

求解多目标不相关并行机调度问题的多群体人工蜂群算法

雷德明[†], 杨海

(武汉理工大学自动化学院, 武汉430070)

摘要: 针对具有预防性维修(PM)和顺序相关准备时间(SDST)的不相关并行机调度问题, 提出了一种多群体人工蜂群算法(MABC)以同时最小化完工时间和总延迟时间. 该算法将雇佣蜂分割成 s 个雇佣蜂群, 除最差雇佣蜂群外, 每个雇佣蜂群都对应一个跟随蜂群. 结合两个目标函数、PM和SDST的特征设计了三种邻域搜索, 采用全局搜索和邻域搜索的不同组合实现雇佣蜂阶段和跟随蜂阶段, 并引入了两种淘汰过程. 通过大量实验测试MABC新策略与搜索性能, 计算结果验证了新策略的有效性和MABC的搜索优势

关键词: 预防性维修; 顺序相关准备时间; 不相关并行机调度; 人工蜂群算法

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0775

Multi-colony artificial bee colony algorithm for multi-objective unrelated parallel machine scheduling problem

LEI De-ming[†], YANG Hai

(College of Automation, Wuhan University of Technology, Wuhan430070, China)

Abstract: To solve the unrelated parallel machine scheduling problem (UPMSP) with preventive maintenance (PM) and sequence dependent setup time (SDST), a multi-colony artificial bee colony (MABC) algorithm is proposed to minimize makespan and total tardiness simultaneously. In this algorithm, s employed bee colonies are obtained by division. Except for the worst employed bee colony, each employed bee colony corresponds to a onlooker bee colony. Combined with the characteristics of two objective functions, PM and SDST, three kinds of neighborhood searches were designed. Different combinations of global search and neighborhood search are used to implement the employed bee phase and the onlooker bee phase, and two elimination processes are applied. Experimental research on the new strategy and search performance of MABC is carried out, and computational results demonstrate the effectiveness of the new strategy and the search advantage of MABC.

Keywords: preventive maintenance; sequence dependent setup time; unrelated parallel machine scheduling; artificial bee colony algorithm

0 引言

并行机调度问题(PMSP)是一种典型的调度问题^[1], 广泛存在于轮胎行业^[2]、纺织行业^[3]和烟草行业^[4]等实际制造业. 它可以看作扩展的单机调度或者柔性多阶段生产系统的基本单元^[5,6]. 根据并行机功能, PMSP可分为同速PMSP, 异速PMSP和UPMSP. 由于工件在不同机器上的加工速度各异, UPMSP求解难度更大, 是目前PMSP研究的焦点, 考虑各种实际加工约束如PM和SDST的UPMSP受到研究者广泛关注.

通常假设并行机在整个生产过程中持续可用, 然而, 实际制造过程中, 经常因为随机故障或可能

失效等导致机器不可用. 由于维修能有效预防机器的潜在故障和严重事故, 有必要研究考虑机器维修可用性约束的UPMSP. 近年来, 相关研究取得了一些进展^[7-12]. 自Parker等^[13]的开创性工作之后, 具有SDST的UPMSP引起广泛关注^[14-24].

以上考虑PM或SDST的UPMSP研究以单目标优化为主^[7,8,11,14-22], 且以优化最大完成时间为主, 很少考虑以总延迟时间等为目标UPMSP. 多目标UPMSP研究取得了一定进展^[9,12,24-25], 主要应用遗传算法^[24]、蚁群算法^[24]和教学优化算法^[25]等优化最大完成时间和提前/延迟惩罚总成本等目标.

由于实际UPMSP通常存在多个目标如最大完成

收稿日期: 2020-06-15; 修回日期: 2021-01-12.

基金项目: 国家自然科学基金项目(61573264)

[†]通讯作者. E-mail: deminglei11@163.com.

时间和总延迟时间, 这些目标分别描述了生产系统的性能和按时交货水平, 而生产性能和按时交货水平等往往彼此冲突, 使得相应的目标函数彼此冲突, 需要在目标间协调折中, 从而增加了问题的求解难度; 此外, 现有研究往往单独处理具有PM的UPMSP或具有SDST的UPMSP, 很少将PM和SDST同时结合到UPMSP中^[11,12]. 实际上, 这两种约束在生产制造过程中常见且往往同时存在. 综上所述, 有必要研究具有PM和SDST的多目标UPMSP.

与遗传算法和蚁群算法相比, ABC具有参数少、实现简单等特点. 该算法已广泛应用于各种调度问题包括PMSP^[17,18,26]、流水车间调度^[27,28]和作业车间调度^[29-31]等的求解. 不过, 尽管ABC在具有PM或SDST的UPMSP的求解方面具有一些应用, 该算法较少解决考虑PM和SDST两种约束的多目标UPMSP.

针对具有PM和SDST的UPMSP, 提出了一种M-ABC算法以同时最小化完工时间和总延迟时间. 该算法采用启发式初始化方法, 将雇佣蜂分割成 s 个雇佣蜂群, 除最差雇佣蜂群外, 每个雇佣蜂群对应一个跟随蜂群, 结合两个目标、PM和SDST的特征设计三种邻域搜索, 运用全局搜索和邻域搜索的不同组合实现雇佣蜂阶段和跟随蜂阶段, 并引入了两种淘汰过程. 通过大量的实验, 验证了新策略对MABC性能的影响以及MABC在UPMSP求解方面的搜索优势.

1 问题描述

具有PM和SDST的UPMSP描述如下: 有 n 个工件 $J_1, J_2, \dots, J_n$ 由 m 台不相关并行机 $M_1, M_2, \dots, M_m$ 加工, 工件可在 m 台机器中任何一台机器加工. 工件 J_j 的加工时间 p_{kj} 取决于其处理机器 M_k 的性能, 不同机器加工时间通常不同. d_j 表示 J_j 的交货期.

为了保证机器持续可用, 在计划阶段确定固定的维修周期, 即对每台机器 M_k , 每运行一段时间进行一次PM活动, w_k 表示每次维修的持续时间, 工件只能在两个连续维修活动之间的时间间隔内加工, 间隔时间为 u_k , 该机器的PM周期表示为 T_k , $T_k = w_k + u_k$ ^[11]. 如果某工件无法在间隔时间内完成加工, 则安排其在下一个时间间隔内完成加工.

对于SDST, st_{kij} 表示机器 M_k 在工件 J_i 加工完毕后再加工工件 J_j 时的准备时间, st_{k0j} 表示机器 M_k 维修完成后第一个加工工件 J_i 时的准备时间, st_{ki0} 是机器 M_k 加工完工件 J_i 后执行维修活动时的准备时间.

此外, 具有如下工件和机器约束: (1)任意工件和机器在零时可用; (2)每个工件只加工一次; (3)工件加

工操作和机器PM不可被中断; (4)每个机器在同一时间只能加工一个工件.

具有PM和SDST的UPMSP由机器分配和调度两个子问题组成, 同时考虑如下两个目标:

$$\min f_1 = C_{\max} = \max \{C_j | j = 1, 2, \dots, n\} \quad (1)$$

$$\min f_2 = \sum_{j=1}^n \max \{C_j - d_j, 0\} \quad (2)$$

其中 f_1 表示最大完成时间, f_2 表示总延迟时间, C_j 表示工件 j 的完成时间.

2 MABC求解考虑PM和SDST的UPMSP

ABC将蜜蜂分为三类: 雇佣蜂、跟随蜂和侦察蜂, 包括三个阶段: 雇佣蜂阶段, 跟随蜂阶段和侦查蜂阶段.

雇佣蜂阶段, 对于每个解 x , 产生新解 y

$$y = x + \phi(x - x') \quad (3)$$

其中 ϕ 是 $[-1,1]$ 内随机数的向量, $x' \neq x$ 表示从种群中随机选择的解.

跟随蜂阶段, 每个跟随蜂通过轮盘赌选择食物源, 然后根据上式产生新解. 若某个雇佣蜂连续未更新的次数超过规定的阈值 $Limit$, 则雇佣蜂变成侦查蜂, 随机产生一个食物源, 替代该雇佣蜂.

用贪心选择确定新解是否替代旧解: 若新解 y 的花蜜量大于 x 的花蜜量, 则 y 替代 x ; 否则, 仍保留 x .

针对所研究的UPMSP, 提出一种MABC, 该算法将所有雇佣蜂分割为 s 个雇佣蜂群, 构建 $s-1$ 个跟随蜂群, 并采用全局搜索和邻域搜索的不同组合实现这些蜂群内的搜索.

2.1 初始化

具有PM和SDST的UPMSP由机器分配和调度两个子问题组成, 常使用双串^[34-37]描述问题的解. 本文采用双串表示法. 对于具有 n 个工件和 m 台机器的UPMSP, 每个解由机器分配串 $[M_{\theta_1}, M_{\theta_2}, \dots, M_{\theta_n}]$ 和调度串 $[\pi_1, \pi_2, \dots, \pi_n]$ 组成. 其中 M_{θ_j} 表示分配给工件 J_j 的并行机, $1 \leq \theta_j \leq m$, $\pi_i \in \{1, 2, \dots, n\}$.

解码过程如下: 首先根据机器分配串确定为每个工件所分配的机器, 然后根据调度串和PM条件, 处理工件加工和每台机器的维修活动.

EB表示所有雇佣蜂的集合, 它由 N 个初始解组成, 每个初始解采用如下方式产生: 随机选择一个目标, 对于选定的目标 f_i , 通过启发式构造解^[12].

2.2 多群体产生

MABC中, N 个雇佣蜂分割成 s 个雇佣群 EB_g , $g = 1, 2, \dots, s$. 具体划分过程如下: 首先, 根据Deb等

[40] 的方法确定所有解的 $rank$ 和拥挤距离.然后对所有解先根据 $rank$ 进行升序排序,相同 $rank$ 的解按拥挤距离降序排序;最后,从第一个解开始,前 $|EB_1|$ 个解分配到 EB_1 中,紧接着的 $|EB_2|$ 个解构成 EB_2 ,依次进行,直到所有解都加入到相应的雇佣蜂群中.

显然, $EB_1, EB_2, \dots, EB_s$ 也是基于 $rank$ 和拥挤距离降序排列的, EB_s 由 EB 的最差解组成,使得其本身也是 s 个雇佣蜂群中最差的群体.

通常跟随蜂的数量等于雇佣蜂的数量.本文中,跟随蜂数量为 $\bar{N}$, $\bar{N} \neq N$,所有跟随蜂划分为 $s-1$ 个跟随蜂群 $OB_h, h=1, 2, \dots, s-1, \bar{N} = \sum_{h=1}^{s-1} |OB_h|$;而且 OB_g 只是跟随 EB_g ,即 OB_g 中的每个跟随蜂只选择 EB_g 中的解,而不是 $EB_h, h \neq g$,这样, EB_s 被排除在跟随蜂的选择之外,即跟随蜂不选择 $x \in EB_s$ 作为食物源,从而避免了在一些低质量解上浪费计算资源.由于至少需一个跟随蜂群即 $(s-1) \geq 1$,因此,至少需要2个雇佣蜂群.

由于采用多个雇佣蜂群和跟随蜂群,雇佣蜂阶段和跟随蜂阶段可实现差异化搜索,为此,引入全局搜索和邻域搜索的多种组合以实现各蜂群内的搜索.

2.3 雇佣蜂阶段

雇佣蜂阶段描述如下:每个雇佣蜂 $x \in EB$

(1) 如果 $x \in EB \setminus EB_s$,则

若 $x \in \bigcup_{g=1}^r EB_g$,则随机选择 $y \in EB_1$;否则随机选择 $y \in \bigcup_{g=1}^r EB_g$.然后依次执行 x 的局部搜索 NS_1 以及 x 和 y 之间的全局搜索,得到新解 z ,如果替换条件成立,则 $x = z$ 并用 x 更新 Ω

(2) 否则,如果随机数 $\alpha < 0.5$,则随机选择 $y \in EB \setminus EB_s$,对 x 执行 NS_1 和 x 和 y 之间的全局搜索,如果新解 z 满足替换条件,则 $x = z$ 并用 x 更新 Ω ;否则,对 x 执行从 NS_2 和 NS_3 随机选择的邻域搜索.得到新解 z .如果替换条件成立,则 $x = z$ 并用 x 更新 Ω .

(3) 更新 γ_x ,并对 x 应用第一种淘汰过程.

其中 γ_x, τ 是整数,如果 $s = 2$,则 $\tau = 1$.如果 $s > 2$, $\tau = 2$.如果 x 被新解更新替代,则 $\gamma_x = 0$;否则, $\gamma_x = \gamma_x + 1$.

替换条件:如果 $z \succ x$ 或 z 和 x 彼此非劣,则 z 替换 x ,同时运用 z 更新档案 Ω .

其中, $z \succ x$ 表示 z 支配 x ,如果 $i = 1, 2, f_i(z) \leq f_i(x)$,且 $\exists i = 1$ 或 $2, f_i(z) < f_i(x)$,则 $z \succ x$.如果 z 不支配 x ,也不被 x 支配,则 z 和 x 彼此非劣.

外部档案 Ω 用来保留MABC搜索过程产生的非劣解,其更新方式如下:将 x 加入 Ω ,根据Pareto支配比较 Ω 内所有解,并从 Ω 中剔除所有受支配解.

邻域搜索 NS_1 依次作用于 x 的两个串.先对调度串,从 π_1 开始,对于每个 π_i ,若随机数 $\alpha < \eta$,互换 π_i 和随机选中的 $\pi_j, j \neq i$;然后对机器分配串中的每个 M_{θ_i} ,若随机数 $\alpha < \eta$,则 M_{θ_i} 替换为随机选取的并行机.通过多次实验 $\eta = 0.1$.

邻域搜索 NS_2 描述如下.生成新解 $z \in \mathcal{N}_1(x)$,若 z 满足替换条件,则 z 替换 x 并更新 Ω ,搜索结束;否则,产生 $z \in \mathcal{N}_2(x)$,若替换条件满足,则利用 z 替代 x 并更新 Ω ;否则, $z \in \mathcal{N}_3(x)$,如果替换条件成立,则更新 x 和 Ω .其中 $\mathcal{N}_i$ 表示邻域结构, $\mathcal{N}_i(x)$ 表示对 x 执行邻域结构 $\mathcal{N}_i$ 而得到的 x 的邻域解集.

邻域结构 $\mathcal{N}_1$ 的具体过程如下.从 f_1, f_2 随机选择一个目标,若 f_1 被选中,确定完成时间最大的机器 M_a 和最小的机器 M_b , $\tilde{C}_a$ 为 M_a 的完成时间,从 M_a 随机选择工件 J_i ,并计算 $\tilde{C}_{i,l,b}$,得到 l^* ,它满足 $\tilde{C}_{i,l^*,b} = \min_l \{\tilde{C}_{i,l,b}\}$.其中 $\tilde{C}_{i,l,b}$ 表示工件 J_i 被插入机器 M_b 的位置 l 时 M_b 的完成时间;若 f_2 被选中,则确定总延迟时间最大的机器 M_a 和最小的机器 M_b ,选取工件 J_i ,计算 $\tilde{T}_{i,l,b}$,得到满足 $\tilde{T}_{i,l^*,b} = \min_l \{\tilde{T}_{i,l,b}\}$ 的位置 l^* .其中 $\tilde{T}_{i,l,b}$ 表示工件 J_i 被插入机器 M_b 的位置 l 时 M_b 的总延迟时间;最后,将工件 J_i 插入到 M_b 的位置 l^* .

$\mathcal{N}_1$ 中,当工件 J_i 从 M_a 上剔除后,剩余工件的加工顺序保持不变, $\tilde{C}'_a$ 为剩余工件的最大完工时间,可知, $\tilde{C}'_a < \tilde{C}_a$.当 J_i 从 M_a 转移到 M_b 后,若 $\tilde{C}_{min,b} < \tilde{C}_a$,

则新解 $z \in \mathcal{N}_1(x)$ 最大完成时间小于 x ,则 z 至少与 x 互不支配.同理, $\tilde{T}'_a$ 为 J_i 从 M_a 上移除后剩余工件的总延迟时间,若 $(\tilde{T}'_a + \tilde{T}_{min,b}) < (\tilde{T}_a + \tilde{T}_b)$,则 $f_2(z)$ 小于 $f_2(x)$.这样,利用 $\mathcal{N}_1$ 具体过程中的计算结果,容易判断新解能否满足替换条件,从而简化计算.

邻域结构 $\mathcal{N}_2$ 的具体步骤如下.随机选择一个目标,若 f_1 被选中,同 $\mathcal{N}_1$ 一样,确定具有完成时间 $\tilde{C}_a$ 最大的机器 M_a 和完成时间 $\tilde{C}_b$ 最小的 M_b ,从 M_a 中随机选择 J_i ,从 M_b 中随机选定工件 J_j ,并从各自机器上移除.计算满足 $\tilde{C}_{min,a} = \tilde{C}_{j,q^*,a}$ 和 $\tilde{C}_{min,b} = \tilde{C}_{i,l^*,b}$ 的位置 q^* 和 l^* ,其中前者在机器 M_a 上,后者在机器 M_b 上;若选中 f_2 ,分别计算 $\tilde{T}_{min,b}$ 和 $\tilde{T}_{min,a}$;确定位置 q^* 和 l^* ;最后,将 J_i 插入 M_b 的位置 l^* , J_j 插入 M_a 位置 q^* .

同样,若 $\tilde{C}_{min,b} < \tilde{C}_a, \tilde{C}_{min,a} < \tilde{C}_a$,则 $f_1(z) < f_1(x)$;或 $(\tilde{T}_{min,a} + \tilde{T}_{min,b}) < (\tilde{T}_a + \tilde{T}_b)$,则 $f_2(z) < f_2(x)$.

$\mathcal{N}_3$ 与 $\mathcal{N}_1$ 类似.选择一个目标,确定 M_a 和 J_i ,计算 $\tilde{C}_{min,a}$ 或 $\tilde{T}_{min,a}$,得到 M_a 上的位置 q^* ,将 J_i 插入 M_a 的位置 q^* . $\mathcal{N}_3$ 只是将机器 M_a 上的工件 J_i 插

入到该机器的新位置上.

邻域搜索 NS_3 的主要步骤包括: 随机选定一台并行机 M_k 和一个目标, 假设选定目标为 f_1 . 若 M_k 加工 $G \geq 3$ 个工件, 则重复如下过程 R 次: 随机选择三个工件 $J_i, J_j, J_r, i \neq j \neq r$, 存在六种不同的组合, 计算每种组合对应的 f_1 , 确定使 f_1 最小的组合并根据该组合调整工件的加工顺序. 通过大量实验确定, 当 $G < 4$ 时, $R = 1$; 当 $4 \leq G < 10$ 时, $R = 2$; 当 $G \geq 10$ 时, $R = 3$.

全局搜索采用循环交叉(cycle crossover, CX^[41]). 对于解 x 和 y , 利用CX得到新解 z , 如果 $\pi_k^x \neq \pi_k^y$ 即 π_k^z 来自 y 的调度串, 则令 $M_{\theta_{\pi_k^z}}$ 等于 $M_{\theta_{\pi_k^y}}$; 否则令 $M_{\theta_{\pi_k^z}}$ 等于 $M_{\theta_{\pi_k^x}}$.

第一种淘汰过程: 对于 $x \in EB_g$ 且 $\gamma_x \geq \mu_g$, 随机选择 $y \in \Omega$, 随机选择 NS_2 和 NS_3 中的一个作用于 y , 生成新解 z , 满足替换条件, 则用 z 替代 x , 同时令 $\gamma_x = 0$. 其中 μ_g 是整数, 经实验得到如下取值.

$$\mu_g = \begin{cases} 20 & 1 \leq g \leq \tau \\ 15 & \tau < g \leq s-1 \\ 10 & g = s \end{cases} \quad (4)$$

如上所示, NS_1 和全局搜索作用于 x , 只生成一个解 z . 执行全局搜索所需的解 y , 当 $x \in EB_1$ 时, $y \in EB_1$; 当 $x \in EB_g (g > 1)$, $y \in EB_j, j = 1, 2, \dots, g-1$, 即 $x \in EB_g (g > 1)$ 总是与质量更好的 y 交叉, 避免所选择的 y 比 x 差, 这样可以有效地改善 x .

2.4 跟随蜂阶段

跟随蜂阶段的主要步骤如下: 对于跟随蜂群 OB_g

(1) 计算每个 $x \in EB_g$ 如公式(6)所示的轮盘赌选择概率 P_x , 令 $t = 1$.

(2) 使用轮盘赌选择一个 $y \in EB_g$.

(3) 若 $g = 1$, 则对 y 执行从 NS_2 和 NS_3 随机选择的邻域搜索; 若 $1 < g \leq \tau$, 如果随机数 $\alpha < \phi$, 对 y 执行 NS_2 , 否则, 对 y 执行 NS_3 ; 若 $g > \tau$, 对 y 执行 NS_2 , 得到新解 z .

(4) 如果替换条件成立, 则运用 z 替代 y 并更新 Ω . 更新 γ_y , 并对 y 执行第一种淘汰过程.

(5) 若 $t \leq |OB_g|$, 则 $t = t + 1$, 转到步骤2.

其中 ϕ 是实数, 根据大量实验确定取值为0.8.

$$fit_{x \in EB_g} = 1 - \beta_x / |EB_g| + \max\{0, 1 - \gamma_x / u_g\} \quad (5)$$

$$P_x = fit_x / \sum_{y \in EB_g} fit_y. \quad (6)$$

其中 β_x 表示 x 是 EB_g 中的第 β_x 个解, fit_x 为

解 x 的适应度.

显然, fit_x 由解的质量和进化质量决定. β_x 越小, fit_x 越高. 若 x 频繁被更新, 则 γ_x 值为0或接近0, 公式第二部分值更大. 进化质量直接影响 fit_x . 而现有ABC研究中, 适应度通常仅由解的质量决定^[30,31].

跟随蜂阶段主要采用邻域搜索, 而雇佣蜂阶段主要采用全局搜索, $\bar{N} \geq N$, EB_s 被排斥在跟随蜂阶段之外, OB_g 中的每个跟随蜂仅从对应的 EB_g 中选择一个解, 从而加强对优秀解的搜索, 淘汰 EB_s 以充分利用计算资源. 这是跟随蜂阶段新的实现方法, 很少出现在以前的ABC中^[30,31].

2.5 算法描述

MABC的主要步骤描述如下:

(1) 采用启发式方法生成 N 个初始解构成 EB , 对于任意 $x \in EB$, 令 $\gamma_x = 0$. Ω 为空.

(2) 构建 s 个雇佣蜂群 $EB_1, EB_2, \dots, EB_s$.

(3) 执行雇佣蜂阶段.

(4) 对 EB 执行非劣排序和拥挤距离计算.

(5) 执行跟随蜂阶段.

(6) 执行第二种淘汰过程.

(7) 对 EB 执行非劣排序和拥挤距离计算.

(8) 重新调整外部档案 Ω .

(9) 满足终止条件, 则搜索结束, 否则转至步骤2.

其中终止条件是目标函数评价的最大值.

第二种淘汰过程如下: 构造一个集合 $\Lambda = \{x \in \Omega | x \notin EB\}$, 如果 $|\Lambda| > 1$, 随机选择一个元素 $y \in \Lambda$, 替换 EB 的最差解.

外部档案重新调整的过程如下. 对于每个非劣解 $x \in EB$, 如果 $x \notin \Omega$, 将其加入 Ω ; 依据Pareto支配, 比较 Ω 中所有元素, 然后剔除所有受支配解.

MABC具有以下特点. (1) 构建了多个雇佣蜂群和跟随蜂群, OB_g 跟随 EB_g , EB_s 没有跟随蜂跟随, 雇佣蜂群 $EB_g, g > 1$ 的解向优于该群的其它雇佣蜂群的解学习. (2) 采用多样化搜索方式实现雇佣蜂阶段和跟随蜂阶段. (3) 采用了两种淘汰过程, 其中第一种淘汰过程相当于侦查蜂阶段, 第二种淘汰过程淘汰 EB 的最差解.

3 计算实验

为了验证MABC在求解考虑PM和SDST的UPMSP方面的有效性和优势, 进行了大量的计算实验, 所有测试实验均用Microsoft visual C++ 2019 编程实现, 在8.0G RAM 2.30GHz CPU环境下运行.

3.1 测试实例, 对比算法以及评价指标

采用了48个实例测试. 小规模实例 $n = 20$,

$m = 2$. 中规模实例 $n = 30$, $m \in \{2, 3, 4\}$. 大规模实例 $n \in \{50, 100, 150, 200\}$, $m \in \{10, 15, 20\}$. $p_{kj} \in [1, 99]$. 有三种准备时间, 分别是 $[1, 9]$, $[1, 99]$ 和 $[1, 124]$. 小规模和大规模的加工时间和准备时间来自 <http://www.cima.uadec.mx/instancias/>, 中规模两种时间随机生成^[11]. $w_k \in [1, 99]$, PM 间隔 u_k 和交货期 d_j 按照论文^[12]的公式设定, u_k 生成公式如下

$$u_k = \max_{j=1,2,\dots,n} \{s_{k0j} + p_{kj} + s_{kj0}\} \quad (7)$$

工件 J_j 的交货期 d_j 生成公式

$$d_j = (0.5 + 2\delta) \times \sum_{k=1}^m \frac{p_{kj}}{m} \quad (8)$$

其中 δ 为介于 0 到 1 的随机实数.

采用如下两个性能测试指标. DI_R ^[42] 通过计算非劣解集 Ω_l 相对于参考集合 Ω^* 的距离衡量算法的收敛性能.

为了计算 DI_R , 所有算法非劣解集合 $\bigcup_l \Omega_l$ 中的非劣解构成 Ω^* , $DI_R(\Omega_l)$ 越小, 该算法收敛性越好. $DI_R(\Omega_l) = 0$ 表明算法产生的解都在参考集 Ω^* 内. ρ_l 指标^[43] 表示集合 $\{x \in \Omega_l | x \in \Omega^*\}$ 与 $|\Omega^*|$ 的大小比例. DI_R 和 ρ_l 经常被应用于度量调度算法的性能^[12,28].

选用多目标多点模拟退火算法^[44](MOMSA), 多目标和声搜索^[45](MOHS) 和新型帝国竞争算法(NICA^[12])作为对比算法.

MOMSA 用于求解多目标 UPMSP. 在解码过程中, 加入准备时间和维修后, MOMSA 可以直接应用于考虑 PM 和 SDST 的多目标 UPMSP 求解. MOHS 中, 应用规则将实数调度串转换为离散工件排列. 机器分配通过对机器分配串元素取整而确定, 在解码过程中采用本文方法处理准备时间和维修后, MOHS 能直接求解本文问题. NICA 本身用来求解具

有 SDST 和 PM 的 UPMSP, 可直接用作对比算法.

MABC 具有以下参数: $N, \bar{N}, s, max_it$. 对于 s , 如前所述, $s \geq 2$, 如果 s 为 2, 则跟随蜂阶段的多样性搜索无法进行, 为此, $s \geq 3$, 为了处理方便, 限定 $s \leq 4$, 这样 s 应等于 3 或 4. 由于 s 限定为 4, 为此采用如下方法确定各蜂群大小: 令 $|EB_s| = N/6$, 若 $s = 4$, $|EB_1| = |EB_2|$, $|EB_3| = |EB_4|$, $|OB_1| : |OB_2| : |OB_3| = 7 : 4 : 1$, 若 $s = 3$, $|EB_1| = |EB_2|$, $|OB_1| : |OB_2| = 7 : 4$.

对算法其它主要参数进行了田口实验, 实验用例选用 $n = 30, m = 4$ 且 SDST 为 $[1, 124]$. 结果表明, 当 $N = 30, \bar{N} = 60, s = 4, max_it = 10^5$ 时, MABC 性能最好, 故选择这组参数.

关于对比算法, 实验发现, 当 $max_it = 10^5$ 时, 几个对比算法都无法产生更好的计算结果, 同时考虑比较的公平性, 对比算法都采用和 MABC 一样的终止条件. 以上算法的参数实验的相关表格由于篇幅限制未列出.

3.2 MABC 策略有效性

将 MABC 与 MABC1 和 MABC2 进行比较, 以验证 MABC 的策略有效性. MABC1 的参数设置与 MABC 相同, 它与 MABC 不同之处在于初始种群随机产生. MABC2 中, $\bar{N} = N, s = 2, EB_2$ 为空, 即没有多群体和全局搜索与邻域搜索的多样化搜索, MABC2 其他步骤和参数与 MABC 相同.

每个算法独立运行 10 次. 表 1 和表 2 描述了 MABC 及其两个变体的计算结果. 结果表明, MABC 关于 41 个实例获得的 DI_R 小于 MABC1, 只是对于 4 个实例所得的 ρ 小于 MABC1, 这表明, 启发式初始化方法能改善 MABC 的性能. MABC2 关于所有实例所得的 DI_R 大于或等于 MABC 的相应结果, 关于 44 个实例, MABC2 的 DI_R 至少比 MABC 的 DI_R 大 10, 即 MA-

表 1 MABC 及其两个变体关于指标 DI_R 的计算结果

Table 1 Computational results of MABC and its two variants on DI_R

Index	MABC	MABC1	MABC2	Index	MABC	MABC1	MABC2	Index	MABC	MABC1	MABC2
1	3.79	0.00	12.12	17	0.00	19.65	37.97	33	3.20	1.80	101.63
2	0.64	1.62	10.63	18	0.00	3.22	62.87	34	0.00	43.34	48.85
3	0.00	49.75	41.70	19	0.00	53.50	15.02	35	0.00	8.34	95.31
4	6.99	1.94	18.30	20	0.00	7.87	41.71	36	0.00	13.30	61.82
5	1.17	8.45	24.62	21	0.00	27.39	22.34	37	0.00	21.22	20.16
6	12.48	7.26	33.74	22	0.00	13.71	50.68	38	0.03	2.98	92.05
7	0.00	17.69	3.57	23	3.50	0.00	82.85	39	0.00	16.71	74.22
8	0.03	4.08	13.11	24	1.31	1.02	62.76	40	0.00	12.87	67.04
9	0.00	13.84	48.06	25	0.00	23.44	26.37	41	0.00	12.40	99.10
10	6.02	7.23	10.40	26	0.00	8.18	88.11	42	0.00	2.68	90.74
11	2.66	0.78	23.45	27	0.00	6.79	70.69	43	0.00	65.73	57.84
12	0.00	15.68	47.46	28	0.00	28.38	31.99	44	0.00	6.10	59.86
13	0.00	2.95	11.19	29	0.37	2.47	33.24	45	0.00	11.47	102.91
14	0.00	7.71	39.85	30	0.00	11.27	51.16	46	0.00	118.99	36.87
15	0.00	20.88	50.60	31	0.00	13.03	54.67	47	0.00	8.78	81.43
16	0.00	79.03	26.53	32	0.00	7.22	103.28	48	0.00	14.36	83.69

表 2 MABC及其两个变体关于指标 ρ 的计算结果Table 2 Computational results of MABC and its two variants on ρ

Index	MABC	MABC1	MABC2	Index	MABC	MABC1	MABC2	Index	MABC	MABC1	MABC2
1	0.50	1.00	0.50	17	1.00	0.00	0.00	33	0.33	0.67	0.00
2	0.78	0.56	0.00	18	1.00	0.00	0.00	34	1.00	0.00	0.00
3	1.00	0.00	0.00	19	1.00	0.00	0.00	35	1.00	0.00	0.00
4	0.75	0.25	0.00	20	1.00	0.00	0.00	36	1.00	0.00	0.00
5	0.80	0.20	0.00	21	1.00	0.00	0.00	37	1.00	0.00	0.00
6	0.50	0.50	0.00	22	1.00	0.00	0.00	38	0.91	0.09	0.00
7	1.00	0.00	0.00	23	0.00	1.00	0.00	39	1.00	0.00	0.00
8	0.88	0.13	0.00	24	0.50	0.50	0.00	40	1.00	0.00	0.00
9	1.00	0.00	0.00	25	1.00	0.00	0.00	41	1.00	0.00	0.00
10	0.87	0.33	0.00	26	1.00	0.00	0.00	42	1.00	0.00	0.00
11	0.33	0.67	0.00	27	1.00	0.00	0.00	43	1.00	0.00	0.00
12	1.00	0.00	0.00	28	1.00	0.00	0.00	44	1.00	0.00	0.00
13	1.00	0.50	0.00	29	0.75	0.25	0.00	45	1.00	0.00	0.00
14	1.00	0.00	0.00	30	1.00	0.00	0.00	46	1.00	0.00	0.00
15	1.00	0.00	0.00	31	1.00	0.00	0.00	47	1.00	0.00	0.00
16	1.00	0.00	0.00	32	1.00	0.00	0.00	48	1.00	0.00	0.00

表 3 MABC与对比算法关于指标 DI_R 的计算结果Table 3 Computational results of MABC and its comparative algorithms on DI_R

Index	MABC	MOHS	MOMSA	NICA	Index	MABC	MOHS	MOMSA	NICA	Index	MABC	MOHS	MOMSA	NICA
1	46.50	3.50	82.21	39.28	17	24.84	20.13	105.14	6.68	33	0.00	111.07	39.04	9.20
2	2.24	13.93	46.88	5.17	18	8.53	6.92	90.85	1.91	34	12.62	96.31	40.71	0.00
3	0.00	53.03	85.17	43.82	19	10.09	0.00	100.00	5.59	35	9.51	92.19	99.00	1.40
4	0.00	46.99	27.58	24.29	20	10.38	10.49	122.59	25.72	36	7.46	92.29	29.68	10.07
5	0.00	50.94	24.64	34.71	21	11.11	11.29	85.80	5.80	37	0.24	95.17	41.56	8.24
6	0.00	50.65	113.48	18.82	22	0.00	84.27	14.52	15.50	38	31.74	74.88	44.69	0.00
7	0.00	13.11	22.13	4.81	23	0.00	104.86	44.76	23.27	39	0.00	71.90	47.40	31.71
8	2.36	58.71	49.48	10.86	24	0.00	117.37	57.95	27.53	40	0.27	74.78	10.49	2.27
9	0.11	58.83	91.25	8.38	25	0.00	60.13	65.19	19.81	41	0.00	81.33	18.04	3.06
10	0.00	13.63	66.96	15.46	26	3.34	79.73	33.75	5.13	42	0.00	92.07	33.58	6.12
11	0.00	40.89	90.72	17.99	27	0.00	99.42	84.62	32.78	43	0.61	103.02	62.83	1.44
12	0.00	59.70	78.47	8.24	28	10.86	33.54	42.07	0.00	44	10.13	122.02	41.27	0.93
13	0.10	28.31	64.56	2.74	29	78.69	31.78	104.95	31.78	45	0.00	112.51	84.68	10.12
14	20.99	65.15	119.35	0.00	30	14.38	70.84	90.96	1.56	46	0.00	69.17	27.97	6.96
15	0.00	38.43	110.73	25.95	31	4.01	103.05	59.24	0.00	47	57.29	86.69	60.41	0.00
16	0.00	1.29	100.00	5.18	32	0.00	102.78	41.55	23.52	48	63.56	125.34	21.50	8.00

表 4 MABC与对比算法关于指标 ρ 的计算结果Table 4 Computational results of MABC and its comparative algorithms on ρ

Index	MABC	MOHS	MOMSA	NICA	Index	MABC	MOHS	MOMSA	NICA	Index	MABC	MOHS	MOMSA	NICA
1	0.17	0.67	0.00	0.17	17	0.20	0.40	0.00	0.40	33	1.00	0.00	0.00	0.00
2	0.63	0.00	0.00	0.38	18	0.17	0.33	0.00	0.50	34	0.00	0.00	0.00	1.00
3	1.00	0.00	0.00	0.00	19	0.00	1.00	0.00	0.00	35	0.20	0.00	0.00	0.80
4	1.00	0.00	0.00	0.00	20	0.67	0.33	0.00	0.00	36	0.50	0.00	0.17	0.33
5	1.00	0.00	0.00	0.00	21	0.33	0.17	0.00	0.50	37	0.83	0.00	0.00	0.17
6	1.00	0.00	0.00	0.00	22	1.00	0.00	0.00	0.00	38	0.00	0.00	0.00	1.00
7	1.00	0.00	0.00	0.00	23	1.00	0.00	0.00	0.00	39	1.00	0.00	0.00	0.00
8	0.67	0.00	0.00	0.33	24	1.00	0.00	0.00	0.00	40	0.83	0.00	0.00	0.17
9	0.83	0.00	0.00	0.17	25	1.00	0.00	0.00	0.00	41	1.00	0.00	0.00	0.00
10	1.00	0.00	0.00	0.00	26	0.60	0.00	0.00	0.40	42	1.00	0.00	0.00	0.00
11	1.00	0.00	0.00	0.00	27	1.00	0.00	0.00	0.00	43	0.75	0.00	0.00	0.25
12	1.00	0.00	0.00	0.00	28	0.00	0.00	0.00	1.00	44	0.25	0.00	0.00	0.75
13	0.67	0.00	0.00	0.33	29	0.00	0.50	0.00	0.50	45	1.00	0.00	0.00	0.00
14	0.00	0.00	0.00	1.00	30	0.25	0.00	0.00	0.75	46	1.00	0.00	0.00	0.00
15	1.00	0.00	0.00	0.00	31	0.00	0.00	0.00	1.00	47	0.00	0.00	0.00	1.00
16	1.00	0.00	0.00	0.00	32	1.00	0.00	0.00	0.00	48	0.00	0.00	0.33	0.67

表 5 四种算法的运行时间

Table 5 Computational times of four algorithms

Index	MABC	MOHS	MOMSA	NICA	Index	MABC	MOHS	MOMSA	NICA	Index	MABC	MOHS	MOMSA	NICA
1	1.84	2.35	1.30	1.58	17	2.31	3.82	2.84	2.65	33	7.15	5.34	4.85	4.88
2	2.02	2.18	1.39	1.39	18	2.60	2.96	3.29	2.25	34	5.81	5.50	5.42	4.62
3	2.07	2.09	0.82	1.67	19	1.75	4.22	3.47	2.90	35	6.35	5.35	5.72	4.57
4	2.46	2.40	1.07	1.53	20	2.03	3.92	2.83	2.71	36	6.48	5.71	5.29	5.23
5	2.52	1.79	0.90	1.28	21	2.18	3.64	2.76	2.91	37	5.94	6.20	6.94	6.11
6	2.57	1.83	1.20	1.46	22	3.98	4.62	2.97	3.46	38	6.70	5.85	5.83	5.84
7	2.21	2.40	1.11	1.59	23	4.41	4.35	2.86	3.24	39	6.23	5.86	5.44	5.78
8	2.33	2.14	0.88	1.34	24	4.40	4.89	3.08	3.50	40	9.74	7.98	7.77	6.70
9	2.40	2.49	1.08	1.60	25	3.72	4.69	3.52	4.34	41	11.35	7.34	7.06	6.07
10	2.05	2.54	1.12	1.69	26	4.06	4.44	3.00	4.23	42	10.87	7.05	6.79	5.98
11	2.21	2.71	1.67	1.56	27	4.16	4.81	3.28	4.29	43	8.75	7.88	7.27	7.26
12	2.16	2.35	1.09	1.40	28	3.41	5.51	6.40	5.62	44	9.67	7.67	7.64	7.49
13	2.45	4.09	3.10	2.55	29	4.32	5.09	3.37	4.96	45	9.71	8.09	6.61	7.67
14	2.58	4.04	2.93	2.31	30	4.04	5.01	3.36	5.03	46	8.74	8.25	8.19	8.21
15	2.54	3.86	2.03	2.62	31	6.55	5.71	3.74	5.17	47	9.67	8.54	7.29	8.07
16	1.84	3.22	3.43	2.83	32	7.53	5.49	5.13	5.03	48	9.70	8.28	6.95	8.05

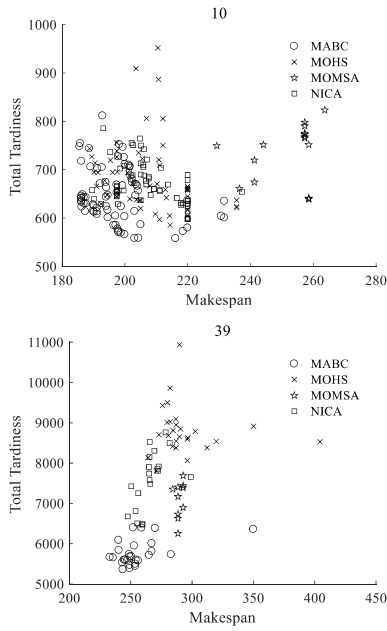


图 1 四种算法在两个实例上的非劣解图

Fig. 1 Non-dominated solutions of tow algorithms on four instances

BC2的非支配解远离MABC的非支配解.MABC关于 ρ 也显著优于MABC2, 这表明多蜂群和差异化搜索是必要且有效的.

3.3 结果与分析

三个对比算法的终止条件和MABC一样以保证公平比较. 每个算法关于每个实例独立运行10次. 表3-5列出了MABC和三种对比算法的结果和计算时间. 图1显示了四种算法的非支配解分布. 图2描述了四种算法的箱线图.

如表3和表4所示, 关于47个实例, MABC的 DI_R 和 ρ 优于MOMSA的相应结果. 对于MOHS, 仅关于5个实例取得了优于MABC的 DI_R , 关于5个实例, MOHS取得优于MABC的 ρ 值. MABC关于32个实例所得的 DI_R 小于NICA, 关于34个实例的 ρ 大于NICA;

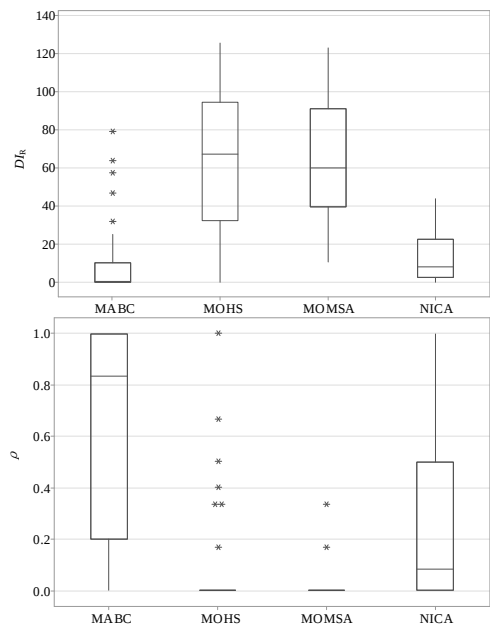


图 2 两种指标的箱线图

Fig. 2 Box plot of two metrics

此外, MABC关于22个实例的 DI_R 等于0且关于22个实例提供了集合 Ω^* 的所有元素. 如图1和图2所示, MABC的搜索性能优于对比算法. 因此, 在相同的计算时间内, MABC在大多数情况下都能得到比三种比较算法更好的结果, 具有较强的搜索优势.

MABC中, 启发式算法解决了初始解质量低的问题, 多雇佣蜂群和多跟随蜂群的引入以及全局搜索和三邻域搜索的差异化组合有效地避免搜索陷入局部最优, 两个淘汰过程提高了某些差解的质量. 这些特点使得差解不断被淘汰并充分利用了优秀解, 有效实现全局搜索和局部搜索的良好平衡. 因此, MABC有效地解决了具有PM和SDST的UPMSP.

4 结论

PM和SDST通常同时存在并行机制造过程, 有必要研究包含PM和SDST的双目标UPMSP, 只是关于这类问题的研究进展较少, 为此, 提出了一种MABC

算法, 该算法将所有雇佣蜂分为 s 个雇佣蜂群, 除最差雇佣蜂群外, 每个雇佣蜂群都对应一个跟随蜂群. 采用全局搜索和三种邻域搜索的不同组合实现雇佣蜂阶段和跟随蜂阶段, 并引入了两种淘汰方法. 进行大量实验测试了MABC的新策略和性能. 计算结果表明MABC在求解具有PM和SDST的UPSMP方面具有较强的搜索优势.

具有SDST等实际加工约束的UPMSP是一个重要的调度研究主题, 广泛存在于实际的制造系统中. 将继续关注这类问题, 并应用蛙跳算法和教学优化算法等解决该类问题. 多工厂分布式绿色调度也是我们未来的研究课题.

参考文献(References)

- [1] 吴楚格, 王凌, 郑晓龙. 求解不相关并行机调度的一种自适应分布估计算法[J]. 控制与决策, 2016, 31(12): 2177-2182.
(WU Chuge, WANG Ling, ZHENG Xiaolong. An adaptive estimation of distribution algorithm for solving the unrelated parallel machine scheduling [J]. *Control and Decision*, 2016, 31(12): 2177-2182.)
- [2] JANS R, DEGRAEVE Z. An industrial extension of the discrete lot-sizing and scheduling problem [J]. *IIE Trans*, 2004, 36(1): 47-58.
- [3] SILVA C, MAGALHAES J A. Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry [J]. *Comput Ind Eng*, 2006, 50(1-2): 76-89.
- [4] PATTLOCH M, SCHMIDT G, KOVALYOV M Y. Heuristic algorithms for lotsize scheduling with application in the tobacco industry [J]. *Comput Ind Eng*, 2001, 39(3-4): 235-253.
- [5] WANG H B, ALIDAEE B. Effective heuristic for large-scale unrelated parallel machines scheduling problems [J]. *Omega-Int J Manage Sci*, 2019, 83: 261-274.
- [6] LI G, LIU M Q, SETHI S P, et al. Parallel-machine scheduling with machine-dependent maintenance periodic recycles [J]. *Int J Prod Econ*, 2017, 186: 1-7.
- [7] YANG D L, CHENG T C E, YANG S J, et al. Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities [J]. *Comput Oper Res*, 2012, 39(7): 1458-1464.
- [8] TAVANA M, ZAROOK Y, SANTOS-ARTEAGA F J. An integrated three-stage maintenance scheduling model for unrelated parallel machines with aging effect and multi-maintenance activities [J]. *Comput Ind Eng*, 2015, 83: 226-236.
- [9] WANG S J, LIU M. Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning [J]. *J Manuf Syst*, 2015, 37: 182-192.
- [10] GARA-ALI A, FINKE G, ESPINOUSE M L. Parallel-machine scheduling with maintenance: Praising the assignment problem [J]. *Eur J Oper Res*, 2016, 252(1): 90-97.
- [11] AVALOS-ROSALES O, ANGEL-BELLO F, ALVAREZ A, et al. Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times [J]. *Comput Ind Eng*, 2018, 123: 364-377.
- [12] WANG M, PAN G. A Novel Imperialist Competitive Algorithm With Multi-Elite Individuals Guidance for Multi-Object Unrelated Parallel Machine Scheduling Problem [J]. *IEEE Access*, 2019, 7: 121223-121235.
- [13] PARKER R G, DEANE R H, HOLMES R A. On the Use of A Vehicle Routing Algorithm for the Parallel Processor Problem with Sequence Dependent Changeover Costs [J]. *AIIE Trans*, 1977, 9(2): 155-160.
- [14] KURZ M E, ASKIN R G. Heuristic scheduling of parallel machines with sequence-dependent set-up times [J]. *Int J Prod Res*, 2001, 39(16): 3747-3769.
- [15] ARNAOUT J-P, RABADI G, MUSA R. A two-stage Ant Colony Optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times [J]. *J Intell Manuf*, 2010, 21(6): 693-701.
- [16] VALLADA E, RUIZ R. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times [J]. *Eur J Oper Res*, 2011, 211(3): 612-622.
- [17] LIN S W, YING K C. ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times [J]. *Comput Oper Res*, 2014, 51: 172-181.
- [18] CANIYILMAZ E, BENLI B, ILKAY M S. An artificial bee colony algorithm approach for unrelated parallel machine scheduling with processing set restrictions, job sequence-dependent setup times, and due date [J]. *Int J Adv Manuf Technol*, 2015, 77(9-12): 2105-2115.
- [19] DIANA R O M, DE FRANCA M F, DE SOUZA S R, et al. An immune-inspired algorithm for an unrelated parallel machines' scheduling problem with sequence and machine dependent setup-times for makespan minimisation [J]. *Neurocomput*, 2015, 163: 94-105.
- [20] WANG L, WANG S Y, ZHENG X L. A Hybrid Estimation of Distribution Algorithm for Unrelated Parallel Machine Scheduling with Sequence-Dependent Setup Times [J]. *IEEE-CAA J Auto Sini*, 2016, 3(3): 235-246.
- [21] EZUGWU A E, AKUTSAH F. An Improved Firefly Algorithm for the Unrelated Parallel Machines Scheduling Problem With Sequence-Dependent Setup Times [J]. *IEEE Access*, 2018, 6: 54459-54478.
- [22] FANJUL-PEYRO L, RUIZ R, PEREA F. Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times [J]. *Comput Oper Res*, 2019, 101: 173-182.
- [23] BEKTUR G, SARAC T. A mathematical model and

- heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server [J]. *Comput Oper Res*, 2019, 103:46-63
- [24] MOJTABA A, JAVAD R. A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches [J]. *Appl Soft Comput*, 2017, 50: 109-123.
- [25] 宋强. 求解异构并行机调度问题的混合多目标教学-学优化算法[J]. 控制理论与应用, 2020, 37(10): 2242-2256.
(SONG Qiang. A hybrid multi-objective teaching-learning-based optimization algorithm for unrelated parallel machine scheduling problem[J]. *Control Theory and Applications*, 2020, 37(10): 2242-2256)
- [26] 王凌,周刚,许烨,王圣尧.求解不相关并行机混合流水线调度问题的人工蜂群算法[J]. 控制理论与应用, 2012, 29(12): 1551-1557.
(WANG Ling, ZHOU Gang, XU Ye, WANG ShengYao. An artificial bee colony algorithm for solving hybrid flow-shop scheduling problem with unrelated parallel machines [J]. *Control Theory and Applications*, 2012, 29(12): 1551-1557.)
- [27] HAN Y Y, GONG D W, SUN X Y. A discrete artificial bee colony algorithm incorporating differential evolution for the flow-shop scheduling problem with blocking [J]. *Eng Optimiz*, 2015, 47(7): 927-946.
- [28] GONG D, HAN Y, SUN J. A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems [J]. *Knowl Based Syst*, 2018, 148:115-130.
- [29] 吴锐,郭顺生,李益兵,等.改进人工蜂群算法求解分布式柔性作业车间调度问题[J]. 控制与决策, 2019, 34(12): 2527-2536.
(WU Rui, GUO Shunsheng, LI Yibing, et al. Improved artificial bee colony algorithm for distributed and flexible jobshop scheduling problem. *Control and Decision*, 2019, 34(12): 2527-2536.)
- [30] LEI D M, GUO X P. Scheduling job shop with lot streaming and transportation through a modified artificial bee colony [J]. *Int J Prod Res*, 2013, 51(16): 4930-4941.
- [31] LI X Y, XIAO S Q, WANG C Y, et al. Mathematical modeling and a discrete artificial bee colony algorithm for the welding shop scheduling problem [J]. *Memet Comput*, 2019, 11(4): 371-389.
- [32] 匡芳君,徐蔚鸿,金忠.自适应Tent混沌搜索的人工蜂群算法[J].控制理论与应用, 2014, 31(11): 1502-1509.
(KUANG Fangjun, XU Weihong, JIN Zhong. A rtificial bee colony algorithm based on self-adaptive Tent chaos search [J]. *Control Theory and Applications*, 2014, 31(11): 1502-1509.)
- [33] 暴励,曾建潮.一种双种群差分蜂群算法[J].控制理论与应用, 2011, 28(02): 266-272.
(BAO Li, ZENG Jianchao. A bi-group differential artificial bee colony algorithm [J]. *Control Theory and Applications*, 2011, 28(02): 266-272.)
- [34] ZHOU J J, YAO X F, CHAN F T S, et al. An individual dependent multi-colony artificial bee colony algorithm [J]. *Infor Sci*, 2019, 485:114-140.
- [35] XIANG Y, ZHOU Y R. A dynamic multi-colony artificial bee colony algorithm for multi-objective optimization [J]. *Appl Soft Comput*, 2015, 35:766-785.
- [36] GAO J Q, HE G X, WANG Y S. A new parallel genetic algorithm for solving multiobjective scheduling problems subjected to special process constraint [J]. *Int J Adv Manuf Technol*, 2009, 43(1-2): 151-160.
- [37] AFZALIRAD M, REZAEIAN J. A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches [J]. *Appl Soft Comput*, 2017, 50:109-123.
- [38] WU G-H, CHENG C-Y, YANG H-I, et al. An improved water flow-like algorithm for order acceptance and scheduling with identical parallel machines [J]. *Appl Soft Comput*, 2018, 71:1072-1084.
- [39] TIGANE M, DAHANE M, BOUDHAR M. Multiobjective approach for deteriorating jobs scheduling for a sustainable manufacturing system [J]. *Int J Adv Manuf Technol*, 2019, 101(5-8): 1939-1957.
- [40] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Trans Evol Comput*, 2002, 6(2): 182-197.
- [41] Oliver I M, Smith D J, Holland J R C. A Study of Permutation Crossover Operators on the Traveling Salesman Problem [C]. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, 1987, 224-230.
- [42] Knowles J, Corne D. On metrics for comparing nondominated sets [C]. In: *Proceedings of 2002 Congress on Evolutionary Computation. Honolulu, USA: IEEE*, 2002, 711-716.
- [43] LEI D. Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems [J]. *Int J Adv Manuf Technol*, 2008, 37(1-2): 157-165.
- [44] LIN S W, YING K C. A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems [J]. *Int J Prod Res*, 2015, 53(4): 1065-1076.
- [45] SHAHIDI-ZADEH B, TAVAKKOLI-MOGHADDAM R, TAHERI-MOGHADAM A, et al. Solving a bi-objective unrelated parallel batch processing machines scheduling problem: A comparison study [J]. *Comput Oper Res*, 2017, 88:71-90.

作者简介

雷德明(1968—),男,教授,学历(博士生导师),从事智能系统优化与控制的研究, E-mail: deminglei11@163.com;
杨海(1996—),男,硕士研究生,从事制造系统智能优化与调度的研究, E-mail: YangHai_2049@163.com.