

基于新型人工蜂群算法的分布式不相关并行机调度

刘美瑶, 雷德明[†]

(武汉理工大学 自动化学院, 湖北 武汉 430070)

摘要: 针对考虑预防性维修的分布式不相关并行机调度问题, 提出了一种新型人工蜂群算法(ABC)以最小化最大完成时间. 为了获得高质量的计算结果, 该算法将整个种群划分为1个引领蜂群和3个跟随蜂群, 跟随蜂有自己的蜜源且采用新方式跟随引领蜂, 4种蜂群运用彼此各异的搜索策略产生新解以增强种群多样性, 提出一种新策略处理侦查蜂的搜索, 并利用优化数据更新整个种群. 通过大量仿真实验验证了新型ABC在求解所研究问题方面的有效性和优势.

关键词: 预防性维修; 人工蜂群算法; 调度; 种群划分

引用格式: 刘美瑶, 雷德明. 基于新型人工蜂群算法的分布式不相关并行机调度. 控制理论与应用, 2020, 37(5): 1080 – 1089

DOI: 10.7641/CTA.2019.90242

New artificial bee colony for distributed unrelated parallel machine scheduling

LIU Mei-yao, LEI De-ming[†]

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

Abstract: To deal with the distributed unrelated parallel machines scheduling problem (DUPMSP) with preventive maintenance (PM), a new artificial bee colony (ABC) algorithm is proposed to minimize the maximum completion time. To obtain new solutions with high quality, the whole population is divided into one employed bee colony and three onlooker bee colonies. Each onlooker bee has its own food source and selects an employed bee using a novel way. In order to intensify the diversity of the whole population, different search strategies are adopted in four bee colonies to generate new solutions. A novel method is proposed to deal with the search of scout and the whole population is updated by using optimization data obtained in the search process of ABC. A number of simulation experiments are conducted on a set of instances to test the performance of the new ABC. The computational results demonstrate that the new ABC is effective and efficient to solve the considered DUPMSP with PM.

Key words: preventive maintenance; artificial bee colony algorithm; scheduling; colony division

Citation: LIU Meiyao, LEI Deming. New artificial bee colony for distributed unrelated parallel machine scheduling. *Control Theory & Applications*, 2020, 37(5): 1080 – 1089

1 引言

随着经济和贸易全球化的不断发展, 制造业从集中式制造转变为分布式制造, 分布式制造已成为常见制造模式, 相应的生产调度问题也变成分布式调度问题^[1–2], 分布式调度比传统的调度更复杂, 在快速响应市场需求方面优势更大. 并行机调度是实际制造过程中一类典型的调度问题, 它在分布式环境下扩展为分布式并行机调度问题(distributed parallel machines scheduling problem, DPMSP), DPMSP在实际分布式制造过程应用广泛^[3–4], 如包括Polyvinyl Chloride管^[4]

在内的化学品生产, 其研究具有重要的理论和现实意义.

近年来, DPMSP研究取得了一些进展^[4–10]. Lei等^[4]设计了一种带记忆的帝国竞争算法. Behnamian等分别应用遗传算法(genetic algorithm, GA), 变邻域搜索和双阶段算法等解决了多种类型的DPMSP^[8–10]. 只是现有研究主要关注了各工厂机器数量和机器加工能力等完全一样的同构工厂DPMSP, 较少关注异构工厂环境下的DPMSP, 且并行机多为同速并行机, 很少考虑不相关并行机. 由于不相关并行机在制造过程

收稿日期: 2019–04–14; 录用日期: 2019–09–27.

[†]通信作者. E-mail: deminglei11@163.com; Tel.: +86 15327311013.

本文责任编辑: 王凌.

国家自然科学基金项目(61573264, 71471151)资助.

Supported by the National Natural Science Foundation of China (61573264, 71471151).

中普遍存在^[4], 且实际多工厂往往为异构工厂, 导致相应的调度问题更复杂, 求解困难更大, 因此, 有必要研究异构工厂下的分布式不相关并行机调度问题(distributed unrelated parallel machines scheduling problem, DUPMSP).

调度问题建模时, 通常假设所有机器在调度过程中一直可用, 而实际制造过程中机器难免会因为老化、违规操作、零件磨损、管理不善和外界因素等需要进行预防性维修(preventive maintenance, PM)^[11-12]. PM的时间窗固定或动态变化. 动态时间窗更灵活, 能有效地减少机器空闲时间, 从而大幅提高机器的利用率, 因此, 具有动态时间窗的PM受到广泛关注. 目前, 考虑PM的调度问题有一些进展^[13-24], 不过, 以上研究主要在单工厂环境下进行, 分布式环境下考虑PM的调度问题, 尤其是具有PM的DUPMSP, 未引起研究者的关注.

人工蜂群算法(artificial bee colony, ABC)^[25]由土耳其学者Karaboga于2005年提出, 它具有控制参数少、操作简单、易于实现等优点. 目前ABC已成功解决了单工厂环境下的多种调度问题^[26-30], 不过, ABC在分布式调度方面的应用研究才刚刚起步^[31-32]. 由于ABC的显著特点及其在求解调度问题方面的优势, 本文探讨ABC在DUPMSP方面的应用可能.

针对考虑PM的DUPMSP, 提出一种新型ABC以最小化最大完成时间. 该算法将整个种群划分成4部分, 其中跟随蜂群划分为3部分, 跟随蜂有自己的蜜源, 引领蜂、跟随蜂和侦查蜂搜索策略各异, 并利用优化数据更新整个种群. 通过大量的仿真实验验证了新型ABC在求解所研究问题方面的有效性和优势.

2 问题描述

DUPMSP描述如下: 假设 n 个工件分配在 F 个工厂内加工, 工厂 f 内有 m_f 台不相关并行机 $M_{s_f+1}, \dots, M_{s_f+m_f}$, 其中 $s_f = \sum_{l=1}^{f-1} m_l (f > 1, s_1 = 0)$, 总共 $W = \sum_{f=1}^F m_f$ 台机器. 每个工件只有一道加工工序, 只需在任一台机器上加工一次, 工件 J_i 在机器 M_k 上的加工时间为 p_{ik} .

每台机器都有多个PM时间窗, 机器 M_k 必须在动态时间窗 $[u_{kl}, v_{kl}]$ 内进行PM, 维护时长为 w_{kl} , 其中 $(v_{kl} - u_{kl}) \geq w_{kl}, l = 1, 2, \dots$. 机器 M_k 第 l 次PM的起始时间, 结束时间分别为 S_{kl} 和 E_{kl} , $S_{kl} \geq u_{kl}$ 且 $E_{kl} \leq v_{kl}$, $E_{kl} = S_{kl} + w_{kl}$.

还需满足以下约束: 所有机器在零时刻均可加工工件, 工件在加工过程中一经打断需重新开始, 一台机器同一时刻只能加工一个工件, 每个工件在同一时刻只能在一台机器上加工等.

DUPMSP包括3个子问题^[9]: 1) 工厂分配, 确定所

有工件在各工厂之间的分配; 2) 机器分配, 在每个工厂为工件确定合适的机器; 3) 调度, 确定各机器上工件的加工顺序. 两个分配子问题之间具有非常紧密的联系, 工厂分配决定了机器分配的内容, 而工厂分配的效果取决于机器分配和调度. 若扩展机器分配, 在所有工厂为工件确定合适的机器, 一旦机器分配确定, 就可统计各工厂内的机器所分配的工件, 而确定工厂分配, 这样可将两个分配问题集成为扩展的机器分配问题.

问题的目标为在满足所有约束条件下最小化最大完成时间.

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

其中: C_i 表示工件 J_i 的完成时间, $C_{\max}$ 为最大完成时间.

表1给出了DUPMSP的一个实例的加工时间, 有2个工厂和30个工件, 其中工厂1有2台不相关并行机, 工厂2有3台不相关并行机, 对应的一种调度方案如图1所示.

表1 加工时间
Table 1 Processing time

工件	机器					工件	机器				
	1	2	3	4	5		1	2	3	4	5
1	50	38	43	49	47	16	50	42	30	42	50
2	46	42	30	49	50	17	45	32	48	34	33
3	44	35	43	36	37	18	47	40	35	38	46
4	38	43	44	48	48	19	48	45	44	34	44
5	39	39	47	36	49	20	43	41	33	48	44
6	43	42	35	50	38	21	33	44	49	36	33
7	38	42	38	39	30	22	48	42	47	35	30
8	34	35	37	34	33	23	32	47	32	34	42
9	47	48	41	39	42	24	42	32	42	45	36
10	32	30	34	38	31	25	34	48	35	39	41
11	41	38	37	48	33	26	38	39	43	50	49
12	37	38	48	49	31	27	38	46	45	40	31
13	40	41	44	45	45	28	37	36	44	40	48
14	43	40	47	37	40	29	31	32	42	39	44
15	50	42	44	39	40	30	35	46	44	38	33

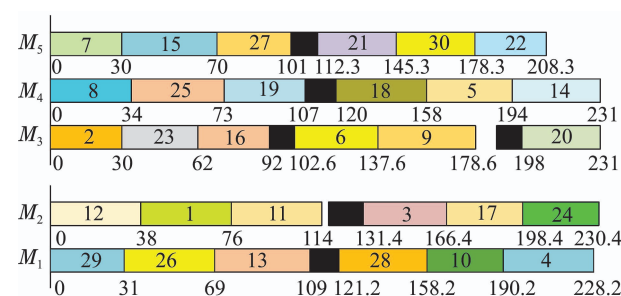


图1 实例的甘特图

Fig. 1 The Gantt chart of the example

3 基于新型ABC的DUPMSP

3.1 初始化

由于工厂分配和机器分配集成为扩展的分配问题, 整个问题由扩展的机器分配和调度组成, 为此, 采用双串编码方式. 对于具有 n 个工件和 W 台机器的问题, 其解由机器分配串和调度串组成, 其中机器分配串 $[M_{h_1} M_{h_2} \cdots M_{h_i} \cdots M_{h_n}]$ 中, M_{h_i} 为加工 J_i 的机器; 调度串 $[q_1 q_2 \cdots q_i \cdots q_n]$ 中, q_i 为工件 J_i 对应的实数.

调度串为实数串, 假设工件 $J_i, J_{i+1}, \dots, J_j$ 在同一台机器上加工, 即 $M_{h_i} = M_{h_{i+1}} = \cdots = M_{h_j}$, 则这些工件的加工顺序由 q_l 的升序决定, 其中 $l \in [i, j]$.

解码过程如下: 对每台机器, 根据第1个串确定所分配的所有工件并按这些工件对应的 q_l 的升序排列, 确定该机器上所有工件的加工顺序. 假设机器 M_k 上工件的加工顺序为 $J_{k_1}, J_{k_2}, \dots, J_{k_g}$, 对于机器的第 l 次PM, 根据如下3种情况之一安排工件加工:

1) 如果 $\alpha_k + p_{k_h k} \leq u_{kl}$, 则 $C_{k_h} = \alpha_k + p_{k_h k}$, $\alpha_k = C_{k_h}$;

2) 如果 $\alpha_k + p_{k_h k} > u_{kl}$ 且 $\alpha_k + p_{k_h k} + w_{kl} \leq v_{kl}$, 则 $S_{kl} = \alpha_k + p_{k_h k}$, $\alpha_k = S_{kl} + w_{kl}$, $l = l + 1$;

3) 如果 $\alpha_k + p_{k_h k} + w_{kl} > v_{kl}$, 则 $S_{kl} = u_{kl}$, $\alpha_k = S_{kl} + w_{kl}$, $l = l + 1$.

其中 α_k 表示 M_k 最早空闲的时间.

如上所示, 部分工件的开始时间和完成时间与PM的开始时间密切相关, 增加了解码过程的复杂性. 如果不考虑PM且目标函数仅为最大完成时间, 则无须优化调度子问题; 随着PM的加入, 需要考虑调度串的优化, 主要在于PM的开始时间会随工件的加工顺序的变化而改变.

图2给出了表1实例的一种可能解, 工件 $J_1, J_3, J_{11}, J_{12}, J_{17}, J_{24}$ 分配给 M_2 , 这些工件对应的 q_i 的升序排列为0.31, 0.33, 0.42, 0.49, 0.68, 0.86, 故先加工 J_{12} , 再加工 J_1 , 依此类推, 对应的甘特图如图1所示.

1	2	3	4	5	6	7	8	9	10
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
2	3	2	1	4	3	5	4	3	1
0.33	0.35	0.49	0.97	0.85	0.78	0.17	0.08	0.79	0.94
11	12	13	14	15	16	17	18	19	20
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
2	2	1	4	5	3	2	4	4	3
0.42	0.31	0.79	0.92	0.33	0.64	0.68	0.49	0.39	0.91
21	22	23	24	25	26	27	28	29	30
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
5	5	3	2	4	1	5	1	1	5
0.57	0.89	0.48	0.86	0.34	0.63	0.42	0.84	0.23	0.61

图2 实例的一个解

Fig. 2 A solution of the example

如图1所示, 解码过程所得到的调度方案是可行的. 实际上, 机器分配串保证了每个工件不能同时分配到两台机器上, 从而保证一个工件无法同时在两台机器上加工; 而每台机器上工件都按调度串决定的顺序依次加工, 不可能在同一时间同时加工两个工件, 也就是编码串已保证每个解都是可行的, 无不可行解存在.

随机产生初始种群 P , 并根据如下过程将种群划分为1个引领蜂群EB和3个跟随蜂群 OB_1, OB_2, OB_3 : 将初始种群的 N 个解根据 $C_{\max}$ 非降序排列, 前 $\beta_1 \times N$ 个解作为引领蜂的蜜源, 剩余解分配给跟随蜂作为蜜源, 并且所有跟随蜂细分为3类, 在剩余解中, 前 $\beta_2 \times N$ 个解组成 OB_1 , 作为第1类跟随蜂的蜜源, 第2类跟随蜂群 OB_2 的蜜源为剩下的解中 $C_{\max}$ 最小的 $\beta_3 \times N$ 个解, 剩余的 $\beta_4 \times N$ 个解分配给第3类跟随蜂群, 其中 $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$. 基于大量实验将参数 $\beta_1, \beta_2, \beta_3, \beta_4$ 分别设置为0.4, 0.1, 0.25, 0.25.

划分种群后, 不同蜂群采用不同的搜索方式, 这样可以运用多种策略产生新解, 保持种群的多样性, 并高效利用质量较好的解.

3.2 引领蜂阶段

通常, ABC不能直接用于离散型问题的求解, 为此, 在两个解之间采用两点交叉进行全局搜索. 两点交叉可作用于两个串, 且作用过程类似, 主要包括随机确定两个位置, 然后将一个解位于两个位置间的基因用另一个解对应位置上的基因替代. 局部搜索包括4个邻域结构 $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4$, 由于调度串对整个解的影响有一定限制, 未设计调度串的邻域结构.

邻域结构 $\mathcal{N}_1$ 描述如下: 先确定完成时间最大的机器 M_l 并任选一台机器 M_r , 从 M_l 上随机选择一个工件转移到 M_r 上. $\mathcal{N}_2$ 过程如下: 确定完成时间最大的机器 M_l 以及 M_l 上加工时间最大的工件 J_h , 任选一台机器 M_r 并确定其上加工时间最大的工件 J_g , 在机器分配串中互换 J_h, J_g 的位置. $\mathcal{N}_3$ 的过程与 $\mathcal{N}_2$ 类似, 只是两台机器是随机选择的, 而 $\mathcal{N}_4$ 则将随机选择的两台机器上随机选择的两个工件互换.

引领蜂群的搜索过程如下: 对于解 $x \in EB$,

1) 如果随机数 $s < \delta_1$, 则随机选择 $y \in EB$, 对解 x, y 执行机器分配串的两点交叉, 产生新解 z_1 , 若 z_1 优于 x , 则 z_1 替代 x 且 $\Omega = \Omega \cup \{x\}$. 否则对解 x, y 执行调度串的两点交叉, 得到 z_2 , $\Omega = \Omega \cup \{z_2\}$ 并更新 x .

2) 如果随机数 $s < \varepsilon_1$, 则 $g = 1$, 重复执行如下过程 R 次: 产生邻域解 $z \in \mathcal{N}_g(x)$, $\Omega = \Omega \cup \{z\}$, 并用 z 更新 x ; 否则 $g = g + 1$ 并令 $g = 1$ 若 $g = 5$.

解的更新规则遵循贪婪选择策略即若新解优于 x , 则新解替代 x . 集合 Ω 用来保留算法搜索过程产生的解.

上述过程中, 参数 δ_1, ε_1 为实数, 通过实验分别设

置为0.7, 0.1.

3.3 跟随蜂阶段

通常, 跟随蜂本身并无蜜源, 而是根据概率选择一个引领蜂的蜜源, 且由于跟随蜂只是利用引领蜂群的蜜源进行搜索, 往往采用和引领蜂同样的搜索方式. 本文为跟随蜂分配了蜜源, 并应用几种彼此各异的搜索策略产生新解.

3种跟随蜂群的具体搜索过程如下:

1) 对于每个解 $x \in OB_1$.

a) 如果随机数 $s < \delta_2$, 则对解 x 和 EB 中的最优解 x^* 执行第3.2节中 x, y 之间的交叉过程;

b) 如果随机数 $s < \varepsilon_2$, 则对解 x 执行第3.2节所述的多邻域搜索.

2) 对于每个解 $x \in OB_2$.

a) 如果 $t \leq T$, 当随机数 $s < \delta_3$ 时, 对解 x 和 EB 中的最优解 x^* 执行第3.2节中 x, y 之间的交叉过程;

b) 否则, 当 $s < \varepsilon_3$ 时, 随机选择 $y \in EB, g=1$, 执行如下过程 R 次: 产生邻域解 $z \in \mathcal{N}_g(y)$, 若 z 优于 x , 则 z 替代 $x, \Omega = \Omega \cup \{x\}$; 否则 $g = g + 1$, 令 $g = 1$ 若 $g = 5$.

3) 对于每个解 $x \in OB_3$.

如果随机数 $s < \varepsilon_4$, 则随机选择 $y \in EB$, 执行和 $x \in OB_2$ 第2种情况下同样的多邻域搜索.

上述过程中, 参数 $\delta_2, \delta_3, \varepsilon_2, \varepsilon_3, \varepsilon_4$ 为实数, 通过实验分别设置为0.9, 0.9, 0.1, 0.6, 0.5. t, T 分别表示实际的目标函数估计次数和给定目标函数估计次数, 基于大量实验设置 $T = 2 \times 10^4$.

如上所述, OB_1 的搜索过程和 EB 中的解搜索过程类似, 只是全局搜索的对象不一样, 其他两个跟随蜂群中的解的搜索过程和方式也各异. 通过多种不同途径产生新解, 有助于避免算法搜索停滞, 增加种群多样性.

3.4 算法描述

新型ABC的具体步骤描述如下:

1) 初始化. 随机产生初始种群 P , 令 $t = 1$.

2) 如果 $t \leq \max_it$, 则

a) 令集合 Ω 为空, 将种群 P 划分为 EB, $OB_1, OB_2, OB_3, \Omega = \Omega \cup EB$;

b) 引领蜂阶段;

c) 跟随蜂阶段;

d) 侦查蜂阶段;

e) 对集合 Ω 中的解非降序排序, 排名靠前的 N 个解组成新种群.

3) 输出最终结果.

其中 $\max_it$ 表示最大目标函数估计次数, 在搜索过程中, 每产生一个解, t 自动增1.

侦查蜂阶段针对 OB_3 中的最差解 x_w , 如果随机数 $s < \theta$, 则随机选择一个解 $x \in EB$, 然后, 对 x 执行跟随蜂搜索过程第2步的第2种情况下的多邻域搜索过程, 如果所得到的最好解优于 x_w 则用前者更新后者, 其中参数 θ 为实数, 通过实验设置为0.4.

由于 OB_3 中的最差解往往是种群内的最差解, 故直接从 OB_3 中选择. 和常规侦查蜂阶段不一样, 本文未采用随机搜索, 而是从 EB 中随机确定一个解进行多邻域搜索, 这样可大概率地保证最差解得到更新和改进.

集合 Ω 用来存储 ABC 搜索过程产生的优化数据, 每一次循环中, Ω 都会清空, 搜索开始后, 先将 EB 中的解加入到 Ω 中, EB 搜索过程中两点交叉产生的新解, 无论是否优于 x , 都放入 Ω 中, 多邻域搜索产生的解也直接放入 Ω 中, 跟随蜂群产生的新解处理方式类似, 这样引领蜂群和跟随蜂群一次搜索至少产生 N 个解, 再加上 EB 中的解, 使得集合 Ω 的解至少 N 个, 且搜索过程得到的好解都保留在集合 Ω 中, 精英解也保留在 Ω 中, 为此, 将 Ω 中 N 个 $C_{\max}$ 最小的解直接替换当前种群内所有解.

新型ABC搜索过程中, PM 的处理始终在解码时进行, 这样无需在两点交叉和邻域搜索时考虑 PM 的处理, 简化了搜索算子的设计; 而且, 两点交叉和邻域搜索不改变两个编码串的可行性, 不会产生不可行解, 其流程图如图3所示, 时间复杂度为 $O(N \times G)$, 其中 G 为步骤2循环的次数.

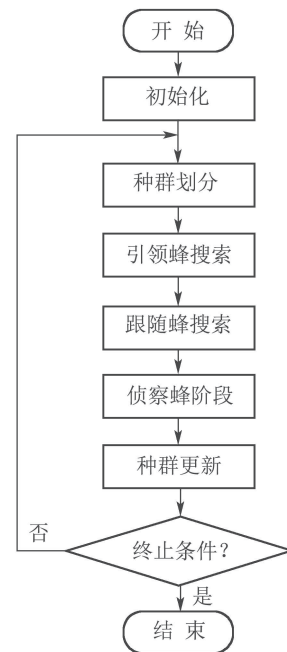


图3 新型ABC流程图

Fig. 3 Flow chart of new ABC

新型ABC具有如下特点: 将种群划分成4部分, 第1部分是由引领蜂的蜜源组成的集合, 其余3部分由跟

随蜂的解组成,不同部分采用不同的搜索策略,侦查蜂阶段用邻域搜索的解替代OB₃中最差解,并利用Ω中保留的优化数据直接更新整个种群.这些特点有助于ABC保持种群多样性,避免陷入局部最优.

4 计算实验

为了验证新型ABC在求解考虑PM的DUPMSP的有效性,进行了大量的计算实验,所有测试实验均通过Microsoft visual C++ 2017编程实现,在4.0 G RAM 2.70 GHz CPU 环境下运行.

4.1 测试实例与对比算法

本文选用64个实例测试所提算法性能,表2描述了实例的基本信息,其中具有 $n = 60, 70, 80$ 的16个实例属于中规模实例,其余为大规模实例. p_{ik} 在区间[30, 50]内随机产生,对于每台机器 M_k ,相关数据为

$$p_{k \max} = \max_{i=1, \dots, n} \{p_{ik}\}, v_{k1} \in [2p_{k \max}, 3p_{k \max}],$$
$$w_{k1} = \frac{v_{k1}}{10}, u_{k1} = v_{k1} - 2w_{k1},$$
$$v_{kl} = r_{k(l-1)} + v_{k1}, l = 2, 3, \dots,$$

其中 $r_{k(l-1)}$ 表示第 $(l-1)$ 次PM结束时间.

表 2 实例信息
Table 2 Information on instances

实例	工厂数	机器数	实例	工件数
1, 2	2	2, 3	1,3,5,7	60
3, 4	3	2, 3, 3	2,4,6,8	70
5, 6	4	2, 3, 3, 2	9, 16, 23, 30, 37, 44, 51, 58	80
7, 8	5	2, 3, 3, 2, 2	10, 17, 24, 31, 38, 45, 52, 59	100
9-15	2	4, 5	11, 18, 25, 32, 39, 46, 53, 60	120
16-22	3	4, 5, 6	12, 19, 26, 33, 40, 47, 54, 61	140
23-29	4	4, 5, 6, 3	13, 20, 27, 34, 41, 48, 55, 62	160
30-36	5	4, 5, 6, 3, 7	14, 21, 28, 35, 42, 49, 56, 63	180
37-43	2	3, 4	15, 22, 29, 36, 43, 50, 57, 64	220
44-50	3	3, 4, 4		
51-57	4	3, 4, 4, 2		
58-64	5	3, 4, 4, 2, 5		

目前, DPMSPP相关研究进展不大,关于DUPMSP的研究更少,故难以找到能直接应用于DUPMSP求解的对比算法. Behnamian和Ghomi^[9]提出了一种基于矩阵编码的GA来求解DPMSPP.在GA中,随机生成初始种群,通过复杂的交叉操作和特殊的变异产生新解,适应度最高的个体自动复制到下一代,并采用局部搜

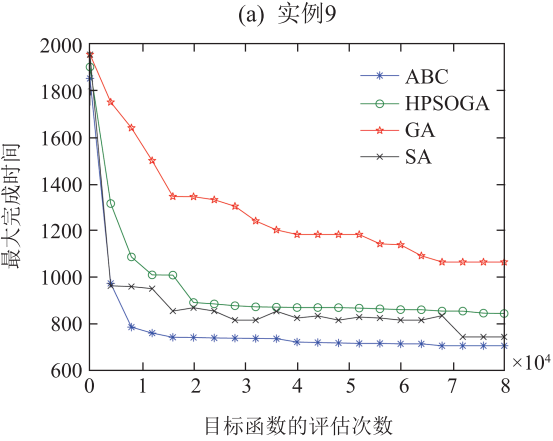
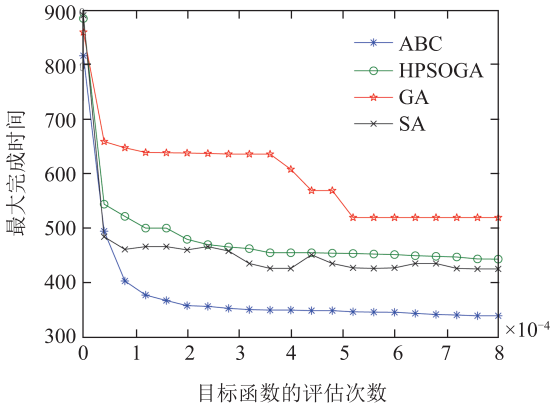
索来提高搜索能力.该算法无须任何修改就可直接用来求解带有PM的DUPMSP,因此选择GA作为对比算法.

Mir等^[33]针对不相关并行机调度问题提出了混合粒子群遗传算法(hybrid particle swarm optimization and genetic algorithm, HPSOGA),在解码过程中加入本文描述的关于PM的处理过程后,该算法可直接应用于DUPMSP的优化,故选择它作为一种对比算法.

模拟退火算法(simulated annealing, SA)具有避免陷入局部最优的特点,已被广泛应用于求解多种组合优化问题. Yeh等^[34]研究了具有学习效应的并行机调度问题,将SA应用于该问题的求解并获得了良好的效果.只需修改该算法的解码部分,便可直接用于求解DUPMSP.为验证本文所提算法性能,故选择该局部搜索算法作为对比算法.

新型ABC的主要策略是种群划分和不同蜂群采用不同的搜索方式.为验证这些策略的有效性,构建了新型ABC算法的变体ABC1,该算法没有种群划分,跟随蜂阶段中,跟随蜂随机选择一只引领蜂,并采用引领蜂相同的方式搜索.

所有算法对每个实例随机运行20次,指标MIN和MAX分别表示最优解和最差解,AVG为20次运行结果的平均值.图4描述了两个实例的收敛曲线.



(b) 实例13

图 4 收敛曲线

Fig. 4 Convergence curves

新型ABC主要参数包括 $\max_it$, N , β_1 和 R , 通过对所有实例进行参数测试实验, 可发现, 当 $N = 100$, $R = 4$, $\max_it = 8 \times 10^4$ 和 $\beta_1 = 0.4$ 时, 算法性能最好, 故采用上述参数设置. HPSOGA、GA和SA除终止条件之外的所有参数直接来自文献[9]和文献[33–34], 所有算法采用和新型ABC相同的终止条件, 实验结果如表3–5所示, 表6描述了算法的计算时间, 时间单位为s.

4.2 实验结果与比较

如表3–5所示, 和ABC1相比, 新型ABC关于64个实例中的56个取得了更优的MIN, 关于56个实例所获得的MAX更小或者等于ABC1的MAX, 关于62个实例取得了更小的AVG, 这表明了新型ABC明显优于ABC1, 从而验证了种群划分和不同部分采用不同搜索策略改进算法性能的有效性. 将种群划分为1个引

领蜂群和3个跟随蜂群, 蜂群的搜索策略各异, 这样有助于避免ABC陷入局部最优, 提高搜索效率. 配对样本均值 t 检验的结果如表7所示, $t\text{-test}(A, B)$ 表示一次配对 $t\text{-test}$, 用来判断算法A是否给出了优于B的样本均值, 假设显著水平为0.05, 若 $p\text{-value} < 0.05$, 说明算法A优于算法B.

如表3–6所示, 和3个对比算法HPSOGA、GA与SA相比, 新型ABC在较短的计算时间内所获得的搜索性能明显占优, 其中关于指标MIN, 新型ABC关于55个实例的结果比HPSOGA最少低70, 关于58个实例比GA最少低150, 关于55个实例的结果比SA最少低75, 对于其他两个指标MAX和AVG也可以得到类似结论, 而且新型ABC关于所有测试实验的计算时间也明显优于HPSOGA和GA. 表7所示的结果也显示, 新型ABC和3种对比算法之间的性能差异是显著的.

表3 5种算法关于指标MIN的计算结果

Table 3 Computational results of five algorithms on metric MIN

实例	ABC	ABC1	HPSOGA	GA	SA	实例	ABC	ABC1	HPSOGA	GA	SA
1	467.0	472.2	572.6	777.4	546.6	33	205.8	213.4	303.6	355.8	307.6
2	547.8	558.0	678.8	867.2	639.6	34	247.8	249.8	363.2	412.5	381.0
3	288.6	291.0	356.5	471.1	405.5	35	272.6	272.6	383.6	448.1	394.4
4	321.0	336.8	420.7	580.7	442.4	36	325.6	333.2	460.6	590.9	447.6
5	212.6	231.1	288.0	447.2	294.0	37	425.4	435.3	560.9	716.8	502.4
6	274.8	276.6	341.6	567.0	374.0	38	549.4	550.6	683.0	859.5	619.2
7	197.2	198.1	244.2	471.1	251.2	39	673.0	690.2	851.5	1074.0	771.6
8	213.6	214.6	273.0	580.7	275.0	40	781.2	793.0	975.2	1286.4	915.5
9	339.0	349.2	444.8	519.4	425.0	41	908.6	923.6	1093.0	1456.4	1122.4
10	430.8	433.7	537.0	673.7	541.2	42	1048.4	1135.2	1282.6	1625.4	1182.1
11	530.0	537.0	646.0	789.0	673.6	43	1247.4	1262.5	1507.4	1946.4	1432.1
12	620.0	624.0	770.6	962.3	693.9	44	284.2	294.0	362.3	476.0	348.3
13	707.6	708.6	845.4	1066.2	745.0	45	343.6	345.6	449.7	581.0	420.0
14	782.8	792.2	978.0	1205.2	921.5	46	415.7	421.4	556.1	725.8	495.2
15	929.8	939.0	1232.5	1586.2	1124.8	47	500.2	502.0	615.4	824.0	618.2
16	200.8	200.8	268.4	350.6	281.4	48	582.8	584.2	729.8	939.3	679.5
17	264.0	267.1	340.5	419.4	327.0	49	607.9	610.4	826.0	1022.3	755.5
18	302.7	305.0	398.6	513.0	393.6	50	801.8	829.2	980.3	1303.6	898.0
19	356.7	349.8	476.2	627.2	502.8	51	229.0	233.8	308.0	351.8	351.4
20	406.2	418.1	539.7	696.2	543.5	52	296.6	292.0	384.8	474.8	381.4
21	471.2	487.2	600.0	746.6	529.4	53	347.9	352.8	471.2	578.8	478.6
22	571.0	569.8	742.4	996.8	662.5	54	418.7	415.0	549.5	676.0	532.8
23	171.3	172.5	233.0	329.5	272.0	55	484.0	478.4	608.0	768.0	593.8
24	208.0	208.9	284.4	402.0	333.7	56	540.2	542.2	669.4	908.0	667.0
25	256.8	260.8	345.9	476.2	388.2	57	660.0	674.5	858.8	1142.8	843.0
26	299.4	304.2	401.5	561.8	381.2	58	171.3	172.5	233.0	307.2	272.0
27	337.2	337.4	455.2	607.8	473.2	59	208.0	208.9	284.4	363.4	333.7
28	391.8	393.8	509.2	713.5	507.2	60	256.8	260.8	345.9	438.2	388.2
29	480.0	484.6	621.0	883.0	610.0	61	299.4	304.2	401.5	513.0	381.2
30	130.5	131.8	175.2	217.0	177.0	62	337.2	337.4	455.2	617.7	473.2
31	151.5	152.9	231.2	226.8	260.0	63	391.8	393.3	509.2	632.8	507.2
32	180.7	181.5	247.0	274.8	272.0	64	484.6	481.0	621.0	824.4	610.0

表 4 5种算法关于指标MAX的计算结果

Table 4 Computational results of five algorithms on metric MAX

实例	ABC	ABC1	HPSOGA	GA	SA	实例	ABC	ABC1	HPSOGA	GA	SA
1	486.4	498.4	608.0	856.8	718.7	33	222.0	219.0	350.6	512.0	470.0
2	576.6	583.0	712.4	1023.0	915.4	34	269.4	278.4	397.0	685.5	575.4
3	308.6	308.6	389.5	537.2	559.2	35	297.0	301.6	403.0	666.3	539.6
4	349.3	360.4	440.4	669.8	621.9	36	344.2	350.8	493.4	797.1	669.6
5	255.0	255.0	312.6	535.0	466.0	37	459.9	465.4	581.5	859.0	759.1
6	278.0	279.0	361.0	604.8	528.0	38	580.6	575.4	719.4	1085.0	816.0
7	210.0	210.0	270.8	537.2	498.0	39	704.0	740.5	867.0	1204.0	1061.6
8	245.6	245.6	309.2	669.8	408.4	40	824.0	867.9	1025.1	1458.5	1156.5
9	355.3	391.0	469.9	629.5	695.0	41	991.8	981.0	1114.0	1761.0	1313.6
10	447.4	448.4	577.0	835.8	857.2	42	1100.2	1212.1	1297.0	1977.2	1613.0
11	580.0	598.2	662.0	972.2	930.0	43	1328.8	1352.2	1554.4	2572.2	1791.6
12	649.5	656.0	791.9	1073.0	1173.0	44	301.0	314.2	388.6	572.0	625.8
13	748.2	786.0	906.7	1261.2	1227.4	45	382.0	393.6	489.6	643.4	646.0
14	847.9	863.2	1010.1	1506.2	1170.8	46	444.7	456.8	585.4	923.6	732.4
15	1083.0	1185.2	1240.1	2135.0	1431.5	47	533.6	531.6	695.8	979.8	892.0
16	207.5	210.0	290.4	411.2	421.0	48	613.8	613.8	791.6	1054.0	904.2
17	269.2	278.0	365.0	466.4	473.2	49	667.3	698.3	892.0	1222.0	906.0
18	327.0	331.0	436.3	609.0	575.4	50	835.8	853.2	1012.3	1567.4	1172.3
19	383.7	383.8	513.5	723.8	708.5	51	271.2	271.3	338.2	456.7	645.6
20	436.6	470.6	581.5	850.0	689.0	52	311.4	323.6	428.8	574.6	604.4
21	493.0	502.5	630.0	945.0	878.0	53	394.6	397.2	502.6	717.4	669.3
22	598.0	604.0	765.1	1170.4	931.0	54	443.0	458.5	581.9	815.0	671.5
23	184.5	185.7	257.1	359.5	408.6	55	499.2	523.2	691.1	987.1	773.4
24	219.0	215.0	300.4	442.0	426.7	56	579.8	595.8	736.9	1135.0	1010.2
25	267.8	288.0	379.8	543.0	635.0	57	728.6	733.5	889.9	1462.0	1232.0
26	317.5	320.4	442.0	668.4	607.4	58	184.5	185.7	257.1	338.0	408.6
27	358.9	381.0	529.5	823.0	660.9	59	219.0	215.0	300.4	421.0	426.7
28	400.2	409.9	541.5	828.0	640.6	60	267.8	288.0	379.8	541.8	635.0
29	516.0	513.6	670.0	1282.6	949.2	61	317.5	320.4	442.0	594.0	607.4
30	141.6	147.0	194.8	258.4	331.0	62	358.9	381.0	529.5	722.1	660.9
31	163.5	169.8	276.6	374.8	410.5	63	400.2	409.9	541.5	831.5	640.6
32	195.3	209.7	318.0	395.6	394.8	64	516.0	514.0	670.0	1020.4	949.2

表 5 5种算法关于指标AVG的计算结果

Table 5 Computational results of five algorithms on metric AVG

实例	ABC	ABC1	HPSOGA	GA	SA	实例	ABC	ABC1	HPSOGA	GA	SA
1	479.9	484.2	590.2	816.4	626.2	33	210.2	215.0	336.6	414.3	368.3
2	564.7	571.0	698.3	955.6	750.3	34	256.8	264.3	383.2	494.2	439.6
3	294.4	297.6	375.1	507.1	453.7	35	275.3	288.3	392.4	536.3	465.2
4	336.7	348.4	435.9	620.4	498.7	36	337.2	342.5	477.3	730.2	551.2
5	238.5	247.4	302.3	501.8	367.0	37	437.1	452.2	572.1	757.1	606.7
6	277.0	277.4	351.3	582.5	443.9	38	556.3	556.7	706.6	979.8	731.7
7	202.4	204.9	253.3	507.1	358.9	39	686.5	708.0	861.4	1146.2	882.1
8	222.1	226.9	294.7	620.4	355.0	40	800.3	814.3	1005.6	1360.2	1014.9
9	350.4	359.1	455.2	579.9	512.1	41	952.8	960.3	1105.6	1582.3	1231.0
10	440.1	442.8	554.1	757.1	695.4	42	1074.2	1177.5	1288.9	1740.8	1367.7
11	558.9	573.7	656.1	873.1	785.4	43	1278.9	1286.3	1534.3	2163.9	1572.6

(转下页)

(接上页)

实例	ABC	ABC1	HPSOGA	GA	SA	实例	ABC	ABC1	HPSOGA	GA	SA
12	635.8	639.6	781.9	1018.7	913.1	44	297.6	300.9	374.4	511.0	460.2
13	724.6	725.0	876.2	1175.1	951.7	45	359.7	367.5	466.6	618.9	538.0
14	801.16	818.7	988.2	1354.4	1025.9	46	425.3	432.1	568.9	803.0	623.2
15	1003.8	1021.6	1236.7	1773.1	1238.4	47	511.2	513.1	643.6	885.6	715.8
16	203.5	206.6	278.4	378.1	356.8	48	598.0	598.7	757.8	994.2	797.1
17	266.7	270.3	349.9	449.8	392.4	49	624.2	644.1	844.5	1131.6	817.7
18	312.1	321.5	413.8	564.2	495.3	50	815.0	849.3	993.2	1383.3	1013.9
19	364.2	370.0	490.1	667.4	593.5	51	242.2	239.5	324.8	416.2	435.9
20	419.4	436.1	564.7	761.7	595.9	52	306.2	309.5	402.7	511.8	474.7
21	490.5	490.8	621.4	846.8	665.6	53	366.6	371.6	483.1	634.4	583.2
22	583.1	591.6	752.7	1085.4	793.7	54	428.2	435.0	559.7	754.9	597.5
23	179.0	180.5	239.3	344.9	347.9	55	491.7	501.2	635.8	863.7	712.8
24	211.6	212.3	293.3	425.5	391.5	56	566.2	572.5	701.4	996.6	772.1
25	262.8	271.4	364.5	510.8	468.4	57	699.5	707.5	871.5	1257.1	952.4
26	305.7	312.4	419.4	608.7	486.8	58	179.0	180.5	239.2	327.2	347.9
27	342.0	349.8	486.5	666.2	547.7	59	211.3	212.2	293.2	389.0	391.5
28	395.6	397.3	523.8	757.2	583.3	60	262.8	271.4	364.4	502.5	468.4
29	496.5	505.6	631.6	1020.4	706.9	61	305.7	314.4	419.3	559.2	486.8
30	139.0	142.0	187.4	242.9	247.6	62	342.0	349.8	486.5	653.8	547.7
31	155.7	158.5	248.1	300.1	327.8	63	395.6	397.3	523.8	692.7	583.3
32	185.4	192.4	276.9	333.7	335.3	64	505.6	496.5	631.6	901.2	706.9

表 6 ABC, HPSOGA, GA和SA的计算时间

Table 6 Computational times of ABC, HPSOGA, GA and SA

[illegible]

和GA相比, 新型ABC的编码方式更简单, 局部搜索和全局搜索之间的平衡更好; 和HPSOGA相比, 新型ABC将种群划分为多个蜂群, 且不同蜂群的搜索策略各异, 这样有利于提高搜索效率, 避免陷入局部最优; 和SA相比, 新型ABC除了局部搜索还运用了全局搜索, 种群内不同解之间进行信息交流, 有助于找到全局最优解. 新型ABC还采取和传统ABC不同的跟随蜂跟随方式, 且跟随蜂自身也有蜜源, 并利用优化过程中产生的数据来更新种群, 构造新种群, 这样策略能有效地实现探索和利用间的良好平衡, 对获得高质量的优化结果有利, 从而表明, 新型ABC是解决具有PM的DUPMSP具有较强竞争力的方法.

表7 配对样本 t 检验Table 7 Results of paired sample t -test

t 检验	p -value (min)	p -value (max)	p -value (AVG)
t -test (ABC, HPSOGA)	0.000	0.008	0.000
t -test (ABC, GA)	0.000	0.000	0.000
t -test (ABC, SA)	0.000	0.000	0.000

5 结论

在分布式环境下的调度问题中考虑机器可用约束如PM更有意义和价值. 针对考虑PM的DUPMSP, 提出一种新型ABC以最小化最大完成时间. 该算法采用全新的种群划分策略, 不同蜂群采用不同的搜索策略, 跟随蜂具有自己的蜜源并采取新的跟随方式, 采用新策略完成侦查蜂搜索和更新整个种群. 运用大量的实例对算法性能进行测试, 实验结果验证了新型ABC的有效性和搜索优势.

分布式调度是目前调度研究的焦点, 但现有研究很少讨论实际加工约束, 未来将深入研究具有随机故障, 零等待和不确定加工条件等的分布式调度问题及其相应的高效智能优化算法.

参考文献:

- [1] MARANDI F, GHOMI S M T F. Network configuration multi-factory scheduling with batch delivery: A learning-oriented simulated annealing approach. *Computers and Industrial Engineering*, 2019, 132: 293 – 310.
- [2] WANG Ling, DENG Jin, WANG Shengyao. Survey on optimization algorithms for distributed shop scheduling. *Control and Decision*, 2016, 31(1): 1 – 11.
(王凌, 邓瑾, 王圣尧. 分布式车间调度优化算法研究综述. 控制与决策, 2016, 31(1): 1 – 11.)
- [3] YAZDANI M, GOHARI S, NADERI B. Multi-factory parallel machine problems: Improved mathematical models and artificial bee colony algorithm. *Computers and Industrial Engineering*, 2015, 81: 36 – 45.
- [4] LEI D M, YUAN Y, CAI J C, et al. An imperialist competitive algorithm with memory for distributed unrelated parallel machines scheduling. *International Journal of Production Research*. 2019 (in press).
- [5] HOOKER J N. A hybrid method for planning and scheduling. *Constraints*, 2004, 10: 385 – 401.
- [6] CHEN Z L, PUNDOOR G. Order assignment and scheduling in a supply chain. *Operations Research*, 2006, 54(3): 555 – 572.
- [7] TERRAZAS-MORENO S, GROSSMANN I E. A multiscale decomposition method for the optimal planning and scheduling of multi-site continuous multiproduct plants. *Chemical Engineering Science*, 2011, 66(19): 4307 – 4318.
- [8] BEHNAMIAN J. Decomposition based hybrid VNS-TS algorithm for distributed parallel factories scheduling with virtual corporation. *Computers and Operations Research*, 2014, 52: 181 – 191.
- [9] BEHNAMIAN J, GHOMI S M T F. The heterogeneous multi-factory production network scheduling with adaptive communication policy and parallel machine. *Information Sciences*, 2013, 219: 181 – 196.
- [10] BEHNAMIAN J. Graph colouring-based algorithm to parallel jobs scheduling on parallel factories. *International Journal of Computer Integrated Manufacturing*, 2016, 29(6): 622 – 635.
- [11] SANLAVILLE E, SCHMIDT G. Machine scheduling with availability constraints. *Acta Informatica*, 1998, 35(9): 795 – 811.
- [12] ALIMIAN M, SAIDI-MEHRABAD M, JABBARZADEH A. A robust integrated production and preventive maintenance planning model for multi-state systems with uncertain demand and common cause failures. *Journal of Manufacturing Systems*, 2019, 50: 263 – 277.
- [13] SHEIKHALISHAHI M, ESKANDARI N, MASHAYEKHI A, et al. Multi-objective open shop scheduling by considering human error and preventive maintenance. *Applied Mathematical Modelling*, 2019, 67: 573 – 587.
- [14] WANG T, BALDACCI R, LIM A, et al. A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine. *European Journal of Operational Research*, 2018, 271(3): 826 – 838.
- [15] HUANG Jingying, WANG Liya. Batch-processing machine production scheduling considering preventive maintenance. *Industrial Engineering and Management*, 2018, 23(4): 129 – 136.
(黄婧颖, 王丽亚. 考虑预防性维护的批处理机调度问题研究. 工业工程与管理, 2018, 23(4): 129 – 136.)
- [16] ZHU Chuanjun, SONG Wenjia, ZHANG Chaoyong, et al. Research on optimization of FJSP based on maintenance time window. *China Mechanical Engineering*, 2016, 27(10): 1337 – 1343.
(朱传军, 宋文家, 张超勇, 等. 基于维修时间窗的柔性作业车间调度优化研究. 中国机械工程, 2016, 27(10): 1337 – 1343.)
- [17] ZHOU Binghai, LIU Zilong. Integrated scheduling method of production and preventive maintenance in flow shops with degradations. *Computer Integrated Manufacturing Systems*, 2016, 22(5): 1272 – 1278.
(周炳海, 刘子龙. 考虑衰退的流水车间生产与预防性维护集成调度方法. 计算机集成制造系统, 2016, 22(5): 1272 – 1278.)
- [18] GAN Jie, ZENG Jianchao. Integrated model of single-machine scheduling and maintenance decision for degrading state systems. *Control and Decision*, 2016, 31(3): 513 – 520.
(甘捷, 曾建潮. 考虑劣化状态的单机调度与维修决策集成模型. 控制与决策, 2016, 31(3): 513 – 520.)
- [19] SOUISSI O, BENMANSOUR R, ARTIBA A. An accelerated MIP model for the single machine scheduling with preventive maintenance. *IFAC Papersonline*, 2016, 49(12): 1945 – 1949.
- [20] KHOUKHI F E, BOUKACHOUR J, ALAOUI A E. The “Dual-AntsColony”: A novel hybrid approach for the flexible job shop scheduling problem with preventive maintenance. *Computers and Industrial Engineering*, 2017, 106: 236 – 255.

- [21] AVALOS-ROSALES O, ANGEL-BELLO F, ALVAREZ A, et al. Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times. *Computers and Industrial Engineering*, 2018, 123: 364 – 377.
- [22] ZANDIEH M, KHATAMI A R, RAHMATI S H A. Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm. *Applied Soft Computing*, 2017, 58: 449 – 464.
- [23] LIU Q M, DONG M, CHEN F F, et al. Single-machine-based joint optimization of predictive maintenance planning and production scheduling. *Robotics and Computer-Integrated Manufacturing*, 2019, 55: 173 – 182.
- [24] MOSHEIOV G, SARIG A, STRUSEVICH V A, et al. Two-machine flow shop and open shop scheduling problems with a single maintenance window. *European Journal of Operational Research*, 2018, 271(2): 388 – 400.
- [25] KARABOGA D. *An idea based on honey bee swarm for numerical optimization*. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [26] ASADZADEH L. A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy. *Computers and Industrial Engineering*, 2016, 102: 359 – 367.
- [27] LIU Gang, HUANG Chongzheng. Permutation flow shop scheduling based on hybrid discrete artificial bee colony algorithm. *Control Engineering of China*, 2017, 24(9): 1925 – 1929.
(刘刚, 黄崇争. 基于HDABC算法的置换流水车间调度策略. 控制工程, 2017, 24(9): 1925 – 1929.)
- [28] WANG Ling, ZHOU Gang, XU Ye, et al. An artificial bee colony algorithm for solving hybrid flow-shop scheduling problem with unrelated parallel machines. *Control Theory & Applications*, 2012, 29(12): 1551 – 1557.
(王凌, 周刚, 许烨, 等. 求解不相关并行机混合流水线调度问题的人工蜂群算法. 控制理论与应用, 2012, 29(12): 1551 – 1557.)
- [29] CHENG Bing, XU Hua, WANG Lingdi, et al. Improved artificial bee colony algorithm for solving multi-objective flexible job-shop scheduling problem. *Information and Control*, 2019, 48(1): 115 – 122, 128.
(程冰, 徐华, 王玲娣, 等. 改进人工蜂群求解多目标柔性作业车间调度问题. 信息与控制, 2019, 48(1): 115 – 122, 128.)
- [30] GONG D W, HAN Y Y, SUN J Y. A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems. *Knowledge-Based Systems*, 2018, 148: 115 – 130.
- [31] YAZDANI M, GOHARI S, NADERI B. Multi-factory parallel machine problems: Improved mathematical models and artificial bee colony algorithm. *Computers and Industrial Engineering*, 2015, 81: 36 – 45.
- [32] PAN Q K, GAO L, WANG L, et al. Effective heuristics and meta-heuristics to minimize total flowtime for the distributed permutation flowshop problem. *Expert Systems with Applications*, 2019, 124: 309 – 324.
- [33] MIR M S 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. *Applied Soft Computing*, 2016, 41: 488 – 504.
- [34] YEH W C, LAI P J, LEE W C, et al. Parallel-machine scheduling to minimize makespan with fuzzy processing times and learning effects. *Information Sciences*, 2014, 269: 142 – 158.

作者简介:

刘美瑶 硕士研究生, 主要从事制造系统智能优化与调度的研究,

E-mail: liumeiyaowhut@126.com;

雷德明 博士, 教授, 主要从事智能系统优化与控制的研究,

E-mail: deminglei11@163.com.