

# 带运输资源约束柔性作业车间调度问题的人工蜂群算法

李俊青<sup>1,2</sup>, 杜宇<sup>1</sup>, 田杰<sup>1</sup>, 段培永<sup>1</sup>, 潘全科<sup>3,2</sup>

(1. 山东师范大学信息科学与工程学院, 山东济南 250014; 2. 聊城大学计算机学院, 山东聊城 252059;  
3. 上海大学机电工程与自动化学院, 上海 200072)

**摘要:** 本文针对一类柔性作业车间调度问题, 综合考虑运输资源约束、工件间准备时间约束等条件, 以最小化最大完工时间和能耗为目标, 提出了一种改进的人工蜂群优化算法. 为求解该问题, 算法采用二维向量编码, 即调度向量记录工件的调度顺序, 机床分配向量记录工件分配可用机床情况, 解码过程充分考虑运输资源、工件间准备时间等约束条件. 在局部搜索策略方面, 提出了五种不同的调度邻域结构, 并根据目标特点, 设计了一种机床分配邻域结构. 围绕人工蜂群算法的三个阶段, 提出了不同的改进策略. 为进一步提升算法的全局搜索能力, 嵌入了模拟退火接受准则. 实验结果验证了所提算法的优势显著.

**关键词:** 人工蜂群; 柔性作业车间; 能耗; 起重机运输; 准备时间

**中图分类号:** TP301.6      **文献标识码:** A      **文章编号:** 0372-2112 (2021)02-0324-07

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20200382

## An Artificial Bee Colony Algorithm for Flexible Job Shop Scheduling with Transportation Resource Constraints

LI Jun-qing<sup>1,2</sup>, DU Yu<sup>1</sup>, TIAN Jie<sup>1</sup>, DUAN Pei-yong<sup>1</sup>, PAN Quan-ke<sup>3,2</sup>

(1. School of Information Science and Engineering, Shandong Normal University, Jinan, Shandong 250014, China;  
2. College of Computer Science, Liaocheng University, Liaocheng, Shandong 252059, China;  
3. School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China)

**Abstract:** The flexible job shop scheduling problem is investigated, where the transportation resource and operation related setup time constraints are considered simultaneously. The objective is to minimize the maximum completion time and the energy consumption. To solve the problem, we propose an improved artificial bee colony algorithm, where each solution is represented by a two-dimensional vector, the scheduling vector is to record the operation processing sequence, and the machine assignment vector is to assign the candidate machine for each operation. In the decoding mechanism, the transportation and setup time constraints are investigated. For the local search approaches, we develop five types of neighborhood structures for the scheduling part, and a well-designed machine assignment neighborhood structure for the machine assignment vector. To enhance the global searching abilities, the simulated annealing acceptance method is embedded. Finally, the experiment comparisons verify the performance of the proposed algorithm.

**Key words:** artificial bee colony; flexible job shop; energy consumption; crane transportation; setup time

### 1 引言

近年来, 柔性作业车间调度问题 (Flexible Job Shop Scheduling Problem, FJSP) 在学术界和工业界都得到了广泛关注<sup>[1-3]</sup>. FJSP 一般作为作业车间调度 (Job Shop

Scheduling Problem, JSSP) 问题的一类特例, 因此 FJSP 也被证明是一类典型 NP-hard 问题<sup>[1]</sup>. 在经典的 FJSP 中, 一般性约束包括: (1) 每台机床在同一时刻只能加工一个工件; (2) 每个工件在同一时刻只能在一台机床上加工; (3) 工件一旦加工不允许抢占. 然而, 在实际工

业生产过程中,往往存在各种特定加工工艺约束,譬如由于工件需要起重机搬运而产生的运输时间和能耗,工件之间的准备时间约束等.经典 FJSP 研究成果由于忽略这些实际约束而不能直接应用于实际工业生产过程中.

针对带起重机运输的 FJSP 问题, Nouri 等人(2016)<sup>[4]</sup>针对多机器人的 FJSP 问题,以最小化 makespan 为目标,提出了一种融合遗传算法、禁忌搜索和多 agent 的混合元启发方法. Liu 等人(2019)<sup>[5]</sup>以最小化总能耗和最大完工时间为目标,提出了一种遗传算法和萤火虫(Glowworm Swarm Optimization, GSO)算法相结合的混合算法(GA-GSO). Karimi 等人(2017)<sup>[6]</sup>设计了一种帝国竞争算法(Imperialist Competitive Algorithm, ICA),并结合模拟退火增强其局部搜索能力.

针对带起重机/机器人运输的能耗指标优化问题: Bukata 等人(2019)<sup>[7]</sup>提出了改进的分支定界算法来优化机器人单元的能量消耗. Gürel 等人(2019)<sup>[8]</sup>研究了机器人的循环时间和能量消耗之间的权衡问题. Kats 和 Levner(2018)<sup>[9]</sup>则针对非等待的多循环调度问题开展了研究.进一步地,诸多启发式和元启发式方法已经被成功用于解决各类优化问题,例如以局部搜索为基础的算法,包括禁忌搜索算法<sup>[10,11]</sup>和模拟退火算法<sup>[12]</sup>;以群体进化为基础的搜索方法,包括帝国竞争算法<sup>[6]</sup>、人工蜂群算法(Artificial Bee Colony, ABC)<sup>[13,14]</sup>等.

综上所述可见,对于带运输约束的 FJSP 问题,需要考虑诸多实际约束条件,如机床加工和起重机运输的能量消耗、因考虑设备高度而增加的起重机抬升操作、工件之间的准备时间等因素.文献分析可见,综合考虑上述实际约束条件的研究目前还未检索到.基于此,本文研究了一类带起重机运输和工件准备时间约束的 FJSP 问题,并以人工蜂群算法为基础,结合模拟退火等局部搜索,提出了一种混合优化算法.

## 2 问题描述

本文考虑的带起重机/机器人运输的柔性作业车间调度问题描述如下:有  $n$  个工件,  $m$  个加工机床,每个工件有  $\theta_j$  ( $j = 1, 2, \dots, n$ ) 个加工工序,  $O_{i,j,k}$  ( $i = 1, 2, \dots, \theta_j$ ) 表示工件  $j$  的第  $i$  道工序在机床  $k$  上加工,  $M_{i,j}$  表示工序  $O_{i,j}$  的可选加工机床集合.所有工件的工序数量和可选机床集合是给定且固定不变的.工序只能在其可选机床集合中选择一台机床进行加工.同一时刻,一台机床只能加工一个工序,一个工序也只能在一台机床上加工.一个工件在当前工序完工后,需要起重机/机器人搬运其到下一道工序所在机床,搬运过程的时间和能耗需要考虑.工序加工过程不允许中断或抢占.针

对该类实际生产调度问题,考虑最小化两类目标,即最大完工时间 makespan 和能耗指标,其中能耗包括机床加工能耗和起重机搬运过程产生的能耗.

### 2.1 假设条件

(1) 在时刻 0 所有工件准备就绪,并且按照分配的机床和加工顺序进行加工.

(2) 在时刻 0 所有机床准备就绪,并且在整个调度过程中处于可用状态.

(3) 每一个工件在同一时刻选择可选机床集合中的一台(且只能为一台)进行加工.

(4) 每一台机床在同一时刻只能加工一个工件.

(5) 不允许抢占情况发生,即工件一旦开工,不允许中断.

(6) 对于每个工件,第一个加工工序不需要起重机设备搬运.

(7) 起重机在同一时刻只能搬运一个工件,即不允许起重机重叠操作的情况.

(8) 起重机的初始位置在坐标(0,0).

(9) 起重机在整个调度过程中处于可使用的状态.

(10) 加工阶段之间有足够多的缓冲区用于存放等待起重机运输的工件.

(11) 同一工件的紧邻工序在同一机床上加工,则不需要起重机运输.

### 2.2 运输能耗

本文考虑的能耗包括机床加工能耗和运输能耗两部分,运输能耗又包括起重机空载能耗、起重机空载等待能耗、起重机制重运输能耗、起重机吊装运输能耗.起重机制重运输能耗用于计算起重机制重移动过程的能耗,起重机制重等待能耗用于计算起重机制重等待工序完工过程的能耗,起重机制重运输能耗用于计算起重机制重运输工序发生的能耗,而起重机吊装运输能耗用于计算起重机制重吊运工序到指定高度所发生的能耗.

与文献[5]不同,本文考虑了由于设备高度而产生的起重机制重时间和能耗.起重机制重在吊运工件  $O_{i,j,k}$  到达机床  $k$  后,由于考虑机床高度,还需要起重机制重吊运工件到指定高度.吊运过程分为三个部分,即垂直方向载重运输启动时间  $T_{vlds}(O_{i,j,k})$  (式(2))、垂直载重运输制动时间  $T_{vldb}(O_{i,j,k})$  (式(3))、垂直载重运输时间  $T_{vldd}(O_{i,j,k})$  (式(4)),  $T_{lf}(O_{i,j,k})$  如式(5)所示因吊运工序  $O_{i,j,k}$  而产生的垂直方向载重运输总时间,由式(6)计算起重机制重垂直方向载重运输  $O_{i,j,k}$  产生的消耗  $E_{lf}(O_{i,j,k})$ .

$$\gamma = \frac{W_s + W_j + W_v}{Q_n + W_v} \quad (1)$$

$$T_{vlds}(O_{i,j,k}) = \gamma \cdot \frac{V_h}{a_{vs}} \quad (2)$$

$$T_{vldb}(O_{i,j,k}) = \gamma \cdot \frac{V_h}{a_{vb}} \quad (3)$$

$$T_{vlds}(O_{i,j,k}) = \sum_{k2=1}^m x_{i,j,k2} \left( \frac{Lz_{k2}}{V_v} - \frac{V_v}{2} \cdot \gamma \cdot \left( \frac{1}{a_{vs}} + \frac{1}{a_{vb}} \right) \right) \quad (4)$$

$$T_{lf}(O_{i,j,k}) = T_{vlds}(O_{i,j,k}) + T_{vldb}(O_{i,j,k}) + T_{vlds}(O_{i,j,k}) \quad (5)$$

$$E_{lf}(O_{i,j,k}) = T_{vlds}(O_{i,j,k}) \cdot P_{ons} + T_{vldb}(O_{i,j,k}) \cdot P_{ovb} + T_{vlds}(O_{i,j,k}) \cdot P_{ov} \quad (6)$$

### 3 求解算法

人工蜂群算法作为群体智能优化算法的一种典型代表,包含雇佣蜂、跟随蜂和侦查蜂三个基本组成部分.雇佣蜂和跟随蜂执行局部搜索过程,而侦查蜂的引入强化了算法的全局搜索能力.针对带运输约束的 FJSP 问题,结合问题约束和目标特点,本节给出了求解 CFJSP 问题的改进人工蜂群算法(Improved Artificial Bee Colony, IABC).

算法主要设计思想是在基本 ABC 算法的基础上,设计适合 CFJSP 问题求解的问题编码、解码、邻域结构.进一步结合问题特点,改进 ABC 算法中的雇佣蜂、跟随蜂和侦查蜂三个基本策略,使上述三种操作算子适合于当前问题求解.为进一步提升算法搜索能力,嵌入了模拟退火接受准则.所提出的 IABC 总体框架如算法 1 所示.

算法 1 IABC 算法框架

输入:算例、算法参数、算法停止条件

输出:最好解

1. 随机方式产生  $P_{size}$  个解构成初始群体,记录当前最好解  $E_{best}$
2. 雇佣蜂阶段
3. For  $i$  to  $P_{size}$
4. 为第  $i$  个雇佣蜂  $E_i$  采用 3.3 节的调度邻域结构生成一个邻域解  $E_i^1$
5. 为  $E_i^1$  采用 3.4 节的机床分配邻域结构生成一个邻域解  $E_i^2$
6. if  $E_i^2 > E_i$ , then 替换当前解
7. if  $E_i^2 > E_{best}$ , then 替换最好解
8. end
9. 跟随蜂阶段
10. For  $i$  to  $P_{size}$
11. 随机选择两个个体,选择较优的为当前解  $E_i$
12. 为  $E_i$  采用调度和机床分配邻域生成一个邻域解  $E_i^2$
13. if  $E_i^2 > E_i$ , then 替换当前解
14. if  $E_i^2 > E_{best}$ , then 替换最好解
15. end
16. 侦查蜂阶段
17. 集合  $TT$  记录  $T_{limit}$  次迭代未更新的解,任选一个作为侦查蜂  $E_s$
18. For  $i \rightarrow S_t$
19. 为  $E_{best}$  采用调度和机床分配邻域生成一个邻域解  $E_i^1$

20. if  $E_i^1$ , then 替换最好解

21. end

22. 选择步骤 19 生成的最好邻域解替换  $E_s$

23. 模拟退火阶段

24. 令  $E_c = E_{best}$

25. For  $i \rightarrow n/4$

为  $E_c$  采用 3.3 节的调度和 3.4 节机床分配邻域生成一个邻域解  $E_c^1$

27. if  $E_c > E_c^1$

28.  $E_c = E_c^1$

29. else

30. 令  $\delta = \text{fitness}(E_c) - \text{fitness}(E_c^1)$

$$31. \quad \text{令 } \text{Temp} = \frac{T \cdot \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^{\theta_j} T_{i,j,k}}{J \cdot K \cdot 10}$$

32. if  $\text{random} \leq e^{-\frac{\delta}{\text{Temp}}}$

33.  $E_c = E_c^1$

34. end

35. end

36. end

#### 3.1 解的表示和编码

在 IABC 算法中,解的编码采用二维向量的表示方法.第一个向量命名为调度向量  $\boldsymbol{\pi} = \{\pi_1, \pi_2, \dots, \pi_\phi\}$ ,向量的长度为所有加工工序的总数  $\phi$ .调度向量中每个元素用工件编号  $\pi_i$  表示,工序在调度向量中出现的顺序即它们的加工顺序.第二个向量为机床分配向量  $\boldsymbol{\omega} = \{\omega_1, \omega_2, \dots, \omega_\phi\}$ ,向量的长度为  $\phi$ .机床分配向量位置  $i$  的元素  $\omega_i$  记录了工件  $\pi_i$  所分配到的机床编号.

#### 3.2 解码策略

解码过程主要完成以下任务,即如何确定每道工序的开始时间和完成时间,以及如何确定起重机移动的路径.

解码策略的具体步骤如下:为了确定每道工序的开始时间和完成时间,我们应从左向右逐一地调度每一道工序.在安排每道工序时,应考虑四个关键因素:起重机的空闲时间、同一工件前道工序的完工时间、加工机床的空闲时间、起重机的移动路径.前三个时间的确定较为容易,而起重机的移动路径需要谨慎考虑.

本文中,起重机的移动路径分为两大类,即平移过程和升降过程.(1)平移过程与文献[5]类似,包括了起重机空载操作、起重机空载等待、起重机载重操作三个过程.起重机的空载操作完成的任务是从当前位置移动到同一工件的前一道工序的机床位置.起重机的空载等待时间,是在等待前道工序完成.起重机的载重运输时间是将工件从前序加工位置移动到当前工序分配机床的时间.(2)升降过程,与文献[7]不同,我们还考虑了因机床的高度而发生的起重机抬升过程.因此,在本研究中首次提出了起重机的抬升操作,抬升操作是起重机装载加工工件并移动至指定高度的过程.

### 3.3 调度邻域结构

结合问题特性,本节给出了五种调度邻域结构.

#### 3.3.1 两点逆转 (TPR) 策略

两点逆转 (Two-Point Reverse, TPR) 策略,通过选定调度向量中两个位置  $r_1$  和  $r_2$ ,并逆序排列位置  $r_1$  和  $r_2$  间的元素来产生新的邻域解.需要注意的是,在改变调度向量中工序的排列位置时,由于工序编号可能会因此改变,因而机床分配向量中对应位置的工序必须做出相应的调整. TPR 策略具体流程如算法 2 所示.

#### 算法 2 TPR 策略

输入:一个可行解

输出:新的邻域解

1. 随机选择当前调度向量中任意两个位置  $r_1, r_2$
2. 记录  $r_1, r_2$  两个位置之间的工序的机床分配情况到集合  $MS$
3. 将调度向量  $\mathbf{II}$  中  $r_1, r_2$  位置之间的工序逆向排列
4. 参照  $MS$  结合,填充机床分配向量  $\mathbf{\Omega}$  中  $r_1, r_2$  位置之间的工序的机床分配

#### 3.3.2 两点交换 (TPS) 策略

两点交换 (Two-Point Swap, TPS) 策略,通过选定调度向量中两点,并交换两个选定位置的工序来产生相邻解. TPS 和 TPR 主要的区别在于上述过程的第 3 步,其中 TPS 是交换加工序列中  $r_1$  和  $r_2$  位置之间的工序.

#### 3.3.3 随机分配机床两点交换 (TPSM) 操作

在 TPS 策略基础之上,随机分配机床两点交换 (Two-Point Swap with Random Machine Assignment, TPSM) 对于交换的两个工序  $r_1$  和  $r_2$ ,随机为之重新分配机床.

#### 3.3.4 两点插入 (TPI) 操作

两点插入 (Two-Point Insertion, TPI) 策略选定调度向量中两点为  $r_1$  和  $r_2$ ,通过将位于  $r_2$  的工序插入到位置  $r_1$  之前来产生相邻解. TPI 和 TPR 主要的不同之处在于上述过程的第 3 步,其中 TPI 是在加工序列中将靠后的  $r_2$  位置的操作插入到  $r_1$  位置之前.

#### 3.3.5 随机分配机床两点插入 (TPIM) 操作

在 TPI 策略基础之上,随机分配机床两点插入 (Two-Point Insertion with Random Machine Assignment, TPIM) 对于选定的两个位置  $r_1$  和  $r_2$ ,随机为之重新分配机床.

### 3.4 机床分配邻域结构

针对能耗目标优化,本节给出了一种能耗启发式策略 (Energy Consumption Heuristic, ECH). 算法主要思想是为工序选择起重机运载能耗较小的机床,具体流程如算法 3 所示.

#### 算法 3 ECH 策略

输入:一个可行解

输出:新的邻域解

1. 随机选择当前调度向量中任意一个位置  $r$ ,记对应工序为  $O_r$
2. For  $i \rightarrow r - 1$
3. 为工序  $O_r$  选择机床分配向量的机床,并计算开工、完工时间
4. end
5. 记工序  $O_r$  当前分配机床为  $m_c$ ,当前能耗  $E_{lof}(O_r, c)$
6. While ( $m \in M_r$ )
7. 假设分配工序  $O_r$  到  $m$ ,按照式(6)计算能耗  $E_{lof}(O_r, m)$
8. if  $E_{lof}(O_r, m) < E_{lof}(O_r, c)$ , then  $m$  加入集合  $M_r$
9. end
10. If ( $M_r \neq \emptyset$ )
11. 选择能耗  $E_{lof}(O_r, m)$  最小的机床  $m^*$  分配给工序  $O_r$
12. 修改机床分配向量  $\mathbf{\Omega}$  对应位置  $r$  的元素为  $m^*$
13. end

### 3.5 雇佣蜂阶段

雇佣蜂的主要任务是在分配的食物上开展挖掘工作,搜索更好的食物源.基本 ABC 中雇佣蜂的操作算子不适合求解调度问题,在改进的离散 ABC 算法中,雇佣蜂的策略如下:

**步骤 1** 为当前解集中每个食物源分配一个雇佣蜂.

**步骤 2** 以指定解为当前解  $E_c$ ,以随机方式选择 3.3 节中调度邻域结构中的一种策略,执行调度邻域局部搜索,得到新的邻域解  $E_c^1$ .

**步骤 3** 对新产生的邻域解  $E_c^1$ ,执行 3.4 节中机床分配邻域策略,得到新的邻域解  $E_c^2$ .

**步骤 4** 如果新的邻域解  $E_c^2$  优于当前解  $E_c$ ,即  $E_c^2 > E_c$ ,则用邻域解  $E_c^2$  替换当前解  $E_c$ .

**步骤 5** 如果新的邻域解  $E_c^2$  优于最好解  $E_b$ ,即  $E_c^2 > E_b$ ,则用邻域解  $E_c^2$  替换  $E_b$ .

### 3.6 跟随蜂阶段

在雇佣蜂挖掘工作结束后,守候在蜂巢的跟随蜂在更新后的解集中以概率选择的方式挑选较优解进一步挖掘搜索.采用轮盘赌注选择方式,需要比较每个解的目标值,因而时间复杂度较高.为了提高算法效率,给出了一种简单的跟随蜂的选择策略,具体描述如下:

**步骤 1** 在当前解集中随机选择两个解  $E_1$  和  $E_2$ .

**步骤 2** 在选中的解中挑选较优解作为当前解  $E_c$ .

**步骤 3** 执行雇佣蜂策略中的步骤 2 ~ 5.

### 3.7 侦查蜂阶段

本节给出了两种侦察蜂策略,具体描述如下:

**策略 1** 随机解策略.若解集中某个解  $E_s$  在指定时间间隔内没有更新,生成一个随机解替换  $E_s$ ,并派出侦察蜂进一步挖掘.需要指出的是,策略一是传统 ABC 算法采用的侦查蜂策略.

**策略 2** 最好解策略.若解集中某个解  $E_s$  在指定时间

间隔内没有更新,则对最好解  $E_b$  进行  $n/4$  次邻域扰动,用扰动后的解替换  $E_s$ ,然后派出侦察蜂进一步挖掘搜索。

#### 4 试验分析

本文所提出的 IABC 算法,编程语言采用 C++, IDE 环境采用 Visual Studio 2017,在 16 GB 内存的 Intel Core i7 3.4-GHz 电脑上运行。为验证 IABC 算法的性能,选择的比较算法包括 Liu 等人(2019)提出的遗传算法和萤火虫群算法的混合算法(GA-GSO)<sup>[5]</sup>,Karimi 等人(2017)提出的帝国竞争算法(ICA)<sup>[6]</sup>。上述两种比较算法中,GA-GSO 算法求解问题与本文类似,都是带有起重约束的 FJSP 问题,文献[6]给出的 ICA 是一种性能较好的求解带有运输时间约束 FJSP 的算法。

综上,上述两种算法是性能较好的,且与本文研究问题类似的算法。由于本文研究带起重机运输和工件间准备时间两种约束的 CFJSP 问题,目前还没有类似文献,也没有经典的算例,因而,我们分别实现了上述两种比较算法。在实现上述两种算法的过程中,参考了相应文献中的局部搜索、全局搜索策略,编码和解码策略采用本文的实现过程。GA-GSO 和 ICA 两种算法的参数设置取自于相应文献的数据。所有算法均独立运行 30 次,得到最优解用于算法比较。算法的性能指标采用相对百分增加值(RPI)来衡量,计算公式为  $RPI(c) = (f_c - f_b) / f_b \times 100$ ,式中  $f_b$  是所有参与比较的算法获得的最优目标值,  $f_c$  是给定比较算法获得的目标值。

##### 4.1 算例描述

调研天津某水泥设备制造生产过程,该企业主要生产回转窑、立磨、辊压机等大型复合水泥设备<sup>[5]</sup>。上述生产过程可以建模为带有起重机的柔性作业车间调度问题。为了测试本文算法的性能,参考上述生产过程数据,并随机生成不同规模的 25 个算例,其中工件数量  $n = \{20, 30, 40, 80, 100\}$ ,机床数量  $m = \{6, 7, 8, 9, 10\}$ 。上述算例可以在网址 [www.researchgate.net/publication/334083698\\_CFJSP\\_instances](http://www.researchgate.net/publication/334083698_CFJSP_instances) 下载。

##### 4.2 算法参数

本文所设计算法的三个主要参数包括:蜂群规模  $P_{size}$ 、侦察蜂解无更新最大迭代次数  $T_{limit}$ 、模拟退火算法

所用退火温度参数  $T$ 。参考文献[14],参数水平设定如下:(1)  $P_{size}$  设定为  $\{20, 50, 100, 200\}$  四个层次;(2)  $T_{limit}$  设定为  $\{2, 5, 10, 20\}$  四个层次;(3)  $T$  设定为  $\{0.1, 0.2, 0.4, 0.8\}$  四个层次。算法终止时间  $T_{stop}$  设定为  $30 \times (n/5)$ 。

为提升调参实验的公平性,随机生成了不同规模的 15 个算例,其中工件数量为  $n = \{5, 6, 7, 8, 9, 10\}$ ,机床数量为  $m = \{2, 3, 4\}$ 。在置信度水平 95% 下,采用方差分析法(Analysis of Variance, ANOVA)分析 16 个不同参数组合的算法参数性能表现,从而确定最优算法参数组合,对每个参数组合在 15 个算例下各独立运行 10 次。

通过主效应分析,蜂群规模参数对算法影响较大,且蜂群规模较小时算法可以获得较优性能。 $T$  参数在水平 3 层次上,  $T_{limit}$  参数在水平 2 层次上,算法可以获得较优性能。由上分析可见,  $P_{size} = 20$ ,  $T = 0.4$ ,  $T_{limit} = 5$ , 此时算法可以获得较优性能。

##### 4.3 多算法比较

为验证 IABC 算法的性能,选择两种典型的求解 FJSP 的算法进行比较,即 GA-GSO 算法和 ICA 算法。上述两种算法是求解 FJSP 问题的高性能算法,其中 GA-GSO 算法所研究的问题与本文的 CFJSP 问题比较接近,不同的是前者没有考虑工件间的准备时间约束,也没有计算起重机垂直方向吊运工件产生的能耗。由于需要实现两种对比算法并求解 CFJSP 问题,为验证所实现算法的可信性,采用所实现的 GA-GSO 算法求解文献[5]中的四个算例,并采用 IABC 算法求解上述四个算例,表 1 给出了两种算法结果对比。

由表 1 可见: IABC 算法得到求解四个算例的结果,明显优于 GA-GSO 算法。譬如,GA-GSO 算法求解四个算例获得平均目标值是 48.136,而 IABC 算法得到的平均目标值是 42.694。

为适应本文所研究的问题,我们对 GA-GSO 和 ICA 两种算法进行实现并用于求解“j20m6”-“j40m10”等 25 个不同规模的算例。参考其文献中的局部搜索、全局搜索策略以及参数设置部分。两种算法的编码和解码策略按照本文所给出的策略进行实现。

表 1 文献[5]中四个算例结果比较

算例	GA-GSO			IABC		
	Makespan	能耗	目标值	Makespan	能耗	目标值
1	75.611	492.987	43.406	63.03	434.56	36.995
2	87.823	539.781	49.288	74.14	517.62	43.738
3	80.814	554.132	47.329	74.00	512.89	43.526
4	90.93	602.179	52.52	79.00	549	46.517
mean	83.794	547.27	48.136	72.54	503.518	42.694

表 2 求解中等规模算例权重值分析( $\omega=0.1,0.5,0.9$ )

算例	$\omega=0.1$			$\omega=0.5$			$\omega=0.9$		
	IABC	Gap <sub>GA-GSO</sub>	Gap <sub>ICA</sub>	IABC	Gap <sub>GA-GSO</sub>	Gap <sub>ICA</sub>	IABC	Gap <sub>GA-GSO</sub>	Gap <sub>ICA</sub>
j20m6	1166.85	0.17	1.82	820.83	3.25	3.07	366.89	3.15	6.60
j20m7	1257.78	0.66	0.93	844.37	0.67	2.61	379.68	-0.29	11.58
j20m8	1669.17	-2.06	-3.99	1129.70	2.85	9.69	507.47	3.80	14.63
j20m9	1953.76	-2.07	2.80	1279.37	5.25	13.66	584.79	2.49	22.29
j20m10	2194.60	-2.77	0.85	1462.40	6.04	15.20	675.41	6.04	21.78
j30m6	2181.66	2.71	0.25	1564.77	3.46	0.75	735.93	6.20	8.49
j30m7	2046.88	-3.76	-1.66	1397.75	0.37	6.44	635.77	3.91	11.78
j30m8	2609.98	2.32	0.16	1808.32	9.21	7.52	819.96	11.44	19.32
j30m9	2682.03	-0.53	-2.44	1850.06	4.19	8.65	796.36	13.74	24.33
j30m10	3714.05	0.80	6.30	2564.40	10.03	15.11	1153.18	14.07	19.65
j40m6	2674.68	4.43	1.11	1988.14	2.70	4.10	952.63	8.30	9.71
j40m7	2824.70	4.46	2.16	2056.48	0.89	3.23	919.73	6.78	12.96
j40m8	3527.37	4.46	5.58	2580.32	4.21	5.38	1147.85	10.46	16.19
j40m9	3915.78	-1.37	1.69	2748.94	5.29	8.04	1201.49	16.22	19.29
j40m10	4222.98	2.35	0.26	3022.66	4.27	8.81	1314.71	15.93	17.46
Mean	2576.15	0.65	1.06	1807.90	4.18	7.48	815.90	8.15	15.74

由表 2 可见. (1) 在权重系数  $\omega=0.1$  的情况下, IABC 算法相比 GA-GSO 平均性能提升了 0.65%, 相比 ICA 算法平均性能提升了 1.06%. 相比 GA-GSO 和 ICA 两种算法, IABC 在求解工件数量等于 40 的相对较大规模算例, 取得了明显较好的结果; 另外, 相比 ICA 算法, IABC 在求解所给定的 15 个中小规模算例, 取得了明显较好的结果. (2) 在权重系数  $\omega=0.5$  的情况下, IABC 相比两种算法, 平均性能分别提升了 4.18% 和 7.48%. (3) 在权重系数  $\omega=0.9$  的情况下, IABC 相比 GA-GSO 和 ICA, 平均性能分别提升了 8.15% 和 15.74%.

图 1 给出了三种比较算法在  $\omega=0.9$  情况下的 ANOVA 分析图, 由图中给出的综合性能可见, 本文所

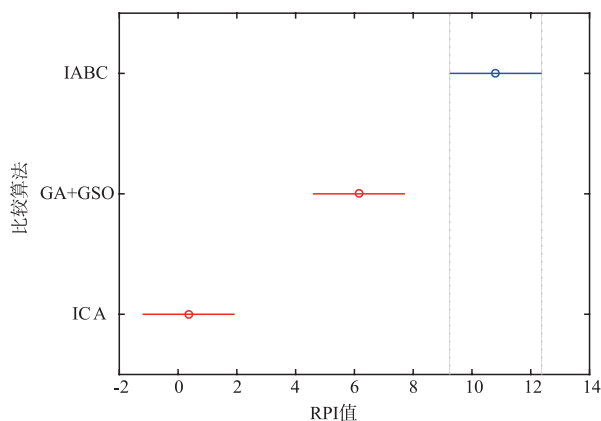


图1 ANOVA性能分析图(多算法比较分析 $\omega=0.9$ )

提出的 IABC 算法相比 GA-GSO 和 ICA 两种算法具有明显的优势.

图 2 给出了三种算法求解 25 个不同规模算例, 在权重系数  $\omega=0.1 \sim 0.9$  情况下, 算法获得的总体目标值(fitness)、最大完工时间(makespan)、能耗(energy)值的三维数据图, 由图可见, IABC 算法无论在总体目标值, 还是在分类目标值中, 都表现了较优的性能.

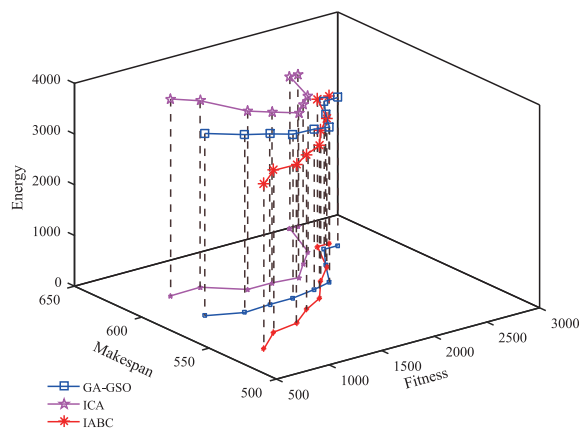


图2 求解25个算例权重值分析图

## 5 结论

针对制造业中存在的一类典型的带有运输资源约束、准备时间约束的柔性作业车间调度问题, 本文首先结合问题特征, 为该类问题建立了数学模型. 其次, 在

人工蜂群算法的框架下,结合问题特点,设计了几类邻域结构,用于提升局部搜索性能.改进了人工蜂群算法中的雇佣蜂、跟随蜂、侦查蜂三类过程,并采用模拟退火策略提升了算法全局搜索能力.在不同规模算例下,验证了所提算法的有效性.未来研究工作主要包括:(1)结合问题特征,进一步改进局部搜索和全局搜索策略,提升算法搜索效率;(2)结合多目标优化算法,包括基于 Pareto 和基于分解机制的多目标优化最新成果,构建一种自适应的多目标优化算法框架;(3)进一步考虑其他实际生产约束,如机床故障,加工时间不确定等,研究不确定条件下的调度优化.

#### 参考文献

- [1] Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2002, 32(1): 1–13.
- [2] 王春, 田娜, 纪志成, 王艳. 求解模糊柔性作业车间调度的多目标进化算法[J]. 电子学报, 2017, 45(12): 2909–2916.  
WANG Chun, TIAN Na, JI Zhi-cheng, WANG Yan. Multi-objective evolutionary algorithm to solve fuzzy flexible job shop scheduling problem[J]. Acta Electronica Sinica, 2017, 45(12): 2909–2916. (in Chinese)
- [3] 雷德明, 杨冬婧. 具有总能耗约束的柔性作业车间调度问题研究[J]. 自动化学报, 2018, 44(11): 2083–2091.  
Lei D M, Yang D J. Research on flexible job shop scheduling problem with total energy consumption constraint[J]. Acta Automatica Sinica, 2018, 44(11): 2083–2091. (in Chinese)
- [4] Nouri H E, Belkahla Driss O, Ghédira K. Simultaneous scheduling of machines and transport robots in flexible job shop environment using hybrid metaheuristics based on clustered holonic multiagent model [J]. Computers & Industrial Engineering, 2016, 102(6): 488–501.
- [5] Liu Z, Guo S, Wang L. Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption [J]. Journal of Cleaner Production, 2019, 211(4): 765–786.
- [6] Karimi S, Ardalan Z, Naderi B, Mohammadi M. Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm [J]. Applied Mathematical Modelling, 2017, 41(1): 667–82.
- [7] Bukata L, cha P, Hanzálek Z. Optimizing energy consumption of robotic cells by a Branch & Bound algorithm [J]. Computers & Operations Research, 2019, 102(2): 52–66.
- [8] Gürel S, Gultekin H, Akhlaghi V E. Energy conscious scheduling of a material handling robot in a manufacturing cell [J]. Robotics and Computer-Integrated Manufacturing, 2019, 58(4): 97–108.
- [9] Kats V, Levner E. On the existence of dominating 6-cyclic schedules in four-machine robotic cells [J]. European Journal of Operational Research, 2018, 268(2): 755–759.
- [10] Liu S Q, Kozan E. A hybrid metaheuristic algorithm to optimise a real-world robotic cell [J]. Computers & Operations Research, 2017, 84(5): 188–194.
- [11] Hurink J, Knust S. Tabu search algorithms for job-shop problems with a single transport robot [J]. European Journal of Operational Research, 2015, 162(1): 99–111.
- [12] Batur G D, Erol S, Karasan O E. Robot move sequence determining and multiple part-type scheduling in hybrid flexible flow shop robotic cells [J]. Computers & Industrial Engineering, 2016, 100(5): 72–87.
- [13] Li J Q, Pan Q K, Duan P Y. An improved artificial bee colony algorithm for solving hybrid flexible flowshop with dynamic operation skipping [J]. IEEE Transactions on Cybernetics, 2016, 46(6): 1311–1324.
- [14] Li J Q, Bai S C, Duan P Y, Sang H Y, Han Y Y, Zheng Z X. An improved artificial bee colony algorithm for addressing distributed flow shop with distance coefficient in a prefabricated system [J]. International Journal of Production Research, 2019, 57(22): 6922–6942.

#### 作者简介



李俊青 男, 1976 年出生, 山东聊城人. 教授、博士生导师, 主要从事智能优化与调度的研究.  
E-mail: lijunqing@lccu-cs.com



杜宇 男, 1992 年出生, 山东泰安人. 博士研究生, 主要从事智能优化与调度的研究.  
E-mail: dyscupse@163.com