控制理论与应用
Control Theory & Applications
ISSN 1000-8152,CN 44-1240/TP

《控制理论与应用》网络首发论文

题目：改进 MOEA/D 算法求解双目标模糊柔性作业车间调度问题

作者：李瑞，龚文引

网络首发日期：2021-09-26

引用格式：李瑞，龚文引．改进 MOEA/D 算法求解双目标模糊柔性作业车间调度问题[J/OL]．控制理论与应用．
https://kns.cnki.net/kcms/detail/44.1240.TP.20210924.1642.002.html

改进 MOEA/D 算法求解双目标模糊柔性作业车间调度问题

李 瑞[1] 龚文引[1,†]

(中国地质大学（武汉） 计算机学院, 湖北 武汉 430074)

摘要: 针对同时考虑最大模糊完工时间和总模糊机器负载的双目标模糊柔性作业车间调度问题(BFFJSP), 提出了一种改进的基于分解的多目标进化算法 (IMOEA/D) 同时最优化最大模糊完工时间和总模糊机器负载, 其主要特点是：(1) 采用三种初始化种群的策略; (2) 提出了非支配解优先策略; (3) 设计了结合五种局部搜索策略的变邻域搜索; (4) 提出了计数器策略预防陷入局部解. 运用大量实例进行了算法策略分析和对比实验, 仿真结果表明, IMOEA/D 在求解 BFFJSP 上具有更优性能.

关键词: 双目标模糊柔性作业车间调度, 非支配解优先策略, 变邻域搜索, 计数器策略, MOEA/D

引用格式: 李瑞, 龚文引. 改进MOEA/D算法求解双目标模糊柔性作业车间调度问题. 控制理论与应用, xxxx, xx(x): xxx – xxx

DOI: 10.7641/CTA.2021.00213

An improved MOEA/D for bi-objective fuzzy flexible job-shop scheduling problem

LI Rui[1] GONG Wen-yin[1]

(School of Computer Science, China University of Geosciences, Wuhan Hubei 430074, China)

Abstract: In this paper, the bi-objective fuzzy flexible job shop scheduling problem (BFFJSP) is considered. FFJSP uses triangle fuzzy number to represent the processing time, which is more practical. An improved multi-objective evolutionary algorithm based on decomposition (IMOEA/D) is proposed to minimize fuzzy maximum completion time and total fuzzy machine workload. In IMOEA/D, a novel initialization strategy non-dominated solution first rule (NDSF) is proposed to obtain high quality and diversity initialize population, which combine three initial rules. Variable neighborhood search fixing five neighborhood structure is used to improve the convergence of population. A counter strategy is designed to control the beginning time of VNS and prevent IMOEA/D from falling into local optimal solution. Comprehensive experiments are conducted to demonstrate the effectiveness of IMOEA/D and compare it with 5 state-of-the-art algorithms. Experimental results show that IMOEA/D has strong advantages for solving BFFJSP.

Key words: Bi-objective fuzzy flexible job shop scheduling, non-dominated solution first rule, variable neighborhood search, counter strategy, MOEA/D

Citation: Rui Li and Wen-Yin Gong. An improved MOEA/D for bi-objective fuzzy flexible job-shop scheduling problem. *Control Theory & Applications*, xxxx, xx(x): xxx – xxx

1 引言

在生产制造的决策和智能制造的分配中, 高效的调度方法是提高企业效率的关键[1–3]. 柔性作业车间调度问题(Flexible job-shop scheduling problem, FJSP)作为调度中的经典问题被广泛研究[4–5]. FJSP 作为车间调度问题上的扩展, 对比于传统车间调度问题更加复杂, 不仅要考虑工序加工的顺序, 还要为每个工序分配机器. 由于车间调度问题是非确定性多项式难(Non-deterministic polynomial hard, NP-hard)问题, 所以 FJSP 同样也是 NP-hard 问题[6].

†通信作者. E-mail: wygong@cug.edu.cn; Tel.: +86-27-67883716.

国家自然科学基金项目(62076225); 湖北省自然科学基金(2019CFA081).

Supported by National Natural Science Foundation of China (62076225); Natural Science Foundation for Distinguished Young Scholars of Hubei (2019CFA081).

1. 中国地质大学（武汉）计算机学院 武汉 430074

1. School of Computer Science, China University of Geosciences, Wuhan 430074

FJSP 在主动生产调度、动态调度、分布式协同调度中起着非常重要的作用. 伴随着群智能优化算法和智能技术发展迅速发展, 新启发式算法方法被应用在 FJSP 求解中. 艾子义等 [7]提出了新型的蛙跳算法采用基于种群和记忆的种群划分高效地求解 FJSP 问题, Lei 等 [8]提出了两阶段的帝国竞争算法结合变邻域搜索, 通过实验证明该算法求解带约束的FJSP高效性, Gao 等 [9]提出了离散的 Jaya 算法结合多种针对性局部搜索策略高效的求解带重排的 FJSP 问题.

FJSP 具有良好的可扩展性, 可与生活中各种实际问题相结合. 例如考虑在实际加工过程中机器老化 [10]带来的加工时间的影响, 在调度中考虑能量消耗的低碳环保调度 [11], 在调度过程中考虑各机器加工速度不同的混合流水调度 [12], 考虑能量峰值约束的调度 [13], 多目标调度问题 [14], 分布式车间调度问题 [15]等.

在上述问题中, 机器加工时间是确定的,但在实际生活中工序加工时间受许多因素影响, 如机器故障, 工件配送失败, 等. 所以工序开始和结束的时间应该在一定时间区间内浮动. Sawaka 等 [16]提出模糊作业车间调度问题(Fuzzy job-shop scheduling problem, fJSP) [17], 采用模糊数的概念来表示加工时间, 将三角函数作为其隶属函数, 设计了三角模糊函数的运算比较规则, 并采用遗传算法求解.这样在智能制造中更加符合实际情况, 具有重要的研究价值. Lei [27]首次提出将 fJSP 和 FJSP 进行结合提出了模糊柔性作业车间调度(Fuzzy flexible job-shop scheduling problem, FFJSP), 相较于前两者更加贴合生成调度中的实际情况. 目前有许多算法求解 FFJSP. Lei 等 [18]把协同进化算法应用于 FFJSP 求解中. Lin 等 [19]提出了一种回溯搜索的方法管理六种启发式动作用于动态产生子代求解 FFJSP. Gao 等 [20]采用离散和声搜索算法求解, 提出新的初始化规则提高初始种群质量. Li 等 [21]采用改进人工免疫算法结合多种局部搜索策略有效求解第二类模糊数的 FFJSP. 上述问题在求解FFJSP时都只考虑最大完工时间或最大剩余时间, 但在现实生产调度中, 要考虑多个目标需求, 平衡多个目标直接的影响并进行取舍, 例如最小空闲时间, 最大机器负载, 总机器负载等.

基于上述考虑, 本文针对同时考虑最大完工时间和总机器负载的双目标模糊柔性作业车间调度问题 (bi-objective FFJSP, BFFJSP), 提出改进的 MOEA/D [22] 算法 (IMOEA/D) 求解该问题. 改进算法的主要特点是: 1) 融合了三种初始化策略, 既保证了初始种群的多样性，又确保能产生高质量的初始种群; 2) 提出了非支配解优先策略, 结合 LS 和 GW 在收敛能力强的特点和 Random 多样性强的特点, 为种群初始化提供了优值种群, 加快收敛速度的同时又保持多样性; 3) 设计了结合五种局部搜索策略的变邻域搜索, 在迭代后期用于加强算法的收敛性; 4) 提出了计数器策略预防算法陷入局部解.

余文框架如下, 第二节简要介绍模糊集合的概念, 相关运算以及优化目标, 第三节介绍改进 MOEA/D 算法求解 BFFJSP, 第四节致力于实验比较和分析, 第五节对本文工作进行小结.

2 背景知识

2.1 问题描述

模糊集合 $\tilde{F}$ 中的元素 x 通常用隶属度函数 $\mu_{\tilde{F}}(x)$ 来衡量元素 x 在集合 $\tilde{F}$ 中的可能性, 其中$x \in X$, 则 X 上的模糊集合 $\tilde{F}$ 定义如下:

$$\tilde{F} = \{x, \mu_{\tilde{F}}(x) | \forall x \in X\}, 0 \leqslant \mu_{\tilde{F}}(x) \leqslant 1. \quad (1)$$

经典集合是确定性集合, 当隶属函数 $\mu_{\tilde{F}}(x) = 1$ 时即为经典集合. 定义如下:

$$F = \{x, \mu_{\tilde{F}}(x) = 1 | \forall x \in X\}. \quad (2)$$

隶属函数有很多种, 在调度问题中常用三角模糊数 (Triangle fuzz number, TFN), 由于隶属函数在图 1 上看起来像三角形, 以此命名. 在图 1 中 t_1 表示最早加工时间, t_2 表示最有可能加工时间, t_3 表示最晚加工时间. 通常用 $TFN = (t_1, t_2, t_3)$ 来表示一个TFN, 三角隶属函数定义如下:

$$\mu_{\tilde{F}}(x) = \begin{cases} 0, & x \leqslant t_1, \\ \dfrac{x - t_1}{t_2 - t_1}, & t_1 < x \leqslant t_2, \\ \dfrac{t_3 - x}{t_3 - t_2}, & t_2 < x < t_3, \\ 0, & x \geqslant t_3. \end{cases} \quad (3)$$

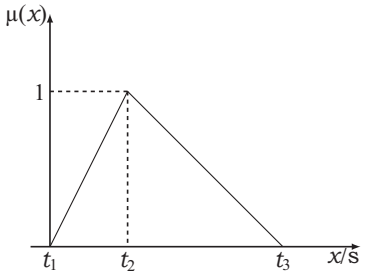

图 1 三角隶属函数

Fig. 1 Triangular membership function

表 1 给出了一个简单的 3 × 3 部分FFJSP的例子共有3个工件和3个机器. 表中 O_{12} 可选的机器只有 M_1 和 M_3, 工件 J_2 的第三工序 O_{23} 选择 M_1, 最早开始时间是 2, 最可能开始的时间为 3, 最晚的开始时间是 4.

表 1 一个3×3的FFJSP例子

Table 1 An example of 3×3FFJSP

工件	工序	M_1	M_2	M_3
J_1	O_{11}	(2,4,7)	(6,8,10)	–
	O_{12}	(1,3,6)	–	(2,8,10)
J_2	O_{21}	–	–	(3,6,9)
	O_{22}	(1,5,7)	(4,5,6)	(7,10,12)
	O_{23}	(2,3,4)	(4,7,8)	(6,7,9)
J_3	O_{31}	(10,15,21)	–	–
	O_{32}	(7,10,12)	(2,5,6)	(3,4,5)

2.2 三角模糊数的运算

在模糊数集合上有三种操作: 加法、比较、取大. 对于两个三角模糊数 (Triangle fuzz number, TFN), $\tilde{s}=(s_1,s_2,s_3), \tilde{t}=(t_1,t_2,t_3)$.

- 加法：模糊数的加法运算定义如下：

$$\tilde{s}+\tilde{t}=(s_1+t_1,s_2+t_2,s_3+t_3). \quad (4)$$

- 比较：根据 Sakawa 提出的排序准则 [16], 采用如下规则比较两个模糊数的大小:

  1) $f_1(\tilde{x})=\frac{x_1+2x_2+x_3}{4}$, 如果 $f_1(\tilde{s})>f_1(\tilde{t})$, 则 $\tilde{s}>\tilde{t}$, 反之 $\tilde{s}<\tilde{t}$.

  2) $f_2(\tilde{x})=x_2$, 当$f_1(\tilde{s})=f_1(\tilde{t})$, 如果 $f_2(\tilde{s})>f_2(\tilde{t})$, 则 $\tilde{s}>\tilde{t}$, 反之 $\tilde{s}<\tilde{t}$.

  3) $f_3(\tilde{x})=s_3-s_1$, 当 $f_2(\tilde{s})=f_2(\tilde{t})$, 如果 $f_3(\tilde{s})>f_3(\tilde{t})$, 则 $\tilde{s}>\tilde{t}$, 反之 $\tilde{s}<\tilde{t}$.

- 取大：两个三角模糊数的近似取大操作定义如下：如果 $\tilde{s}>\tilde{t}$, 则 $\tilde{s}\vee\tilde{t}=\tilde{s}$; 否则 $\tilde{s}\vee\tilde{t}=\tilde{t}$.

2.3 符号说明

- 指数：M: 总的机器数; N: 总的工件数; Θ_i: 工件数 i 的总工序数; i: 工件的索引值; j: 工序的索引值; k: 机器的索引值.

- 变量：$O_{i,j}$:工件 i 的第 j 个工序; $\tilde{S}_{i,j}$: 工序 $O_{i,j}$ 的开始时间, 用三角模糊数表示$\tilde{S}_{i,j}=(s_1,s_2,s_3)$; $\tilde{C}_{i,j}$: 工序 $O_{i,j}$ 的结束时间, 用三角模糊数表示 $\tilde{C}_{i,j}=(c_1,c_2,c_3)$.

- 参数: t_i: 工件 i 需要进行的工序数; $\tilde{P}_{i,j,k}$: 工件i第j个工序在机器k上的处理时间,用三角模糊数表示 $\tilde{P}_{i,j,k}=(p_1,p_2,p_3)$.

- 决策变量: $\mathbf{x}_{i,j,k}$: 工件i第j个工序在机器k上加工则为 1 否则为 0; $\mathbf{u}_{i_1,j_1,i_2,j_2}$: 工件$i_1$第$j_1$个工序和工件$i_2$第$j_2$个工序都在同一个机器上则为 1 否则为 0;

2.4 BFFJSP模型

$$minF_1=\max\left\{\tilde{C}_{i,t_i}\right\}, 1\le i\le N. \quad (5)$$

$$minF_2=\sum_{k=1}^{M}W_k. \quad (6)$$

$$W_k=\sum_{i=1}^{N}\sum_{j=1}^{\Theta_i}\tilde{P}_{i,j,k}*\mathbf{x}_{i,j,k}. \quad (7)$$

$$\tilde{C}_{i,j}=\tilde{S}_{i,j}+\tilde{P}_{i,j,k}. \quad (8)$$

$$\tilde{S}_{i,j}\geqslant\tilde{C}_{i-1,j}. \quad (9)$$

$$(\tilde{S}_{i_1,j_1}-\tilde{C}_{i_2,j_2})*\mathbf{u}_{i_1,j_1,i_2,j_2}\geqslant 0. \quad (10)$$

$$\sum_{1}^{K}\mathbf{x}_{i,j,k}=1. \quad (11)$$

$$\mathbf{x}_{i,j,k}=\begin{cases}1, if\ O_{ij}\ processed\ on M_k,\\ 0, otherwise.\end{cases} \quad (12)$$

$$\mathbf{u}_{i_1,j_1,i_2,j_2}=\begin{cases}1, if\ \mathbf{x}_{i_1,j_1,k}=\mathbf{x}_{i_2,j_2,k}=1,\\ 0, otherwise.\end{cases} \quad (13)$$

公式 5,6 描述的是两个优化目标,分别是最大完工时间和总的机器负载, 都是 TFN. 公式 7 描述机器负载计算公式. 公式 8 采用模糊加操作计算完成时间. 约束 9 表示该工序的开始时间必须小于上个工序完成时间. 约束 10 表示如果两个工序在一个机器上, 则开始时间必须小于正在加工工序的完成时间, 其中所有的时间和时刻都是 TFN. 约束 11 表示一个工件一时间只能由一个机器加工.约束 12, 13 表示决策变量的取值范围.

2.5 实例说明

图 2 中给出了一个甘特图例子, 该实例中一共有10个机器, 10个工件. 所有的开始时间和完成时间都是用TFN, 在图上都用三角形表示, 每条直线上表示该机器上的加工顺序, 直线下的三角形为开始时间, 直线上的为结束时间. 例如工序$O_{6,5}$的开始时间为(29,39,54) 结束时间为(31,46,63). 计算完所有工序的完工时间, 就可以算出最大完工时间.

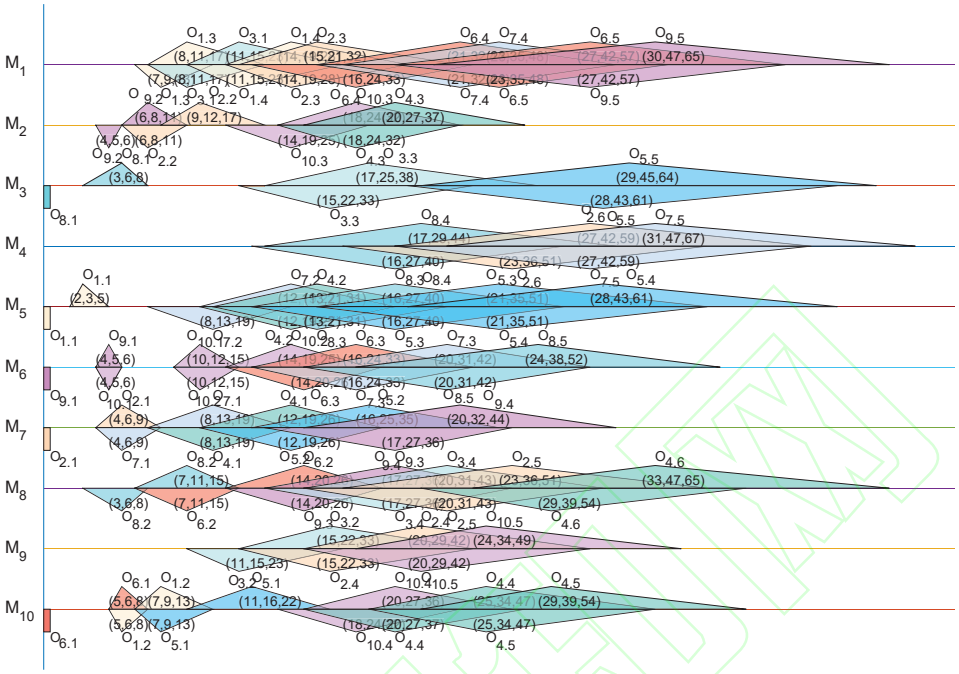

图 2 一个实例的甘特图

Fig. 2 A Gantt chart of an instance

3 改进的MOEA/D算法

3.1 编码与解码

采用传统的双层编码形式表示种群中的个体, 工序码表示每个工序的加工顺序, 如图 3 所示:

工序码	3	2	1	2	1	3	1	3	2
机器码	1	2	3	3	3	1	2	3	3

图 3 编码表示举例

Fig. 3 An example of solution representation

工序顺序为 $O_{31}, O_{21}, O_{11}, O_{22}, O_{12}, O_{32}, O_{13}, O_{33}, O_{23}$, 对应选择的机器为 $M_1, M_2, M_3, M_3, M_3, M_1, M_2, M_3, M_3$, 工序码在机器码同索引位对应的是该工序选择的机器号.

3.2 种群初始化

为了获得优质的初始化种群, 确保群体中个体的质量和群体多样性, 本文结合三种初始化策略: 局部加工时间最小原则 (Local minimum processing time rule, LS) [24], 全局机器负载最小原则 (Global minimum workload rule, GW) [21], 以及随机初始化原则 (Random) [20], 提出了一种新的初始化策略, 即非支配解优先原则 (Non-dominated solution first rule, NDSF).

- 随机初始化原则: 具有高多样性特点, 工序码将工件 j 重复 n_j 次, 再重排整个向量. 机器码选择一个工序 O_{ij} 随机选择该工序的可选机器 M_{ij} 并存储到机器码;
- 局部加工时间最小原则: 根据模糊数比较原则, 每个工序 O_{ij} 选择当前可用机器 M_{ij} 中处理时间最小的;
- 全局机器负载最小原则: 根据模糊数比较原则, 每个工序 O_{ij} 选择当前可用机器 M_{ij} 中机器负载最小的, 如果有多个可选机器, 则选最小处理时间的;
- 非支配解优先原则: 执行 LS 和 GW 原则分别产生 Np 大小的两个种群$\mathbb{P}_1, \mathbb{P}_2$, 然后把两个种群合并. 对于合并种群中的个体利用快速非支配排序 (Fast non-dominated sorting) [28]进行排序, 剔除其中适应值冗余的个体, 如果种群中剩余个体的数量大于 Np, 则选择前 Np 个个体作为初始父种

群$\mathbb{P}$; 否则, 采用随机初始化原则补齐剩余个体.

3.3 交叉和突变

在 2.1 节中采用了双层编码的方式, 染色体中每个基因值都必须是整数, 同时染色体的结构需要满足两个约束: 1) 在工序码中每个基因值表示工序号, 工序号 O_{ij} 必须在 $\{1,\ldots,N\}$ 的范围内取离散的整数值, 且工序 O_{ij} 必须在染色体中重复出现 t 次; 2) 在机器码中每个基因值表示工序号 O_{ij} 选择的机器 M_k, 如表 1 所示有的工序如 O_{12} 只能选择机器 M_1, M_3, 当改变 O_{12} 改变所选机器, 备选的机器号只能在 M_1, M_3 中变动. 在进行交叉操作时需要保证产生的新解依旧满足以上的两个约束. 而传统的交叉操作会将基因值变成实数域的小数值, 违背了上述约束, Gao 等[25]提出了一种新的交叉策略, 将两个不同的染色体的基因一部分做保留, 一部分做交换. 同时不违背两个约束条件.

假设有四个工件$J_1, J_2, J_3, J_4 \in \mathbb{J}$, 每个工件只加工一个工序, 每个工件号 J_i 在染色体中只出现一次, 工序码交叉操作的具体步骤如下:

步骤 1　随机将工件集合$\mathbb{J}$分为两个子集 $\mathbb{A} = \{J_1, J_2\}, \mathbb{B} = \{J_3, J_4\}$.

步骤 2　选取两个解原解 1 和原解 2. 如图 4 所示, 图中较短的直线表示, 将原解 1 中工件 1 和工件 2 的工序拷贝到新解 1. 同样的将原解 2 中工件 3 和工件 4 的工序存入新解2.

步骤 3　经过步骤 2 后. 两个新解中存在较多空白处. 图中较长的直线表示, 将原解 1 中工件 1 和工件 2 的工序按照从左到右的顺序填入新解 2 的空白处. 同样的将原解 2 中的工件3 和 4 的工序存入新解 1.

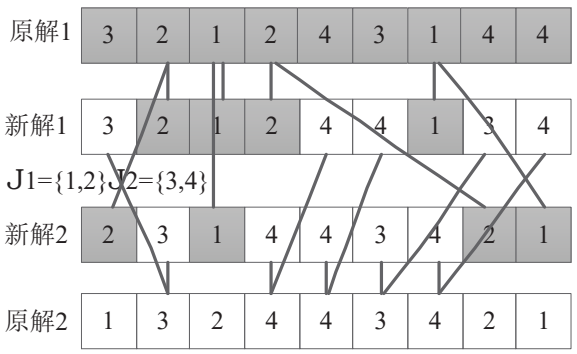

图 4 交叉策略举例

Fig. 4 An example of crossover strategy

突变操作: 1)工序变异: 随机选两个工序码上的基因点, 交换两点基因值. 2)机器变异: 随机选两个变异点位, 在备选机器中随机替换一个.

3.4 变邻域搜索

为了增加算法的后期收敛能力, 设计了结合五种邻域动作的变邻域搜索 (Variable neighborhood search, VNS)[8]. 选取一个解 $\mathbf{x}$, 采用邻域动作 $NS_i, i \in \{1,2,3,4,5\}$ 在解 $\mathbf{x}$ 的附近局部搜索, 产生两个邻域解 $\mathbf{x}'$, $\mathbf{x}''$. 五个局部搜索一共产生十个邻域解, 对于所有邻域解进行评估, 如果有解可以更新参考点则更新当前参考点. 随后根据 1.2 节中的模糊数比较原则, 选取当前邻域解中适应值最好的个体 $\mathbf{x}_{lbest}$, 采用聚合函数的评价方式, 如果 $\mathbf{x}_{lbest}$ 优于$\mathbf{x}$ 则替换 $\mathbf{x}$. 五种邻域动作描述如下:

1) 邻域动作 NS_1: 找到所有工序中最后一个完成的工序 O_{final}, 如果该工序有多个备选机器, 则替换备选机器中处理时间最小的那个.

2) 邻域动作 NS_2: 在所有工序中随机选择一个工序 O_{ij}, 如果该工序有多个备选机器, 则替换备选机器中处理时间最小的那个.

3) 邻域动作 NS_3: 找到负载最大的机器 M_{max}, 在所有 M_{max} 上加工的工序中随机选择工序 O_{ij}, 在该工序的备用机器中随机选择一个替换.

4) 邻域动作 NS_4: 随机选择两个工序 O_{ij} 和 O'_{ij}, 交换两个工序在工序码中的位置.

5) 邻域动作 NS_5: 随机选择两个工序O_{ij} 和 O'_{ij}, 将 O'_{ij} 插入到 O_{ij} 前一个位置.

3.5 IMOEA/D算法描述

本节将 MOEA/D[22]算法, 与前文提到的种群初始化策略, 新的交叉变异操作, 变邻域搜索结合, 最后提出了改进的基于分解的多目标进化算法 (IMOEA/D). 算法具体步骤如下：

步骤 1　初始化 Np 个参考向量, 使其均匀的分布在解空间中, 根据向量间的欧式距离, 计算每个参考向量的T个邻居参考线, 存储邻居信息为邻域矩阵.

步骤 2　用 NDSF 策略初始化种群$\mathbb{P}$大小为Np.

步骤 3 初始化种群 $\mathbb{P}$ 中每个个体的计数器 $F_i = 0, i = (1, \cdots, Np)$.

步骤 4 对于当前种群$\mathbb{P}$中每个个体执行交叉和突变操作,产生两个新解 $\mathbf{x}_{new1}$ 和 $\mathbf{x}_{new2}$. 计算 $\mathbf{x}_{new1}$ 和 $\mathbf{x}_{new2}$ 的适应值并更新全局参考点 $refPoint$.

步骤 5 根据邻域矩阵找到当前解 $\mathbf{x}$ 的 T 个邻域解,根据切比雪夫聚合函数比对新解$\mathbf{x}_{new1}$ 和 $\mathbf{x}_{new2}$ 与 T 个邻域解的聚合函数值, 如果新解更好则替换原邻域个体, 否则当迭代次数大于 $V * iterMax$ 时, 邻域个体的计数器 $Counter_i++$. $iterMax$ 表示最大进化代数.

步骤 6 如果个体计数器累计超过阈值 $limit$ 且迭代次数大于 $V * iterMax$ 且当前个体 $\mathbf{x} \notin PF$, 则执行变邻域搜索. 如果 VNS 成功则替换当前解, 否则个体计数器 F_i++. PF表示算法求得的非支配解集 (Pareto optimal front, PF) [22].

步骤 7 更新当前种群的 PF, 如果满足终止条件, 则搜索结束; 否则,转至步骤4.

在步骤2中NDSF的非支配排序中由于适应值都是TFN, 所以比较大小需要用到2.2节的比大操作. 在计算适应值解码时, 计算每个工序的完工时间需要将处理时间加上开始时间, 需要用到 2.2 节的模糊加操作. 在步骤5中需要计算切比雪夫聚合函数需要用 2.2 节中$f_1(\tilde{x})$将TFN映射为函数值进行计算.

4 实验分析

4.1 实验设计

在第3节已经详细说明了提出的算法.本节将验证 IMOEA/D 的收敛性和分布性. 本文算法和对比算法都采用 MATLAB 语言编程, 在 core-i7 6700 CPU@3.40 GHz 和 8 GB RAM 的电脑上运行. 公平起见, 每个算法独立运行 30 次, 通过30次运行的最好结果和平均结果来验证 IMOEA/D 的分布性和收敛性.

对比算法主要分为三类：(1)基于种群的算法. 新提出的 IAIS [21]; (2)基于参考线的多目标优化算法. 包括 MOEA/D [22]和 MOEA/D-M2M [23]; (3)基于支配关系的多目标优化算法. 包括 NSGA-II [28]和 NSGA-III [29]

为了对比六个算法所获得 PF 的收敛性和分布性, 选用超体积度量(Hypervolum, HV) [30]公式如下:

$$HV(P, r) = \bigcup_{\mathbf{x} \in P}^{P} v(\mathbf{x}, r). \tag{14}$$

P 为算法求得的 PF, r 为该解集的参考点, $r = (1, 1)$, $\mathbf{x} = (\frac{f_1 - f_{1min}}{f_{1max} - f_{1min}}, \frac{f_2 - f_{2min}}{f_{2max} - f_{2min}})$, v 表示在 PF 上的点 $\mathbf{x}$ 与 r 坐标围成的超立方体的体积, HV 将 PF 上各个点 $\mathbf{x}$ 和参考点 r 围成超立方体的体积 v 求并集, 即 P 与 r 组成的不规则超立方体的总体积. HV 适用于在不知真实 PF 前沿的情况, 度量统一坐标系下解的分布性和收敛性. HV 的值越大, 算法性能越好.

4.2 实验数据集

实验测试集参考 [26]选择两个标准测试集, 两组数据集都是模糊柔性作业车间调度的标准测试集, 区别在于工序在选择机器时是否全部机器备选. 第一组数据选自 [18, 27], 第二组数据选自 [20].

第 1 组测试集是完全 FFJSP, 即各工序可以在所有的机器上加工. 共有 5 个实例, 最小规模的实例工件数是 10, 机器数是 10, 总工序数为 40. 规模最大的实例工件数是 15, 机器数是 10, 总工序数为 80.

第 2 组测试集是部分 FFJSP, 即各工序在加工时仅能选择部分机器, 共有 8 个实例, 最小规模的实例工件数是 5, 机器数是 4, 总工序数为 23, 规模最大的实例工件数是 20, 机器数是 15, 总工序数为 355.

表 2 参数的Friedman秩和检验

Table 2 Friedman rank-sum test of the parameters

Algorithm	Ranking	Algorithm	Ranking
T05	2.3846	M03	**1.9231***
T10	**2.3077***	M05	2.6154
T15	2.9231	M08	3
T20	2.3846	M10	2.4615
Algorithm	Ranking	Algorithm	Ranking
V10	3.3846	R04	4.0769
V20	2.769	R05	2.9231
V30	2.6154	R06	2
V50	**1.2308***	R08	**1.6154***

*means the best algorithm of this column

4.3 实验参数

本文提出的算法涉及的参数主要涉及 MOEA/D 的邻域个数 T, 统计启动 VNS 的计数器阈值 M (即 iterMax), 还有在最后 V%迭代轮

次,启动 VNS, 进化过程中变异的概率 R. 为了测试哪一种参数是更好的选择, 本文采用控制变量的方法, 设计了四组参数对比试验, 设置如下:

T=5, 10, 15, 20. M=3, 5, 8, 10.

V=10, 20, 30, 50. R=0.4, 0.5, 0.6, 0.8.

每一组实验中代码在两个测试集上独立运行30次, 计算30轮独立运行的平均HV值, 将得到的结果用Friedman秩和检验, 将每一组参数做个排序. 由表 2 结果可知 $T=10, M=3, V=50, R=0.8$ 是最好的选择, 表 2 中标星号的是该列中最好的算法. 在两个数据集上, 所有算法相关参数都是选择上述最优参数.

4.4 初始化策略的有效性

种群初始化策略在3.2节已经讨论过了, 为了证明种群初始化策略的有效性, 本文编码两种算法, 一种是 MOEA/D加上随机初始化策略, 另一种是 MOEA/D加上非支配解优先策略, 其他参数设置相同, 在两组测试集上独立运行30次计算HV平均值和最优值即对比能够找到最优的前沿面. 由表 3 结果可知, NDSF初始化策略在求解的问题上表现更好.

表 3 MOEA/D 和 MOEA/D+NDSF 对比结果

Table 3 Comparison result between MOEA/D and MOEA/D+NDSF

实例	MOEA/D+Random		MOEA/D+NDSF	
	aver	best	aver	best
D1	0.088394	0.093111	**0.09074**	**0.095866**
D2	0.087689	**0.093819**	**0.088444**	0.093101
D3	0.061554	0.066035	**0.063381**	**0.068624**
D4	0.060536	0.066232	**0.061442**	**0.067661**
D5	0.046982	0.054037	**0.051365**	**0.057103**
R1	**0.042643**	0.044132	0.042499	**0.044132**
R2	0.035812	0.042099	**0.036814**	**0.043283**
R3	0.035828	0.03912	**0.036181**	**0.041213**
R4	0.040066	0.045096	**0.040915**	**0.045862**
R5	0.03471	0.039485	**0.035866**	**0.040251**
R6	0.032388	0.040401	**0.036483**	**0.044226**
R7	0.036565	0.047181	**0.047831**	**0.053561**
R8	0.033841	0.046013	**0.064988**	**0.071104**

4.5 变邻域搜索策略的有效性

在3.4节中描述了变邻域搜索的细节, 为了证明变邻域搜索的有效性, 本文编码两种算法, 一种是 IMOEA/D去掉变邻域搜索,使用随机局部搜索策略,称为IMOEA/DNV, 另一种是使用变邻域搜索的 IMOEAD, 其他参数设置相同, 在两组测试集上独立运行30次计算HV平均值和最优值即对比能够找到最优的前沿面. 由表 4 结果可知, 变邻域搜索在求解问题上表现更好.

表 4 IMOEA/DNV 和 IMOEA/D对比结果

Table 4 Comparison result between IMOEA/DNV and IMOEA/D

实例	IMOEA/DNV		IMOEA/D	
	aver	best	aver	best
D1	0.084032	**0.089744**	**0.085351**	0.088257
D2	0.097715	0.102836	**0.098164**	**0.105348**
D3	0.055954	0.06087	**0.056251**	**0.064308**
D4	0.058506	0.064192	**0.059513**	**0.067065**
D5	0.055942	0.061425	**0.057426**	**0.063475**
R1	0.036291	0.037651	**0.036619**	**0.037651**
R2	0.047897	0.05355	**0.048088**	**0.053568**
R3	**0.036773**	**0.041887**	0.036505	0.041022
R4	**0.042948**	0.047799	0.042694	**0.049463**
R5	0.033407	0.037489	**0.034636**	**0.040411**
R6	0.041878	**0.049256**	**0.043516**	0.048924
R7	0.041247	0.0469	**0.043451**	**0.050458**
R8	0.06282	0.068734	**0.065229**	**0.071376**

4.6 对比实验

为了进一步证明IMOEA/D的收敛性和分布性, 本文选择了如下算法作为对比: MOEA/D, NSGA-II, MOEA/D-M2M, NSGA-III, IAIS. 每个对比算法都在两组测试集上独立运行30次. 统计每个测试实例上HV值的均值以及HV最优值. IMOEA/D的参数设置参照4.3节中最优参数选择. 由表 5 结果可知在所求解的测试集上, IMOEA/D算法都有着良好的收敛性和分布性.

在图 5 中给出了六个算法在所有测试集上的所找的最优前沿面, 前沿面是根据30轮独立运行中最优HV值的轮次选取, 图中横坐标表示最大完工时间, 纵坐标表示总机器负载, 由于两个目标函数都是TFN是三维向量, 为了展示方便经过 2.2 节中$f_1(\tilde{x})$处理, 将三维向量映射为均值. 由图 5 结果可知, IMOEA/D对比其他算法具有良好的收敛性和分布性.

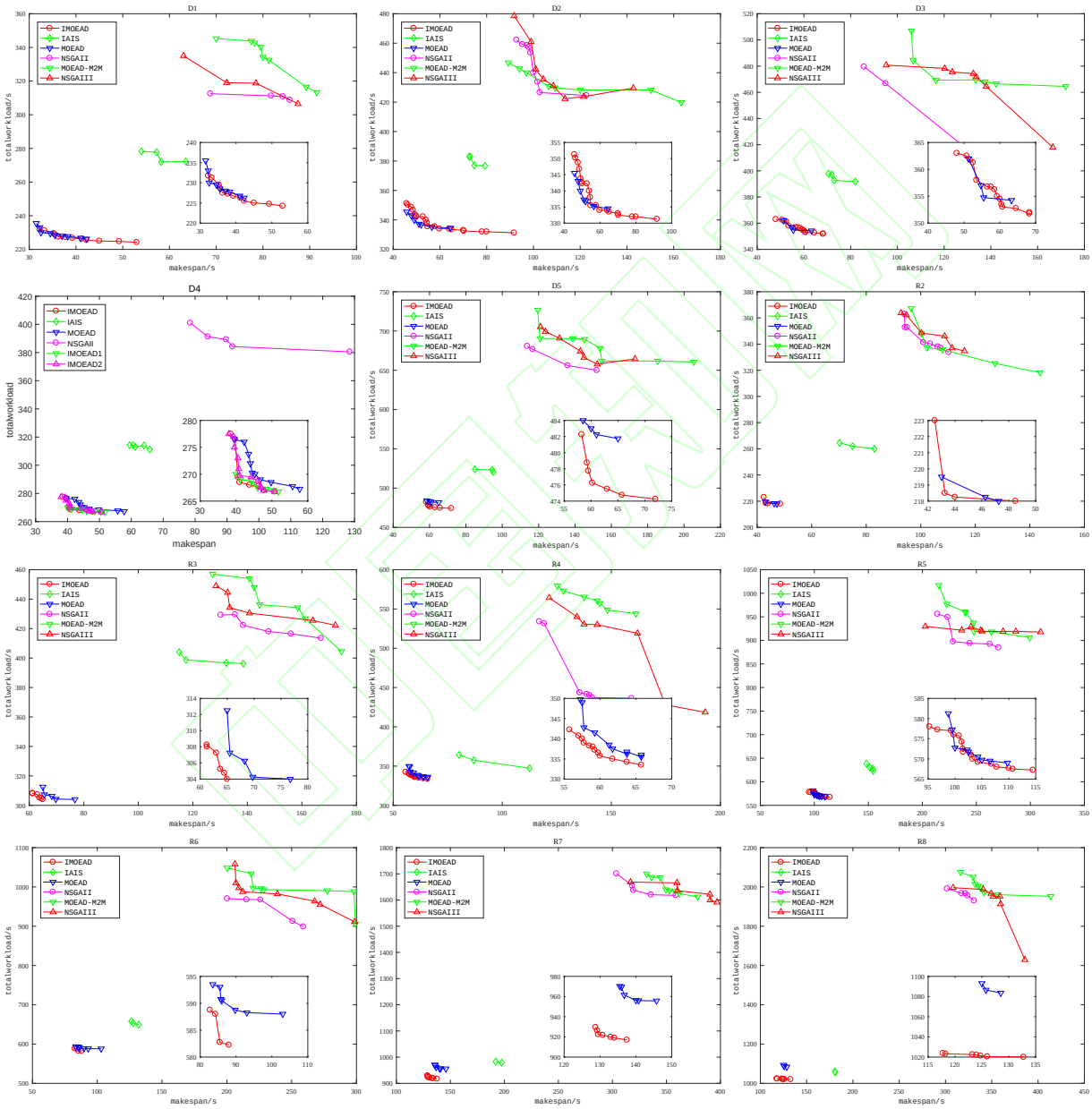

图 5 两个标准测试集的帕里托前沿结果

Fig. 5 Pareto Front results of two benchmarks

表 5 六个算法的HV值对比结果

Table 5 HV value results of six comaprsion algorithms

实例	IMOEA/D		IAIS		MOEA/D		NSGAII		MOEA/D-M2M		NSGAIII	
	aver	best	aver	best	aver	best	aver	best	aver	best	aver	best
D1	**0.34664**	**0.351052**	0.207838	0.268335	0.342602	0.349697	0.133522	0.148387	0.126514	0.137847	0.124076	0.142776
D2	**0.279613**	0.289817	0.180792	**0.319032**	0.278851	0.287572	0.117376	0.132501	0.122591	0.239099	0.116328	0.211559
D3	**0.281324**	**0.293162**	0.177692	0.201502	0.278628	0.283894	0.102613	0.132943	0.094743	0.105166	0.105003	0.126938
D4	**0.371355**	**0.380504**	0.253692	0.304137	0.368871	0.376695	0.146552	0.155583	0.145802	0.17853	0.140604	0.176314
D5	**0.310545**	**0.317087**	0.194729	0.234844	0.302175	0.310063	0.094137	0.112154	0.091317	0.102813	0.10339	0.270691
R1	**0.32301**	**0.324721**	0.209474	0.279117	0.322686	0.324721	0.15783	0.188688	0.143673	0.175514	0.146147	0.19016
R2	**0.358787**	**0.366725**	0.244213	0.548902	0.357455	0.365199	0.109059	0.132416	0.101929	0.117248	0.118969	0.517473
R3	0.303859	**0.310521**	0.171322	0.22335	**0.303901**	0.30851	0.09052	0.09052	0.09789	0.204732	0.086485	0.195758
R4	**0.35935**	**0.369295**	0.217932	0.296265	0.358586	0.366126	0.09389	0.14772	0.102896	0.146727	0.106153	0.149921
R5	**0.345592**	**0.354724**	0.236876	0.274182	0.34166	0.348676	0.072579	0.08679	0.07454	0.2772	0.076582	0.330675
R6	**0.384762**	**0.391823**	0.273012	0.311862	0.37639	0.387096	0.08384	0.09452	0.082528	0.092448	0.083052	0.096226
R7	**0.380975**	0.388763	0.294521	**0.471762**	0.361105	0.37539	0.058732	0.070027	0.062223	0.282657	0.063068	0.268007
R8	**0.415481**	**0.422458**	0.321148	0.349924	0.371536	0.38885	0.053597	0.066826	0.062454	0.091977	0.072413	0.0836

5 结论

为了降低生产过程中的机器负载和能量消耗, 优化最大完工时间和总的机器负载, 本文提出了改进的MOEA/D算法来求解BFFJSP. 该算法结合MOEA/D良好多样性, 增加了优化初始化种群的非支配解优先策略, 并在迭代过程中加入计数器策略和变邻域搜索增加算法的收敛能力. 为了验证算法的性能, 选取两个标准数据集进行测试, 并与其他相近算法进行对比. 实验结果表明, 对于大部分数据实例, IMOEA/D的收敛性和分布性优于对比算法.

后续工作将考虑目标函数的扩展, 考虑更多实际问题因素, 结合超多目标优化算法求解模糊柔性车间调度问题.

参考文献:

[1] 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.)

[2] XU Wenjie, ZHU Guangyue. Genetic algorithm based on similarity of intuitionistic fuzzy sets for many-objective flow shop scheduling problems. *Control Theory & Applications*, 2019, 36(7): 1057-1066.
(徐文婕, 朱光宇. 直觉模糊集相似度遗传算法求解多目标车间调度问题. 控制理论与应用, 2019, 36(7): 1057-1066.)

[3] LI Ming, LEI Deming. Novel imperialist competitive algorithm for many-objective flexible job shop scheduling. *Control Theory & Applications*, 2019, 36(6): 893-901.
(李明, 雷德明. 基于新型帝国竞争算法的高维多目标柔性作业车间调度. 控制理论与应用, 2019, 36(6): 893-901.)

[4] WANG Fang, TANG Qiuhua, RAO Yunqing, et al. Efficient estimation of distribution for flexible hybrid flow shop scheduling.*Acta Automatica Sinica*, 2017, 43(2): 280-293
(王芳, 唐秋华, 饶运清, 等. 求解柔性流水车间调度问题的高效分布估算算法. 自动化学报, 2017, 43(2): 280-293)

[5] ZHENG Xiaocao, GONG Wenyin. An improved artificial bee colony algorithm for fuzzy flexible job-shop scheduling problem. *Control Theory & Applications*, 2020, 37(6):1284-1292
(郑小操, 龚文引. 改进人工蜂群算法求解模糊柔性作业车间调度问题. 控制理论与应用, 2020, 37(6):1284-1292)

[6] PAVLOV A A, MISURA E B, MELNIKOV O V, et al. NP-Hard Scheduling Problems in Planning Process Automation in Discrete Systems of Certain Classes. *Advances in Computer Science for Engineering and Education*, 2019, 429-436

[7] AI Ziyi, LEI Deming, A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling. *Control Theory & Applications*, 2017, 34(10):1361-1368
(艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度. 控制理论与应用, 2017, 34(10):1361-1368)

[8] LEI D M, LI M, WANG L. A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold *IEEE Transactions on Cybernetics*, 2019, 49, 1097-1109

[9] GAO K Z, YANG F J, ZHOU M C, et al. Flexible job-shop rescheduling for new job insertion by using discrete jaya algorithm. *IEEE Transactions on Cybernetics*, 2019, 49, 1944-1955

[10] LI J Q, SONG M X, WANG L, et al. Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs. *IEEE Transactions on Cybernetics*, 2020, 50, 2425-2439

[11] PAN Zixiao, LEI Deming. Research on property-based distributed low carbon parallel machines scheduling algorithm. *Acta Automatica Sinica*, 2020, 46(11): 2427-2438
(潘子肖, 雷德明. 基于问题性质的分布式低碳并行机调度算法研究. 自动化学报, 2020, 46(11): 2427-2438)

[12] LEI D M, GAO L, ZHENG Y L. A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop.*IEEE Transactions on Engineering Management*, 2018, 65, 330-340

[13] LEI Deming, YANG Dongjing. Research on flexible job shop scheduling problem with total energy consumption constraint. *Acta Automatica Sinica*, 2018, 44(11): 2083-2091
(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. 自动化学报, 2018, 44(11): 2083-2091)

[14] JIA Zhaohong, WANG Yan, Zhang Yiwen. A bi-objective synergy optimization algorithm of ant colony for scheduling on non-identical parallel batch machines. *Acta Automatica Sinica*, 2020, 46(6): 1121-1135
(贾兆红, 王燕, 张以文. 求解差异机器平行批调度的双目标协同蚁群算法. 自动化学报, 2020, 46(6): 1121-1135)

[15] WANG Ling, DENG Jing, WANG Shengyao. Survey on optimization algorithms for distributed shop scheduling. *Control & 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, 控制与决策, 2016, 31(1): 1-11)

[16] SAKAWA M, MORI T. An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy duedate. *Computers & Industrial Engineering*, 1999, 36(2):325-341

[17] ABDULLAH S, ABDOLRAZZAGH N M. Fuzzy job-shop scheduling problems: A review. *Information Sciences*, 2014, 278, 380-407

[18] LEI D M. Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Applied Soft Computing*, 2012, 12, 2237-224.

[19] LIN J. Backtracking search based hyper-heuristic for the flexible job-shop scheduling problem with fuzzy processing time. *Engineering Applications of Artificial Intelligence*, 2019, 77, 186-196

[20] GAO K Z, SUGANTHAN P N, PAN Q K, et al. An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time. *International Journal of Production Research*, 2015, 53, 5896-5911

[21] LI J Q, LIU Z M, LI C D, et al. Improved artificial immune system algorithm for Type-2 fuzzy flexible job shop scheduling problem. *IEEE Transactions on Fuzzy Systems*, 2020, 1-1

[22] ZHANG Q F, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11, 712-731

[23] LIU H L, GU F Q, ZHANG Q F. Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems. *IEEE Transactions on Evolutionary Computation*, 2014, 18(3), 450-455

[24] SHAHEED I M, SHUKOE S A, ABDULLAH S. Population initialisation methods for fuzzy job-shop scheduling problems: issues and future trends. *International Journal on Advanced Science, Engineering and Information Technology*, 2018, 8, 1820-1828

[25] GAO K Z, SUGANTHAN P N, CHUA T J, CHONG C S, et al. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion. *Expert Systems with Applications*, 2015, 42(21):7652-7663

[26] PALACIOS J J, PUENTE J, VELA C R, et al. Benchmarks for fuzzy job shop problems. *Information Sciences*, 2016, 329, 736-752

[27] LEI D M. A genetic algorithm for flexible job shop scheduling with fuzzy processing time. *International Journal of Production Research*, 2010, 48 , 2995-3013

[28] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6, 182-197

[29] DEB K, JAIN H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4), 577-601

[30] WHILE L, HINGSTON P, BARONE L, et al. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 2006, 10(1):29-38

作者简介:

李 瑞 硕博连读研究生, 目前研究方向为智能优化与应用, E-mail: liruicug@163.com;

龚文引 教授, 博士, 博士生导师, 目前研究方向为智能优化与应用, E-mail: wygong@cug.edu.cn.