

12-30-2023

Unrelated Parallel Machine Scheduling with Additional Resource and Learning Effect

Youlian Zheng

Faculty of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China,
zhengyl@hubu.edu.cn

Deming Lei

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

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

Unrelated Parallel Machine Scheduling with Additional Resource and Learning Effect

Abstract

Abstract: To solve unrelated parallel machine scheduling problem(UPMSP) with additional resource and learning effect, a dynamical artificial bee colony(DABC) algorithm is proposed to minimize the makespan. A new representation and decoding process is given and two initial bee swarms are constructed. A swarm evaluation method is applied to dynamically decide employed bee swarms and onlooker bee swarms. Employed bee phase and onlooker bee phase are implemented in different ways to increase exploration ability. The experimental results show that the new strategies of DABC are effective and reasonable, and can obtain results with better convergence, average value and stability, which d has high search performance in solving the considered UPMSP.

Keywords

unrelated parallel machine scheduling, additional resource, learning effect, artificial bee colony

Recommended Citation

Zheng Youlian, Lei Deming. Unrelated Parallel Machine Scheduling with Additional Resource and Learning Effect[J]. Journal of System Simulation, 2023, 35(12): 2560-2569.

考虑附加资源和学习效应的不相关并行机调度

郑友莲¹, 雷德明²

(1. 湖北大学 计算机与信息工程学院, 湖北 武汉 430062; 2. 武汉理工大学 自动化学院, 湖北 武汉 430070)

摘要: 针对考虑附加资源和学习效应的不相关并行机调度问题(unrelated parallel machine scheduling problem, UPMSP), 提出一种动态人工蜂群算法(dynamical artificial bee colony, DABC), 实现最小化最大完成时间。给出一种新的编码方法和解码过程, 并构建2个初始蜂群; 提出一种蜂群评估策略, 以动态确定雇佣蜂群和跟随蜂群; 在雇佣蜂阶段和跟随蜂阶段采用不同搜索策略, 以增强算法的探索能力。实验结果表明: DABC的新策略合理有效, 且该算法在求解UPMSP时收敛性、平均值和稳定性更强, 显示出较强的搜索性能。

关键词: 不相关并行机调度; 附加资源; 学习效应; 人工蜂群算法

中图分类号: TP273 文献标志码: A 文章编号: 1004-731X(2023)12-2560-10

DOI: 10.16182/j.issn1004731x.joss.22-0791

引用格式: 郑友莲, 雷德明. 考虑附加资源和学习效应的不相关并行机调度[J]. 系统仿真学报, 2023, 35(12): 2560-2569.

Reference format: Zheng Youlian, Lei Deming. Unrelated Parallel Machine Scheduling with Additional Resource and Learning Effect[J]. Journal of System Simulation, 2023, 35(12): 2560-2569.

Unrelated Parallel Machine Scheduling with Additional Resource and Learning Effect

Zheng Youlian¹, Lei Deming²

(1. Faculty of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China;

2. School of Automation, Wuhan University of Technology, Wuhan 430070, China)

Abstract: To solve unrelated parallel machine scheduling problem(UPMSP) with additional resource and learning effect, a dynamical artificial bee colony(DABC) algorithm is proposed to minimize the makespan. A new representation and decoding process is given and two initial bee swarms are constructed. A swarm evaluation method is applied to dynamically decide employed bee swarms and onlooker bee swarms. Employed bee phase and onlooker bee phase are implemented in different ways to increase exploration ability. The experimental results show that the new strategies of DABC are effective and reasonable, and can obtain results with better convergence, average value and stability, which d has high search performance in solving the considered UPMSP.

Keywords: unrelated parallel machine scheduling; additional resource; learning effect; artificial bee colony

0 引言

调度算法广泛应用于制造业以提高效率。并行机调度问题(parallel machine scheduling problem,

PMSP)是一种为所有工件分配合适并行机, 并确定每台机器上工件的加工顺序, 以达到相关性能指标最优的典型制造调度问题^[1], 广泛应用于纺织、半导体、轮胎、陶瓷和烟草等制造业。不相

收稿日期: 2022-07-07 修回日期: 2023-01-14

基金项目: 国家自然科学基金(61573264)

第一作者: 郑友莲(1972-), 女, 副教授, 博士, 研究方向为智能优化与调度。Email: zhengyl@hubei.edu.cn

关并行机调度问题(unrelated PMSP, UPMSP)是 PMSP 的一种, 其工件在各机器上的加工时间各异。考虑各种实际工况与约束(如附加资源和学习效应)的 UPMSP 受到研究者广泛关注, 出现了大量相关研究成果。

通常, 除了机器资源外, 在许多并行机制造过程中, 还需考虑自动导引小车、工人、工具、托盘、模具和工业机器人等附加资源, 考虑附加资源的 UPMSP 研究取得了较大进展。Zheng 等^[2]结合启发式方法和知识引导搜索提出了一种双阶段自适应果蝇算法。Fanjul-Peyro 等^[3]建立了 2 种整数规划模型和 3 种数学启发式方法。Yepes-Borrero 等^[4]针对考虑准备时间和准备过程中的附加资源的 UPMSP, 给出了 3 种启发式方法和贪婪随机自适应搜索过程。Lei 等^[5]提出一种自适应人工蜂群算法(artificial bee colony, ABC), 解决了考虑附加资源和维修的 UPMSP。

学习效应是指由于反复加工同种或类似的工件, 同一台机器上工件的加工时间随着位置的延后而逐渐减小。考虑学习效应的 UPMSP 受到研究者较多关注。Soleimani 等^[6]针对考虑恶化、学习效应和顺序相关准备时间的问题, 提出了遗传算法、猫群算法和交互 ABC。Zhang 等^[7]应用一种组合进化算法解决了考虑准备时间、有限工人和学习效应的 UPMSP。轩华等^[8]运用改进的模拟退火算法求解了考虑恶化和学习效应的 UPMSP。

综上所述, 考虑附加资源的 UPMSP 和考虑学习效应的 UPMSP 相关研究都取得了进展, 不过, 现有研究很少同时考虑附加资源和学习效应。由于实际并行机制造过程中, 附加资源和学习效应往往同时存在, 考虑附加资源和学习效应的 UPMSP 与制造过程实际情况更接近, 相应的优化结果应用价值更高, 因此, 有必要研究同时考虑这 2 种实际约束的 UPMSP。

由于 UPMSP 的 NP 难特性, 智能算法已成为解决这类问题的主要方法。ABC 是一类具有参数

少且容易实现的智能算法, 它已广泛应用于单工厂和多工厂环境下的各种制造调度问题。研究者提出了包括改进 ABC^[9-16]、多群体 ABC^[17]和混合 ABC^[18]在内的各种算法求解 UPMSP。ABC 自身特点以及在 UPMSP 求解, 尤其是考虑附加资源或学习效应的 UPMSP 求解方面具有潜在优势。

为此, 本文针对考虑附加资源和学习效应的 UPMSP, 提出了一种动态人工蜂群算法(dynamical ABC, DABC)以最小化最大完成时间。该算法运用一种新的编码策略产生 2 个蜂群, 并运用一种蜂群评估策略动态确定雇佣蜂群和跟随蜂群, 且雇佣蜂阶段和跟随蜂阶段搜索策略不同, 可增强算法的探索能力。

1 问题描述

考虑附加资源和学习效应的 UPMSP 描述: 存在 n 个工件 J_1, J_2, \dots, J_n 和 m 台不相关并行机 M_1, M_2, \dots, M_m , 每个工件可在任意一台并行机上加工。学习效应指工件 J_j 在机器 M_k 上的加工时间 p_{kjr} 取决于 M_k 的性能, 以及在给定调度下该工件在 M_k 上的加工位置, 即 $p_{kjg} = p_{kj}(W + (1 - W)g^\delta)$, t_{kj} 是工件初始加工时间, $\delta \leq 0$ 是学习效应, W 是不可压缩因子, g 是在给定调度下 J_j 在 M_k 上的位置。考虑单附加可再生资源, 工件 J_j 在 M_k 上加工时需要 r_{kj} 单位的附加资源。在任意时刻, 任意机器可使用的最大附加资源量为 R_{\max} 。

该问题具有如下工件和机器相关的约束:

- (1) 所有机器在零时刻均可加工;
- (2) 每个工件同一时刻只能由一台机器加工;
- (3) 一台机器同一时刻只能加工一个工件;
- (4) 不允许中断加工过程;
- (5) 不允许加工抢占。

该问题由调度子问题和机器分配子问题组成, 问题的目标是为每个工件分配合适的机器, 并确定每台机器上工件的加工顺序, 以最小化最大完成时间。

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

式中： C_j 为工件 J_j 的完成时间。

2 求解考虑附加资源和学习效应的UPMSP的DABC

ABC中，存在3种类型蜜蜂：雇佣蜂、跟随蜂和侦察蜂。种群 P 初始化之后，整个搜索过程包括3个阶段：雇佣蜂阶段，种群中每个解 x 进行全局搜索，产生新解，并进行贪婪选择以保留适应度高的解；跟随蜂阶段，每个跟随蜂通过轮盘赌选择食物源，采用与雇佣蜂阶段同样的方式产生新解，并利用贪婪原则；侦察蜂阶段，若某个蜜源连续未更新的次数超过阈值 L ，则雇佣蜂变为侦察蜂，随机产生1个新蜜源替换原蜜源。

ABC中，雇佣蜂群和跟随蜂群在初始化阶段一旦形成，就在整个搜索过程中保持不变。DABC中，形成2个蜂群，然后运用一种策略对蜂群进行评估，根据评估结果确定雇佣蜂群和跟随蜂群，且雇佣蜂阶段和跟随蜂阶段搜索策略不同。

2.1 编码方法

本文提出一种双串编码方法。针对具有 n 个工件和 m 台机器的问题，问题的解由机器分配串 $[M_{h_1}, M_{h_2}, \dots, M_{h_n}]$ 和调度串 $[\pi_1, \pi_2, \dots, \pi_n]$ 表示，其中， M_{h_i} 表示工件 J_i 分配的机器， $\pi_i \in \{1, 2, \dots, n\}$ 。

解码过程：从调度串的 π_1 开始，依次对每个工件 J_{π_i} ，由机器分配串确定工件 J_{π_i} 的加工机器 M_k ，在 M_k 上依次检查每一个空闲时间段，当加工时间和附加资源约束同时满足时，将 J_{π_i} 插入开始加工时间最小的空闲时间段内，否则，将 J_{π_i} 作为 M_k 上当前最后加工的工件。

J_{π_i} 的加工开始时间 $B_{\pi_i,k} = \max\{\eta_k, \theta_{\pi_i,k}\}$ ，其中， η_k 为 M_k 上最早可加工 J_{π_i} 的时间； $\theta_{\pi_i,k}$ 为满足资源约束条件下 M_k 最早可加工工件 J_{π_i} 的时间。当工件 J_{π_i} 插入机器空闲时间段时，工件 J_{π_i} 后面工件的位置

会后移，导致这些工件的加工时间变短，为此，让这些工件的开始加工时间保持不变，完成时间缩短。

问题的解由上述2个串表示。循环执行如下步骤后产生 P 个初始解：随机产生2个串，构成1个初始解。这 P 个初始解构成初始种群 P_1 。初始种群 P_2 也采用同样方式由随机产生 P 个初始解组成，种群 $P = P_1 \cup P_2$ 。

2.2 全局搜索和邻域搜索

全局搜索过程：对解 x_i ，随机选择解 $y \in P$ ，执行 x_i, y 间关于机器分配串的两点交叉，得到新解 x_i^{new} ，如果 x_i^{new} 优于 x_i ，则用 x_i 更新记忆集 Ω ， x_i^{new} 替换 x_i ；否则，执行 x_i, y 间关于调度串的部分映射交叉(partial mapped crossover, PMX^[19])，得到新解 x_i^{new} ，如果新解 x_i^{new} 满足上述条件，则更新 Ω 并由新解替代 x_i 。

记忆集 Ω 用来保留搜索过程的中间数据，若 Ω 的实际大小未超过其最大值，则直接添加 x_i 到 Ω 中；否则，如果 x_i 优于 Ω 中的最差解，则 x_i 替代最差解。

本文提出5种邻域结构。 \mathcal{N}_1 ：分别确定完成时间最大和最小的机器 M_k, M_l ，从 M_k 上随机选择工件 J_i ，将其转移到 M_l 上。 \mathcal{N}_2 ：确定完成时间最大的机器 M_k ，任选机器 $M_l, l \neq k$ ，互换从2台上随机选择的一个工件。 \mathcal{N}_3 与 \mathcal{N}_2 类似，只是2台机器都是随机选择的。 \mathcal{N}_4 用来随机确定1台机器，互换该机器上随机选择的2个工件。 \mathcal{N}_5 ：随机确定1台机器以及该机器上的2个工件 J_{π_i}, J_{π_j} ，将 π_i 插入调度串的位置 $j-1$ ，如果 $j=1$ ，则插入到位置 j 。

多邻域搜索过程：对解 x_i ，依次执行邻域结构 $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4, \mathcal{N}_5$ ，对每个 \mathcal{N}_g ，产生的新解 $x_i^{\text{new}} \in \mathcal{N}_g(x_i)$ ，若 x_i^{new} 优于 x_i ，则更新 Ω 并由新解替代 x_i ，其中， $\mathcal{N}_g(x)$ 表示由 \mathcal{N}_g 产生的邻域解集。

全局搜索和多邻域搜索过程中, 一旦 x_i 被新解 x_i^{new} 所替代, 则对应连续未更新的次数 $trail_i$ 为0; 否则, $trail_i+1$ 。

2.3 蜂群动态确定

蜂群动态确定的具体过程。

步骤1: 令 $\alpha_i=0, i=1,2$ 。

步骤2: 构建集合 $\theta_i, i=1,2$, 由 P_i 中最大完成时间最小的 \bar{p}_e 个解组成, 然后对于每个解 $x_i \in \theta_i$, 计算 $\alpha_i = \alpha_i + |\{y_j \in \theta_{3-i} | x_i \text{ 优于 } y_j\}|$ 。

步骤3: 执行以下操作 \bar{p}_t 次。从2个蜂群中各随机选择一个解 $x \in P_1$ 和 $y \in P_2$, 如果 x 优于 y , 则 $\alpha_1 = \alpha_1 + 1$; 若 y 优于 x , 则 $\alpha_2 = \alpha_2 + 1$ 。

步骤4: 若 $\alpha_1 > \alpha_2$, 则 P_1 是 EB , P_2 为 OB ; 若 $\alpha_1 < \alpha_2$, 则 P_1 是 OB , P_2 为 EB ; 若 $\alpha_1 = \alpha_2$, 则从 P_1 和 P_2 中随机选择一个为 EB , 另一个为 OB 。其中, α_1 和 α_2 为正整数, 用于评价 P_1 和 P_2 中解的质量, α 值较大的蜂群解的质量较好, \bar{p}_e 和 \bar{p}_t 为整数。

假设 $\bar{p}_e=3, P=50$ 即种群 P_i 由50个解组成, 每个解对应一个目标值即最大完成时间。从 P_i 的50个解里面选出目标值最小的3个解, 构成集合 θ_i , 即 θ_i 由种群 P_i 中最好的3个解组成。

假设 $\theta_1 = \{x_1, x_2, x_3\}$, x_1 的目标值是500, x_2 的目标值是456, x_3 的目标值是567, 这3个目标值都是有量纲的。假设 $\theta_2 = \{y_1, y_2, y_3\}$, y_1 的目标值是480, y_2 的目标值是512, y_3 的目标值是523。目标值越小, 解越优。

利用 θ_1 中的每个解计算 α_1 。对于 x_1 , 确定集合 $\{y_j \in \theta_2 | x_1 \text{ 优于 } y_j\}$, 即由 θ_2 中的解组成且 x_1 优于这些解。 x_1 的目标值是500, 比 y_2, y_3 的目标值小, 即 x_1 优于 y_2, y_3 , 上述集合由 y_2, y_3 组成, 集合大小为2, $\alpha_1 = 0 + 2 = 2$ 。对于 x_2 , 对应的集合 $\{y_j \in \theta_2 | x_2 \text{ 优于 } y_j\} = \theta_2$, 集合大小为3, $\alpha_1 = 2 + 3 = 5$ 。对于 x_3 , 对应的集合 $\{y_j \in \theta_2 | x_3 \text{ 优于 } y_j\} = \emptyset$, $\alpha_1 = 5 + 0 = 5$ 。

α_2 的计算过程和上述过程一样, 而且无论目标值的量纲是h、s、ms, 结果都一样。

步骤3用来比较种群 P_1, P_2 , 步骤4根据比较结果确定哪个种群是 EB , 哪个是 OB 。通常情况下, ABC的2个蜂群 EB 和 OB 是固定的, 即一旦指定为 EB 和 OB , 在整个搜索过程中, 它们始终为 EB 和 OB 。而DABC根据解的质量动态从蜂群 P_1, P_2 中确定一个蜂群为 EB , 另一个为 OB , 这样蜂群在搜索过程中所执行的搜索阶段是动态变化的, 即这一代 P_i 执行雇佣蜂阶段, 下一代可能执行跟随蜂阶段。

2.4 算法描述

DABC的具体步骤描述。

步骤1: 产生初始种群 P_1 和 P_2 。

步骤2: 根据蜂群评估策略确定 EB 和 OB 。

步骤3: 执行雇佣蜂阶段, 对于每个解 $x_i \in EB$, 运行全局搜索; 若新解未更替 x_i , 则执行多邻域搜索。

步骤4: 运行跟随蜂阶段, 计算 OB 中所有解的平均目标值 A 。对每个解 $x_i \in OB$, 若其目标函数值小于 A , 则对 x_i 执行多邻域搜索; 否则, 若随机数 $r < \min\{\alpha_1, \alpha_2\} / (\alpha_1 + \alpha_2)$, 对 x_i 执行多邻域搜索, 否则, 则随机选择 EB 中的解 y 直接替换 x_i , 并对 x_i 执行多邻域搜索。

步骤5: 执行侦察蜂阶段。

步骤6: 若终止条件未满足, 则转到步骤2; 否则停止搜索。

侦察蜂阶段描述: 当种群 P 中存在满足 $trial_i > L$ 的解时, 对种群内所有解根据目标值升序排序, 对于满足条件的每个解 x_i , 如果 $i \leq \gamma \times 2 \times P$, 则对记忆集 Ω 每个解计算其概率 $(C_{\max}^x)^{-1} / \sum_{y \in \Omega} (C_{\max}^y)^{-1}$, 根据该概率采用轮盘赌方式选择解 $y \in \Omega$, 然后依次对 y 执行 $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ 和 $\mathcal{N}_4, \mathcal{N}_5$, 得到5个邻域解, 令其中最好的邻域解替代 x_i ; 否则, 对排在前 $2\gamma \times P$ 位的解, 采用上述方式计算概率, 并选择

一个解 y ，采用同样方式得到最好的邻域解替代 x_i 。其中， C_{\max}^x 表示解 x 的最大完成时间。

DABC具有不同于现有ABC的新特征：采用动态方式确定蜂群 EB 和 OB ，且 EB 和 OB 中的解搜索方式各异，这样2个种群的搜索方式也将动态地改变。

3 计算实验

为了测试DABC的搜索性能，采用文献[4]的方法产生60个实例，每个实例表示为 $n \times m \times 1$ 或 $n \times m \times 2$ ，其中，1、2指采用第1种或第2种方式产生附加资源数据，所有实例都采用文献[4]中第1种方式产生加工时间数据。 $W=0$ 、 $\delta=-0.1$ 、 $R_{\max}=5m$ 。所有实验采用Microsoft Visual C++ 2019实现并运行于8 G RAM 2.3 GHz CPU的PC机上。

选用双阶段自适应果蝇算法(two-phase adaptive fruit fly optimization algorithm, TAFOA^[2])和MPH(multi-pass heuristic^[19])作为对比算法，它们本身用来解决考虑附加资源的UPMSP，能直接用来求解本文的UPMSP。在混合粒子群遗传算法(hybrid PSO genetic algorithm, HPSOGA^[20])中，直接利用DABC解码后，该算法能直接求解本文的UPMSP，因此，选择以上算法作为对比算法。

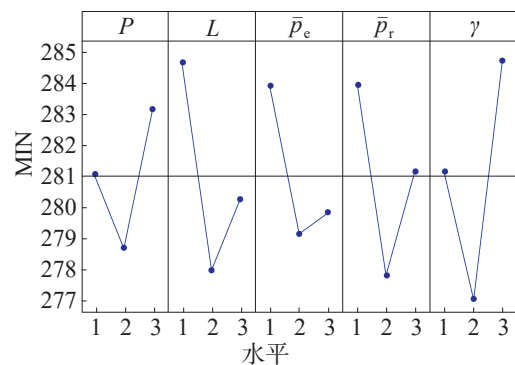
构建ABC以验证DABC策略的有效性。该算法的 EB 和 OB 固定，雇佣蜂阶段与DABC一样，跟随蜂阶段则采用DABC侦察蜂阶段的方式计算概率，然后选择蜜源，执行和雇佣蜂阶段一样的搜索策略，侦察蜂阶段则对满足条件的解直接由随机产生的解替换。

DABC具有参数：终止条件， P 、 L 、 γ 、 \bar{p}_e 、 \bar{p}_r 。终止条件，实验发现，当终止条件设为 $0.3n$ s CUP时间时，DABC能收敛，当该CPU时间达到时，对比算法和ABC也处于收敛状态，故设置终止条件为 $0.3n$ s CUP时间，采用田口方法确定其他参数，表1给出了这些参数的设置。根据正交表 $L_{27}(3^5)$ ，对实例 $50 \times 10 \times 1$ 和每组参数组合，

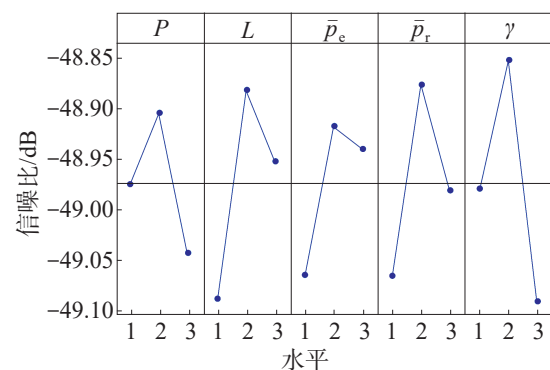
DABC随机运行10次，图1描述了相应的信噪比，信噪比定义为 $-10 \times \lg(MIN^2)$ ， MIN 表示10次运行所得的最优解。

表1 参数及其水平
Table 1 Parameters and their levels

参数	水平		
	1	2	3
P	40	50	60
L	10	15	20
\bar{p}_e	8	10	12
\bar{p}_r	30	35	40
γ	0.2	0.3	0.4



(a) MIN



(b) 信噪比

图1 MIN和信噪比
Fig. 1 MIN and SNR

如图1所示，对于参数 γ ，其水平为2时，对应的信噪比最大，故 γ 最好设置为0.3。对于其他参数，采用同样的方式确定信噪比最大的水平及其对应的设置，最终得到最佳参数组合为 $\gamma=0.3$ 、 $P=50$ 、 $L=15$ 、 $\bar{p}_e=10$ 、 $\bar{p}_r=35$ 。

除了终止条件, ABC的其他参数 P 、 L 和 DABC一样。HPSOGA参数: 种群规模为40、最大迭代为400、交叉概率为0.8、变异概率为0.03、 $c_1=c_2=2.05$ 。TAFOA的参数: $NS=30$ 、 $S=5$ 。MPH终止条件与DABC一样。

针对每个实例、每个算法随机运行10次, 表2~4描述了所有算法的计算结果。每次运行得到一个精英解, avg是这些精英解的平均值, max是最大差值, min是最好值。DABC关于54个实例获得了优于ABC的min值, 且随着实例规模变大,

DABC与ABC的min值差距加大, DABC的收敛性能明显优于ABC; 平均结果avg, DABC仅关于4个实例所得的结果和ABC一样, 其他实例的相应结果均优于ABC。DABC的搜索性能明显优于ABC, 这表明, DABC的新策略如动态评估策略等有效地改善了算法性能, 策略合理有效。

图2描述了2个实例计算结果的对比图, D、A、T、H和M分别表示DABC、ABC、TAFOA、HPSOGA、MPH。图3给出了所有算法关于2个实例的收敛曲线。

表2 5种算法关于min的计算结果
Table 2 Computational results of five algorithms on min

实例	DABC	ABC	TAFOA	HPSOGA	MPH	实例	DABC	ABC	TAFOA	HPSOGA	MPH
8×2×1	210.5	210.5	237.9	290.8	210.5	30×2×1	800.3	854.1	845.8	1 012.6	876.1
8×2×2	246.2	246.2	246.3	253.5	246.2	30×2×2	643.2	668.5	656.1	713.0	663.4
8×4×1	114.3	114.3	120.3	129.1	126.0	30×4×1	354.2	380.0	375.5	410.1	416.1
8×4×2	114.8	114.8	133.4	146.1	135.7	30×4×2	326.6	370.2	348.5	363.4	316.7
8×6×1	103.0	103.0	103.0	103.0	103.0	30×6×1	292.1	319.2	317.8	341.3	343.9
8×6×2	106.0	106.0	106.0	106.0	106.0	30×6×2	246.5	260.7	261.4	285.4	267.0
12×2×1	228.2	230.2	257.4	288.6	241.6	50×2×1	266.8	306.4	296.6	339.8	289.4
12×2×2	401.9	478.5	428.0	443.5	402.1	50×2×2	260.2	287.6	281.1	311.5	287.8
12×4×1	209.6	252.2	215.5	231.8	236.5	50×4×1	191.0	222.6	213.5	249.8	222.6
12×4×2	186.0	209.0	195.2	204.6	185.5	50×4×2	149.8	177.7	163.5	202.3	170.6
12×6×1	140.0	143.0	149.3	217.6	152.3	50×6×1	122.0	144.5	133.9	194.8	142.8
12×6×2	139.3	141.1	155.5	212.6	153.3	50×6×2	115.2	143.8	140.1	208.6	143.4
16×2×1	390.3	493.7	391.8	503.7	407.5	150×2×1	822.1	901.2	882.7	1 047.2	1 000.5
16×2×2	422.4	451.4	422.4	507.9	432.9	150×2×2	721.9	775.4	730.6	869.8	730.7
16×4×1	265.3	337.8	274.1	339.1	318.2	150×4×1	484.7	494.1	446.8	587.3	502.9
16×4×2	195.3	210.5	198.2	221.1	196.8	150×4×2	424.9	446.9	440.1	523.2	553.0
16×6×1	133.5	154.4	147.7	159.7	140.1	150×6×1	331.2	347.8	334.5	482.9	470.8
16×6×2	141.9	163.3	159.4	165.8	147.5	150×6×2	275.1	352.8	307.0	427.4	515.4
20×2×1	451.2	547.9	502.2	500.4	453.2	250×2×1	1 283.6	1 521.3	1 373.1	1 615.2	1 400.2
20×2×2	463.4	496.5	465.7	539.1	529.7	250×2×2	1 173.6	1 263.1	1 248.3	1 420.5	1 561.5
20×4×1	322.3	422.3	334.7	399.7	386.7	250×4×1	746.4	835.2	725.8	860.0	945.1
20×4×2	263.3	290.3	276.6	290.3	287.8	250×4×2	728.7	884.2	863.4	868.6	743.3
20×6×1	199.3	209.2	210.1	239.3	207.9	250×6×1	592.8	784.3	648.5	801.7	614.6
20×6×2	186.3	197.7	178.2	198.1	194.3	250×6×2	561.3	642.6	581.2	721.8	745.7
25×2×1	573.4	720.2	593.1	685.6	592.1	350×2×1	1 923.2	2 263.8	2 031.1	2 334.9	2 139.3
25×2×2	581.2	589.6	587.0	600.7	624.5	350×2×2	1 786.0	1 980.5	1 874.2	2 257.2	1 854.7
25×4×1	305.3	430.1	369.7	363.8	352.9	350×4×1	1 078.8	1 305.3	1 154.1	1 341.6	1 052.0
25×4×2	362.2	398.6	389.5	374.3	378.3	350×4×2	947.1	1 156.1	1 013.2	1 083.2	1 337.1
25×6×1	211.2	241.3	221.3	260.3	246.8	350×6×1	782.4	1 266.5	866.7	1 038.7	899.4
25×6×2	194.0	208.0	215.0	237.3	216.9	350×6×2	675.8	914.6	780.1	861.6	834.0

表 3 5 种算法关于 avg 的计算结果
Table 3 Computational results of five algorithms on avg

实例	DABC	ABC	TAFOA	HPSOGA	MPH	实例	DABC	ABC	TAFOA	HPSOGA	MPH
8×2×1	210.5	211.0	239.8	300.9	216.2	30×2×1	813.3	893.6	858.6	1 020.1	879.5
8×2×2	246.2	246.2	248.2	261.5	246.2	30×2×2	652.6	681.6	664.2	720.2	680.7
8×4×1	114.3	114.8	132.3	131.1	126.0	30×4×1	362.1	390.3	388.9	430.9	439.1
8×4×2	114.8	114.8	148.1	148.0	144.9	30×4×2	331.9	377.6	353.5	370.9	329.0
8×6×1	103.0	103.0	108.2	104.8	104.2	30×6×1	299.0	335.2	327.8	359.2	368.7
8×6×2	106.0	106.0	106.2	106.5	106.3	30×6×2	248.7	271.8	281.7	288.2	284.9
12×2×1	228.6	235.3	264.8	295.7	260.3	50×2×1	286.8	320.3	308.0	361.9	306.3
12×2×2	408.4	481.2	431.8	452.3	404.8	50×2×2	265.3	295.3	290.0	313.3	293.8
12×4×1	215.3	264.2	222.4	246.3	243.9	50×4×1	199.6	230.0	222.5	262.2	228.8
12×4×2	187.9	216.2	207.4	209.4	191.0	50×4×2	159.2	187.4	178.2	216.9	180.0
12×6×1	142.1	145.5	158.1	229.0	159.4	50×6×1	125.7	150.3	145.0	211.1	148.6
12×6×2	141.9	147.8	164.0	217.0	161.7	50×6×2	121.1	148.6	146.8	222.9	150.2
16×2×1	394.8	498.3	397.3	516.5	421.6	150×2×1	829.9	918.5	901.9	1 055.7	1 004.1
16×2×2	424.2	455.1	428.5	512.1	438.4	150×2×2	737.7	788.8	746.0	875.2	754.4
16×4×1	273.1	347.9	287.8	347.0	335.6	150×4×1	495.0	507.3	455.5	591.2	507.5
16×4×2	198.1	222.8	205.4	231.5	205.4	150×4×2	426.9	470.0	441.8	544.0	561.5
16×6×1	139.1	165.2	162.9	168.7	150.2	150×6×1	337.9	364.0	345.8	487.2	485.2
16×6×2	143.7	170.0	169.3	173.2	151.6	150×6×2	285.3	357.3	322.8	437.3	524.8
20×2×1	455.0	551.6	517.4	503.3	463.6	250×2×1	1 288.9	1 552.1	1 397.5	1 624.8	1 406.5
20×2×2	467.4	510.8	477.8	544.2	538.3	250×2×2	1 178.6	1 267.9	1 252.2	1 427.2	1 569.3
20×4×1	330.7	429.1	345.7	421.0	388.1	250×4×1	755.0	841.0	739.5	863.5	952.8
20×4×2	266.6	309.1	280.1	299.3	310.9	250×4×2	735.7	888.6	867.6	874.7	752.9
20×6×1	201.3	216.9	240.5	259.2	217.4	250×6×1	597.2	790.4	651.3	808.8	620.4
20×6×2	189.4	202.5	183.1	204.0	202.8	250×6×2	565.2	649.8	591.4	745.3	757.7
25×2×1	582.2	740.0	600.7	693.3	601.7	350×2×1	1 929.7	2 275.8	2 052.2	2 340.9	2 144.7
25×2×2	584.6	596.2	593.7	608.9	649.2	350×2×2	1 819.3	1 985.2	1 882.6	2 264.8	1 861.6
25×4×1	310.6	440.5	391.3	376.1	363.6	350×4×1	1 084.5	1 310.9	1 158.2	1 346.7	1 055.7
25×4×2	369.1	405.5	398.4	383.2	389.9	350×4×2	959.8	1 167.3	1 022.0	1 089.2	1 346.9
25×6×1	216.9	252.9	230.7	266.0	253.9	350×6×1	790.6	1 271.7	870.3	1 048.9	908.4
25×6×2	198.1	211.9	227.7	244.6	229.8	350×6×2	683.4	919.6	792.2	870.2	839.4

表 4 5 种算法关于 max 的计算结果
Table 4 Computational results of five algorithms on max

实例	DABC	ABC	TAFOA	HPSOGA	MPH	实例	DABC	ABC	TAFOA	HPSOGA	MPH
8×2×1	210.5	211.2	240.2	310.2	220.3	30×2×1	831.1	917.3	1 003.0	1 089.4	1 176.8
8×2×2	246.2	246.2	249.3	263.0	246.2	30×2×2	655.3	685.9	716.0	746.8	778.6
8×4×1	114.3	115.9	140.3	165.4	126.0	30×4×1	364.2	395.7	426.7	458.4	491.1
8×4×2	114.8	114.8	155.9	197.7	240.5	30×4×2	335.3	385.3	434.8	485.0	334.9
8×6×1	103.0	103.0	111.1	119.9	129.7	30×6×1	305.2	353.9	402.1	451.0	500.9
8×6×2	106.0	106.0	106.8	108.3	110.8	30×6×2	249.3	280.5	311.2	342.6	375.0
12×2×1	229.0	239.7	277.3	315.6	354.9	50×2×1	299.2	334.2	368.7	403.9	440.1
12×2×2	412.9	485.5	557.6	630.4	704.2	50×2×2	270.5	300.9	330.8	361.4	393.0
12×4×1	218.3	269.3	319.8	371.0	423.2	50×4×1	203.2	237.7	271.7	306.4	342.1
12×4×2	188.9	220.1	250.8	282.2	314.6	50×4×2	162.3	195.9	229.0	262.8	297.6

续表

实例	DABC	ABC	TAFOA	HPSOGA	MPH	实例	DABC	ABC	TAFOA	HPSOGA	MPH
12×6×1	144.3	148.3	151.8	156.0	161.2	50×6×1	130.7	155.3	179.4	204.2	230.0
12×6×2	143.6	150.3	156.5	163.4	171.3	50×6×2	124.8	153.7	182.1	211.2	241.3
16×2×1	398.3	500.2	601.6	703.7	806.8	150×2×1	837.5	926.3	1 014.6	1 103.6	1 193.6
16×2×2	425.9	459.3	492.2	525.8	560.4	150×2×2	750.3	803.5	856.2	909.6	964.0
16×4×1	277.3	356.1	434.4	513.4	593.4	150×4×1	506.1	511.9	517.2	523.2	530.2
16×4×2	200.7	232.4	263.6	295.5	328.4	150×4×2	429.3	482.3	534.8	588.0	642.2
16×6×1	142.7	169.9	196.6	224.0	252.4	150×6×1	340.2	377.3	413.9	451.2	489.5
16×6×2	145.2	180.8	215.9	251.7	288.5	150×6×2	295.3	362.8	429.8	497.5	566.2
20×2×1	456.0	556.3	656.1	756.6	858.1	250×2×1	1 299.8	1 578.1	1 855.9	2 134.4	2 413.9
20×2×2	469.1	515.1	560.6	606.8	654.0	250×2×2	1 182.2	1 280.0	1 377.3	1 475.3	15 74.3
20×4×1	335.2	533.2	730.7	928.9	1 128.1	250×4×1	759.6	855.1	950.1	1 045.8	1 142.5
20×4×2	268.7	319.2	369.2	419.9	471.6	250×4×2	743.2	899.3	1 054.9	1 211.2	1 368.5
20×6×1	203.6	221.8	239.5	257.9	277.3	250×6×1	599.9	799.1	997.8	1 197.2	1 397.6
20×6×2	193.7	208.7	223.2	238.4	254.6	250×6×2	567.8	657.3	746.3	836.0	926.7
25×2×1	585.3	766.3	946.8	1 128.0	1 310.2	350×2×1	1 935.2	2 288.2	2 640.7	2 993.9	3 348.1
25×2×2	588.3	600.3	611.8	624.0	637.2	350×2×2	1 840.9	1 990.5	2 139.6	2 289.4	2 440.2
25×4×1	320.6	453.2	585.3	718.1	851.9	350×4×1	1 095.0	1 314.9	1 534.3	1 754.4	1 975.5
25×4×2	375.2	411.1	446.5	482.6	519.7	350×4×2	968.1	1 177.6	1 386.6	1 596.3	1 807.0
25×6×1	220.3	262.3	303.8	346.0	389.2	350×6×1	800.1	1 277.2	1 753.8	2 231.1	2 709.4
25×6×2	199.3	213.4	227.0	241.3	256.6	350×6×2	689.4	922.3	1 154.7	1 387.8	1 621.9

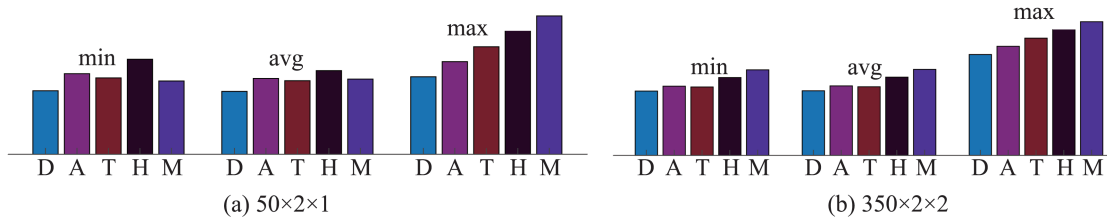


图2 2个实例的对比图

Fig. 2 Comparisons of two instances

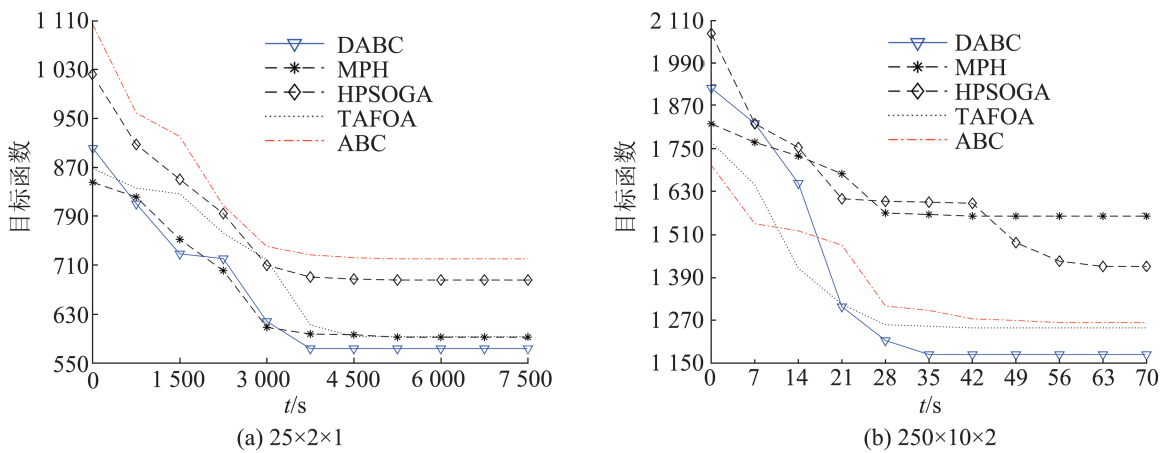


图3 所有算法的收敛曲线

Fig. 3 Convergence curves of all algorithms

<http://www.china-simulation.com>

表 2~4 也显示了 DABC 与其他 3 个对比算法之间的性能差异。DABC 算法中 50 个实例取得了优于对比算法的 min 值, DABC 算法中 49 个实例所得到的 avg 均小于对比算法, DABC 算法中 56 个实例的 max 值均小于对比算法, DABC 在收敛性、平均结果和稳定性方面性能占优, 且该结论在统计意义下也是成立的。

4 结论

本文针对考虑附加资源和学习效应的 UPMSPP, 提出了一种动态人工蜂群算法, 以最小化最大完成时间, 该算法采用新的编码和解码过程, 建立 2 个蜂群, 并根据一种新的蜂群动态评估策略确定雇佣蜂群和跟随蜂群, 且雇佣蜂阶段和跟随蜂阶段搜索策略各异。利用 60 个实例进行仿真实验, 实验结果表明, DABC 的新策略合理有效, 且 DABC 在求解所研究的 UPMSPP 方面具有较强的搜索优势。

考虑多种实际工况与约束的 UPMSPP 和制造过程实际情况更为接近, 其研究成果应用价值更高。未来将继续研究这类 UPMSPP。另外, 同时考虑多种工况与约束的混合流水车间调度问题也值得深入研究, 尤其是考虑多种实际约束的分布式混合流水车间调度问题。

参考文献:

- [1] Cheng T C E, Sin C C S. A State-of-the-art Review of Parallel-machine Scheduling Research[J]. *European Journal of Operational Research*, 1990, 47(3): 271-292.
- [2] Zheng Xiaolong, Wang Ling. A Two-stage Adaptive Fruit Fly Optimization Algorithm for Unrelated Parallel Machine Scheduling Problem with Additional Resource Constraints[J]. *Expert Systems with Applications*, 2016, 65: 28-39.
- [3] Fanjul-Peyro L, Perea F, Rubén Ruiz. Models and Matheuristics for the Unrelated Parallel Machine Scheduling Problem with Additional Resources[J]. *European Journal of Operational Research*, 2017, 260(2): 482-493.
- [4] Yepes-Borrero J C, Villa F, Perea F, et al. GRASP Algorithm for the Unrelated Parallel Machine Scheduling Problem with Setup Times and Additional Resources[J]. *Expert Systems with Applications*, 2020, 141: 112959.
- [5] Lei Deming, He Shaosi. An Adaptive Artificial Bee Colony for Unrelated Parallel Machine Scheduling with Additional Resource and Maintenance[J]. *Expert Systems with Applications*, 2022, 205: 117577.
- [6] Soleimani H, Ghaderi H, Tsai P W, et al. Scheduling of Unrelated Parallel Machines Considering Sequence-related Setup Time, Start Time-dependent Deterioration, Position-dependent Learning and Power Consumption Minimization[J]. *Journal of Cleaner Production*, 2020, 249: 119428.
- [7] Zhang Like, Deng Qianwang, Lin Ruihang, et al. A Combinatorial Evolutionary Algorithm for Unrelated Parallel Machine Scheduling Problem with Sequence and Machine-dependent Setup Times, Limited Worker Resources and Learning Effect[J]. *Expert Systems with Applications*, 2021, 175: 114843.
- [8] 轩华, 张寒, 王薛苑. 带恶化和学习效应的不相关并行机调度优化[J]. *控制工程*, 2023, 30(5): 769-778.
Xuan Hua, Zhang Han, Wang Xueyuan. Optimization on Unrelated Parallel Machine Scheduling with Deteriorating and Learning Effects[J]. *Control Engineering of China*, 2023, 30(5): 769-778.
- [9] Sharma N, Sharma H, Sharma A. Beer Froth Artificial Bee Colony Algorithm for Job-shop Scheduling Problem[J]. *Applied Soft Computing*, 2018, 68: 507-524.
- [10] Lin W C, Xu Jianyou, Bai Danyu, et al. Artificial Bee Colony Algorithms for the Order Scheduling with Release Dates[J]. *Soft Computing*, 2019, 23(18): 8677-8688.
- [11] Lei Deming, Yuan Yue, Cai Jingcao. An Improved Artificial Bee Colony for Multi-objective Distributed Unrelated Parallel Machine Scheduling[J]. *International Journal of Production Research*, 2021, 59(17): 5259-5271.
- [12] Zhang Rui, Chang P C, Song Shiji, et al. A Multi-objective Artificial Bee Colony Algorithm for Parallel Batch-processing Machine Scheduling in Fabric Dyeing Processes[J]. *Knowledge-Based Systems*, 2017, 116: 114-129.
- [13] Cao Yang, Shi Haibo. An Adaptive Multi-strategy Artificial Bee Colony Algorithm for Integrated Process Planning and Scheduling[J]. *IEEE Access*, 2021, 9: 65622-65637.
- [14] Liu Qihao, Li Xinyu, Gao Liang, et al. Mathematical Model and Discrete Artificial Bee Colony Algorithm for Distributed Integrated Process Planning and Scheduling[J]. *Journal of Manufacturing Systems*, 2021, 61: 300-310.

- [15] Ying K C, Lin S W. Unrelated Parallel Machine Scheduling with Sequence-and Machine-dependent Setup Times and Due Date Constraints[J]. International Journal of Innovative Computing, Information & Control, 2012, 8(5): 3279-3297.
- [16] Lin S W, Ying K C. ABC-based Manufacturing Scheduling for Unrelated Parallel Machines with Machine-dependent and Job Sequence-dependent Setup Times[J]. Computers & Operations Research, 2014, 51: 172-181.
- [17] 雷德明, 杨海. 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[J]. 控制与决策, 2022, 37(5): 1174-1182.
Lei Deming, Yang Hai. Multi-colony Artificial Bee Colony Algorithm for Multi-objective Unrelated Parallel Machine Scheduling Problem[J]. Control and Decision, 2022, 37(5): 1174-1182.
- [18] Lu Shaojun, Liu Xinbao, Pei Jun, et al. A Hybrid ABC-TS Algorithm for the Unrelated Parallel-batching Machines Scheduling Problem with Deteriorating Jobs and Maintenance Activity[J]. Applied Soft Computing, 2018, 66: 168-182.
- [19] Villa F, Vallada E, Fanjul-Peyro L. Heuristic Algorithms for the Unrelated Parallel Machine Scheduling Problem with One Scarce Additional Resource[J]. Expert Systems with Applications, 2018, 93: 28-38.
- [20] Salehi Mir M S, Rezaeian J. A Robust Hybrid Approach Based on Particle Swarm Optimization and Genetic Algorithm to Minimize the Total Machine Load on Unrelated Parallel Machines[J]. Applied Soft Computing, 2016, 41: 488-504.