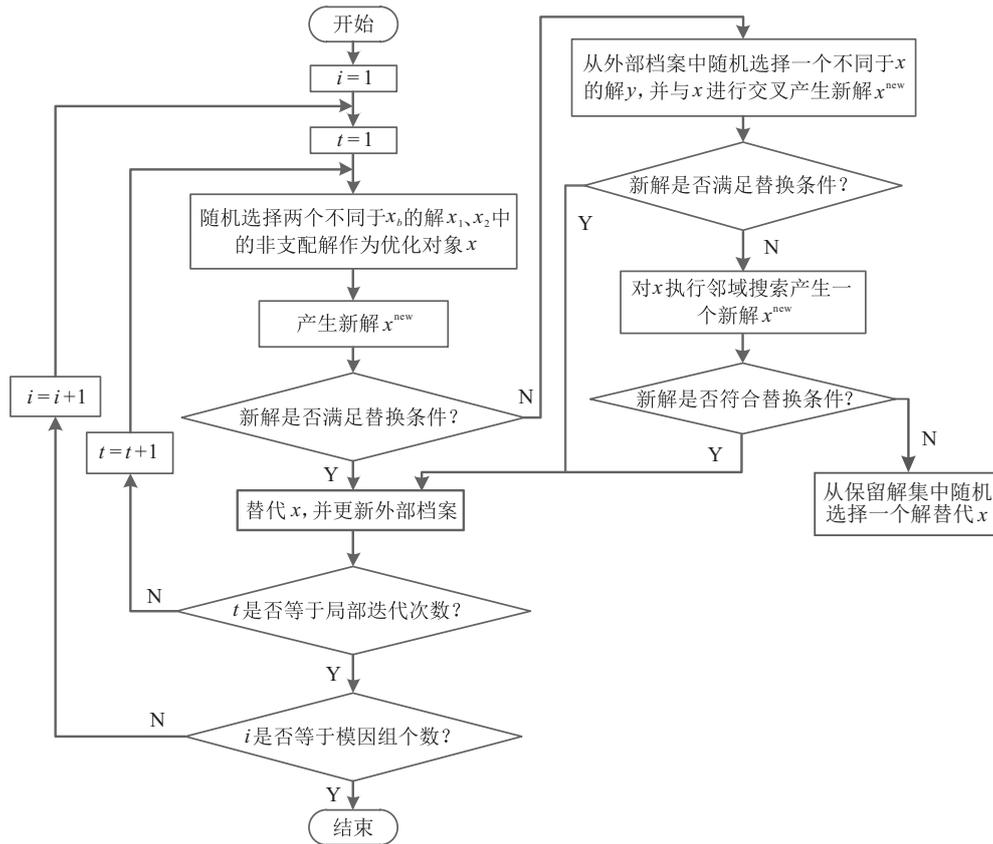


图2 新型SFLA模因组内搜索过程

6) 将保留解集 Ψ 与集合 Δ 合并在一起,进行非劣排序,选择其中rank最小的 λ 个解,然后与进化后的模因组混合,执行种群重构。

7) 如果满足终止条件,则输出外部档案;否则转到步骤3)。

终止条件为最大目标估计次数 \max_it 。

算法流程如图1和图2所示。该算法的时间复杂度为 $O(G \times N^2 \times L)$ 。其中:种群规模为 N , L 表示算法循环部分的循环次数, G 表示目标函数的个数。

种群重构先将保留解集与最差解集合并以更新最差解,再与模因组组合在一起构建新种群。由于部分最差解排斥在模因组外,新型SFLA采取模因组构建新方式、模因组搜索新过程以及种群重构的新策略,这样既利用了搜索过程中的优化数据,提高了模因组内解的质量,又在同样的终止条件下增加了模因组搜索的次数,从而提升了算法搜索效率,有助于改进求解质量。

3 计算实验

为了测试新型SFLA在求解低碳HFSP方面的性能,运用一系列实例进行大量的实验。所有实验在Microsoft Visual C++ 2015编程实现,程序运行环境为8.0G RAM 2.50 GHz CPU的个人计算机。

3.1 测试实例和对比算法

考虑具有 $n = 20, 30, 40, 50, 60, 70, 80, 100, 120$ 和 $m = 2, 4, 8$ 的27个实例,这些实例的数据见文献[18],其中交货期设置为工件在每个阶段的最大加工时间的总和。

选择NSGA-II^[22]和VNS作为对比算法。NSGA-II由Deb等^[23]于2002年提出,该算法在每一代首先对种群 P 进行遗传操作,得到种群 Q ;然后将两个种群合并,通过非劣排序和拥挤距离排序形成新的种群 P ,反复进行直到结束。NSGA-II在多目标柔性作业车间调度^[25]、多目标流水车间调度^[22]等应用广泛,计算结果具有优势。为了解决低碳HFSP,对NSGA-II作如下改变:在文献[22]原有编码方法的基础上增加速度选择部分,速度选择子问题的变异操作为speed。

VNS作为一种智能算法,已成功地用于解决各种调度问题^[37-38],并显示了较强的搜索性能,为此,利用2.3节所描述的4种邻域结构构造一种VNS。

VNS的具体步骤描述如下:

1) 随机产生一个初始解。

2) 重复如下过程直到达到最大目标估计次数:随机产生一个新解 $z \in \mathcal{N}_i(x)$,若 x 和 z 满足 z 支配 x 或与 x 彼此非劣,则 z 替代 x , $i \leftarrow 1$,并更新外部档案;否则 $i \leftarrow i + 1$ 。当 $i = 5$ 时, $i \leftarrow 1$ 。

采用如下3个指标评价算法的计算结果.

1) 距离指标 $DI_R^{[39]}$, 用来衡量非劣解集 Ω_l 中的元素相对于参考集 Ω^* 的距离, 表示为

$$DI_R(\Omega_l) = \frac{1}{|\Omega^*|} \sum_{Y \in \Omega^*} \min\{\phi_{XY} | X \in \Omega_l\}. \quad (5)$$

其中: ϕ_{XY} 表示解 X 与参考集 Ω^* 中元素 Y 在归一化目标空间内的距离,

$$\phi_{XY} = \sqrt{(f_1^*(X) - f_1^*(Y))^2 + \dots + (f_F^*(X) - f_F^*(Y))^2},$$

F 为目标函数的个数, f_i^* 为第 i 个归一化目标, 归一化方法见文献 [40]; 参考集 Ω^* 由所有算法非劣解集并集中的非劣解组成.

显然, $DI_R(\Omega_l)$ 越小, Ω_l 所对应的算法越好.

2) 指标 nd_l , 用来表示 Ω_l 所提供的非劣解的个数.

3) 指标 $\rho_l^{[41]}$, 用来衡量 Ω_l 所提供的非劣解在整个参考集 Ω^* 中所占的比例.

3.2 结果分析

新型 SFLA 具有 3 个参数: 种群规模 N 、模因组个数 s 和最大目标估计次数 \max_it . 根据大量实验, 确定如下参数设置: $N = 60, s = 6, \max_it = 10^5$.

VNS 终止条件与新型 SFLA 完全相同.

关于 NSGA-II, 基于大量实验得到如下参数: 最大代数为 $\max_it/100$, 种群规模为 100, 交叉概率为 0.8, 变异概率为 0.1. 因每代要进行 100 次目标函数估计, 故 NSGA-总的目标函数估计次数也为 10^5 .

3 种算法具有相同的目标函数估计次数, 即终止条件一样.

每个算法关于每个实例运行 10 次. 计算结果如表 1 和表 2 所示, 其中 Ω^* 由并集 $\Omega_1 \cup \Omega_2 \cup \Omega_3$ 中的非劣解组成, $\Omega_1, \Omega_2, \Omega_3$ 分别为 SFLA、NSGA-II 和 VNS 的非劣解集, $DI_R(\Omega_l)$ 、 nd_l 、 ρ_l 分别为 Ω_l 的 3 个指标值, 表 3 给出了算法的计算时间.

表 1 SFLA、NSGA-II 和 VNS 关于指标 $DI_R(\Omega)$ 的计算结果

Instance	SFLA	NSGA-II	VNS	Instance	SFLA	NSGA-II	VNS
20 × 2	0.000	25.14	28.58	60 × 8	19.95	28.79	53.40
20 × 4	0.000	68.57	58.28	70 × 2	11.91	12.54	54.56
20 × 8	0.000	34.96	68.50	70 × 4	0.000	61.33	69.97
30 × 2	1.070	21.23	42.24	70 × 8	0.000	16.45	73.51
30 × 4	0.000	38.92	38.48	80 × 2	9.908	11.53	49.61
30 × 8	0.000	29.04	74.89	80 × 4	56.95	11.16	17.90
40 × 2	1.904	20.93	49.92	80 × 8	7.263	11.45	53.82
40 × 4	16.68	37.49	55.24	100 × 2	4.666	5.510	58.04
40 × 8	17.68	18.43	66.24	100 × 4	55.74	11.04	18.46
50 × 2	5.332	23.74	43.61	100 × 8	6.932	1.270	44.95
50 × 4	0.000	72.38	59.30	120 × 2	1.329	3.050	62.66
50 × 8	23.71	11.93	38.55	120 × 4	53.95	2.088	11.24
60 × 2	0.000	27.74	53.03	120 × 8	0.000	46.51	71.66
60 × 4	25.25	31.94	28.47				

表 2 SFLA、NSGA-II 和 VNS 关于指标 nd_l 和 ρ_l 的计算结果

Instance	SFLA	NSGA-II	VNS	Instance	SFLA	NSGA-II	VNS
20 × 2	77, 1.000	0, 0.000	0, 0.000	60 × 8	31, 0.620	19, 0.380	0, 0.000
20 × 4	32, 1.000	0, 0.000	0, 0.000	70 × 2	34, 0.618	21, 0.382	0, 0.000
20 × 8	18, 1.000	0, 0.000	0, 0.000	70 × 4	35, 1.000	0, 0.000	0, 0.000
30 × 2	58, 0.967	2, 0.033	0, 0.000	70 × 8	17, 1.000	0, 0.000	0, 0.000
30 × 4	41, 1.000	0, 0.000	0, 0.000	80 × 2	25, 0.510	24, 0.490	0, 0.000
30 × 8	25, 1.000	0, 0.000	0, 0.000	80 × 4	10, 0.118	53, 0.624	22, 0.259
40 × 2	39, 0.951	2, 0.049	0, 0.000	80 × 8	8, 0.615	5, 0.384	0, 0.000
40 × 4	27, 0.771	8, 0.229	0, 0.000	100 × 2	33, 0.767	10, 0.233	0, 0.000
40 × 8	10, 0.526	9, 0.474	0, 0.000	100 × 4	11, 0.190	47, 0.810	0, 0.000
50 × 2	33, 0.84	6, 0.154	0, 0.000	100 × 8	11, 0.297	26, 0.703	0, 0.000
50 × 4	31, 1.000	0, 0.000	0, 0.000	120 × 2	21, 0.700	9, 0.300	0, 0.000
50 × 8	28, 0.364	49, 0.636	0, 0.000	120 × 4	7, 0.130	42, 0.778	5, 0.093
60 × 2	47, 1.000	0, 0.000	0, 0.000	120 × 8	7, 1.000	0, 0.000	0, 0.000
60 × 4	16, 0.552	13, 0.448	0, 0.000				

表3 SFLA、NSGA-II和VNS的运行时间

单位: s

Instance	SFLA	NSGA-II	VNS	Instance	SFLA	NSGA-II	VNS
20 × 2	2.848 6	2.928 5	1.530 7	60 × 8	22.844 2	15.015 4	16.525
20 × 4	2.838 8	2.460 8	2.049 2	70 × 2	9.086 8	5.523 1	6.600 8
20 × 8	5.129 4	3.913 7	3.379 6	70 × 4	13.781 5	8.316 4	9.7
30 × 2	3.717 5	2.654 5	2.726 1	70 × 8	27.331	15.164 1	19.760 8
30 × 4	6.196 7	3.869 9	3.477	80 × 2	9.429 1	5.450 3	6.957 5
30 × 8	8.762 3	5.604 6	6.018 9	80 × 4	17.917 6	10.075	12.138 8
40 × 2	4.665 4	3.334 6	3.348 3	80 × 8	30.942 8	18.675 4	25.175 5
40 × 4	6.802 2	5.090 4	5.317 7	100 × 2	11.846	7.022 5	11.080 5
40 × 8	12.81	8.550 6	9.854 9	100 × 4	20.783 6	14.035	16.350 3
50 × 2	6.396 7	4.052 6	4.356 8	100 × 8	43.837 8	30.096 2	38.040 9
50 × 4	8.368 9	7.551 2	6.336 7	120 × 2	16.163 2	10.087 5	14.299 6
50 × 8	15.776 2	14.238 8	12.545 1	120 × 4	29.151 4	20.589 1	26.908 3
60 × 2	7.694 2	5.003 9	5.593 7	120 × 8	56.099 9	63.063 6	48.149 8
60 × 4	11.570 2	7.866 1	7.860 7				

新型SFLA的计算时间比NSGA-II和VNS稍长. 图3为3种算法关于20 × 2、30 × 2、30 × 4、60 × 2运行10次所得所有非劣解的分布图. 从图3中可知,新

型SFLA的大部分解均优于另外两种算法所产生的解. 总之,新型SFLA是求解低碳HFSP竞争力较强的算法.

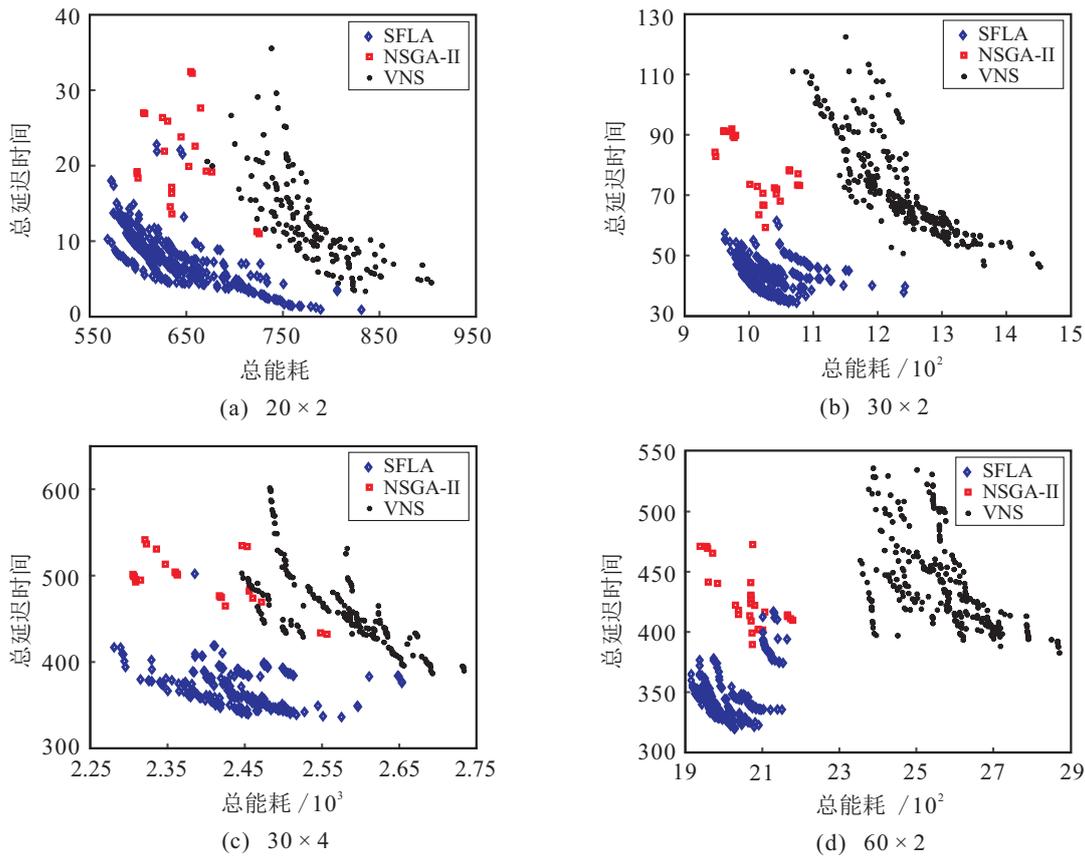


图3 3种算法关于4个实例的非劣解分布图

4 结论

研究低碳HFSP具有非常重要的理论和现实意义. 本文针对具有总能耗和总延迟时间的低碳HFSP, 在对其3个子问题单独编码后, 运用一种新的蛙跳算法对问题进行求解. 该算法将部分最差解排斥在模

因组之外, 并运用优化数据即保留解更新种群中的部分最差解, 而在现有SFLA中, 这些优化数据通常都被丢弃. 最后通过实例验证了新型SFLA的有效性和可行性.

本文将继续深入研究低碳调度问题, 包括低碳分

布式调度,并进一步研究帝国竞争算法等在低碳生产调度方面的应用.另外,分布式调度也是未来研究的主题之一.

参考文献(References)

- [1] 王凌,王晶晶,吴楚格.绿色车间调度优化研究进展[J].控制与决策,2018,33(3):385-391.
(Wang L, Wang J J, Wu C G. Advances in green shop scheduling and optimization[J]. Control and Decision, 2018, 33(3): 385-391.)
- [2] Fang K, Uhan N, Zhao F, et al. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction[J]. Journal of Manufacturing Systems, 2011, 30(4): 234-240.
- [3] Rahimifard S, Seow Y, Childs T. Minimising embodied product energy to support energy efficient manufacturing[J]. CIRP Annals—Manufacturing Technology, 2010, 59(1): 25-28.
- [4] Ding J Y, Song S, Wu C. Carbon-efficient scheduling of flow shops by multi-objective optimization[J]. European Journal of Operational Research, 2015, 248(3): 758-771.
- [5] Gahm C, Denz F, Dirr M, et al. Energy-efficient scheduling in manufacturing companies: A review and research framework[J]. European Journal of Operational Research, 2016, 248(3): 744-757.
- [6] Zheng H Y, Wang L. Reduction of carbon emissions and project makespan by a Pareto-based estimation of distribution algorithm[J]. International Journal of Production Economics, 2015, 164: 421-432.
- [7] Wang J J, Wang L. A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, DOI: 10.1109/TSMC.2017.2788879.
- [8] Li J Q, Sang H Y, Han Y Y, et al. Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions[J]. Journal of Cleaner Production, 2018, 181: 584-598.
- [9] Yildirim M B, Mouzon G. Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm[J]. IEEE Transactions on Engineering Management, 2012, 59(4): 585-597.
- [10] Shrouf F, Ordieres-Meré J, García-Sánchez A, et al. Optimizing the production scheduling of a single machine to minimize total energy consumption costs[J]. Journal of Cleaner Production, 2014, 67(6): 197-207.
- [11] Masmoudi O, Yalaoui A, Ouazene Y, et al. Solving a capacitated flow-shop problem with minimizing total energy costs[J]. International Journal of Advanced Manufacturing Technology, 2017, 90(9): 2655-2667.
- [12] Yin L, Li X, Gao L, et al. A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem[J]. Sustainable Computing Informatics & Systems, 2017, 13: 15-30.
- [13] 艾子义,雷德明.基于新型蛙跳算法的低碳柔性作业车间调度[J].控制理论与应用,2017,34(10):1361-1368.
(Ai Z Y, Lei D M. A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling[J]. Control Theory & Applications, 2017, 34(10): 1361-1368.)
- [14] Lei D, Zheng Y, Guo X. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption[J]. International Journal of Production Research, 2017, 55(11): 3126-3140.
- [15] 张晓星,王艳,严大虎,等.改进的混合蛙跳算法求解柔性作业车间调度[J].系统仿真学报,2017,29(9):2093-2099.
(Zhang X X, Wang Y, Yan D H, et al. Improved shuffled frog-leaping algorithm for solving flexible job shop scheduling problem[J]. Journal of System Simulation, 2017, 29(9): 2093-2099.)
- [16] Liu Q, Chekem F O, Zhan M, et al. A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint[J]. Journal of Cleaner Production, 2017, 168: 668-678
- [17] Tang D, Dai M, Salido M A, et al. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization[J]. Computers in Industry, 2016, 81(C): 82-95.
- [18] Lei D M, Gao L, Zheng Y. A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop[J]. IEEE Transactions on Engineering Management, 2018, 65(2): 330-340.
- [19] Lin W, Yu D Y, Zhang C, et al. A multi-objective teaching-learning-based optimization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint[J]. Journal of Cleaner Production, 2015, 101: 337-347.
- [20] Luo H, Du B, Huang G Q, et al. Hybrid flow shop scheduling considering machine electricity consumption cost[J]. International Journal of Production Economics, 2013, 146(2): 423-439.
- [21] Mansouri S A, Aktas E, Besikci U. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption[J]. European Journal of Operational Research, 2016, 248(3): 772-788.
- [22] Zeng Z Q, Hong M N, Man Y, et al. Multi-objective optimization of flexible flow shop scheduling with batch process-consideration total electricity consumption and material wastage[J]. Journal of Cleaner Production, 2018, 183: 925-939.
- [23] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [24] Dai M, Tang D B, Giret A, et al. Energy-efficient

- scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm[J]. *Robotics and Computer-Integrated Manufacturing*, 2013, 29(5): 418-429.
- [25] 蒋增强, 左乐. 低碳策略下的多目标柔性作业车间调度[J]. *计算机集成制造系统*, 2015, 21(4): 1023-1031. (Jiang Z Q, Zuo L. Multi-objective flexible job-shop scheduling based on low-carbon strategy[J]. *Computer Intergrated Manufacturing Systems*, 2015, 21(4): 1023-1031.)
- [26] 唐立力. 求解低碳调度问题的改进型候鸟优化算法[J]. *计算机工程与应用*, 2016, 52(17): 166-171. (Tang L L. Improved migrating birds optimization algorithm to solve low-carbon scheduling problem[J]. *Computer Engineering and Applications*, 2016, 52(17): 166-171.)
- [27] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. *Journal of Water Resources Planning and Management*, 2003, 129(3): 210-225.
- [28] 王松, 孙振忠, 郭建文, 等. 基于混合蛙跳算法的复杂产品装配序列规划[J]. *计算机集成制造系统*, 2014, 20(12): 2991-2999. (Wang S, Sun Z Z, Guo J W, et al. Research on assembly sequence planning based on shuffled frog leaping algorithm[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(12): 2991-2999.)
- [29] 刘琼, 许金辉, 张超勇. 基于改进蛙跳算法的鲁棒性车间布局[J]. *计算机集成制造系统*, 2014, 20(8): 1879-1886. (Liu Q, Xu J H, Zhang C Y. Robust layout of floor shop based on improved shuffled frog leaping algorithm[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(8): 1879-1886.)
- [30] 罗雪晖, 杨焯, 李霞. 改进混合蛙跳算法求解旅行商问题[J]. *通信学报*, 2009, 30(7): 130-135. (Luo X H, Yang Y, Li X. Modified shuffled frog-leaping algorithm to solve traveling salesman problem[J]. *Journal on Communications*, 2009, 30(7): 130-135.)
- [31] Lei D M, Guo X P. A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents[J]. *Expert Systems with Applications*, 2015, 42(23): 9333-9339.
- [32] Li J, Pan Q, Xie S. An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems[J]. *Applied Mathematics & Computation*, 2012, 218(18): 9353-9371.
- [33] 潘玉霞, 潘全科, 李俊青. 蛙跳优化算法求解多目标无等待流水线调度[J]. *控制理论与应用*, 2011, 28(10): 1363-1370. (Pan Y X, Pan Q K, Li J Q. Shuffled frog-leaping algorithm for multi-objective no-wait flowshop scheduling[J]. *Control Theory & Applications*, 2011, 28(10): 1363-1370.)
- [34] Piroozfard H, Wong K Y, Wong W P. Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm[J]. *Resources Conservation & Recycling*, 2018, 128: 267-283.
- [35] Ruizab R, Vazquez-Rodriguez J A. The hybrid flow shop scheduling problem[J]. *European Journal of Operational Research*, 2010, 205(1): 1-18.
- [36] Rashidi E, Jahandar M, Zandieh M. An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines[J]. *International Journal of Advanced Manufacturing Technology*, 2010, 49(9/10/11/12): 1129-1139.
- [37] Vanchipura R, Sridharan R, Babu A S. Improvement of constructive heuristics using variable neighbourhood descent for scheduling a flow shop with sequence dependent setup time[J]. *Journal of Manufacturing Systems*, 2014, 33(1): 65-75.
- [38] Siqueira E C D, Souza M J F, Souza S R D. A multi-objective variable neighborhood search algorithm for solving the hybrid flow shop problem[J]. *Electronic Notes in Discrete Mathematics*, 2018, 66: 87-94.
- [39] Knowles J, Corne D. On metrics for comparing nondominated sets[C]. *Proceedings of the 2002 Congress on Evolutionary Computation*. Honolulu, 2002: 711-716.
- [40] Ishibuchi H, Yoshida T, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 204-223.
- [41] Lei D M. Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems[J]. *International Journal of Advanced Manufacturing Technology*, 2008, 37(1/2): 157-165.

作者简介

雷德明(1968—), 男, 教授, 博士生导师, 从事智能系统优化与控制等研究, E-mail: deminglei11@163.com;

杨冬婧(1994—), 女, 硕士生, 从事制造系统智能优化与调度的研究, E-mail: niceyjdj@163.com.

(责任编辑: 李君玲)