

超启发式遗传算法求解模糊柔性作业车间调度

李尚函¹, 胡蓉^{1,2†}, 钱斌^{1,2}, 张梓琪¹, 金怀平¹

(1. 昆明理工大学 信息工程与自动化学院, 云南 昆明 650500;

2. 昆明理工大学 机电工程学院, 云南 昆明 650500)

摘要: 本文提出一种混合超启发式遗传算法(HHGA), 用于求解一类采用三角模糊数表示工件加工时间的模糊柔性作业车间调度问题(FFJSP), 优化目标为最小化最大模糊完工时间(即makespan). 首先, 详细分析现有三角模糊数排序准则性质, 并充分考虑取大操作的近似误差和模糊度, 设计一种更为准确的三角模糊数排序准则, 可合理计算FFJSP和其他各类调度问题解的目标函数值. 其次, 为实现对FFJSP解空间不同区域的有效搜索, HHGA将求解过程分为两层, 高层利用带自适应变异算子的遗传算法对6种特定操作(即6种有效邻域操作)的排列进行优化; 低层将高层所得的每种排列作为一种启发式算法, 用于对低层相应个体进行操作来执行紧凑的变邻域局部搜索并生成新个体, 同时加入模拟退火机制来避免搜索陷入局部极小. 最后, 仿真实验和算法比较验证了所提排序准则和HHGA的有效性.

关键词: 三角模糊数排序; 模糊加工时间; 柔性作业车间; 超启发式算法

引用格式: 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

DOI: 10.7641/CTA.2019.80813

Hyper-heuristic genetic algorithm for solving fuzzy flexible job shop scheduling problem

LI Shang-han¹, HU Rong^{1,2†}, QIAN Bin^{1,2}, ZHANG Zi-qi¹, JIN Huai-ping¹

(1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming Yunnan 650500, China;

2. School of Mechanical and Electronic Engineering, Kunming University of Science and Technology, Kunming Yunnan 650500, China)

Abstract: In this paper, a hybrid hyper-heuristic genetic algorithm (HHGA) is proposed to minimize the maximum fuzzy completion time (i.e., makespan) for the fuzzy flexible job shop scheduling problem (FFJSP), in which the job's processing time is represented by using triangular fuzzy number. Firstly, after analyzing the properties of the existing sorting rules on triangular fuzzy number in detail, and fully considering the approximate error and the ambiguity of the operation of taking the bigger, a more accurate triangular fuzzy number sorting rule is designed, which can reasonably calculate the objective function values of the solutions for FFJSP and other various scheduling problems. Secondly, to realize the effective search in different regions of FFJSP's solution space, HHGA divides the solving process into two layers. The upper layer uses the genetic algorithm with adaptive mutation operator to optimize the permutation of six special operations, i.e., six effective neighbor operations. The lower layer uses each permutation obtained from the upper layer as a heuristic to perform operations on the corresponding individual of the lower layer for executing a compact variable neighborhood local search and generating new individual, and meanwhile adds the simulated annealing mechanism to overcome the local-optimality trap. Finally, simulation experiments and algorithm comparisons verify the effectiveness of the proposed sorting rules and HHGA.

Key words: triangular fuzzy number sorting; fuzzy machining time; flexible job shop; hyper-heuristic algorithm

Citation: 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

收稿日期: 2018-10-21; 录用日期: 2019-06-04.

†通信作者. E-mail: ronghu@vip.163.com; Tel.: +86 13508719500.

本文责任编辑: 丛爽.

国家自然科学基金项目(51665025, 61963022), 云南省应用基础研究计划重点项目, 云南省教育厅科学研究基金项目(2017ZZX149)资助.

Supported by the National Natural Science Foundation of China (51665025, 61963022), the Applied Basic Research Key Project of Yunnan Province and the Scientific and the Research Foundation Project of Yunnan Education Department (2017ZZX149).

1 引言

生产调度是智能制造中管理决策和资源分配的核心. 高效的调度方法是实现智能制造的基础和提高企业效率的关键^[1-5]. 柔性作业车间调度问题(flexible job shop scheduling problem, FJSP)是一类典型的生产调度问题. 由于传统作业车间调度问题(job shop scheduling problem, JSP)具有非确定多项式难(non-deterministic polynomial hard, NP-hard)的属性^[6], 且该问题归约于(reduce to)FJSP, 故FJSP为NP-hard问题. 在实际生产过程中, 加工环境复杂多变, 往往难以获得准确的加工时间. 利用模糊理论中的三角模糊数表征不确定的加工时间, 进而建立模糊柔性作业车间调度问题(fuzzy flexible job shop scheduling problem, FFJSP)模型, 可更客观地描述实际生产过程. 因此, 研究FFJSP有很高的理论和实际价值.

在计算模糊生产调度问题不同工件排列或解对应的目标函数值时, 需按问题模型进行模糊数(即加工时间)的加、减、取大运算; 同时评价不同解的优劣时, 也需对各解的目标函数值进行比较运算. 其中, 取大和比较运算均可由模糊数排序实现. 由于加、减运算简单直观, 已有研究都采用模糊数学的相关定义直接执行. 但是, 模糊数排序较为复杂, 存在多种不同的排序准则. 优化算法要有效求解模糊调度问题, 不仅要求算法本身具有较强搜索能力, 也需设计或采用合理的排序准则以帮助算法确定优质解区域. 对于模糊作业车间系列调度问题的研究, 最先出现在模糊JSP上, 目前在排序准则和求解算法上已有一定的研究成果^[7-10]. Sakawa等^[7]以最大化最小客户满意度为目标函数, 用模糊数表示加工时间和交货期, 设计一种基于隶属度参数值的三角模糊数排序准则, 并提出融入个体相似度信息的遗传算法求解. Sakawa等^[8]在文献[7]的问题中同时以最大化最小客户满意度和平均客户满意度、最小化最大模糊完工时间为目标函数, 并在文献[7]排序准则的基础上, 设计三条排序准则来综合确定模糊数的序或大小, 进而采用类似文献[7]的遗传算法求解. Lei^[9]在文献[7]的问题中加入机器可用约束, 采用文献[8]的排序准则并证明其优于文献[7]的准则, 进而提出随机键遗传算法进行求解. Tsujimura等^[10]以最小化最大模糊完工时间为目标函数, 用模糊数表示加工时间, 采用文献[8]的排序准则, 并设计对解部分位置进行交换的交叉操作, 进而提出遗传算法求解.

对于FFJSP, 现有的单目标问题研究均以最小化最大模糊完工时间为目标函数, 用三角模糊数表示加工时间, 并采用文献[8]中的排序准则, 差别在于所提求解算法不同. Lei 分别提出分解-集成遗传算法(decomposition-integration genetic algorithm, DIGA)^[11]和协同进化遗传算法(co-evolutionary genetic algorithm, CGA)^[12]求解. Xu等^[13]提出教与学算法(teaching-lea-

ring-based optimization algorithm, TLBO)求解, 并采用一种分别处理操作序列和机器分配序列的局部搜索来增强算法性能. Lin^[14]提出混合生物地理学优化算法(hybrid biogeography-based optimization, HBBO)求解, 并采用路径重链技术(path relinking technique, PRT)来提升算法局部搜索能力, 取得较好效果. Wang等^[15]提出分布式估计算法(estimation of distribution algorithm, EDA)求解, 并在解码时采用左移插空操作来改善算法性能. Wang等^[16]提出混合人工蜂群算法(hybrid artificial bee colony algorithm, HABC)求解, 算法利用多种策略生成初始解, 并采用变领域搜索(variable neighborhood search, VNS)来增强算法局部搜索能力. 对于多目标FFJSP, Gao等^[17]以同时最小化最大模糊完工时间和最大模糊机器负荷为目标函数, 用三角模糊数表示加工时间, 并采用文献[8]中的排序准则, 提出人工蜂群算法(artificial bee colony algorithm, ABC)求解. 由文献调研可知, 一方面, 已有研究基本都采用文献[8]中的排序准则来确定模糊数大小, 尚无对其进行改进的研究. 另一方面, 现有求解FJSP和FFJSP的算法均基于某种常规的元启发式算法框架, 并在其中融合一些特定操作或局部搜索, 该类问题的求解算法仍存在提升性能的空间.

超启发式算法(hyper-heuristic algorithm)是一类新型智能优化算法. 该算法通过某种高层策略(high-level strategy, HLS)动态控制多种不同低层启发式操作(low-level heuristics, LLH)的执行, 有利于实现对问题解空间不同区域的搜索. 近年来这类算法在组合优化领域已获得较广泛的成功应用^[18-21]. 但是, 超启发式算法在车间调度方面的研究仍比较有限. Rodr等^[22]针对带序相关设置时间混合流水车间调度问题, 以最小化最大完工时间为目标函数, 提出一种超启发式算法求解. 该算法高层利用带随机插入(Insert)变异操作的遗传算法对13种启发式规则的排列进行优化; 低层依据高层所得每种排列中的规则顺序, 依次执行相应规则来生成解并执行搜索. Park等^[23]针对动态作业车间调度问题, 以最小化平均加权延迟时间为目标函数, 提出一种基于遗传编程的超启发式算法求解. 该算法高层利用遗传编程来构造不同的分派规则; 低层执行高层确定的分派规则以生成解并实现搜索. Yan等^[24]针对作业车间调度问题, 以最小化最大完工时间为目标函数, 提出一种超启发式遗传算法求解. 该算法高层利用轮盘赌策略对16种交叉变异组合进行选择; 低层使用高层选择的组合来执行遗传算法并实现搜索. Wang等^[25]针对随机作业车间调度问题, 以最小化期望最大完工时间为目标函数, 提出一种超启发式遗传算法求解. 该算法高层采用带随机交换(Interchange)变异操作的遗传算法, 对不同故障场景下的4种启发式规则的排列进行优化; 低层依据高层所得每种排列

中的规则顺序,依次执行相应规则来生成解并执行搜索.上述算法虽能较好求解相关问题,但仍存在一定的不足:1)算法高层仅对规则组合或简单操作组合进行优化,这导致算法在执行低层搜索时未能执行局部搜索,一定程度上限制了算法的性能;2)这些算法简单采用较小的固定变异概率,这使得算法容易过早陷入局部极小.目前尚无超启发式算法求解FFJSP的相关研究.

本文提出一种两层结构的混合超启发式遗传算法(hybrid hyper-heuristic genetic algorithm, HHGA),用于求解FFJSP,采用三角模糊数表示工件加工时间,优化目标(即目标函数)为最小化最大模糊完工时间(即makespan).首先,详细分析了现有模糊调度问题中使用的三角模糊数排序准则性质,并充分考虑取大操作的近似误差和模糊度,设计一种更为准确的三角模糊数排序准则,可合理计算FFJSP和其他各类调度问题的目标函数值.其次,将问题求解算法分为两层,高层策略域利用带自适应变异算子的遗传算法来优化6种低层启发式操作(即6种有效邻域操作)的排列,低层问题域将高层所得的每种操作排列作为一种启发式算法,用于对低层相应个体进行紧凑的变邻域局部搜索,以此来对问题解空间的不同区域进行有效搜索.同时在搜索过程中加入模拟退火机制,利用模拟退火算法的概率跳突特性(在局部最优附近以一定概率接受较差解)来避免搜索陷入局部极小.最后,仿真实验和算法比较验证了所提三角模糊数排序准则和HHGA的有效性.

2 模糊柔性作业车间调度问题

2.1 问题描述

由于生产过程往往伴随着很多不确定性因素,每个工件的加工时间只能确定为一个大概的范围,因此在模糊柔性作业车间调度问题中,通常使用三角模糊数(triangular fuzzy number, TFN)来表示每个工序的加工时间 $TFN = (t_1, t_2, t_3)$,隶属度函数图像如图1所示.

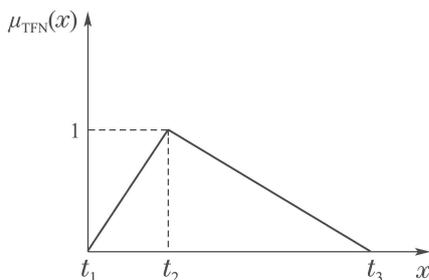


图1 三角模糊数的隶属度函数

Fig. 1 Membership function of triangular fuzzy number

图中: t_1 为最乐观的加工时间, t_2 为最可能的加工时间,隶属度为1,也就是这个模糊数的核, t_3 为最保

守的加工时间. TFN的隶属度函数表达式^[26-27]如下:

$$\mu_{TFN}(x) = \begin{cases} 0, & x \leq t_1, \\ \frac{x - t_1}{t_2 - t_1}, & t_1 < x \leq t_2, \\ \frac{t_3 - x}{t_3 - t_2}, & t_2 < x < t_3, \\ 0, & x \geq t_3. \end{cases} \quad (1)$$

会有如下的几种特殊情况:

1) 若 $t_2 = t_3$ 时,

$$\mu_{TFN}(x) = \begin{cases} 0, & x \leq t_1, \\ \frac{x - t_1}{t_2 - t_1}, & t_1 < x \leq t_2 = t_3, \\ 0, & x > t_2 = t_3. \end{cases} \quad (2)$$

2) 若 $t_1 = t_2$ 时,

$$\mu_{TFN}(x) = \begin{cases} 0, & x < t_1 = t_2, \\ \frac{t_3 - x}{t_3 - t_2}, & t_1 = t_2 \leq x < t_3, \\ 0, & x \geq t_3. \end{cases} \quad (3)$$

3) 若 $t_1 = t_2 = t_3$ 时,

$$\mu_{TFN}(x) = \begin{cases} 1, & x = t_1 = t_2 = t_3, \\ 0, & \text{其他}. \end{cases} \quad (4)$$

因为加工时间采用TFN表示,所以模糊完工时间 C_i 也是TFN,表示为 $C_i = (c_1, c_2, c_3)$, c_1, c_2, c_3 分别代表第 i 个工件的最乐观完工时间(即 C_i 的下界),最可能完工时间和最保守完工时间(即 C_i 的上界).

FFJSP的排序模型描述如下: n 个相互独立的工件在 m 台机器上加工,其中第 i 个工件具有 u_i 道工序,并且每道工序至少可由一台机器加工, J_i 为第 i 个工件, $O_{i,j}$ 为工件 J_i 的第 j 道工序,必须满足如下约束条件:

- 1) 任一时刻同一个工件最多只能在一台机器上进行加工;
- 2) 任一时刻同一台机器最多只能处理一道工序;
- 3) 任一操作在加工期间不可以被中断;
- 4) 同一工件的工序必须要在上一个工序加工完才能进行下一个工序的加工.

要求为每一道工序分配合适的机器,并确定每台机器上工件的加工路径,问题的目标函数为式(5),即最小化最大模糊完工时间,其中: C_i 代表第 i 个工件的模糊完工时间, n 代表工件的数量,

$$C_{\max} = \min(\max_{i=1,2,\dots,n} C_i). \quad (5)$$

本文采用问题的排序模型,该类模型的决策变量为基于操作或工件的排列编码.对于任何一个具体的排列(即问题的一个解),可按一定的解码方式确定并计算各工件在各台机器上的完工时间,进而计算问题的目标函数(譬如本文的 C_{\max}).本文采用的编码和解码方法在第3.1节介绍.

表1给出了一个3台机器3个工件的模糊柔性作业车间工序加工时间的例子, 其中 $O_{1,1}$ 和 $O_{1,2}$ 代表工件 J_1 的加工工序, $O_{2,1}$ 和 $O_{2,2}$ 为工件 J_2 的加工工序, $O_{3,1}$, $O_{3,2}$ 和 $O_{3,3}$ 为工件 J_3 的加工工序, M_1 , M_2 和 M_3 为3台加工机器, 其中*表示该工序无法在这一机器上加工.

表 1 模糊柔性作业车间工序加工时间

Table 1 Processing times of the operations in fuzzy flexible job shop

工件	工序	M_1	M_2	M_3
J_1	$O_{1,1}$	6, 7, 10	8, 10, 11	9, 11, 14
	$O_{2,1}$	1, 2, 4	6, 8, 9	*
J_2	$O_{2,1}$	6, 7, 9	5, 6, 8	7, 10, 12
	$O_{2,2}$	7, 9, 11	5, 9, 12	4, 7, 9
	$O_{3,1}$	18, 21, 24	*	16, 19, 22
J_3	$O_{3,2}$	*	10, 14, 17	7, 10, 11
	$O_{3,3}$	7, 10, 13	4, 6, 9	4, 5, 7

2.2 模糊数的运算

在解决FFJSP时, 因为生成模糊调度和计算目标值时需要TFN进行运算和排序, 因此还需要考虑TFN的相加, 相减, 取大等操作. 对于两个TFN $\tilde{s} = (s_1, s_2, s_3)$ 和 $\tilde{t} = (t_1, t_2, t_3)$, 根据模糊扩张原理, 模糊数的加法运算定义如下:

$$\tilde{s} + \tilde{t} = (s_1 + t_1, s_2 + t_2, s_3 + t_3). \quad (6)$$

同理可得减法运算:

$$\tilde{s} - \tilde{t} = (s_1 - t_1, s_2 - t_2, s_3 - t_3). \quad (7)$$

而比较模糊完工时间则需要使用模糊数的取大操作, 用以对模糊数进行比较排序. 由扩张原理可知, 两个TFN $\tilde{s} = (s_1, s_2, s_3)$ 和 $\tilde{t} = (t_1, t_2, t_3)$ 取大隶属度函数 $\mu_{s \vee t}(z)$ 为

$$\mu_{s \vee t}(z) = \bigvee_{z=x \vee y} (\mu_s(x) \wedge \mu_t(y)). \quad (8)$$

然而在实际运算过程中, 有时使用式(8)无法得到一个规则的TFN隶属度函数, 如图2阴影部分所示.

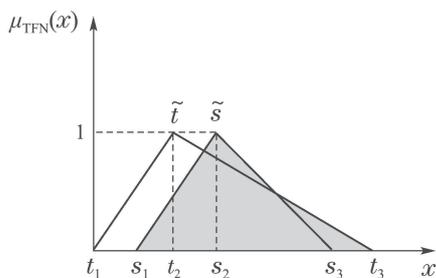


图 2 $\tilde{s} \vee \tilde{t}$ 隶属度函数

Fig. 2 Membership function of $\tilde{s} \vee \tilde{t}$

显然这在建立调度的过程中是不可取的, 当进行模糊数排序时, 排序的结果必须是非此即彼的, 才能计算调度的目标值. 所以在模糊调度中对两个模糊数

进行比较时, 通常取其中的一个作为近似最大值, 根据Lei^[9]的定义, 对于两个TFN $\tilde{s} = (s_1, s_2, s_3)$ 和 $\tilde{t} = (t_1, t_2, t_3)$, 近似取大操作的误差(下文简称取大近似误差)等于

$$\frac{S\{[d(\mu_{s \vee t}(z)) - d(\mu'_{s \vee t}(z))]\}}{S\{d(\mu_{s \vee t}(z))\}} \cup \frac{S\{[d(\mu'_{s \vee t}(z)) - d(\mu_{s \vee t}(z))]\}}{S\{d(\mu_{s \vee t}(z))\}},$$

其中: $d[\mu_{s \vee t}(z)]$ 为取大隶属度函数与x轴围成的图形, $d[\mu'_{s \vee t}(z)]$ 为近似最大值的隶属度函数与x轴围成的图形, 运算符“-”为计算两个图形差集, $S\{i\}$ 为图形*i*的面积. 显然, 近似最大值的取大近似误差越小, 它就越接近真实的最大值. 针对TFN的排序问题, Sakawa等^[8]提出一种排序准则(下文简称为Sakawa准则)可以对模糊数进行比较, 已被广泛应用于模糊调度的研究中, 该准则具体如下:

准则 1 计算 $Z_1(x) = \frac{x_1 + 2x_2 + x_3}{4}$, 其中 x_1 是最乐观值, x_2 是最可能值, x_3 是最保守值. 将两个TFN的 Z_1 作为排序的首要依据.

准则 2 当两个TFN的 Z_1 相等时, 令 $Z_2 = x_2$, 并使用它作为次要依据.

准则 3 当 Z_1 和 Z_2 都相等时, 令 $Z_3 = x_3 - x_1$, 使用它进行判断.

根据以上3条准则对两个TFN进行比较. 例如 \tilde{t} 的 $Z_1 > \tilde{s}$ 的 Z_1 , 记作 $\tilde{t} > \tilde{s}$. 进而, 两个模糊数的近似取大操作可以定义如下: 若 $\tilde{t} > \tilde{s}$, 则 $\tilde{t} \vee \tilde{s} = \tilde{t}$; 否则 $\tilde{t} \vee \tilde{s} = \tilde{s}$. 使用Sakawa准则在部分情况下可准确分辨出较大的模糊数. 但是在有些取大隶属度函数为不规则多边形的情况下, 该准则依旧无法正确的进行判断. 本节将根据几种不同的三角模糊数位置情况, 对Sakawa准则的不足进行分析并给出相应的定理. 同时, 提出一种新的排序准则, 可以更合理地对模糊数进行排序.

对于两个不相等的TFN \tilde{s} 和 \tilde{t} , 图3描述了3种取大隶属度函数为不规则多边形的情况, 从中可以看到, 若两个TFN的对应边交叉, 就会出现不规则的取大隶属度函数图像. 表2给出了本节后续使用的符号及其说明.

表 2 本节用到的符号及其含义说明表

Table 2 Symbols used in this section and their meanings

符号	说明
A_i	图形 <i>i</i> 的面积
a	两个TFN隶属度函数交叉点
x_a	a 点的横坐标
y_a	a 点的纵坐标
s_n	其中 $n=1, 2, 3$, 依次为 \tilde{s} 的最乐观值、最可能值、最保守值
t_n	其中 $n=1, 2, 3$, 依次为 \tilde{t} 的最乐观值、最可能值、最保守值

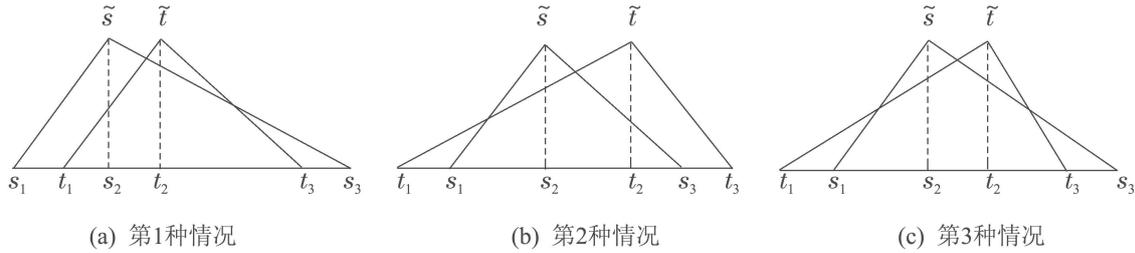


图3 3种取大隶属度函数为不规则多边形情况

Fig. 3 Three cases where each function of taking bigger fuzzy number is an irregular polygon

定理1 针对图3(a)的情况, 取大隶属度函数如图4(a)的阴影部分所示. 若两个模糊数的参数关系满足 $t_1 > s_1, t_2 > s_2, t_3 < s_3$, 且

$$y_a > \frac{1}{2} \times \left(\frac{t_3 - s_1}{s_3 - t_1} + 1 \right),$$

Sakawa准则会错误地判断取大的近似值为 \tilde{t} (即取大

近似误差较大的TFN).

定理2 针对图3(a)的情况. 若两个模糊数的参数关系满足 $t_1 > s_1, t_2 > s_2, t_3 < s_3$, 且 Sakawa 准则判断取大的近似值为 \tilde{s} 时, 不会选择出错.

定理1-2的证明见附录.

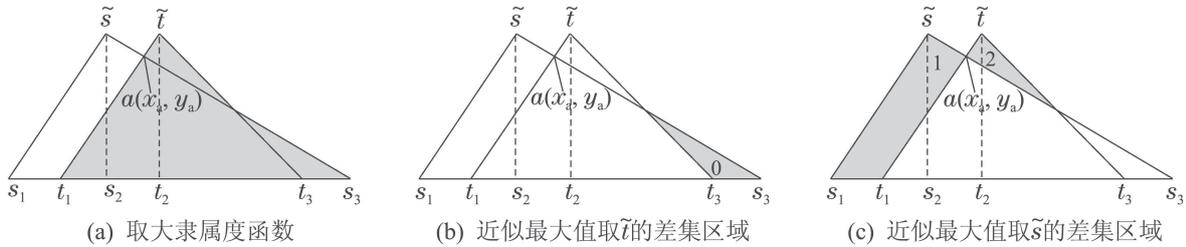


图4 图3(a)情况下的取大隶属度函数和近似取大隶属度函数比较

Fig. 4 Comparison of membership function and approximate membership functions of taking bigger fuzzy number in the case of Fig. 3(a)

推论1 根据定理1, 对于图3(b)的情况. 若两个模糊数的参数关系满足 $t_1 < s_1, t_2 > s_2, t_3 > s_3$, 且 $y_a > \frac{1}{2} \times \left(\frac{t_3 - s_1}{s_3 - t_1} + 1 \right)$ 时, Sakawa准则会错误地判断取大的近似值为 \tilde{t} (即取大近似误差较大的TFN).

推论1的证明过程与定理1类似, 这里略过.

推论2 根据定理2, 若两个模糊数的参数关系满足 $t_1 < s_1, t_2 > s_2, t_3 > s_3$, 且 Sakawa 准则判断取大的近似值为 \tilde{s} 时, 不会判断出错.

所需满足的条件和图3(a)的情况相同, 证明过程与定理2类似, 这里略过. 由于 $s_3 = t_3$ 是这一情况下的特例, 可得相同的结论这里不再分析.

推论3 根据定理1, 对于图3(c)的情况. 若两个模糊数的参数关系满足 $t_1 < s_1, t_2 > s_2, t_3 < s_3$, 且 $y_a > \frac{1}{2} \times \left(\frac{t_3 - s_1}{s_3 - t_1} + 1 \right)$ 时, Sakawa准则会错误地判断取大的近似值为 \tilde{t} (即取大近似误差较大的TFN).

推论3的证明过程与定理1类似, 这里略过.

推论4 根据定理2, 若两个模糊数的参数关系

满足 $t_1 < s_1, t_2 > s_2, t_3 < s_3$, 且 Sakawa 准则判断取大的近似值为 \tilde{s} 时, 不会判断出错.

所需满足的条件和图3(a)的情况相同, 证明过程与定理2类似, 这里略过.

通过实验分析发现, Sakawa准则主要依靠准则1进行判断, 且准则1和准则2都认为模糊数的核较重要, 因此, 当某个模糊数的模糊度较高时, 会被核所误导, 判断出错. 基于上述证明和分析, 本文提出一种新的排序准则, 对于两个TFN, 即 $\tilde{s} = (s_1, s_2, s_3)$ 和 $\tilde{t} = (t_1, t_2, t_3)$, 具体如下所示:

准则1 若两个TFN有对应边交叉时, 使用

$$L_0 = \left(\frac{1}{2} - y_a \right) \times (s_3 - t_1) + \frac{1}{2} (t_3 - s_1)$$

判断(即两者取大近似误差的差值, 推导过程见附录式(A3)), L_0 大于0则 $\tilde{t} > \tilde{s}$, 小于则 $\tilde{t} < \tilde{s}$, 其中 y_a 等于两个隶属度函数的交点的 a 纵坐标, 这里是假设 \tilde{t} 的核大于 \tilde{s} , 如果小于, 只需将公式中的 t 和 s 互换即可. 若 L_0 等于0或没有对应边交叉则使用TFN的均值 $L_1 = \frac{x_1 + x_2 + x_3}{3}$ 进行判断, 如 \tilde{t} 的 L_1 大于 \tilde{s} 的

L_1 , 则 $\tilde{t} > \tilde{s}$.

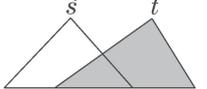
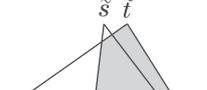
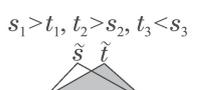
准则 2 当均值相等时, 则使用 $L_2 = x_3 - x_1$ 判断, 如 \tilde{t} 的 L_2 大于 \tilde{s} 的 L_2 , 则 $\tilde{t} < \tilde{s}$.

该准则直接利用取大近似误差比较, 在取大隶属度函数为不规则图形时, 可保证判断的准确性, 也更为直观. 而当取大隶属度函数为正规TFN时, 使用均值判断大小, 可在保证准确性的前提下, 减少计算量. 当两个TFN的核相等, 且 $s_1 > t_1, s_3 < t_3$, 或 $s_1 < t_1, s_3 > t_3$ 时会出现均值相等无法判断的情况, 但此时两个模糊数的取大近似误差都相等, 所以使用准则 L_2 , 即模糊数的模糊度来判断, 模糊度

越小模糊数越大. 使用表3中随机生成的4种不同位置情况的TFN算例验证本文所提准则的有效性, 每种情况5组. 表3中计算取大近似误差时TFN的隶属度最大为1, 表中带*的两列代表使用本文所提取大准则的取大结果和相应的取大近似误差, 不带*的两列则是使用Sakawa准则, 粗体代表较小的取大近似误差. 由表3可知, Sakawa准则在TFN重叠时, 会出现判断出错, 而本文所提出的准则在几种情况下均选择到了取大近似误差较小的那一个模糊数. 显然, 误差越小, 模糊完工时间的计算和比较就越准确, 因此新准则在模糊调度方面比Sakawa准则更为合理.

表 3 不同情况下模糊数取大结果对比

Table 3 Comparison of the results of taking bigger fuzzy number in difference cases

位置情况	模糊数	取大结果	取大近似误差	取大结果*	取大近似误差*
$t_1 \geq s_1, t_2 \geq s_2, t_3 \geq s_3$ 	$\tilde{s}(1, 3, 6), \tilde{t}(2, 4, 7)$	\tilde{t}	0	\tilde{t}	0
	$\tilde{s}(4, 6, 9), \tilde{t}(5, 7, 12)$	\tilde{t}	0	\tilde{t}	0
	$\tilde{s}(2, 7, 10), \tilde{t}(4, 8, 13)$	\tilde{t}	0	\tilde{t}	0
	$\tilde{s}(11, 14, 17), \tilde{t}(12, 16, 19)$	\tilde{t}	0	\tilde{t}	0
	$\tilde{s}(13, 19, 21), \tilde{t}(14, 20, 21)$	\tilde{t}	0	\tilde{t}	0
$s_1 > t_1, t_2 > s_2, t_3 > s_3$ 	$\tilde{s}(8, 9, 11), \tilde{t}(5, 10, 12)$	\tilde{t}	0.47	\tilde{s}	0.41
	$\tilde{s}(7, 8, 12), \tilde{t}(1, 11, 13)$	\tilde{t}	0.5	\tilde{s}	0.46
	$\tilde{s}(4, 6, 11), \tilde{t}(1, 7, 12)$	\tilde{t}	0.26	\tilde{s}	0.24
	$\tilde{s}(13, 16, 19), \tilde{t}(8, 17, 21)$	\tilde{t}	0.45	\tilde{s}	0.34
	$\tilde{s}(5, 6, 11), \tilde{t}(2, 7, 16)$	\tilde{t}	0.19	\tilde{t}	0.19
$s_1 \leq t_1, t_2 > s_2, t_3 < s_3$ 	$\tilde{s}(1, 3, 9), \tilde{t}(1, 4, 7)$	\tilde{t}	0.18	\tilde{s}	0.15
	$\tilde{s}(1, 3, 8), \tilde{t}(2, 4, 5)$	\tilde{t}	0.43	\tilde{s}	0.37
	$\tilde{s}(7, 9, 14), \tilde{t}(8, 11, 13)$	\tilde{t}	0.06	\tilde{t}	0.06
	$\tilde{s}(3, 6, 11), \tilde{t}(4, 7, 8)$	\tilde{t}	0.36	\tilde{s}	0.03
	$\tilde{s}(2, 5, 8), \tilde{t}(3, 6, 7)$	\tilde{t}	0.11	\tilde{t}	0.11
$s_1 > t_1, t_2 > s_2, t_3 < s_3$ 	$\tilde{s}(2, 7, 10), \tilde{t}(1, 8, 9)$	\tilde{t}	0.13	\tilde{s}	0.1
	$\tilde{s}(5, 8, 11), \tilde{t}(4, 9, 10)$	\tilde{t}	0.17	\tilde{s}	0.13
	$\tilde{s}(5, 6, 11), \tilde{t}(3, 8, 9)$	\tilde{t}	0.33	\tilde{s}	0.2
	$\tilde{s}(13, 14, 19), \tilde{t}(10, 15, 17)$	\tilde{s}	0.06	\tilde{s}	0.06
	$\tilde{s}(35, 42, 59), \tilde{t}(30, 45, 58)$	\tilde{t}	0.13	\tilde{s}	0.11

3 混合超启发式遗传算法

3.1 编码与解码

在HHGA中, 首先在高层策略域利用遗传算法对6种操作的组合顺序(每种顺序为1个高层种群个体)进行优化, 从而得到高层策略域新个体; 然后, 在低层问题域将每个高层策略域个体作为独立的启发式算法, 用于对低层相应个体进行变邻域局部搜索, 同时加入模拟退火机制来避免低层搜索陷入局部较小, 从而实现多种启发式算法的并行搜索. 图5为HHGA两层结构的示意图.

对于高层策略域, 遗传算法的种群中每个个体

都由策略集中的6种低层启发式操作构成, 个体中每一位染色体都是一种简单的启发式操作, 染色体的长度等于策略集中低层启发式操作的个数, 且每条染色体中允许出现相同的低层启发式操作. 解码该染色体时, 对一个低层问题域中的解, 按从左到右的顺序依次执行染色体中不同基因位对应的低层启发式操作, 每执行一种低层启发式操作, 就将得到的新解与旧解进行对比, 若新解的适应值(即目标函数值)小于旧解的适应值, 则用新解替换旧解且不再执行剩余的低层启发式操作, 否则继续执行. 这可实现对低层解的变邻域局部搜索. 高层策略域中

每个个体的适应值等于用它更新问题的解后, 这个解的适应值, 图6为一个高层策略域个体的示意图.

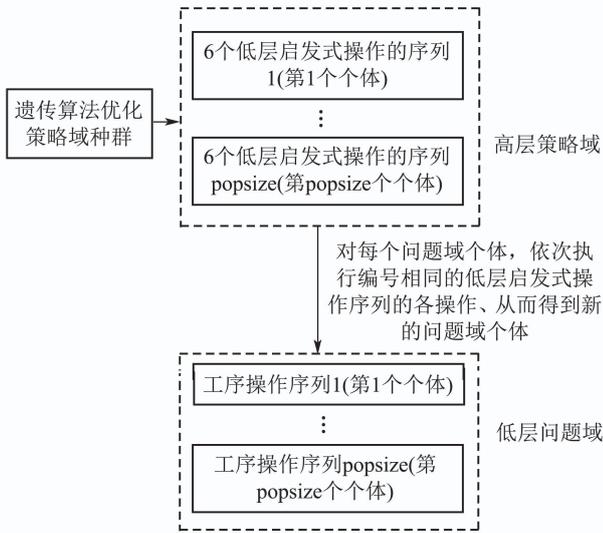


图5 HHGA的两层结构

Fig. 5 Two-layer structure of HHGA

更新操作					
1	3	3	6	2	4
随机交叉	后项插入	后项插入	截取	前项插入	相邻交叉

图6 高层策略域的个体示意图

Fig. 6 Schematic diagram of an individual in the upper layer

对于低层问题域, 每个个体就是原问题的一种工序排列序列, 即原问题的解, 高层策略域的一种新算法对应更新一个解, 因此低层问题域的种群大小与高层策略域相同. 柔性作业车间调度问题和作业车间调度问题相比, 不同主要在于每个工序可以在多个机器上进行加工, 总工序数

$$U = \sum_{i=1}^n u_i,$$

其中 u_i 为第 i 个工件的工序数. 该问题的工序排列编码长度为 U , 可表示为 (p_1, p_2, \dots, p_u) , $p_u \in \{1, 2, \dots, n\}$, n 为工件数, 如果 $p_u = 3$, 则3在序中总共要出现 u_3 次, 如果3是第3次出现, 则代表 p_u 对应工件 J_3 的第3道工序 $O_{3,3}$, 使用表1的数据可得一个编码个体示例为 $(3, 1, 3, 2, 1, 2, 3)$. 由于柔性加工还需选择合适的机器, 所以在解码时采用贪婪策略, 依次将工序排列序列中的每一个工序放在能加工该工序的所有机器上进行活动化解码^[28], 然后选择能让当前最大模糊完工时间最短的那台机器进行加工, 形成具体调度. 这种方法与文献[11]所用的两阶段编码解码相比, 编码序列更短, 可在选择较优机器

的同时充分利用每台机器上的空闲时间间隔, 减少加工时间的浪费, 进而提高解的质量.

下面以上述的编码个体为例阐述该解码方法的具体过程, 解码时按照顺序先加工 $O_{3,1}$, 从表1可知 $O_{3,1}$ 可在 M_1 和 M_3 上加工, 此时若在 M_3 上加工, 则当前最大模糊完工时间为 $(16, 19, 22)$, 比在 M_1 上加工得到的当前最大模糊完工时间短, 因此选择在 M_3 上加工 $O_{3,1}$. 然后加工 $O_{1,1}$, 该工序3台机器均可加工, 若在 M_1 上加工, 由于 M_1 上没有空闲时间间隔, 所以在该机器上加工得到的当前最大模糊完工时间为 $(6, 7, 10)$, 同理可得若在 M_2, M_3 上加工当前最大模糊完工时间分别为 $(8, 10, 11)$ 和 $(25, 30, 36)$, 因此选择在 M_1 上加工 $O_{1,1}$. 后面的解码过程以此类推, 这里不再赘述, 最终得到的模糊甘特图如图7所示, 其中下三角为模糊开始加工时间, 上三角为模糊完工时间.

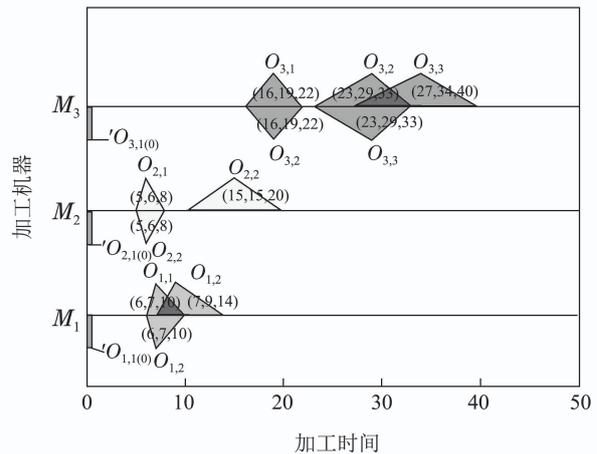


图7 贪婪策略下活动化解码得到的模糊甘特图

Fig. 7 Fuzzy Gantt chart obtained by the active decoding with greedy strategy

3.2 自适应变异算子的遗传算法

遗传算法(genetic algorithm, GA)是一种借鉴生物界适者生存、优胜劣汰机制的元启发式算法, 具有算法简单、应用范围广和全局搜索能力较强等特点. 该算法主要包括3个操作: 选择、交叉和变异.

选择操作采用轮盘赌的方法, 根据上一代每个个体的适应值建立轮盘, 由于本问题所针对的是最小化目标, 所以适应值越小的精英个体被选中的概率越大, 选中的个体被放入到交叉池中, 每次迭代时不断重复上述选择操作, 直到交叉池中的个体数达到两倍的种群数为止.

交叉操作则是从交叉池一半的个体中随机选出一个作为父代1, 然后再从另一半中随机选出一个作为父代2, 采用两点交叉, 随机产生两个位置, 将父

代1中两个位置之外的染色体放入实验个体 X_{test} 的对应位置中, 再在父代2中从头检索每一个染色体, 将 X_{test} 中缺少的染色体根据父代2中的顺序补充进 X_{test} 中. 交叉操作是否进行由交叉算子(crossover operator, CO)控制, 具体操作如式(9), 式中: X_{new} 为下一代的新个体, X_{old} 为上一代的老个体, random为一个0到1之间的实数.

$$X_{\text{new}} = \begin{cases} X_{\text{old}}, & \text{random} > \text{CO}, \\ X_{\text{test}}, & \text{random} < \text{CO}. \end{cases} \quad (9)$$

Leung等^[29]通过详细的分析表明种群多样性的降低直接导致了遗传算法的“早熟收敛”, 而高效的变异操作可以帮助算法提高种群的多样性, 本文采用随机突变的策略, 当进行变异时随意位置会变为一个随机的合法染色体. 变异操作由自适应变异算子(mutation operator, MO)控制, 当产生的一个0到1之间的随机值小于MO时, 进行变异. 自适应变异算子由式(10)计算:

$$\text{MO} = 0.1 \times \frac{\alpha \times t}{\text{MaxIters}}, \quad (10)$$

式中: t 代表当前的迭代次数; MaxIters代表总的迭代次数; α 代表变化速率, 其越大则变异概率增大的速率也就越大. 这里随着迭代次数的增加, 变异的概率会越来越大, 原因在于当算法迭代到后期时, 种群相似度过高趋于收敛, 必须加大扰动的频率才有可能帮助算法跳出局部最优.

3.3 低层启发式操作

由低层启发式操作组成的新算法用于搜索解空间, 它的好坏决定了搜索能力和解的质量, 但是在设计低层启发式操作时不宜过于复杂, 如果复杂度较高, 会大大降低算法的运行效率, 使得求解时间太长. 本文低层启发式操作的策略集包括6种经典的邻域变换操作, 具体如下:

LLH₁: 随机交叉操作, 从工序排列序列中随机选取两位进行交换;

LLH₂: 前项插入操作, 随机从工序排列序列中选择一个位置, 将随机选择的一个工序插入到这个位置之前;

LLH₃: 后项插入操作, 随机从工序排列序列中选择一个位置, 在这个位置之后插入一个随机选择的工序;

LLH₄: 相邻交叉操作, 随机选择一个位置, 将它前一个或后一个工序与它交换;

LLH₅: 逆序操作, 随机在工序排列序列中选择3到4位连续的子序列, 将它们逆序排列;

LLH₆: 截取操作, 随机选择1到5位连续的子序

列, 将它们插入到工序排列序列前端.

为了进一步提高算法中每个低层启发式操作跳出局部最优的能力, 本文在算法之中嵌入了模拟退火机制. 具体而言, 在算法之中, 设置冷却控制表, 每次迭代时根据冷却系数 η 来降温, 温度变化式(11)lk为

$$T_0 = T \times \eta, \quad (11)$$

式中: T_0 是下一次迭代所用的温度, T 是当前的温度, 温度通过如式(12)来影响低层启发式操作:

$$P = \exp\left[-\frac{\Delta f}{T}\right], \quad (12)$$

其中 Δf 为每次的新解与旧解适应值之差. 由于本文所针对的问题目标是模糊数, 这里使用两个模糊数的均值来去模糊化, 得到一个实数. 迭代时, 每个新算法中的每种低层启发式操作, 都要更新对应的解 i_{number} 次, 每次得到的新解若优于旧解, 则将旧解替换, 否则随机生成一个0到1之间的实数 r , 若 r 小于概率 P 则接受这个差解, 将之替换旧解. 可见在迭代后期每一个低层启发式操作都会退化为贪婪搜索, 从而使算法最终收敛到一个全局较优值.

3.4 算法流程

根据上述算法描述, 整个算法的流程图如图8所示, 具体可描述如下.

步骤1 初始化高层策略域和低层问题域种群. 第1次迭代时, 随机生成长度为6的高层策略域个体, 并随机生成长度为总工序数 U 的工序排列(即底层问题的解或个体), 高层和低层对应的种群大小均为popsize.

步骤2 设置模拟退火初始温度 $T_S = 200$.

步骤3 评价两个种群. 用每一组低层启发式操作来更新问题的解, 根据目标函数计算得到解的适应值. 高层策略域的个体与对应的解适应值相同, 即某个解的好坏也间接代表了这种搜索方式的好坏.

步骤4 使用轮盘赌选出策略种群中的精英解, 进行交叉变异.

步骤5 使用新产生的策略个体中的6种低层启发式操作依次更新问题的解, 每次更新时若产生的新解适应值优于旧解, 则替换旧解, 并不再使用后续低层启发式操作进行搜索, 否则根据当前温度以一定概率接受差解.

步骤6 如果接受了当前低层启发式操作得到的较差解, 则判断是否达到该操作的内循环次数 $i_{\text{number}} = 5$, 没达到则继续使用该启发式操作更新

问题的解, 否则进行下一步.

步骤 7 分别对两个域中的新旧种群进行保优操作, 将每个域的精英解放入对应种群.

步骤 8 根据降温系数降温, 并判断迭代次数是否达到总代数MaxIters, 若没有则转到步骤4, 否则终止循环.

在混合超启发式遗传算法中, 遗传算法的交叉算子CO, 自适应变异算子MO中的变化速率 α 和模拟退火机制中的降温系数 η 为3个主要的参数, 为了考察参数设置对算法效果的影响, 采用文献[15]中的实验设计方法, 各参数设置水平如表4所示.

表 4 主要参数与水平

Table 4 Main parameters and levels

主要参数	水平设置			
	1	2	3	4
交叉算子CO	0.6	0.7	0.8	0.9
变化速率 α	20	15	12	10
降温系数 η	0.85	0.9	0.95	0.98

针对各参数的水平设置, 使用正交表 $L_{16}(3^4)$ 进行实验, 共有16组不同参数不同水平的组合. 使用算例3进行实验, 对每种不同参数组合的HHGA独立运行20次, 每组得到20个模糊最大完工时间, 取它们的平均值作为平均响应值ARV, 平均响应值ARV 越小意味着这组参数下的算法性能越强, 参数设置的正交表如表5所示.

表 5 参数设置的正交表

Table 5 Orthogonal table of parameter settings

组合编号	水平			ARV
	CO	α	η	
1	1	1	1	29.6, 43.8, 57.9
2	1	2	2	28.9, 44.0, 58.3
3	1	3	3	30.2, 44.0, 57.8
4	1	4	4	31.1, 43.7, 57.7
5	2	1	2	30.1, 43.9, 57.4
6	2	2	1	29.2, 43.7, 58.3
7	2	3	4	30.3, 43.9, 58.0
8	2	4	3	30.9, 43.6, 57.6
9	3	1	3	30.3, 43.9, 57.5
10	3	2	4	30.0, 43.7, 58.3
11	3	3	1	29.4, 43.8, 58.1
12	3	4	2	30.1, 43.7, 57.6
13	4	1	4	29.9, 44.1, 58.0
14	4	2	3	30.5, 43.6, 57.8
15	4	3	2	30.9, 43.9, 56.9
16	4	4	1	30.3, 43.5, 57.4

根据上述实验, 得到3个主要参数的平均响应值和影响力等级如表6, 其中影响力一栏越小代表该参数的影响力排名越高, 括号内的值为本文所提排序准则下, 该参数不同水平的平均响应值排名, 平均

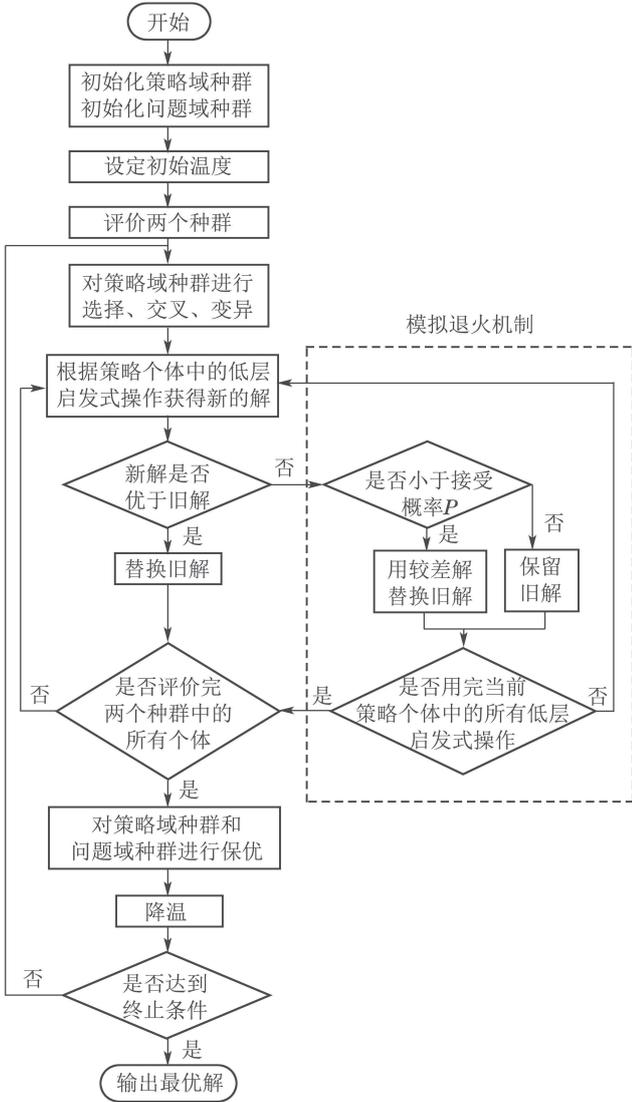


图 8 混合超启发式遗传算法流程图

Fig. 8 Flow chart of hybrid hyper-heuristic genetic algorithm

4 实验分析与比较

4.1 参数设定

第4.1节和第4.2节的实验均使用文献[11-12]中所提供的5种FFJSP测试算例*. 其中: 算例1-2含10个工件、10台机器和40道工序; 算例3-4含10个工件、10台机器和50道工序; 算例5含15个工件、10台机器和80道工序.

*本文仿真实验所用测试算例及案例分析数据可在<https://pan.baidu.com/s/16OjUp8WncOUAIEY40OIYcg>下载, 若无法下载可联系通讯作者.

响应值越小排名越高, 根据表6, 3个参数的变化对算法性能的影响分析如下:

交叉算子CO影响着信息的交换频率, 随着CO的增大, 策略个体之间交流的次数就多, 使得全局的搜索能力变强;

自适应变异算子MO中的变化速率 α 过大的话将导致高层策略域的遗传算法过于发散, 较为适合的策略还未完全发挥作用就被改变, 使得对问题解空间的搜索能力下降, 但是过小又会导致遗传算法早熟, 陷入局部最优后无法及时更换策略进行搜索, 因此应取适中值;

模拟退火机制中的降温系数 η 对优化结果影响最大, 原因在于降温系数作用于低层启发式操作, 而低层启发式操作的效率直接影响着算法对问题解空间的搜索能力, 降温系数过大将导致低层启发式操作过多的接受劣解, 前期虽然有助于跳出局部最优, 但是到后期算法收敛性差, 若过小则会导致算法过早退化到贪婪搜索, 大大降低局部搜索的能力.

在之后的实验中3个主要参数设置如下: CO = 0.9, $\alpha = 15$, $\eta = 0.85$.

表 6 各参数不同水平下的平均响应值和影响力

Table 6 Average response values and influence at different levels of each parameter

水平	主要参数		
	CO	α	η
1	29.9, 43.9, 57.9(4)	30.0, 43.9, 57.7(3)	29.6, 43.7, 57.9(1)
2	30.1, 43.8, 57.8(3)	29.7, 43.8, 58.1 (1)	30.0, 43.9, 57.6(2)
3	30.0, 43.8, 57.9(2)	30.2, 43.9, 57.7(4)	30.5, 43.8, 57.6(3)
4	30.4, 43.8, 57.5(1)	30.6, 43.6, 57.6(2)	30.3, 43.9, 58.0(4)
极差影响力	-0.5, 0.1, 0.4 3	-0.4, 0.1, 0.5 2	0.7, 0.2, 0.1 1

4.2 仿真结果比较与分析

为验证所提算法中嵌入模拟退火机制的有效性, 分别使用不加入模拟退火机制的HHGA(HHGA without simulated annealing mechanism, HHGA-noSA)和完整的HHGA进行仿真实验, 每种算法其他参数均保持一致, 每种规模的算例独立运行20次, 每种算法在20次中的最优值、最差值和平均值如表7所示, 其中粗体为本文所提排序准则下较小的值.

根据表7可知, 使用模拟退火机制的HHGA在不同规模的算例中, 性能均优于或等于HHGA-noSA,

这也验证了这一机制能进一步改善HHGA的设想, 模拟退火机制强大的突变性可以有效地避免每个低层启发式操作陷入局部最优, 增强了低层启发式操作的局部搜索能力, 从而提升HHGA的整体性能.

表 7 模拟退火机制性能验证

Table 7 Performance verification of the simulated annealing mechanism

算例	算法	平均值	最好值	最差值
1	HHGA	20.8, 28, 37.2	21, 28, 37	19, 28, 39
	HHGA-noSA	20.3, 28, 37.7	21, 28, 37	19, 28, 39
2	HHGA	35, 42, 59	35, 42, 59	35, 42, 59
	HHGA-noSA	35, 42, 59	35, 42, 59	35, 42, 59
3	HHGA	30.5, 44, 57.4	29, 44, 58	31, 45, 57
	HHGA-noSA	31, 43.9, 57.8	29, 44, 58	33, 43, 59
4	HHGA	22.2, 34, 48	23, 33, 47	22, 35, 48
	HHGA-noSA	22.3, 34.1, 47.9	20, 34, 49	23, 35, 48
5	HHGA	37.3, 54.3, 73.5	37, 53, 73	37, 56, 75
	HHGA-noSA	36.3, 54.9, 76.4	34, 54, 75	36, 56, 82

为了测试提出的HHGA的性能, 利用5种算例进行了多组实验. 然后将实验结果与现有的5种较强算法进行比较, 包括CGA^[12], TLBO^[13], HBBO^[14], EDA^[15]和HABC^[16]. 对比结果如表8所示, 由于另外5种算法均使用Sakawa准则进行TFN排序, 为了保证比较makespan时的公平性, 括号中NR对应在HHGA计算过程中使用本文提出的排序准则(new rule, NR)得到的结果, OR对应在HHGA计算过程中使用Sakawa排序准则(old rule, OR)得到的结果, 表中的粗体为使用Sakawa排序准则判断的几种算法中最小makespan, 其中, 上标为“#”则代表使用本文排序准则判断的几种算法中最小makespan. 其中包括算法独立运行20次求解得到的最好值, 平均值和最差值, 这5种算法的结果均来源于文献[13-14], 其中HBBO, TLBO, ABC评价次数为100000次, 另外两种为150000次. HHGA采用英特尔核心2.6 GHz处理器, 4 G的RAM, 编程语言为Pascal, 编程环境为Delphi2010. 实验中种群数为100, 迭代50次, 由于算法中使用策略个体对应的低层启发式操作搜索时, 只要找到更优解就不再进行该染色体中后续低层启发式操作, 所以评价次数最差情况需要150000次. 从表8可以看到, 对于5个不同规模的算例, HHGA除了部分与HBBO接近外, 无论是本文的排序规则下还是Sakawa排序规则下, 均强于其他算法. 具体而言, 算例1-2中HBBO和HHGA均找到了同一个解, 算例2中由于评价准则有所差别, 导致最终的模糊完工时间不同, 在本文的排序准则下, (35, 42,

59)大于(30, 45, 58), 所以认为(35, 42, 59)才应是最终模糊完工时间, 经过计算也可以验证(30, 45, 58)和(35, 42, 59)比较时, 后者取大的近似误差更小, 后者作为最终模糊完工时间更为准确; 而算例3-5中HHGA均强于其他5种算法, 处于领先地位, 并且这

种优势在问题规模扩大时更为明显, 可以看出HHGA在FFJSP中有很强的竞争力和有效性, 而在HHGA中使用本文的排序准则得到的结果也更加合理. 图9给出了本文所提排序准则下, HHGA求解规模最大的算例5所得的模糊甘特图.

表8 两种规则下HHGA与5种算法的对比结果

Table 8 Comparison of HHGA and five algorithms under two rules

算例	算法	平均值	最好值	最差值
1(10 × 10)	HHGA (NR)	21.0, 28.0, 37.0[#]	21, 28, 37[#]	21, 28, 37[#]
	HHGA (OR)	21.0, 28.0, 37.0[#]	21, 28, 37[#]	21, 28, 37[#]
	HBBO	20.8, 28.0, 37.2	21, 28, 37[#]	19, 28, 39
	TLBO	20.3, 29.9, 40.9	19, 28, 39	21, 32, 42
	EDA	20.3, 30.5, 41.6	20, 28, 40	22, 32, 43
	HABC	21.0, 32.0, 43.6	19, 30, 43	23, 33, 46
	CGA	23.1, 33.1, 43.4	21, 29, 41	25, 37, 47
2(10 × 10)	HHGA (NR)	35.0, 42.0, 59.0	35, 42, 59	35, 42, 59
	HHGA (OR)	30.0, 45.0, 58.0 [#]	30, 45, 58 [#]	30, 45, 58 [#]
	HBBO	30.0, 45.0, 58.0 [#]	30, 45, 58 [#]	30, 45, 58 [#]
	TLBO	32.6, 46.4, 58.5	30, 45, 58 [#]	36, 49, 63
	EDA	33.7, 46.9, 57.9	32, 46, 57	34, 48, 58
	HABC	33.0, 47.8, 62.2	33, 46, 58	36, 48, 65
	CGA	35.0, 47.1, 60.6	32, 47, 57	38, 49, 64
3(10 × 10)	HHGA (NR)	29.7, 44.1, 57.7 [#]	30, 43, 57[#]	31, 44, 57[#]
	HHGA (OR)	29.8, 43.6, 58.6	30, 42, 60	29, 44, 60
	HBBO	30.2, 43.8, 58.1	30, 42, 60	31, 45, 57
	TLBO	31.5, 46.7, 62.2	30, 45, 60	33, 50, 70
	EDA	32.8, 47.2, 62.9	31, 46, 60	34, 49, 66
	HABC	33.9, 50.8, 67.3	33, 47, 64	36, 54, 70
	CGA	36.4, 50.8, 66.0	34, 47, 63	38, 53, 71
4(10 × 10)	HHGA (NR)	22.6, 33.6, 47.7[#]	20, 34, 48 [#]	24, 34, 48
	HHGA (OR)	22.2, 34.0, 48.0	24, 33, 47	21, 34, 50[#]
	HBBO	22.8, 34.0, 47.9	24, 33, 47	23, 35, 48
	TLBO	24.9, 36.5, 50.8	21, 36, 50	26, 40, 57
	EDA	24.8, 37.2, 51.9	21, 36, 50	24, 39, 57
	HABC	25.5, 40.0, 56.3	23, 38, 53	25, 44, 59
	CGA	27.4, 40.4, 55.0	26, 37, 51	29, 42, 59
5(15 × 10)	HHGA (NR)	36.1, 53.3, 73.6	36, 52, 71	35, 55, 76[#]
	HHGA (OR)	35.5, 53.2, 74.1[#]	35, 52, 71[#]	37, 54, 78
	HBBO	37.2, 54.0, 74.3	36, 54, 70	37, 55, 75
	TLBO	36.1, 57.5, 78.2	36, 55, 72	37, 61, 82
	EDA	38.6, 56.9, 78.3	36, 55, 73	40, 60, 81
	HABC	—	—	—
	CGA	47.0, 65.4, 86.0	42, 62, 82	49, 70, 91

4.3 实际案例分析

目前, 已有一些学者考虑到将模糊理论应用到实际JSP生产调度中^[30-31]. 为了验证本文所提出的

模糊排序规则和HHGA在实际柔性JSP中的有效性, 本节将其用于求解云南玉溪某钢铁公司的冷轧厂生产调度问题.

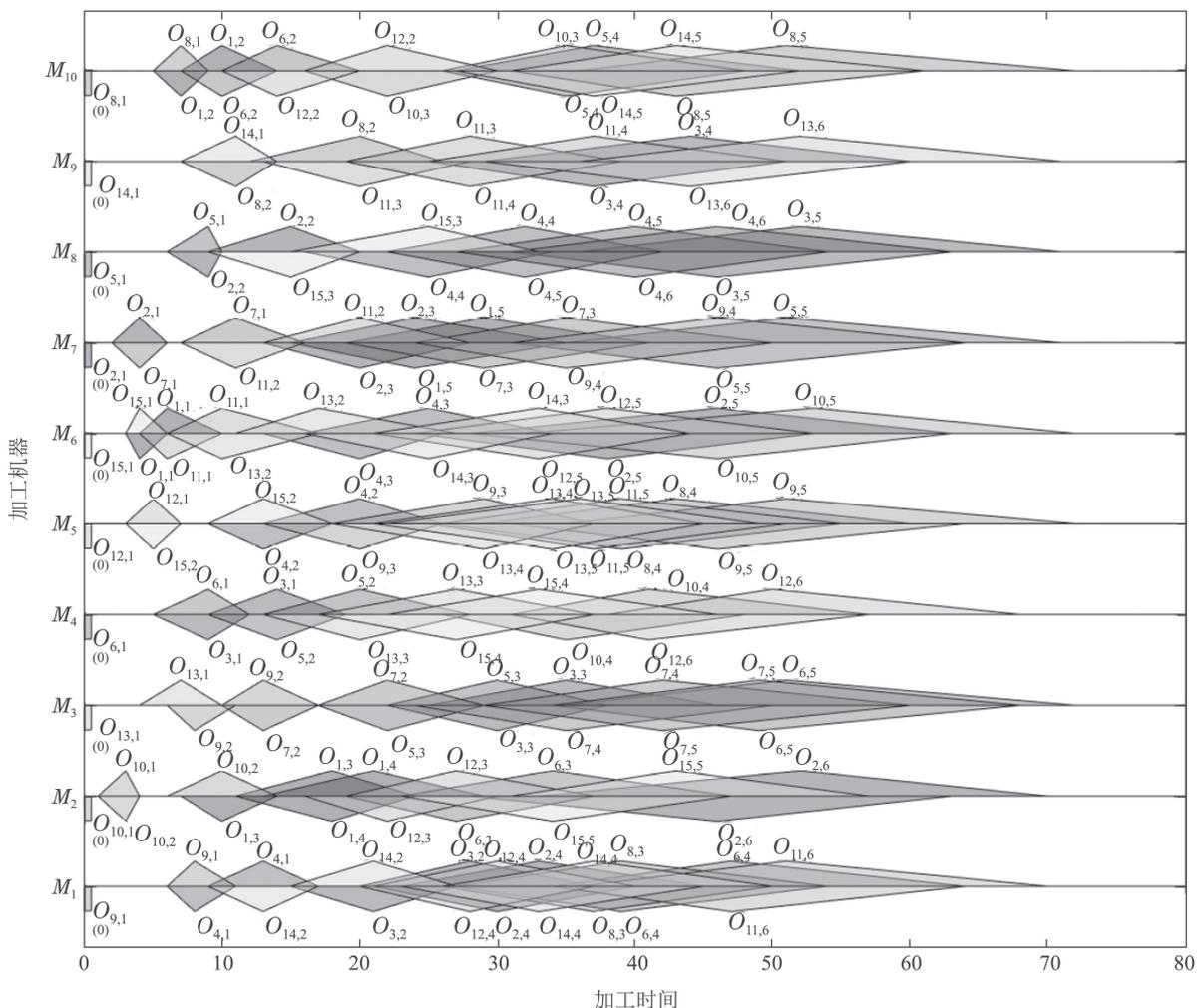


图 9 所提排序规则求解实例5得到的模糊甘特图

Fig. 9 Fuzzy Gantt chart obtained by solving Case 5 via the proposed sorting rules

该厂主要生产冷轧钢板, 其产品包括轧硬板、冷轧板、极薄板、涂镀板, 每种产品的加工过程如表 9所示.

表 9 产品加工流程

Table 9 Product manufacturing process

产品	加工过程
轧硬板	酸洗→冷轧
冷轧板	酸洗→冷轧→脱脂→退火→平整
极薄板	酸洗→冷轧→脱脂→退火→平整→二次冷轧
涂镀板	酸洗→冷轧→脱脂→退火→平整→涂镀→精整

为了满足冷轧钢板各个环节的制造工艺需求, 厂内配备有1250, 1450酸洗机组各一套(编号分别为 M_1, M_2), 1250五连轧机组一套(编号为 M_3), 光亮罩式退火炉两套(编号分别为 M_4, M_5), 黑退火炉两

套(编号分别为 M_6, M_7), 脱脂机组一套(编号为 M_8), 精整机组一套(编号为 M_9), 平整机组两套(编号分别为 M_{10}, M_{11}), 热镀锌机线一套(编号为 M_{12}). 通过对冷轧钢板制造过程的现场调研, 发现在整个制造过程中, 由于存在设备故障、机组定期检修, 工人的熟练度有所不同以及原料品质不完全一样等不确定性因素, 所以每个工序的加工时间都不是一个确定值(譬如每卷原料在酸洗时的速度是由操作工人根据酸液浓度的变化、原料氧化程度的不同而动态调整的, 从而也就导致每次酸洗的时间不是一个定值).

另外需要说明的是, 为了提高生产效率, 冷轧厂采用合同组批的形式来进行生产调度. 所谓合同组批, 就是将未处理的生产合同中具有相同产品种类且同一周提交的合同组合为一个批次, 然后按批次

进行集中加工. 显然, 冷轧过程中每个批次都有不同的加工路径, 且每道工序都有一套或一套以上的机组可加工, 如果采用三角模糊数来表示各批次的加工时间, 将每个批次作为一种工件, 每套机组作为一台机器, 那么此冷轧厂的组批调度问题就是一个典型的FFJSP.

本节使用该冷轧厂某月10个批次的实际加工数据作为测试实例, 用HHGA进行求解. 首先, 考虑到冷轧钢板加工过程的不确定性, 需要先计算出每个工序的模糊加工时间TFN = (t₁, t₂, t₃). 对于每个工序最乐观的加工时间t₁(机组一直都处于正常运行且原材料完全一致的理想状态), 利用文献[32]中的方法进行计算. 而对于每个工序最可能的加工时间t₂则使用机组产能平均效率公式^[31]进行计算:

$$t_2 = \frac{\sum t_{kM}}{\sum W_{kM}} \times W_i, \quad (13)$$

其中: $\sum t_{kM}$ 为k类产品近一个月来在M机组上的总加工时间, $\sum W_{kM}$ 为k类产品近一个月来在M机组上加工的总重量, W_i 为本月第i个批次(该批次为k类产品)的重量. 对于每个工序最保守的加工时间t₃则需考虑各种停机状况(计划内停机和计划外停机), 因此使用式(14)进行合理的预测, 其中l_M为近一个

月来M机组的最长停机时间.

$$t_3 = t_2 + l_M. \quad (14)$$

得到模糊加工时间后, 再使用HHGA求解测试实例, 程序计算过程中均使用本文所提出的模糊排序准则进行模糊完工/加工时间的排序, 运行时间为5 s, 同时, 请厂中有经验的调度员10 min内给出调度方案(该厂调度方案一般由调度员基于某种调度规则得到, 这里是先用最小加工时间规则确定每个工序使用的机组, 然后再使用最大剩余加工时间规则确定各个工序的加工顺序), 两种调度方案如表10所示.

调度方案中的机器一栏中每一个数字, 依次代表每个批次各个工序所使用的加工机组, 例如第1个数字2代表第1个批次的第1个工序在M₂上加工, 第2个数字3代表第1个批次的第2个工序在M₃上加工, 以此类推; 操作一栏为工序的加工顺序, 加工时按从左到右的顺序依次加工, 其中每个数字代表相应的批次, 每个数字第几次出现就是该批次的第几道工序. 表中: R₁为调度员得到的调度方案, R₂为HHGA得到的调度方案, 可以看到R₂明显优于R₁, 说明所提模糊排序规则和HHGA在实际制造过程中具有较好的应用前景.

表 10 调度方案

Table 10 Scheduling scheme

		调度方法																		最小化最大模糊完工时间/h					
R ₁	机器	2	3	2	3	2	3	2	3	8	4	10	2	3	8	5	10	2	3	8	5	11	2	3	(615, 708, 841)
	操作	10	10	10	10	5	9	6	10	7	4	9	5	6	10	9	4	7	9	8	5	2	6	4	
	机器	8	5	10	2	3	8	5	10	3	2	3	8	5	10	12	9	2	3	8	5	10	12	9	
	操作	7	3	5	8	6	9	1	4	7	8	9	8	10	2	8	6	3	4	5	7	1	8	9	
R ₂	机器	2	3	1	3	2	3	1	3	8	7	10	2	3	8	5	10	1	3	8	4	11	2	3	(526, 565, 631)
	操作	10	9	4	7	10	9	8	6	5	5	8	2	1	7	10	9	3	10	9	9	6	7	8	
	机器	8	4	11	1	3	8	6	11	3	1	3	8	7	10	12	9	2	3	8	5	10	12	9	
	操作	4	5	10	10	9	1	5	7	8	8	9	6	7	4	8	5	10	6	2	4	6	4	3	

5 结论

本文提出一种两层结构的混合超启发式遗传算法(HHGA), 用于求解FFJSP, 采用三角模糊数表示工件加工时间, 优化目标为最小化最大模糊完工时间(即makespan). 首先, 通过详细分析现有模糊调度问题中使用的三角模糊数排序准则, 指出文献[8]中的排序准则在某些情况下会不够准确, 进而充分考虑取大操作的近似误差和模糊度, 设计一种更为准确的三角模糊数排序准则, 可合理计算FFJSP和其他各类调度问题解的目标函数值. 其次, 针对现有遗传算法和超启发式算法相结合方法的不足, 即底

层搜索中缺乏有效局部搜索和算法整体容易过早陷入局部极小, 将问题求解算法分为两层, 在高层对6种低层启发式操作(即6种有效邻域操作)的排列进行优化, 使得低层可执行紧凑的变邻域局部搜索, 同时在高层和低层分别设计自适应变异算子和模拟退火机制来避免算法陷入局部极小. 最后, 采用实验设计方法对算法中的关键参数进行了确定, 并通过仿真实验和算法比较验证了所提三角模糊数排序准则和HHGA的有效性.

后续工作将把所设计的排序准则应用于其他模糊调度问题, 并将所提算法扩展用于求解低碳FFJSP

问题.

参考文献:

- [1] XUAN Hua, LI Bing, WANG Xueyuan, et al. Multi-stage dynamic reentrant hybrid flow shop scheduling with transportation considerations. *Control Theory & Applications*, 2018, 35(3): 357 – 366. (轩华, 李冰, 王薛苑, 等. 带运输考虑的多阶段动态可重入混合流水线车间调度. *控制理论与应用*, 2018, 35(3): 357 – 366.)
- [2] ZHANG Hao, ZHU Yunlong, PEI Xiangbo. Scheduling of high-precision copper ingot smelting based on multi-objective root growth algorithm. *Control Theory & Applications*, 2018, 35(1): 121 – 128. (张浩, 朱云龙, 亓祥波. 基于多目标根系生长算法的高精铜锭熔炼作业调度. *控制理论与应用*, 2018, 35(1): 121 – 128.)
- [3] YANG Xiaolin, HU Rong, QIAN Bin, et al. Enhanced distribution estimation algorithm for low-carbon distributed pipeline scheduling. *Control Theory & Applications*, 2019, 36(5): 803 – 815. (杨晓林, 胡蓉, 钱斌, 等. 增强分布估计算法求解低碳分布式流水线调度. *控制理论与应用*, 2019, 36(5): 803 – 815.)
- [4] QIAN B, LI Z C, HU R. A copula-based hybrid estimation of distribution algorithm for m-machine reentrant permutation flow-shop scheduling problem. *Applied Soft Computing*, 2017, 61(6): 921 – 934.
- [5] LI Z C, QIAN B, HU R, et al. An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups. *Knowledge-Based Systems*, 2019, 173(6): 83 – 112.
- [6] LENSTRA J K, RINNOOY KAN A H G, BRUCKER P. Complexity of machine scheduling problems. *Annals of Discrete Machines*, 1977, 1(4): 343 – 362.
- [7] 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.
- [8] SAKAWA M, KUBOTA R. Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *European Journal of Operational Research*, 2000, 120(2): 393 – 407.
- [9] LEI D M. Fuzzy job shop scheduling problem with availability constraints. *Computers & Industrial Engineering*, 2010, 58(4): 610 – 617.
- [10] TSUJIMURA Y, GEN M, KUBOTA E. Solving job-shop scheduling problem with fuzzy processing time using genetic algorithm. *Journal of Japan Society for Fuzzy Theory & Systems*, 2017, 7(5): 1073 – 1083.
- [11] LEI D M. A genetic algorithm for flexible job shop scheduling with fuzzy processing time. *International Journal of Production Research*, 2010, 48(10): 2995 – 3013.
- [12] LEI D M. Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Applied Soft Computing*, 2012, 12(8): 2237 – 2245.
- [13] XU Y, WANG L, WANG S Y, et al. An effective teaching – learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Neurocomputing*, 2015, 148(19): 260 – 268.
- [14] LIN J. A hybrid biogeography-based optimization for the fuzzy flexible job-shop scheduling problem. *Knowledge-Based Systems*, 2015, 78(1): 59 – 74.
- [15] WANG S Y, WANG L, XU Y, et al. An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *International Journal of Production Research*, 2013, 51(12): 3778 – 3793.
- [16] WANG L, ZHOU G, XU Y, et al. A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem. *International Journal of Production Research*, 2013, 51(12): 3593 – 3608.
- [17] GAO K Z, SUGANTHAN P N, PAN Q K, et al. Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion. *Knowledge-Based Systems*, 2016, 109(1): 1 – 16.
- [18] KOULINAS G, KOTSIKAS L, ANAGNOSTOPOULOS K. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. *Information Sciences*, 2014, 277(2): 680 – 693.
- [19] SALCEDO-SANZ S, MATÍAS-ROMÁN J M, Jiménez-Fernández S, et al. An evolutionary-based hyper-heuristic approach for the job-breaker puzzle. *Applied Intelligence*, 2014, 40(3): 404 – 414.
- [20] YAFRANI M E, MARTINS M, WAGNER M, et al. A hyperheuristic approach based on low-level heuristics for the travelling thief problem. *Genetic Programming & Evolvable Machines*, 2018, 19(1/2): 121 – 150.
- [21] RABBANI M, KAZEMI S M, MANAVIZADEH N. Mixed model U-line balancing type-1 problem: A new approach. *Journal of Manufacturing Systems*, 2012, 31(2): 131 – 138.
- [22] RODR J A V, PETROVIC S, SALHI A. A combined metaheuristic with hyper-heuristic approach to the scheduling of the hybrid flow shop with sequence dependent setup times and uniform machines. *Multidisciplinary International Scheduling Conference: Theory and Applications*. Paris, France: MISTA, 2007: 506 – 513.
- [23] PARK J, MEI Y, SU N, et al. Investigating the generality of genetic programming based hyper-heuristic approach to dynamic job shop scheduling with machine breakdown. *Australasian Conference on Artificial Life and Computational Intelligence*. Berlin, German: Springer International Publishing, 2017: 301 – 313.
- [24] YAN J, WU X. A Genetic based hyper-heuristic algorithm for the job shop scheduling problem. *International Conference on Intelligent Human-machine Systems & Cybernetics*. Hangzhou, China: IEEE, 2017: 161 – 164.
- [25] WANG H. An adaptive hyper-heuristics genetic algorithm for stochastic job shop scheduling problem. *Applied Mechanics and Materials*, 2014, 556(6): 3956 – 3960.
- [26] BORTOLAN G, DEGANI R. A review of some methods for ranking fuzzy subsets. *Fuzzy Sets & Systems*, 1985, 15(1): 1 – 19.
- [27] RAN Jingxue. Research on the ranking method of triangular fuzzy numbers. *Journal of The Central University for Nationalities (Journal of Natural Science)*, 2011, 20(4): 37 – 42. (冉静学. 三角模糊数排序方法的研究. *中央民族大学学报(自然科学版)*, 2011, 20(4): 37 – 42.)
- [28] PINEDO M. *Scheduling: Theory, Algorithms and Systems*. New York: Springer, 2008.
- [29] LEUNG Y, GAO Y, XU Z B. Degree of population diversity—a perspective on premature convergence in genetic algorithms and its Markov chain analysis. *IEEE Transactions on Neural Networks*, 1997, 8(5): 1165 – 1176.
- [30] YU Shengping, PANG Xinfu, CHAI Tianyou, et al. Research on production scheduling problem of steelmaking and continuous casting with uncertain processing time. *Control and Decision*, 2009, 24(10): 1467 – 1472. (俞胜平, 庞新富, 柴天佑, 等. 加工时间不确定的炼钢连铸生产调度问题研究. *控制与决策*, 2009, 24(10): 1467 – 1472.)
- [31] ZHAO Jun. *Research and application of production scheduling and optimization algorithm for rolling process*. Dalian: Dalian University of Technology, 2008. (赵珺. 轧钢过程生产调度及其优化算法的研究与应用. 大连: 大连理工大学, 2008.)
- [32] QIN Yining, WANG Wei, WANG Jianjun, et al. Object modeling method for process industry production planning and scheduling system. *Journal of Computer Integrated Manufacturing Systems*, 2006, 12(1): 44 – 49.

(覃一宁, 王伟, 王建军, 等. 流程工业生产计划与调度系统的对象建模方法. 计算机集成制造系统, 2006, 12(1): 44 - 49.)

附录 定理证明

定理1的证明.

证 从图4(b)中可知, 若取大的近似值是 \tilde{t} 则取大近似误差为图形0的面积与取大的隶属度函数面积之比; 若取大的近似值是 \tilde{s} 则取大近似误差为图形1和2的面积比上取大隶属度函数的面积, 如图4(c)所示.

取大的近似值为 \tilde{t} 意味着图4(c)中图形1和2的面积应大于图4(b)中图形0的面积. 两者的面积之差可以通过式(A1)得到, $A_{\Delta s_3 a t_1}$ 为三角形 $s_3 a t_1$ 的面积, $A_{\Delta s s_1 s_3}$ 为TFN \tilde{s} 的隶属度函数面积, $A_{\Delta t t_1 t_3}$ 为TFN \tilde{t} 的隶属度函数面积

$$A_1 + A_2 = (A_{\Delta s s_1 s_3} + A_{\Delta t t_1 t_3}) - 2 \times (A_{\Delta s_3 a t_1} - A_0) - A_0. \quad (A1)$$

将坐标值代入式(A1), 并且隶属度最大值为1, 则可得到

$$(A_1 + A_2) - A_0 = \frac{1}{2} \times (s_3 - s_1 + t_3 - t_1) - y_a \times (s_3 - t_1). \quad (A2)$$

将上式化简可得

$$(A_1 + A_2) - A_0 = \left(\frac{1}{2} - y_a\right) \times (s_3 - t_1) + \frac{1}{2}(t_3 - s_1). \quad (A3)$$

由取大近似误差的定义可知, 两个TFN取大近似误差的差异直接取决于式(A3), 若 $\tilde{t} > \tilde{s}$, 即取大的近似值为 \tilde{t} , 式(A3)等号右边的项应大于0. 但是容易证明, 在使用Sakawa准则判断 $\tilde{t} > \tilde{s}$ 时, 若式(A3)右边的 y_a 大于0.5, 且满足下面不等式:

$$-\left(\frac{1}{2} - y_a\right) \times (s_3 - t_1) > \frac{1}{2} \times (t_3 - s_1), \quad (A4)$$

其中 $s_3 - t_1, t_3 - s_1$ 在这种位置情况下大于0, 化简后可得

$$y_a > \frac{1}{2} \times \left(\frac{t_3 - s_1}{s_3 - t_1} + 1\right). \quad (A5)$$

将导致式(A3)等号右边的项小于0, 这就与之前Sakawa准则的判断相悖, 可见在这种情况下, Sakawa准则将会进行错误的判断. 证毕.

定理2的证明.

证 Sakawa准则判断取大的近似值为 \tilde{s} , 就意味着 $\tilde{t} < \tilde{s}$, 式(A3)右边项应小于0, 但是在 y_a 小于等于 $\frac{1}{2}$ 或 y_a 大于 $\frac{1}{2}$ 且满足不等式(A6)时, Sakawa准则就会判断出错:

$$-\left(\frac{1}{2} - y_a\right) \times (s_3 - t_1) < \frac{1}{2} \times (t_3 - s_1). \quad (A6)$$

同样地, $s_3 - t_1, t_3 - s_1$ 大于0, 化简后可得

$$y_a < \frac{1}{2} \times \left(\frac{t_3 - s_1}{s_3 - t_1} + 1\right), \quad (A7)$$

其中 y_a 可通过将 x_a 代入 \tilde{t} 的隶属度函数并化简得到, 如下式:

$$y_a = \frac{s_3 - t_1}{s_3 - t_1 + t_2 - s_2}. \quad (A8)$$

易知Sakawa准则判断 $\tilde{t} < \tilde{s}$ 需符合 \tilde{t} 的 Z_1 小于 \tilde{s} 的 Z_1 , 即满足

$$\begin{cases} 2 \times (t_2 - s_2) + t_3 - s_3 - s_1 + t_1 < 0, \\ t_2 > s_2, \end{cases} \quad (A9)$$

式中 $2 \times (t_2 - s_2)$ 在这种位置情况下大于0, 通过对上式进行变换可得到 $t_2 - s_2$ 的不等式:

$$t_2 - s_2 < \frac{(s_3 - t_1) - (t_3 - s_1)}{2}. \quad (A10)$$

由此可知, 式(A8)的分母在这一情况下一定小于2倍的 $s_3 - t_1$, 即 y_a 必定大于 $\frac{1}{2}$. 另外当 y_a 大于 $\frac{1}{2}$ 时, 由不等式(A10)可知, $\frac{(s_3 - t_1) - (t_3 - s_1)}{2}$ 就是 $t_2 - s_2$ 的上限, 将之代入到式(A8)即可得到 y_a 的下限 y_a^1 , 化简后表达式如下:

$$y_a^1 = \frac{2 \times (s_3 - t_1)}{3 \times (s_3 - t_1) - (t_3 - s_1)}. \quad (A11)$$

若要满足不等式(A7), 则至少应满足下式:

$$y_a^1 < \frac{1}{2} \times \left(\frac{t_3 - s_1}{s_3 - t_1} + 1\right). \quad (A12)$$

将式(A11)代入上式后化简可得

$$((s_3 - t_1) - (t_3 - s_1))^2 < 0, \quad (A13)$$

显然式(A13)是不成立的, 由此可知式(A7)也不可能成立. 而之前已证明 y_a 在Sakawa准则判断取大的近似值为 \tilde{s} 时必大于 $\frac{1}{2}$, 所以在两个模糊数的参数关系满足 $t_1 > s_1, t_2 > s_2, t_3 < s_3$ 条件下, Sakawa准则若判断取大的近似值为 \tilde{s} , 则不会出错. 证毕.

当 $t_1 = s_1$ 时, 其实是图4(a)情况的特例, 可以得到相同结论, 这里不再赘述.

作者简介:

李尚函 硕士研究生, 目前研究方向为智能算法与优化调度, E-mail: 491626402@qq.com;

胡蓉 副教授, 硕士生导师, 目前研究方向为智能优化调度、物流优化, E-mail: ronghu@vip.163.com;

钱斌 教授, 博士生导师, 目前研究方向为优化调度理论与方法、智能优化方法, E-mail: bin.qian@vip.163.com;

张梓琪 博士研究生, 目前研究方向为优化调度理论与方法、智能优化方法, E-mail: Albert.ziqi@outlook.com;

金怀平 讲师, 硕士生导师, 目前研究方向为复杂过程建模与机器学习, E-mail: jinhuaping@gmail.com.