

# 改进生物地理学优化算法求解模糊分布式柔性作业 车间调度问题

孙美玲, 顾幸生<sup>†</sup>

(华东理工大学 能源化工过程智能制造教育部重点实验室, 上海 200237)

**摘要:** 经济全球化推动制造企业从单一工厂向多工厂协同模式转变, 模糊分布式柔性作业车间调度问题(FDFJ-SP)成为调度领域的研究热点. 为最小化FDFJSP的最大模糊完工时间, 本文提出了一种基于模拟退火和局部搜索策略的生物地理学优化算法(BBOSL). 根据问题特点, 设计了工厂-随机键的新型编解码方案; 通过调度规则生成半数初始种群以提高种群质量; 提出了基于模拟退火算法的新解接受方法和基于关键工厂的局部搜索策略以增强搜索能力; 通过对算法参数调优提升了算法性能. 实验结果验证了改进策略的有效性, 并与现有算法进行了对比实验, 验证了其在模糊集中式和模糊分布式柔性作业车间调度问题上的优越性.

**关键词:** 生产调度; 模糊分布式; 柔性作业车间; 生物地理学优化算法; 调度规则; 模拟退火

**引用格式:** 孙美玲, 顾幸生. 改进生物地理学优化算法求解模糊分布式柔性作业车间调度问题. 控制理论与应用, 2025, 42(4): 713–721

DOI: 10.7641/CTA.2023.30395

## An improved biogeography-based optimization algorithm for fuzzy distributed flexible job-shop scheduling problem

SUN Mei-ling, GU Xing-sheng<sup>†</sup>

(Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education,  
East China University of Science and Technology, Shanghai 200237, China)

**Abstract:** The globalization of the economy has prompted manufacturing enterprises to transition from a single factory to a multi-factory collaborative model, making the fuzzy distributed flexible job-shop scheduling problem (FDFJSP) a research hotspot in the scheduling field. In this paper, a novel biogeography-based optimization algorithm based on simulated annealing and local search strategy (BBOSL) is proposed to minimize the maximum fuzzy completion time of FDFJSP. Based on the characteristics of the problem, a new factory-random key encoding and decoding scheme is designed. Scheduling rules are used to generate half of the initial population to improve the population quality. A new solution acceptance method based on a simulated annealing algorithm and a local search strategy based on a critical factory are proposed to enhance the search capability. The algorithm parameters are tuned to improve algorithm performance. The experimental results validate the effectiveness of the improved strategy and compare it with the existing algorithms to verify its superiority in fuzzy centralized and fuzzy distributed flexible job-shop scheduling problems.

**Key words:** production scheduling; fuzzy distributed; flexible job-shop; biogeography-based optimization algorithm; scheduling rule; simulated annealing

**Citation:** SUN Meiling, GU Xingsheng. An improved biogeography-based optimization algorithm for fuzzy distributed flexible job-shop scheduling problem. *Control Theory & Applications*, 2025, 42(4): 713–721

## 1 引言

随着经济全球化的兴起, 现代制造业正积极寻求转型升级, 其中分布式制造成为一种新兴的生产模式. 在这种模式下, 产品不再只由单一工厂进行加工生产, 而是分配到多个工厂进行加工. 这种生产模式能够更

有效地分配资源, 显著提高生产效率并降低成本. 由此分布式柔性作业车间调度问题 (distributed flexible job-shop scheduling problem, DFJSP) 受到了越来越多的关注. Li等人<sup>[1]</sup>提出了一种改进的灰狼优化器求解DFJSP, 通过灰狼群体的协同搜索和自适应调整有

收稿日期: 2023–06–07; 录用日期: 2023–12–20.

<sup>†</sup>通信作者. E-mail: xsqu@ecust.edu.cn; Tel.: +86 18964992383.

本文责任编辑: 丛爽.

国家自然科学基金项目(61973120, 62076095)资助.

Supported by the National Natural Science Foundation of China (61973120, 62076095).

效地优化了调度方案. Du等人<sup>[2]</sup>设计了一种新的解码策略处理带有起重机运输的DFJSP, 并提出了一种混合分布估计算法. Xu等人<sup>[3]</sup>考虑了带有运营外包和碳排放的DFJSP, 提出了一种基于混合遗传算法和禁忌搜索的调度方法. Luo等人<sup>[4]</sup>提出了一种基于非支配排序遗传算法II(non-dominated sorting genetic algorithm II, NSGA-II)结构的改进记忆算法求解具有人工安排的DFJSP.

上述文献中, 学者们在求解DFJSP时均假设产品在机器上的加工时间是确定的. 然而, 在实际生产中, 机器故障、工人操作和预防性维护等不确定因素可能导致产品加工时间的波动, 这将严重影响生产进度. 因此, 许多学者考虑了加工时间不确定的因素, 对模糊柔性作业车间调度问题(fuzzy flexible job-shop scheduling problem, FFJSP)进行了深入研究. 李尚函等人<sup>[5]</sup>提出了一种混合超启发式遗传算法, 该算法采用三角模糊数表示工件的模糊加工时间, 并结合了超启发式策略和遗传算法, 有效地提高了算法对全局最优解的搜寻能力. 为解决带有新工作插入约束的FFJSP, Seyyedi等人<sup>[6]</sup>采用NSGA-II元启发式算法构建了模糊线性规划模型. Li等人<sup>[7]</sup>设计了一种基于分解的混合自适应多目标进化算法. 该算法采用多层次编码策略和自适应交叉与变异算子, 增强了算法的搜索能力. 之后, Li等人<sup>[8]</sup>又提出了一种基于强化学习的基于分解机制的多目标进化算法(multi-objective evolutionary algorithm based on decomposition, MOEA/D), 设计了基于Q-learning的参数自适应策略, 提高了了解的质量.

目前, 大多数关于模糊柔性作业车间调度的研究仍集中在集中式调度上, 对分布式调度的研究相对较少. 针对模糊分布式柔性作业车间调度问题(fuzzy DFJSP, FDFJSP), 陈文洲<sup>[9]</sup>提出了一种改进的人工蜂群算法, 设计了基于车间-工序-机器的三层编码方法, 并使用多种混合搜索策略提高了算法的搜索能力. 唐红涛等人<sup>[10]</sup>提出了一种改进的灰狼优化算法, 采用基于非线性函数和动态权重的更新机制来协调算法的全局探索和局部搜索能力, 成功优化了FDFJSP的最大模糊完工时间. Li等人<sup>[11]</sup>研究了加工时间为2型模糊的分布式绿色柔性作业车间调度问题, 并提出了两阶段知识驱动进化算法. Yang等人<sup>[12]</sup>提出了一种混合分解变量邻域模因算法, 该算法通过利用多个进化算子和权重向量实现高效挖掘.

综上所述, 现有的关于柔性作业车间调度的研究主要集中在DFJSP和FFJSP, 而FDFJSP更符合实际生产. 因此, 本文在DFJSP和FFJSP的研究成果的基础上, 考虑了多工厂分布和产品加工时间的不确定性问题, 深入研究了FDFJSP.

传统算法如遗传算法、禁忌搜索算法等在解决FDFJSP时常常受到复杂高维问题、局部最优、收敛速

度慢、参数难调等限制<sup>[13]</sup>. 作为一种新兴算法, 生物地理学优化算法(biogeography-based optimization, BBO)<sup>[14]</sup>因具有原理简单、参数少及易于实现等特点被广泛应用于各个领域<sup>[15-18]</sup>. 尽管BBO具有上述优点, 但在解决FDFJSP等问题时仍存在需要兼顾全局搜索能力和局部搜索能力的挑战, 引入模拟退火算法可以帮助BBO跳出局部最优解, 增强全局搜索能力. 同时, 局部搜索策略可以加快收敛速度并优化解的质量. 因此本文提出了一种基于模拟退火和局部搜索策略的生物地理优化算法(BBO algorithm based on simulated annealing and local search strategy, BBOSL)优化FDFJSP的最大模糊完工时间(makespan).

## 2 模糊分布柔性作业车间调度问题

### 2.1 符号定义

为了便于描述问题, 定义了以下数学符号:

$i$ : 工件索引,  $i \in \{1, 2, \dots, n\}$ ,  $n$ 为工件总数;

$j$ : 工序索引,  $j \in \{1, 2, \dots, q_i\}$ ,  $q_i$ 为工件 $i$ 的工序总数;

$f$ : 工厂索引,  $f \in \{1, 2, \dots, p\}$ ,  $p$ 为工厂总数;

$k$ : 机器索引,  $k \in \{1, 2, \dots, m_f\}$ ,  $m_f$ 为工厂 $f$ 的机器总数;

$r$ : 机器加工阶段索引,  $r \in \{1, 2, \dots, P_{f,k}\}$ ;

$q$ : 所有工件的所有工序数;

$J_i$ : 第 $i$ 个工件;

$U_f$ : 第 $f$ 个工厂;

$O_{ij}$ : 工件 $i$ 的第 $j$ 道工序;

$M_{f,k}$ : 工厂 $f$ 的第 $k$ 台机器;

$P_{f,k}$ : 在机器 $M_{f,k}$ 上加工的工序总数;

$\tilde{t}_{i,j}^{f,k}$ : 工序 $O_{ij}$ 在机器 $M_{f,k}$ 上的模糊加工时间;

$\tilde{S}_{i,j}$ : 工序 $O_{ij}$ 的模糊开始加工时间;

$\tilde{E}_{i,j}$ : 工序 $O_{ij}$ 的模糊结束加工时间;

$\tilde{H}_{f,k,r}$ : 机器 $M_{f,k}$ 上第 $r$ 个阶段的模糊开始加工时间;

$\tilde{C}_i$ : 工件 $i$ 的模糊完工时间;

$\tilde{C}_{\max}$ : 最大模糊完工时间;

$x_i^f$ : 决策变量, 如果工件 $i$ 在工厂 $f$ 中加工则为1, 否则为0;

$y_{i,j}^{f,k}$ : 决策变量, 如果工序 $O_{ij}$ 在机器 $M_{f,k}$ 上加工则为1, 否则为0;

$z_{i,j}^{f,k,r}$ : 决策变量, 如果工序 $O_{ij}$ 在机器 $M_{f,k}$ 上的第 $r$ 个阶段加工则为1, 否则为0.

### 2.2 问题描述

FDFJSP是FFJSP的扩展, 包含多个工厂, 每个工厂都是典型的FFJSP. 该问题可以被表述为: 一组工件 $J = \{J_1, J_2, \dots, J_n\}$ 需要被分配到位于不同地区的工厂 $U = \{U_1, U_2, \dots, U_p\}$ 进行加工, 每个工厂都配备了一组机器 $M = \{M_1, M_2, \dots, M_{m_f}\}$ , 每个工

件 $J_i$ 由多道工序 $O_{ij} = \{O_{i1}, O_{i2}, \dots, O_{iq_i}\}$ 组成. 属于同一工件的工序受到顺序约束, 且每道工序可以在一个或多个机器上加工. 同一工件的所有工序必须在同一个工厂完成加工任务.

### 2.3 调度模型

本文以最小化最大模糊完工时间(makespan)为目标, 考虑以下假设:

- 1) 在零时刻所有工厂中的机器均可用, 所有工件都可以被加工;
- 2) 每道工序在不同机器上的模糊加工时间不同;
- 3) 所有工序的模糊加工时间已知;
- 4) 工件在工厂内加工时不能在工厂之间转移;
- 5) 已经开始加工的工序直到完成前不能中断;
- 6) 不考虑准备时间、释放时间、运输时间等.

基于以上符号定义和假设, 最小化模糊makespan的FDFJSP数学模型<sup>[10]</sup>如下:

$$\min \tilde{C}_{\max} = \min (\max_{i=1}^n \{\tilde{C}_i\}), \quad (1)$$

$$\tilde{C}_i = \tilde{E}_{i,q_i}, \quad (2)$$

$$\tilde{E}_{i,j} = \tilde{S}_{i,j} + \sum_{j=1}^{q_i} \sum_{k=1}^n x_i^f \times y_{i,j}^{f,k} \times \tilde{t}_{i,j}^{f,k}, \quad (3)$$

$$\tilde{S}_{i,j+1} \geq \tilde{E}_{i,j}, \quad (4)$$

$$\tilde{H}_{f,k,r+1} \geq \tilde{H}_{f,k,r} + \sum_{i=1}^n \sum_{j=1}^{q_i} \tilde{t}_{i,j}^{f,k} \times z_{i,j}^{f,k,r}, \quad (5)$$

$$\sum_{j=1}^{q_i} \sum_{k=1}^{m_f} y_{i,j}^{f,k} \in \{0, q_i\}, \quad (6)$$

$$\sum_{f=1}^p \sum_{k=1}^{m_f} y_{i,j}^{f,k} = 1, \quad (7)$$

$$x_i^f \in \{0, 1\}, y_{i,j}^{f,k} \in \{0, 1\}, z_{i,j}^{f,k,r} \in \{0, 1\}, \quad (8)$$

式(1)表示目标函数, 最大模糊完工时间是所有工件模糊完工时间的最大值; 式(2)定义了工件的模糊完工时间为其最后一道工序的模糊完工时间; 式(3)表示工件的模糊完工时间等于其模糊开始加工时间加上实际模糊加工时间; 式(4)是工序的约束条件, 确保属于同一工件的工序按顺序进行加工; 式(5)表示机器的约束条件, 确保同一台机器一次只能加工一道工序; 式(6)要求属于同一工件的所有工序必须在同一工厂加工; 式(7)是确保每道工序只能在一个工厂内选择一台机器进行加工; 式(8)定义了决策变量的取值范围.

### 2.4 三角模糊数的运算规则

本文采用三角模糊数 $\tilde{t}_{i,j}^{f,k} = (t_1, t_2, t_3)$ 表示加工时间的不确定性, 其中:  $t_1$ 是最乐观值即最优加工时间,  $t_2$ 是最可能值即最可能加工时间,  $t_3$ 是最悲观值即最差加工时间. 处理三角模糊数涉及的基本操作包括加法、近似取大和比较操作. 已知两个三角模糊数:  $\tilde{X} = (x_1, x_2, x_3), \tilde{Y} = (y_1, y_2, y_3)$ .

1) 加法运算.

$$\tilde{X} + \tilde{Y} = (x_1 + y_1, x_2 + y_2, x_3 + y_3). \quad (9)$$

2) 近似取大运算.

设 $\tilde{X} = (x_1, x_2, x_3)$ 为工件 $i$ 第 $j$ 道工序的模糊完工时间,  $\tilde{Y} = (y_1, y_2, y_3)$ 为机器 $k$ 第 $r$ 个阶段的模糊完工时间. Sakawa准则<sup>[19]</sup>和Lei准则<sup>[20]</sup>分别被表示为式(10)和式(11):

$$\tilde{X} \vee \tilde{Y} \cong (x_1 \vee y_1, x_2 \vee y_2, x_3 \vee y_3), \quad (10)$$

$$\tilde{X} \vee \tilde{Y} = \begin{cases} \tilde{X}, & \tilde{X} > \tilde{Y}, \\ \tilde{Y}, & \text{其他}. \end{cases} \quad (11)$$

图1是Sakawa准则的近似取大值、 $\tilde{X}$ 和 $\tilde{Y}$ 实际取大值、Lei准则的近似取大值. 其中, 灰色填充部分为最后的取大值结果, 可以看出Lei准则的近似值相较于Sakawa准则更接近实际值. 因此, 本文选择采用Lei准则来确定模糊开始加工时间.

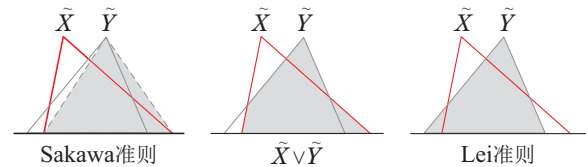


图1 实际取大值与两个准则近似取大值的比较  
Fig. 1 Comparison between real max and approximate max by two criterion

3) 比较运算.

基于三角模糊数的加法运算可以计算出工件的模糊完工时间 $\tilde{C} = (c_1, c_2, c_3)$ , 其隶属函数为

$$\mu_C(x) = \begin{cases} \frac{x - c_1}{c_2 - c_1}, & c_1 < x \leq c_2, \\ 0, & x \leq c_1 \text{ 或 } x > c_3, \\ \frac{c_3 - x}{c_3 - c_2}, & c_2 < x \leq c_3. \end{cases} \quad (12)$$

根据文献[21],  $\tilde{C} = (c_1, c_2, c_3)$ 的左积分和右积分分别被定义为

$$I_L(\tilde{C}) = c_2 - \int_{c_1}^{c_2} \mu_C(x) dx = \frac{c_1 + c_2}{2}, \quad (13)$$

$$I_R(\tilde{C}) = c_2 + \int_{c_2}^{c_3} \mu_C(x) dx = \frac{c_2 + c_3}{2}, \quad (14)$$

总积分为

$$I_\beta(\tilde{C}) = \beta I_L(\tilde{C}) + (1 - \beta) I_R(\tilde{C}), \quad (15)$$

其中乐观系数 $\beta \in [0, 1]$ 由调度决策者根据当前生产情况进行调整,  $\beta$ 值越高表示决策者持越积极的态度. 如果 $I_\beta(\tilde{C}_1) < I_\beta(\tilde{C}_2)$ , 则 $\tilde{C}_1$ 优于 $\tilde{C}_2$ .

## 3 基于模拟退火和局部搜索的生物地理学优化算法

### 3.1 编码解码

针对FDFJSP, 本文设计了一种基于工厂-随机键

的新型编解码方法,该方法包含工厂分配向量(factory allocation vector, FAV)和随机键向量(random key vector, RKV). 其中: FAV的向量长度等于工件数, RKV的向量长度等于所有工件的工序总数. 以表1数据为例,可以得到可行解的编码如图2所示.

表 1 FDFJSP示例

Table 1 The example FDFJSP instance

工件 工序	工厂1			工厂2			
	$M_{11}$	$M_{12}$	$M_{13}$	$M_{21}$	$M_{22}$	$M_{23}$	
$J_1$	$O_{11}$	(2,4,5)	(1,3,4)	(3,5,7)	(6,8,10)	—	(3,5,7)
	$O_{12}$	(6,8,10)	(2,4,5)	—	(4,7,11)	(5,8,13)	(6,8,10)
	$O_{21}$	(3,5,7)	(2,4,5)	(4,7,11)	(3,5,7)	(4,7,11)	—
$J_2$	$O_{22}$	(2,4,5)	—	(5,8,13)	(5,8,13)	(1,3,4)	(2,4,5)
	$O_{23}$	(4,7,11)	(1,3,4)	(2,4,5)	—	(6,8,10)	(5,8,13)
$J_3$	$O_{31}$	(2,4,5)	—	(6,8,10)	(6,8,10)	(4,7,11)	(3,5,7)
	$O_{32}$	(6,8,10)	(1,3,4)	(4,7,11)	(1,3,4)	(4,7,11)	—

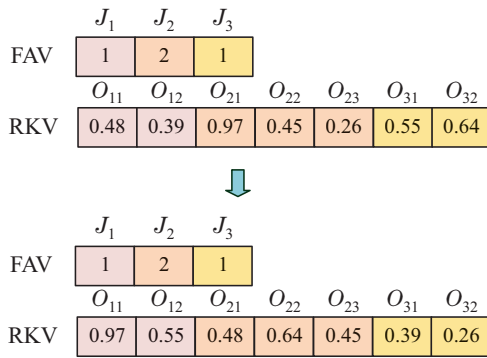


图 2 编码表示

Fig. 2 Encoding representation

FAV为1时表示工件 $J_1$ 被分配到工厂1. RKV是[0, 1]的随机数, 随机数越大表示对应工序的优先级越高. 首先, 比较每个工件第1道工序的优先级, 即比较 $O_{11}, O_{21}, O_{31}$ 随机键值的大小. 例如 $0.97 > 0.55 > 0.48$ , 所以先加工 $O_{21}$ . 然后继续比较 $O_{11}, O_{22}, O_{31}$ 随机键值的大小, 以此类推. 由于 $J_1$ 和 $J_3$ 在工厂1加工,  $J_2$ 在工厂2加工, 可以确定工厂1的加工顺序为 $O_{31} \rightarrow O_{11} \rightarrow O_{32} \rightarrow O_{12}$ , 工厂2的加工顺序为 $O_{21} \rightarrow O_{22} \rightarrow O_{23}$ . 机器分配由规则解码得到. 对FAV和RKV解码得到工序顺序后, 为每道工序选择工作负载小的机器. 这种编解码方式可以有效避免非法解, 节省检查非法解的时间.

3.2 种群初始化

为保证种群多样性并平衡工厂负载, 设计了一种基于规则的初始化方法. 50% FAV 随机生成, 剩余50%根据规则生成; RKV全部随机生成. 以下是规则生成FAV的具体步骤:

**步骤 1** 对RKV中每个工件的第1道工序对应随机键值进行降序排序, 确定工件的加工顺序;

**步骤 2** 计算每个工件在所有工厂中的平均模糊加工时间;

**步骤 3** 根据加工顺序和平均模糊加工时间, 依次为每个工件选择加工时间最短的工厂. 如果有多个工厂符合条件, 则随机选择其中一个.

3.3 迁移操作

基于余弦迁移模型<sup>[22]</sup>, 迁入率和迁出率的计算公式如式(16)–(17), 其中:  $N - k$ 为栖息地 $k$ 的物种数量,  $N - 1$ 是栖息地适宜性指数 (habitat suitability index, HSI)最高的栖息地物种最大容量,  $I$ 和 $E$ 分别表示最大迁入率和最大迁出率.

$$\lambda_k = \frac{I}{2} \times (1 + \cos(\frac{N - k}{N - 1} \pi)), \quad (16)$$

$$\mu_k = \frac{E}{2} \times (1 - \cos(\frac{N - k}{N - 1} \pi)). \quad (17)$$

针对工厂分配向量FAV和随机键向量RKV, 分别采用0–1均匀交叉和IPOX进行迁移. FAV的迁移过程如图3所示. 生成0–1均匀向量, 长度与工件数相同, 迁入栖息地 $H_i$ 中与0对应的个体保持不变, 将迁出栖息地 $H_j$ 中与1对应的个体迁移到 $H_i$ , 得到新栖息地 $H_k$ .

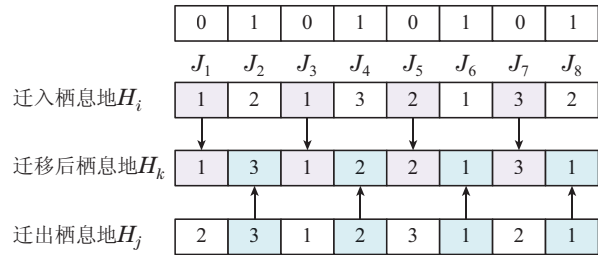


图 3 FAV的迁移操作

Fig. 3 Migration operations on FAV

RKV的迁移过程如图4所示. 将工件随机分为Job1和Job2两部分,  $H_i$ 中属于Job1的工序的随机键保持不变,  $H_j$ 中属于Job2的工序的随机键值迁移到 $H_i$ , 由此得到一组新的RKV.

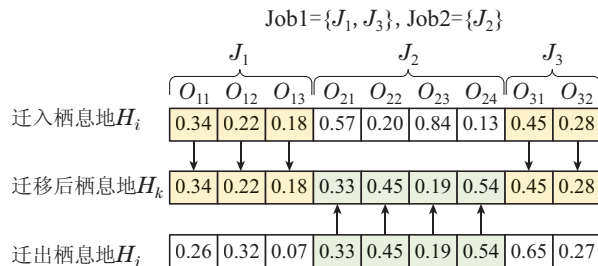


图 4 RKV的迁移操作

Fig. 4 Migration operations on RKV

为增强算法的全局搜索能力和扩大解空间, 在迁移操作中引入了一种更灵活且全局性更强的搜索策略, 即模拟退火算法, 并提出了一种基于模拟退火算法(simulated annealing, SA)的新解接受方法, 用于更有效地评估和接受生成的新解. 与SA算法的原温度

设定不同,本文的退火温度是基于三角模糊数设置的.定义初始温度 $T_0$ 为初始种群的HSI均值, $T_1$ 为迁移后解的HSI值, $T_2$ 为模拟退火产生新解的HSI值.公式定义如下:

$$T_0 = \frac{1}{N} \sum_{i=1}^N I_{\beta}(\tilde{C}_0) = \frac{1}{N} \sum_{i=1}^N (\beta I_L(\tilde{C}_0) + (1 - \beta) I_R(\tilde{C}_0)), \quad (18)$$

$$T_1 = I_{\beta}(\tilde{C}_1) = \beta I_L(\tilde{C}_1) + (1 - \beta) I_R(\tilde{C}_1), \quad (19)$$

$$T_2 = I_{\beta}(\tilde{C}_2) = \beta I_L(\tilde{C}_2) + (1 - \beta) I_R(\tilde{C}_2), \quad (20)$$

式中: $N$ 为种群大小即栖息地大小, $\beta$ 为乐观因子, $\tilde{C}_0, \tilde{C}_1, \tilde{C}_2$ 分别表示初始栖息地、迁移后栖息地和模拟退火后栖息地的HSI值即对应调度方案的模糊makespan.

### 3.4 基于关键工厂的局部搜索策略

BBO算法具有很强的全局搜索能力,但在搜索过程中可能陷入局部最优解.为了充分利用问题的局部信息并提高算法的收敛速度和解的质量,提出了基于关键工厂的局部搜索策略.关键工厂是整个调度方案中模糊完工时间最长的工厂,通过减小关键工厂的模糊完工时间可以提高解的质量.此外,根据BBOSL的迁移操作和迁移公式可知,具有最高HSI的栖息地的迁入率为0,该栖息地不会进行迁移操作,因此,将每次迭代后HSI最高的解设置为精英解.基于此,设计了一种基于关键工厂的局部搜索策略.

### 3.5 BBOSL算法整体框架

本文提出了一种用于求解FDFJSP的基于模拟退火和局部搜索策略的生物地理学优化算法,记为BBOSL.在规则初始化后,该算法采用迁移算子进行探索,并结合SA以一定概率接受劣解,有效避免算法陷入局部最优.每次迭代结束,在精英解中进行基于关键工厂的局部搜索操作,增加了算法的局部搜索能力.BBOSL算法的具体流程如图5所示.

## 4 实验分析

本节进行实验以分析BBOSL算法在求解FFJSP和FDFJSP时的性能.首先,分析算法的复杂度;然后,介绍实验1和实验2的相关算例;对算法相关参数进行调优;将BBOSL算法与现有算法进行对比,验证其在求解模糊调度问题时的性能.此外,将不含基于SA的新解接受方法的算法命名为BBOSL1,不含基于关键工厂的局部搜索策略的算法命名为BBOSL2,用于验证基于SA的新解接受方法和局部搜索策略的有效性.实验的运行环境为: Intel Core i5-1240P CPU 1.70 GHz, 16GB RAM, Windows 10, 64 位操作系统, 编程语言为C++.

### 4.1 算法复杂度分析

BBOSL算法的计算时间主要分为3个部分:迁移

操作、目标函数和局部搜索.迁移操作包括两个阶段:假设每个栖息地的所有物种都完成了迁移,则迁移操作的复杂度为 $O(N \times q^2)$ ;基于SA的新解接受方法的时间复杂度为 $O(N \times q)$ .目标函数的时间复杂度为 $O(N \times q^2)$ .局部搜索的时间复杂度为 $O(q^2)$ .因此,在每次迭代中BBOSL算法的时间复杂度为 $O(N \times q^2)$ .

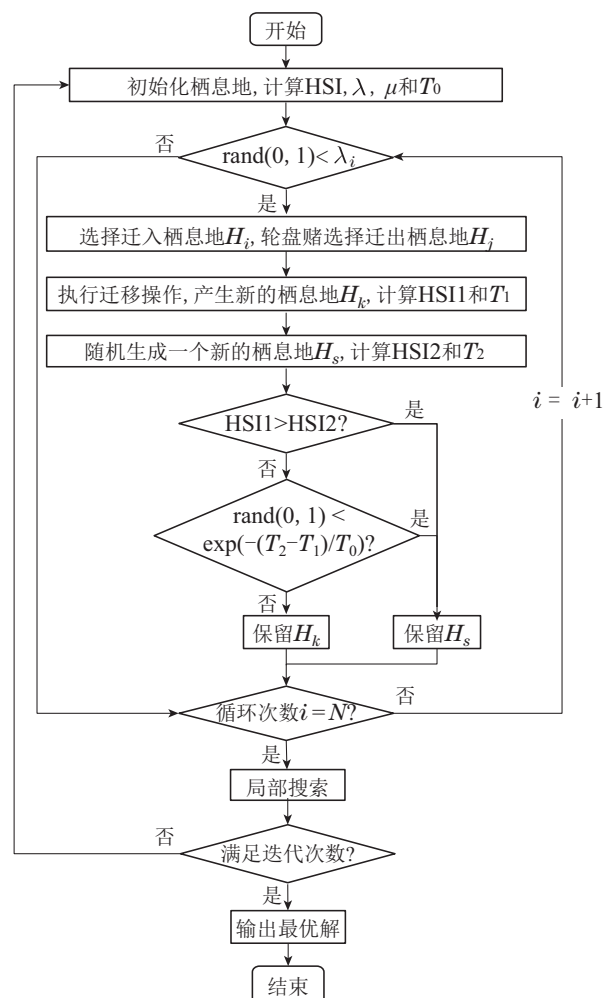


图5 BBOSL算法流程图

Fig. 5 Flow chart of the BBOSL algorithm

### 4.2 算例介绍

实验1的数据选自LEI<sup>[20]</sup>的4个单工厂模糊算例 instance1–instance4, 实验与3种现有算法对比: 分解集成遗传算法(decomposition integrated genetic algorithm, DIGA)<sup>[20]</sup>、改进离散粒子群优化算法(improved discrete particle swarm optimization, IDPSO)<sup>[23]</sup>和混合协同进化算法(hybrid cooperative evolutionary algorithm, hCEA)<sup>[24]</sup>.

实验2的数据是一组由唐红涛等人<sup>[10]</sup>基于FFJSP标准测试集生成的8个多工厂算例.实验将BBOSL与4种算法对比: 遗传算法(GA-JS)<sup>[25]</sup>、离散粒子群算法(discrete particle swarm algorithm, DPSO)<sup>[26]</sup>、灰狼算法(grey wolf algorithm, GWO)<sup>[27]</sup>、改进灰狼算法(improved GWO, IGWO)<sup>[10]</sup>.



### 4.3 参数设置

本文提出的BBOSL算法的主要参数包括种群规模( $N$ )、迁入率( $\lambda$ )、迁出率( $\mu$ )、迭代次数( $t$ )和乐观系数( $\beta$ )。其中: 迭代次数 $t$ 设置为300, 迁入率 $\lambda$ 和迁出率 $\mu$ 均为1<sup>[17]</sup>。为测试不同种群规模及不同乐观系数

对算法性能的影响, 在实验2的8个算例中进行实验, 种群规模( $N$ )分别设置为50, 100, 150, 200, 250和300, 乐观系数 $\beta$ 设置为0, 0.3, 0.5, 0.7, 1。每个算例独立运行30次, 记录平均最大模糊完工时间, 实验结果记录于表2和表3, 其中粗体数值表示最优解。

表2 不同种群规模下的平均模糊完工时间  
Table 2 Average fuzzy makespan for different population sizes

算例	$N = 50$	$N = 100$	$N = 150$	$N = 200$	$N = 250$	$N = 300$
R1-2	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>
R1-3	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>
R2-2	(20.5, 35.8, 50.9)	(21.0, 35.0, 49.6)	(20.9, 34.7, 49.1)	<b>(20.3, 34.3, 48.4)</b>	(20.2, 34.6, 48.9)	(21, 34.3, 48.7)
R2-3	(19.2, 33.4, 48.2)	(18.3, 33.1, 48.7)	(18.2, 33, 48.8)	<b>(18.0, 33, 49)</b>	<b>(18.0, 33, 49)</b>	<b>(18.0, 33, 49)</b>
R3-2	(31.1, 51.8, 70.6)	(29.4, 48.5, 65.8)	(27.7, 46.4, 62.8)	<b>(27.2, 44.7, 61.4)</b>	(28.1, 45.9, 63.6)	(29, 47.1, 62.6)
R3-3	(25.7, 42.6, 58.5)	(25.1, 41.7, 56.8)	(25.5, 41, 55.9)	<b>(23.6, 39.5, 54.1)</b>	(23.4, 40.8, 56.3)	(25.4, 40.6, 55.2)
R4-2	(28.1, 45.5, 64.1)	(26.2, 44.1, 62.4)	(26.0, 43.4, 61.7)	<b>(25.6, 42.1, 60.1)</b>	(25.7, 42.7, 60.6)	(26, 42.9, 60.8)
R4-3	(24.6, 41.7, 59.5)	(24.4, 41.3, 59.8)	(24.1, 41.2, 59.8)	<b>(24, 41, 60)</b>	(24.1, 41.1, 59.9)	(24.1, 41.3, 59.7)

表3 不同乐观系数下的平均模糊完工时间  
Table 3 Average fuzzy makespan for different optimism coefficients

算例	$\beta = 0$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.7$	$\beta = 1$
R1-2	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	(9.1, 23.9, 32)	(9.4, 23.9, 32.5)
R1-3	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	(10.5, 23, 32)	(10, 23.5, 32.4)
R2-2	(23.5, 34.8, 46.6)	(21.7, 34.6, 47.1)	<b>(20.3, 34.3, 48.4)</b>	(18.1, 34.7, 50.4)	(17.3, 35.3, 51.8)
R2-3	(22.3, 33.3, 45.3)	(21.4, 33.4, 45.3)	<b>(18, 33, 49)</b>	(17.2, 33.4, 49.3)	(16.7, 33.9, 49.6)
R3-2	(29.6, 45.9, 60.7)	(29.4, 45.3, 60.5)	<b>(27.2, 44.7, 61.4)</b>	(26.3, 45.3, 62.4)	(25.5, 46.7, 64.7)
R3-3	(26.3, 40.5, 53.9)	(24.6, 40.5, 55.2)	<b>(23.6, 39.5, 54.1)</b>	(22.6, 40.3, 56)	(21, 41.4, 58.6)
R4-2	(29.3, 42.6, 56.7)	(26.2, 42.5, 58.7)	<b>(25.6, 42.1, 60.1)</b>	(21.7, 42.7, 62.7)	(19.9, 43.6, 65.9)
R4-3	(28.4, 42, 55.3)	(27.7, 41.5, 56.4)	<b>(24, 41, 60)</b>	(20.6, 41.8, 62.5)	(18.8, 42.9, 64.8)

根据表2数据, 可以观察到在R1-2和R1-3小规模算例中, 种群规模 $N$ 对结果的影响较小, 平均最大模糊makespan相同。在R2-3中, 由于工厂数量相对较多, 工件可以被有效分配, 得到最优的调度方案。在大规模算例中, 不同的种群规模对模糊makeapan的数值表现出明显差异, 当种群大小为200时, 模糊makespan最优, 因此, 将种群规模设置为200。从表3得知, 当 $\beta = 0.5$ 时, 各算例的平均最大模糊makespan最优, 因此, 设置 $\beta = 0.5$ 。

### 4.4 结果分析

#### 4.4.1 实验1分析

实验1旨在验证BBOSL算法求解FFJSP的性能, 表4为4种算法的实验结果。每个算例运行30次, 记录最大模糊完工时间的最优解、平均解、最差解及30次运行的平均CPU时间( $T$ )于表4, 粗体表示最优结果。

从表4的实验结果看, BBOSL算法的最优解、平均解及最差解均优于DIGA, IDPSO和hCEA算法, 运行时间也基本优于其他算法。这表明BBOSL算法在求解FFJSP时具有很好的竞争力。采用95%置信区间下的Friedman检验来验证各算法之间是否存在显著差

异, 表5显示了实验1的4种算法的平均秩和 $p$ 值。统计结果表明, BBOSL算法的平均秩为1,  $p$ 值为 $7.35E-07$ 远远小于0.05, 表明BBOSL算法与其他3种算法相比具有显著性优势。

#### 4.4.2 实验2分析

实验2通过8个算例验证BBOSL算法改进策略的有效性, 并检验算法在模糊分布式柔性作业车间调度问题下的表现能力, 每个算例运行30次, 实验结果列于表6。

通过对比表6中BBOSL1和BBOSL的结果, 可以看出在小规模算例中, BBOSL获得与BBOSL1相同的最优解、平均解和最差解; 在相对较大规模的R2-3算例中, BBOSL得到与BBOSL1相同的最优解, 并且在平均解和最差解上表现更好; 在R2-2中, BBOSL的最优解、平均解和最差解都优于BBOSL1; 在除R4-3之外的大规模算例中, BBOSL的最优解均优于BBOSL-1。这表明BBOSL能明显改善平均解和最差解, 进一步证实了模拟退火策略的有效性。通过对比BBOSL-2和BBOSL的数据发现, 在小规模算例中, BBOSL是一种优秀的改进方法, 能够显著改善平均解和最差解

的质量. 在R2-3和R4-3中, 由于工厂和机器相对较多, 工序可被合理分配得到最优解决方案. 在其他大规模算例中, BBOSL在最优解、平均解和最差解方面都优于BBOSL2, 验证了基于关键工厂的局部搜索策略的有效性.

表 4 FFJSP 4种算法的比较结果  
Table 4 Comparison results of four algorithms for FFJSP

算例	算法	最优解	平均解	最差解	运行时间/s
1	DIGA	(20, 31, 40)	(22.8, 33.4, 44.6)	(25, 37, 49)	9.91
	IDPSO	(19, 28, 40)	(20.6, 29.9, 40.2)	(21, 31, 41)	2.93
	hCEA	(19, 31, 43)	(21.6, 30.7, 42.1)	(24, 31, 40)	2.8
	BBOSL	<b>(17, 28, 41)</b>	<b>(20.3, 29.2, 39.5)</b>	<b>(21, 31, 39)</b>	<b>2.27</b>
2	DIGA	(33, 48, 57)	(35.4, 48.4, 62.3)	(37, 50, 65)	9.96
	IDPSO	(30, 46, 58)	(32.5, 46.2, 57.5)	(35, 47, 57)	<b>2.9</b>
	hCEA	(43, 60, 77)	(46.3, 65.2, 84.7)	(49, 71, 92)	3.14
	BBOSL	<b>(30, 45, 58)</b>	<b>(31.9, 45.3, 57.4)</b>	<b>(31, 46, 58)</b>	2.93
3	DIGA	(37, 49, 64)	(37.3, 53.0, 66.9)	(41, 58, 75)	13.58
	IDPSO	(33, 45, 60)	(32.5, 46.2, 60.9)	(35, 49, 64)	3.74
	hCEA	(31, 48, 65)	(33.7, 47.3, 62.5)	(37, 48, 60)	3.92
	BBOSL	<b>(29, 45, 59)</b>	<b>(31.0, 45.2, 59.8)</b>	<b>(31, 46, 61)</b>	<b>3.57</b>
4	DIGA	(29, 41, 56)	(29.2, 42.9, 57.5)	(29, 46, 60)	13.21
	IDPSO	(23, 35, 49)	(24.9, 36.7, 51.0)	(27, 38, 51)	3.6
	hCEA	(28, 36, 50)	(26.6, 38.2, 52.5)	(24, 40, 56)	3.79
	BBOSL	<b>(23, 35, 48)</b>	<b>(22.3, 35.8, 49.2)</b>	<b>(25, 35, 49)</b>	<b>3.2</b>

表 5 实验1的Friedman检验结果  
Table 5 Friedman test results for experiment 1

算法	平均秩	p值
DIGA	3.67	7.35E-07
IDPSO	2.08	
hCEA	3.25	
BBOSL	1	

图6为R3-3中各算法的收敛曲线. BBOSL相比于BBOSL1和BBOSL2具有更优的初始值, 在约165代时搜寻到最优解, 表明将基于模拟退火的新解接受方法

与基于关键工厂的局部搜索策略相结合, 可以加速对最优解的搜索. BBOSL在约120代时找到较优解后仍继续搜索最终获得更好的解, 说明基于模拟退火的新解接受方法有助于BBOSL跳出局部最优扩展解空间. 此外, 未进行局部搜索的BBOSL2算法得到的解是最差的, 明显不如BBOSL, 验证了所提出的局部搜索策略在增强BBOSL的局部搜索能力方面的有效性.

根据表7中5种算法求解FDFJSP的实验对比结果, 可以看出BBOSL算法在求解大规模模糊问题时表现出色, 其最优解、平均解和最差解明显优于其他4个算法. 特别在R4-3中, BBOSL直接搜索到了(24, 41, 60). 在小规模算例中, 该算法的优势不太明显, 这是因为其局部搜索策略与工序数量密切相关, 工序少时搜索能力受限. 在R1-3的小规模算例中, BBOSL搜索到了与IGWO相同的最优解、平均解和最差解, 展现了良好的稳定性. 此外, 当工序数量相同时, BBOSL在3个工厂中的最大模糊makespan明显小于2个工厂的. 在R3-2和R3-3中, 随着工厂数量的增加, 最大模糊makespan的最优解从(22, 43, 61)减小到(25, 39, 50), 表明生产任务分配更加合理, 提高了生产效率.

本文的模糊makespan采用三角模糊数表示, 通过式(15)进行去模糊处理, 映射为一个均值. 以R4-3为例, 对5个算法的最大模糊makespan进行去模糊处理, 绘制它们的收敛曲线见图7. 从图中明显可见, BBOSL的迭代初始值明显小于其他4个算法, 说明混合初始化策略有效地提升了初始解的质量. BBOSL的迭代拐点出现在其他算法之前, 表明该算法具有强大的搜索能力, 能够快速寻找到最优解. 此外, BBOSL的迭代收敛曲线位于所有曲线的最低位置, 且模糊makespan最优, 表明了该算法在求解FDFJSP方面具有出色的性能.

利用Friedman检验对实验2中的5个算法进行显著性差异分析, 置信区间设为95%. 表8中的统计分析结果显示BBOSL算法具有最小的平均秩, 且p值为1.54E-18, 远远小于显著性水平0.05, 表明该算法与其他4个算法相比具有显著优越性.

表 6 验证改进策略有效性的对比结果  
Table 6 Comparative results to verify the effectiveness of improved strategies

算例	BBOSL1				BBOSL2				BBOSL			
	最优解	平均解	最差解	T/s	最优解	平均解	最差解	T/s	最优解	平均解	最差解	T/s
R1-2	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	2.4	(15,22,33)	(20.3,28.8,38.3)	(23,31,42)	3	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	3.6
R1-3	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	3.6	<b>(15, 22, 29)</b>	(18, 26, 34.4)	(20,28,38)	3.3	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	3.6
R2-2	(21, 34, 47)	(20.8, 35.1, 49.9)	(26, 36, 51)	10.8	(17, 35, 48)	(20.4,34.9,48.6)	(23, 35, 50)	8.1	<b>(22, 33, 46)</b>	<b>(20.9, 34.5, 48)</b>	<b>(22, 35, 49)</b>	13.5
R2-3	<b>(18, 33, 49)</b>	<b>(18.3, 33, 48.8)</b>	<b>(22, 33, 46)</b>	9	<b>(18, 33, 49)</b>	<b>(18, 33, 49)</b>	<b>(18, 33, 49)</b>	8.7	<b>(18, 33, 49)</b>	<b>(18, 33, 49)</b>	<b>(18, 33, 49)</b>	10.2
R3-2	(25, 43, 59)	(27.4, 46, 62.7)	(30, 47, 69)	12.6	(29, 43, 60)	(28.6, 46.6, 63.6)	(34, 48, 65)	10.5	<b>(22, 43, 61)</b>	<b>(27.2, 44.7, 61.4)</b>	<b>(32, 45, 61)</b>	15.6
R3-3	(26, 40, 53)	(23.3, 40.3, 55.5)	(22, 44, 61)	13.2	(21, 39, 55)	(23.4, 39.7, 55.2)	(22, 41, 58)	10.8	<b>(25, 39, 50)</b>	<b>(23.6, 39.5, 54.1)</b>	<b>(26, 39, 50)</b>	13.5
R4-2	(24, 41, 61)	(26.6, 43.3, 61.6)	(26, 45, 65)	25.2	(25, 42, 59)	(25.8, 42.9, 60.8)	(26, 46, 63)	14.4	<b>(24, 41, 60)</b>	<b>(25.6, 42.1, 60.1)</b>	<b>(25, 44, 64)</b>	27.3
R4-3	<b>(24, 41, 60)</b>	(24, 41.3, 60)	(25, 42, 61)	21.6	<b>(24, 41, 60)</b>	<b>(24, 41, 60)</b>	<b>(24, 41, 60)</b>	14.4	<b>(24, 41, 60)</b>	<b>(24, 41, 60)</b>	<b>(24, 41, 60)</b>	21

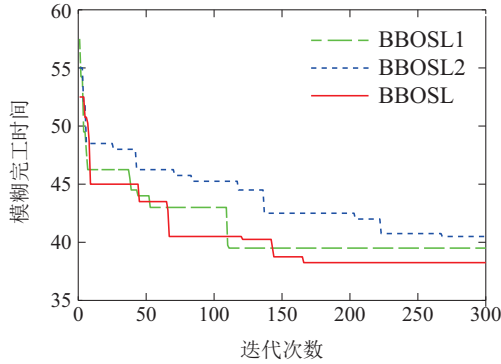


图6 R3-3中BBOSL1, BBOSL2和BBOSL的收敛曲线  
Fig.6 Convergence curves of BBOSL1, BBOSL2 and BBOSL in R3-3

表7 FDFJSP 5种算法的比较结果

Table 7 Comparison results of five algorithms for FD-FJSP

算例	算法	最优解	平均解	最差解
R1-2	GA-JS	(16, 21, 30)	(16.6, 24.3, 32.9)	(21, 27, 36)
	DPSO	(19, 28, 38)	(18.1, 29.8, 44.3)	(18, 30, 45)
	GWO	(17, 25, 34)	(19.1, 27.5, 36.2)	(21, 29, 39)
	IGWO	<b>(16, 21, 30)</b>	<b>(15.2, 21.8, 29.2)</b>	<b>(15, 22, 29)</b>
	BBOSL	(15, 22, 29)	(15, 22, 29)	<b>(15, 22, 29)</b>
R1-3	GA-JS	(15, 22, 29)	(15.4, 22.6, 30.2)	(17, 26, 35)
	DPSO	(15, 23, 31)	(16.5, 25.5, 34.1)	(17, 26, 37)
	GWO	(18, 24, 31)	(18.2, 24.9, 32.7)	(19, 26, 34)
	IGWO	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>
	BBOSL	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>	<b>(15, 22, 29)</b>
R2-2	GA-JS	(23, 41, 61)	(29.9, 45.6, 62.9)	(37, 51, 63)
	DPSO	(41, 60, 84)	(44.3, 65.4, 89.7)	(43, 70, 95)
	GWO	(41, 57, 79)	(42.3, 61.7, 82.5)	(42, 64, 91)
	IGWO	(24, 36, 47)	(23.3, 37.5, 51.7)	(26, 39, 56)
	BBOSL	<b>(22, 33, 46)</b>	<b>(20.3, 34.3, 48.4)</b>	<b>(22, 35, 49)</b>
R2-3	GA-JS	(25, 39, 53)	(27.0, 42.0, 58.1)	(29, 45, 63)
	DPSO	(36, 57, 74)	(39.7, 58.1, 79.6)	(42, 62, 84)
	GWO	(35, 54, 74)	(37.7, 56.4, 75.9)	(36, 59, 81)
	IGWO	(22, 33, 46)	(21.4, 34.6, 47.7)	(23, 38, 50)
	BBOSL	<b>(18, 33, 49)</b>	<b>(18, 33, 49)</b>	<b>(18, 33, 49)</b>
R3-2	GA-JS	(39, 51, 65)	(39.2, 57.3, 76.3)	(47, 65, 85)
	DPSO	(46, 70, 97)	(52.7, 74.1, 98.7)	(53, 76, 105)
	GWO	(51, 73, 95)	(55.3, 79.0, 102.5)	(63, 84, 110)
	IGWO	(25, 44, 60)	(30.2, 44.4, 60.0)	(31, 45, 62)
	BBOSL	<b>(22, 43, 61)</b>	<b>(27.2, 44.7, 61.4)</b>	<b>(32, 45, 61)</b>
R3-3	GA-JS	(27, 44, 63)	(33.0, 48.6, 65.9)	(38, 55, 67)
	DPSO	(46, 66, 93)	(52.1, 72.9, 96.2)	(54, 77, 101)
	GWO	(42, 64, 85)	(48.4, 71.4, 93.6)	(49, 76, 101)
	IGWO	(24, 41, 54)	(26.0, 42.0, 55.8)	(25, 44, 59)
	BBOSL	<b>(25, 39, 50)</b>	<b>(23.6, 39.5, 54.1)</b>	<b>(26, 40, 56)</b>
R4-2	GA-JS	(37, 60, 84)	(43.1, 63.5, 86.9)	(42, 66, 95)
	DPSO	(52, 77, 110)	(57.7, 81.0, 109.6)	(58, 83, 118)
	GWO	(50, 74, 99)	(57.8, 79.8, 106.2)	(56, 82, 117)
	IGWO	(29, 46, 65)	(30.4, 47.0, 66.2)	(30, 51, 68)
	BBOSL	<b>(24, 41, 60)</b>	<b>(25.6, 42.1, 60.1)</b>	<b>(25, 44, 60)</b>
R4-3	GA-JS	(39, 55, 74)	(40.4, 58.3, 78.6)	(43, 63, 85)
	DPSO	(53, 78, 100)	(56.7, 79.1, 104.6)	(62, 80, 107)
	GWO	(53, 77, 101)	(54.3, 81.4, 106.0)	(51, 84, 114)
	IGWO	(30, 43, 58)	(29.3, 44.5, 60.4)	(31, 46, 63)
	BBOSL	<b>(24, 41, 60)</b>	<b>(24, 41, 60)</b>	<b>(24, 41, 60)</b>

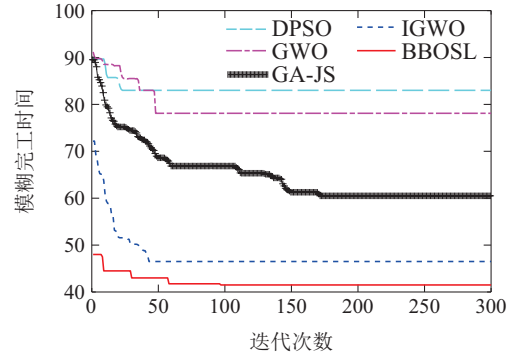


图7 R4-3中5个算法的收敛曲线  
Fig.7 Convergence curves of the five algorithms in R4-3

表8 实验2的Friedman检验结果

Table 8 Friedman test results for experiment 2

算法	平均秩	p值
GA-JS	2.92	
DPSO	4.75	
GWO	4.25	1.54E-18
IGWO	1.88	
BBOSL	1.21	

### 5 总结

本文提出了基于模拟退火和局部搜索策略的生物学优化算法,用于求解模糊分布式柔性作业车间调度问题.设计了一种基于工厂-随机键的双向量编解码方法,利用调度规则进行种群初始化,并提出了基于模拟退火的新解接受方法和基于关键工厂的局部搜索策略.实验结果表明, BBOSL算法在FFJSP和FDFJSP算例中表现出色.在可持续发展和绿色制造的背景下,下一步计划将BBO算法应用于解决节能型FDFJSP,建立适当的数学模型和调度规则,以最大限度地降低能耗和废弃物产生,提高生产效率和经济效益.

### 参考文献:

- [1] LI X Y, XIE J, MA Q J, et al. Improved gray wolf optimizer for distributed flexible job shop scheduling problem. *Science China Technological Sciences*, 2022, 65(9): 2105 - 2115.
- [2] DU Y, LI J, LUO C, et al. A hybrid estimation of distribution algorithm for distributed flexible job shop scheduling with crane transportations. *Swarm and Evolutionary Computation*, 2021, 62: 100861.
- [3] XU W, HU Y, LUO W, et al. A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission. *Computers & Industrial Engineering*, 2021, 157: 107318.
- [4] LUO Q, DENG Q, GONG G, et al. A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm. *Expert Systems with Applications*, 2022, 207: 117984.
- [5] LI Shanghan, HU Rong, QIAN Bin, et al. Hyper-heuristic genetic algorithm for solving fuzzy flexible job shop scheduling problem. *Control Theory & Applications*, 2020, 37(2): 316 - 330. (李尚函, 胡蓉, 钱斌, 等. 超启发式遗传算法求解模糊柔性作业车间调度. *控制理论与应用*, 2020, 37(2): 316 - 330.)



- [6] SEYYEDI M H, SAGHIH A M F, AZIMI Z N. A fuzzy mathematical model for multi-objective flexible job-shop scheduling problem with new job insertion and earliness/tardiness penalty. *International Journal of Industrial Engineering*, 2021, 28(3): 256 – 276.
- [7] LI R, GONG W, LU C. Self-adaptive multi-objective evolutionary algorithm for flexible job shop scheduling with fuzzy processing time. *Computers & Industrial Engineering*, 2022, 168: 108099.
- [8] LI R, GONG W, LU C. A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling. *Expert Systems with Applications*, 2022, 203: 117380.
- [9] CHEN Wenzhou. Optimization of fuzzy distributed flexible job shop scheduling problem. *Modern Manufacturing Engineering*, 2022, 496(1): 10 – 18.  
(陈文洲. 模糊分布式柔性作业车间调度优化. 现代制造工程, 2022, 496(1): 10 – 18.)
- [10] TANG Hongtao, LI Yue, WANG Lei. An improved GWO algorithm for fuzzy distributed flexible job shop scheduling problem. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 2022, 50(6): 81 – 88.  
(唐红涛, 李悦, 王磊. 模糊分布式柔性作业车间调度问题的求解算法. 华中科技大学学报(自然科学版), 2022, 50(6): 81 – 88.)
- [11] LI R, GONG W, WANG L, et al. Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time. *Swarm and Evolutionary Computation*, 2022, 74: 101139.
- [12] YANG J, XU H. Hybrid memetic algorithm to solve multiobjective distributed fuzzy flexible job shop scheduling problem with transfer. *Processes*, 2022, 10(8): 1517.
- [13] WANG X W, HUA Y, GAO J, et al. Digital twin implementation of autonomous planning arc welding robot system. *Complex System Modeling and Simulation*, 2023, 3(3): 236 – 251.
- [14] SIMON D. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 2008, 12(6): 702 – 713.
- [15] ZHANG Z, GAO Y. Solving large-scale global optimization problems and engineering design problems using a novel biogeography-based optimization with Levy and brownian movements. *International Journal of Machine Learning and Cybernetics*, 2023, 14(1): 313 – 346.
- [16] RAJESHWARAN S, KUMAR C, GANAPATHY K. Hybrid optimization based PID controller design for unstable system. *Intelligent Automation & Soft Computing*, 2023, 35(2): 1611 – 1625.
- [17] ZHANG Qiwen, YANG Yongchao. Improved biogeography-based optimization algorithm introducing species evolution. *Application Research of Computers*, 2022, 39(2): 367 – 373, 378.  
(张其文, 杨永超. 引入物种演化的改进生物地理学优化算法. 计算机应用研究, 2022, 39(2): 367 – 373, 378.)
- [18] ZHANG Y, GU X. A biogeography-based optimization algorithm with modified migration operator for large-scale distributed scheduling with transportation time. *Expert Systems with Applications*, 2023, 231: 120732.
- [19] SAKAWA M, MORI T. An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date. *Computers & Industrial Engineering*, 1999, 36(2): 325 – 341.
- [20] LEI D. A genetic algorithm for flexible job shop scheduling with fuzzy processing time. *International Journal of Production Research*, 2010, 48(10): 2995 – 3013.
- [21] GHRAYEB O A. A bi-criteria optimization: Minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems. *Applied Soft Computing*, 2003, 2(3): 197 – 210.
- [22] MA H. An analysis of the equilibrium of migration models for biogeography-based optimization. *Information Sciences*, 2010, 180(18): 3444 – 3464.
- [23] HUANG S, TIAN N, WANG Y, et al. An improved version of discrete particle swarm optimization for flexible job shop scheduling problem with fuzzy processing time. *Mathematical Problems in Engineering*, 2016, 24(5): 3645 – 3658.
- [24] SUN L, LIN L, GEN M, et al. A Hybrid cooperative coevolution algorithm for fuzzy flexible job shop scheduling. *IEEE Transactions on Fuzzy Systems*, 2019, 27(5): 1008 – 1022.
- [25] LU P H, WU M C, TAN H, et al. A genetic algorithm embedded with a concise chromosome representation for distributed and flexible job-shop scheduling problems. *Journal of Intelligent Manufacturing*, 2018, 29(1): 19 – 34.
- [26] ZHANG J, WANG W, XU X. A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility. *Journal of Intelligent Manufacturing*, 2017, 28(8): 1961 – 1972.
- [27] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69: 46 – 61.

#### 作者简介:

孙美玲 硕士研究生, 目前研究方向为智能优化算法与生产调度, E-mail: 13061260501@163.com;

顾幸生 教授, 博士生导师, 目前研究方向为智能优化算法、生产计划与调度、工业过程建模控制与优化、故障检测与诊断等, E-mail: xsgu@ecust.edu.cn.