考虑客户满意度的模糊柔性作业车间调度优化方法

刘 设 李佳欣 田志强* 张诗曼
沈阳工业大学机械工程学院,沈阳,110870

摘要:针对离散制造企业单件小批量、高定制化导致完工时间不确定、客户满意度差的问题,建立了以完工时间、平均及最小客户满意度为目标的模糊柔性作业车间调度模型,提出一种改进多目标进化算法对模型进行求解,引入基于种群分布的自适应交叉变异策略平衡全局及局部搜索。设计了基于满意度结构的知识驱动邻域搜索策略,通过调整关键路径的关键块与非关键块工序,以缩短完工时间、提高客户满意度。基于多组基准算例的对比分析结果验证了所提模型及算法的有效性。

关键词:模糊柔性作业车间调度;客户满意度;多目标优化;知识驱动邻域搜索
中图分类号:TH162
DOI:10.3969/j.issn.1004-132X.2026.07.023 **开放科学(资源服务)标识码(OSID)**:

Fuzzy Flexible Job-shop Scheduling Optimization Approach Considering Customer Satisfaction

LIU She LI Jiaxin TIAN Zhiqiang* ZHANG Shiman
School of Mechanical Engineering, Shenyang University of Technology, Shenyang, 110870

Abstract: To address uncertain completion times and low customer satisfaction in discrete manufacturing with single-piece, small-batch, and highly customized production, this paper establishes a fuzzy flexible job-shop scheduling model that minimizes makespan, maximizes average customer satisfaction, and maximizes the minimum customer satisfaction. An improved multi-objective evolutionary algorithm is proposed, which incorporates an adaptive crossover-mutation strategy based on population distribution to balance global and local search, and a knowledge-driven neighborhood search strategy that exploits the structure of customer satisfaction. By adjusting critical and non-critical blocks on the critical path, the algorithm reduces makespan and enhances customer satisfaction. Comparative results on multiple benchmark instances confirm the effectiveness of the proposed model and algorithm.

Key words: fuzzy flexible job-shop scheduling problem; customer satisfaction index; multi-objective optimization; knowledge-driven neighborhood search

0 引言

当前,以航空航天、汽车为代表的离散制造企业已从“以产品为中心”转向“以客户为中心”,客户满意度已成为其可持续发展的基石。离散制造企业车间生产高度柔性化及定制化,产品品种多、批量小、工艺过程复杂导致车间生产效率低、工件加工时间及完工时间难以准确估计[1]。此外,工件完工时间与客户满意度之间存在协同优化冲突,而交货窗口提前或延后都会降低客户满意度[2],因此如何平衡生产效率与客户满意度已成为企业发展面临的核心问题。高效的车间调度是压缩制造提前期、确保承诺交货期的关键。因此,开展考虑客户满意度的模糊柔性作业车间调度优化方法研究已势在必行。

模糊柔性作业车间调度问题(fuzzy flexible job-shop scheduling problem, FFJSP)已被证明是NP-hard问题[3]。目前,学者针对不同场景的FFJSP,在模型建立、算法求解领域开展了大量研究。模型建立领域,JIMÉNEZ TOVAR等[4]采用线性加权和协同优化了完工时间与平均流程时间,李瑞等[5]构建了以完工时间与机器负载为优化目标的FFJSP模型,LIU等[6]构建了考虑总负载及最大机器负载的多目标FFJSP模型。在此基础上,DENG等[7]引入能耗指标,将退化效应转化为模糊数内部的线性平移。LI等[8]为更好捕捉复杂制造过程中加工时间的不确定性,引入区间二型模糊集来表征加工时间,建立了集成开关机策略的节能调度混合整数线性规划(mixed-integer linear programming, MILP)模型。CHEN等[9]研究了设备加工速度可变的FFJSP,将加工时间表征为速度选择的动态函数。SUN等[10]针

收稿日期:2025-07-04
基金项目:辽宁省科技计划(2024JH2/102600216);沈阳工业大学翔源学者青年项目(XLYCQNO3)

对集成批量分割的FFJSP，采用具有非线性隶属度的L-R模糊数表征加工及装配时间。SOOFI等[11]研究了考虑机器故障的双资源FFJSP，以所有故障场景的平均完工时间及最坏情况的完工时间加权和为混合目标函数，提出一种鲁棒模糊随机规划模型。WU等[12]在生产调度的基础上引入AGV运输，以模糊数表征运输时间，构建了集成加工及运输的调度模型。ZHANG等[13]采用对称三元随机数表征加工时间，设计了一种集成调度方案生成与瓶颈识别的分层模型，依据多场景性能指标动态识别瓶颈。上述研究多聚焦于生产系统的效率与资源利用，鲜有揭示调度方案与客户满意度间的制约机理。

为准确量化客户满意度与交付期望，MA等[14]、YANG等[15]将完工时间与交货期间的偏差转化为满意度损失以提高客户满意度。在此基础上，CHEN等[2]引入设备维护约束，以区间数表征维护周期，构建了以完工时间、碳排放量、生产成本及客户满意度为目标的FFJSP模型。WANG等[16]考虑作业插入，建立了协同优化完工时间、客户满意度及总能耗的FFJSP模型。ZHU等[17]设计了一种基于完工时间相对交货期的延迟指数来表征客户交付期望，同时优化了完工时间、工作量及延迟指数。

求解算法方面，分支定界法、整数规划、约束规划等传统精确算法在求解大规模问题时难以在合理时间内求得满意可行解[18]，以进化算法、群智能优化算法为代表的元启发式算法已成为求解FFJSP模型的主流方法[9]。针对不同FFJSP的特点，学者们提出多种改进策略。唐红涛等[19]采用非线性动态权重更新机制平衡全局与局部搜索。LI等[20]提出了一种改进免疫算法，采用双向量局部搜索策略来提高搜索效率，并引入基于模拟退火算法的概率性接受劣解策略来避免早熟。PAN等[21]提出了一种带有反馈机制的双种群进化算法，并基于历史种群信息动态评估个体质量并调整种群规模，以提升解集质量。LI等[22]提出了一种基于成功/失败记忆的局部搜索策略，依据历史成功概率自主学习有效局部搜索操作，显著提高了搜索效率。GNANAVELBABU等[23]将蒙特卡洛模拟融入多目标改进回溯搜索算法，增强不确定性的处理能力。FANG等[24]设计了一种基于混沌与贪婪算法的混合初始化方法，提高了初始解质量。ZHANG等[25]设计了一种基于分解的多目标进化算法，通过基于惩罚的边界交集分层评估与聚类自适应分解策略动态调整解集的选择压力及权重向量，增强Pareto前沿的逼近能力。

针对考虑客户满意度的FFJSP，目前存在满意度目标维度单一与满意度驱动的优化算子缺失的问题，难以有效提升客户满意度。基于此，本文针对考虑客户满意度的FFJSP，建立以完工时间、平均客户满意度及最小客户满意度为优化目标的数学模型，提出改进多目标进化算法（improved multi-objective evolutionary algorithm，IMOEA）进行求解，采用自适应交叉与变异概率以平衡算法前后期的搜索能力；设计知识驱动的邻域搜索策略，引入N8与客户满意度（CSI）邻域结构进行局部搜索，进一步提高客户满意度、缩短工件完工时间。最后，通过田口试验选取算法最优参数组合，并基于多组基准算例验证了算法的有效性。

1　问题描述及数学模型

1.1　问题描述

本文所研究的考虑客户满意度的FFJSP可描述如下：有n个工件需在m台机器上加工，机器$M_k(k=1,2,\cdots,m)$可加工多道工序，其中，工件$J_i(i=1,2,\cdots,n)$有N_i道工序。工序的顺序是预先给定的，即后道工序必须在其前道工序完成后才能开始加工。工序加工时间以三角模糊数$\tilde{T}_1=(t_1,t_2,t_3)$表示，其中，$t_1$、$t_2$、$t_3$分别为最乐观值即最优加工时间、最可能值即最可能加工时间、最悲观值即最差加工时间。工件交货期以梯形模糊数$\tilde{D}_1=(d_1,d_2,d_3,d_4)$表示，其中，$d_1$、$d_2$、$d_3$、$d_4$分别为最早交货期、最理想交货期下限、最理想交货期上限及最晚交货期。图1给出的考虑客户满意度的FFJSP示例包含4个工件和3台机器。

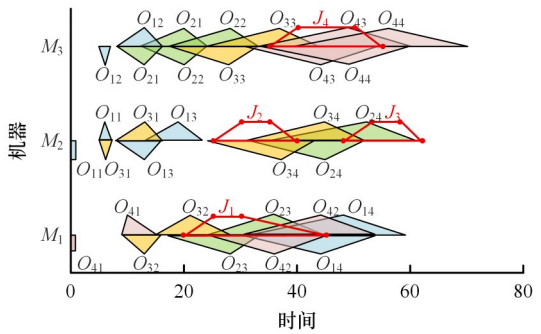


图1　考虑客户满意度的FFJSP示例

Fig.1　An illustration of FFJSP considering customer satisfaction

考虑客户满意度的FFJSP基于如下假设：①初始调度时刻，所有机器均可用，所有工件均可被加工；②各工件相互独立，同一个工件的工序有先后关系；③机器开始工作后一直处于工作状态，直

到完成分配的所有作业；④每道工序由不同机器完成的模糊加工时间不同且已知；⑤不考虑机器损坏、订单插入等特殊情况。

1.2 模型建立

模型中相关参数及含义如表1所示。

表1 符号说明

Tab.1 Description of symbols

符号	含义
i	工件索引
j	工序索引
k	机器索引
n	工件总数量
m	机器总数量
O_{ij}	工件J_i的第j道工序
N_i	工件J_i的工序总数量
$\tilde{D}_i$	工件J_i的交货期
$\tilde{t}_{ijk}$	工序O_{ij}在机器M_k的完成时间
$\tilde{s}_{ij}$	工序O_{ij}的开始时间
$\tilde{f}_{ij}$	工序O_{ij}的结束时间
$\tilde{C}_i$	工件J_i的完工时间
$\tilde{C}_{\max}$	所有工件中的最大完工时间
$\tilde{M}$	一个足够大的模糊正数
x_{ijk}	0-1变量，工序O_{ij}在机器M_k上加工为1，否则0
Y_{ijhpk}	0-1变量，若工序O_{ij}先于工序O_{hp}在机器M_k上加工，则为1，否则为0
A_i	工件J_i的客户满意度

1.2.1 优化目标

最小化完工时间即所有工件的最大完工时间：

$$f_1=\min \tilde{C}_{\max} \tag{1}$$

最大化平均客户满意度即指完工时间与交货期之间的符合程度：

$$f_2=\max\left(\frac{1}{n}\sum_{i=1}^{n}A_i\right) \tag{2}$$

$$A_i=\frac{area(\tilde{C}_i\cap\tilde{D}_i)}{area\tilde{C}_i} \tag{3}$$

式中：$area\tilde{C}_i$为模糊数$\tilde{C}_i$隶属度函数曲线下的面积；$area(\tilde{C}_i\cap\tilde{D}_i)$为$\tilde{C}_i$与$\tilde{D}_i$交集的隶属度函数曲线下的面积。

最大化最小客户满意度：

$$f_3=\max\min(A_1,A_2,\cdots,A_n) \tag{4}$$

1.2.2 约束条件

$$\tilde{s}_{ij}+\tilde{t}_{ijk}=\tilde{f}_{ij}\quad \forall i,j,k \tag{5}$$

$$\tilde{f}_{ij}\leqslant\tilde{s}_{i(j+1)}\quad \forall i,j,k \tag{6}$$

$$\tilde{f}_{i(N_i)}\leqslant\tilde{C}_{\max}\quad \forall i,j,k \tag{7}$$

$$\tilde{s}_{ij}+\tilde{t}_{ijk}\leqslant\tilde{s}_{ph}+\tilde{M}(1-Y_{ijphk})\quad \forall i,j,k,p,h \tag{8}$$

$$\sum_{k=1}^{m}x_{ijk}=1\quad \forall i,j,k \tag{9}$$

$$\tilde{s}_{ij},t_{ijk}\geqslant 0\quad \forall i,j,k \tag{10}$$

式(5)、式(6)约束各工序的先后顺序；式(7)表示每工件的完工时间必须小于总完工时间；式(8)表示一台机器一次只能完成一道工序；式(9)表示某一时刻的一道工序仅能在一台机器上完成。

1.3 三角模糊数的运算

本文所建FFJSP数学模型中的工序完成时间均以三角模糊数表示。求解前必须明确三角模糊数的运算规则即加法、比较和取大。对于三角模糊数$\tilde{T}_1=(t_1,t_2,t_3)$与$\tilde{T}_2=(t_1^*,t_2^*,t_3^*)$，加法运算规则为

$$\tilde{T}_1+\tilde{T}_2=(t_1+t_1^*,t_2+t_2^*,t_3+t_3^*) \tag{11}$$

比较准则如下：

准则1 对于$C_1(\tilde{T}_1)=(t_1+2t_2+t_3)/4$，$C_1(\tilde{T}_2)=(t_1^*+2t_2^*+t_3^*)/4$，若$C_1(\tilde{T}_1)>C_1(\tilde{T}_2)$，则$\tilde{T}_1>\tilde{T}_2$，反之，则$\tilde{T}_1<\tilde{T}_2$。

准则2 对于$C_2(\tilde{T}_1)=t_2$，$C_2(\tilde{T}_2)=t_2^*$，$C_1(\tilde{T}_1)=C_1(\tilde{T}_2)$时，若$C_2(\tilde{T}_1)>C_2(\tilde{T}_2)$，则$\tilde{T}_1>\tilde{T}_2$，反之，则$\tilde{T}_1<\tilde{T}_2$。

准则3 对于$C_3(\tilde{T}_1)=t_3-t_1$，$C_3(\tilde{T}_2)=t_3^*-t_1^*$，$C_2(\tilde{T}_1)=C_2(\tilde{T}_2)$时，若$C_3(\tilde{T}_1)>C_3(\tilde{T}_2)$，则$\tilde{T}_1>\tilde{T}_2$，反之，则$\tilde{T}_1<\tilde{T}_2$。

模糊数的近似取大规则如下：

$$\tilde{T}_1\vee\tilde{T}_2=\begin{cases}\tilde{T}_1 & \tilde{T}_1>\tilde{T}_2\\ \tilde{T}_2 & \text{其他}\end{cases} \tag{12}$$

2 基于IMOEA的模型求解

2.1 算法设计

相较于传统FJSP，考虑客户满意度的FFJSP具有加工时间不确定、多目标耦合、目标量级差异大等特点，问题复杂度高，以进化算法与群智能优化算法为代表的元启发式算法是主要求解方式。第三代非支配排序遗传算法(NSGA-Ⅲ)基于参考点的选择机制能有效维持目标空间中的解集分布性，并通过归一化处理与参考点设定有效解决不同目标量级差异大的问题，确保所有目标在Pareto前沿进化过程中的收敛性[26]。基于此，本文提出一种基于参考点的IMOEA对模型进行求解。首先，采用两段式整数编码表征FFJSP的解向量，基于贪婪解码将解向量中的编码值转化为具体的调度方案。然后，针对模型目标量级差异大的问题，引入基于参考点关联的自适应交叉与变异概率，并根据参考点的稀疏度与聚集度动态调整不同个体的交叉与变异概率，平衡算法的全局搜索能力与局部搜索能力。最后，根据问题特征设计基于N8与CSI邻域结构的知识驱动邻域搜索策略，通过有限次调整关键路径上的工序，实现精英个体周围解空间的精细化搜索，进一步缩

短完工时间，提高客户满意度。算法总体流程如图2所示。

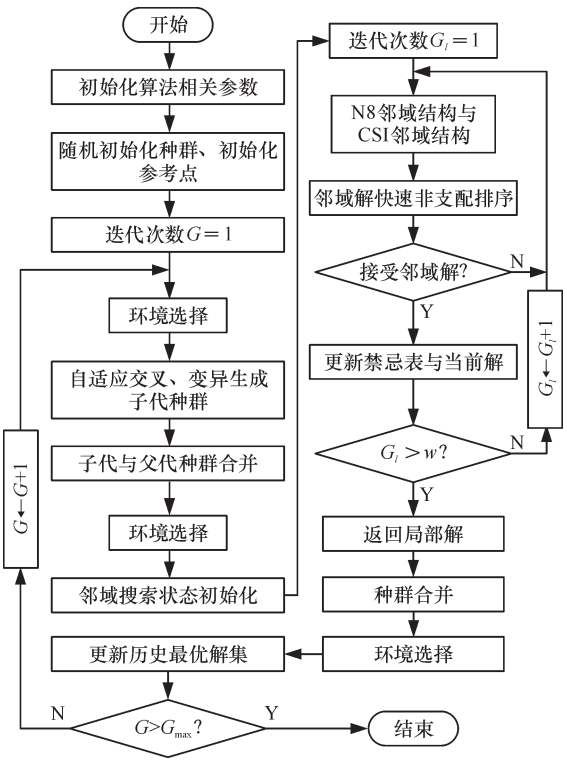


图2　算法流程图

Fig.2　The framework of the proposed algorithm

2.2　初始种群与编码解码

本文所提IMOEA采用随机初始化策略生成种群个体，基于两段式整数编码表征种群个体。每条染色体都由工序排序(OS)、机器选择(MS)两部分组成，其编码长度等于所有工件的工序总和。OS部分的元素值为工件的编号，同一编号在序列中出现的次数对应该工件的工序序号。MS部分的元素值为工件加工的机器编号。图1(甘特图)对应的编码如图3所示，OS部分的4个“4”表示工件4的第1-4道工序；MS部分的第一个“1”表示将工序O_{41}安排到机器M_1上完成。

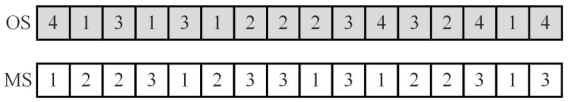


图3　编码示意图

Fig.3　Encoding diagram

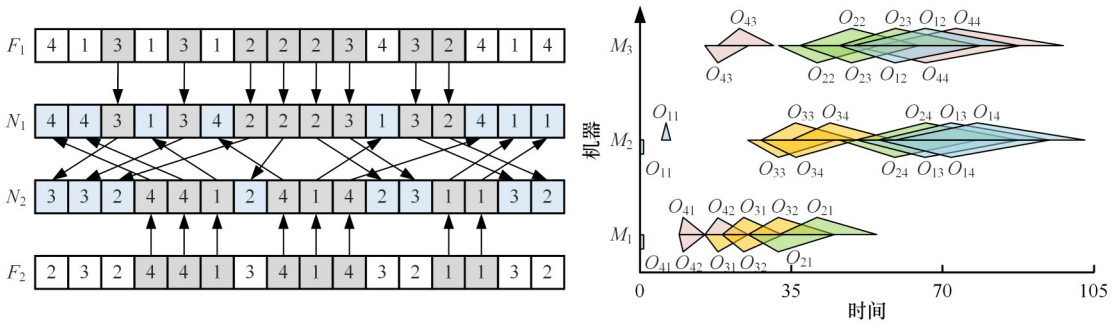
2.3　自适应交叉与变异

2.3.1　交叉算子

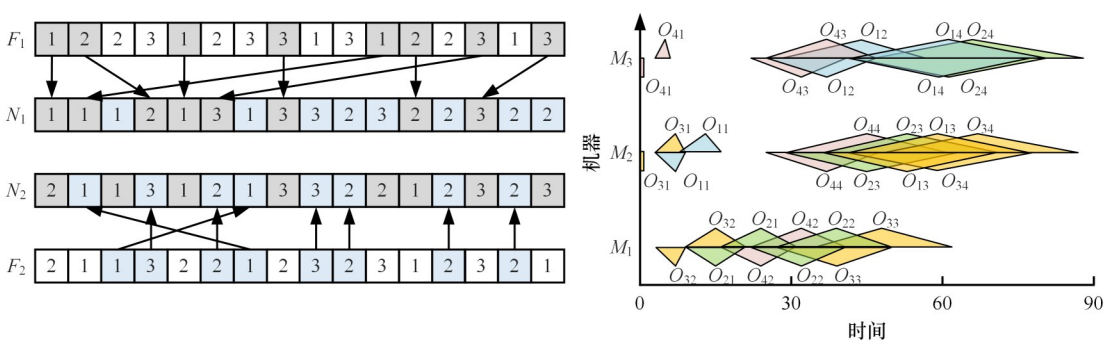
交叉操作中，OS编码采用POX交叉算子，MS编码采用MPX交叉算子。

1)POX交叉算子具体步骤如下：①将所有工件号随机分到非空子集S_1、S_2；②在父代F_1中找到与S_1内的工件号相同的编码，将其复制到子代N_1的相同位置上；③在父代F_2中找到与S_2内的工件号相同的编码，按从左到右的顺序复制到子代N_1中的空余位置；④以同样操作生成子代N_2，具体操作如图4所示。

2)MPX交叉算子具体步骤如下：①生成一个长度为所有工序总和的0-1集合B；②若B集合中



(a)POX交叉算子

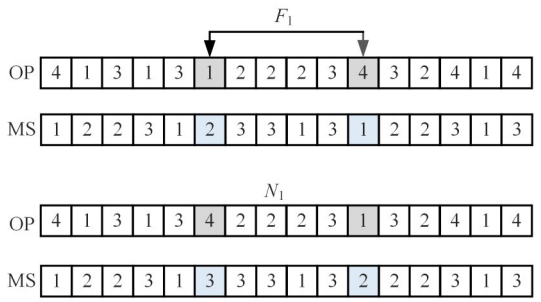


(b)MPX交叉算子

图4　交叉算子示意图

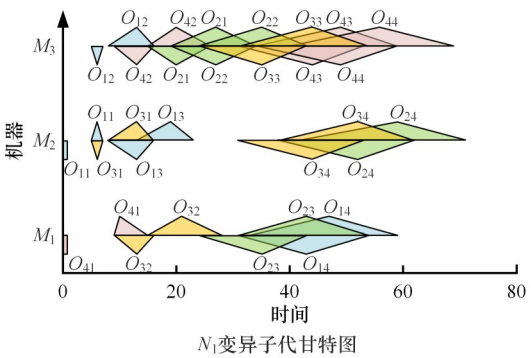
Fig.4　Schematic diagram of crossover operator

的某元素为1，则将父代F_1中对应位置的编码复制到子代N_1中的相同位置；③若B集合中某元素为0，则将父代F_2中对应位置的编码编复制到子代N_1中的相同位置；④以同样操作生成子代N_2，具体操作如图4所示。



2.3.2 变异算子

变异操作中，随机选取OS部分的2个编码并交换编码值；随机选取MS部分的两个工序并从对应候选机器中选择一台机器替代原有机器，如图5所示。



N_1变异子代甘特图

图5 变异算子示意图

Fig.5 Schematic diagram of mutation operator

2.3.3 自适应机制

在求解考虑客户满意度的FFJSP时，固定的交叉与变异概率往往难以适应算法在进化过程中的种群变化。进化初期，交叉与变异概率过大可能破坏优质解，过小则会削弱种群的探索能力，使算法过早收敛到次优解。算法后期，种群多样性降低时，过小的变异概率可能会使算法陷入局部最优解。因此，为减小固定参数可能导致的误差，平衡算法的全局及局部搜索能力，采用自适应策略来动态调整交叉与变异概率。该策略根据种群在进化过程中的动态信息调整不同个体的交叉与变异概率，以适应不同进化阶段的需求。第α个个体的交叉概率$P_c^{(\alpha)}$与变异概率$P_m^{(\alpha)}$分别为

$$P_c^{(\alpha)}=\begin{cases}P_{ca}-0.4C_\alpha & C_\alpha>0.25\\ P_{ci} & \text{其他}\end{cases} \tag{13}$$

$$C_\alpha=C_{i,\mathrm{num}}/Z$$

$$P_m^{(\alpha)}=P_{mi}+(P_{ma}-P_{mi})\frac{D_\alpha-D_{\min}}{D_{\max}-D_{\min}} \tag{14}$$

$$D_\alpha=\frac{1}{G}\sum_{g=1}^{G}(\|\boldsymbol{f}_g-\boldsymbol{R}_y\|-\mu)^2$$

$$\mu=\frac{1}{G}\|\boldsymbol{f}_g-\boldsymbol{R}_y\|$$

$$\|\boldsymbol{f}_g-\boldsymbol{R}_y\|=\sqrt{\sum_{h=1}^{H}(f_{g,h}-r_{y,h})^2}$$

式中：P_{ci}、P_{ca}分别为最小、最大交叉概率；$C_{i,\mathrm{num}}$为参考点R_y的关联个体数；Z为种群规模；P_{mi}、P_{ma}分别为最小、最大变异概率；D_α为参考点$\boldsymbol{R}_y$邻域稀疏程度；G为参考点$\boldsymbol{R}_y$邻域内的个体数；$\boldsymbol{f}_g$为第g个体目标向量；μ为邻域距离均值；H为目标函数个数；$f_{g,h}$、$r_{y,h}$分别为个体目标向量与参考点向量在第h个目标维度的分量。

2.4 知识驱动的邻域搜索策略

为进一步探索问题局部解的空间，提出基于禁忌搜索(tabu search，TS)的知识驱动邻域搜索策略。将禁忌搜索算法作为搜索框架，在此基础上，基于问题特征设计CSI邻域结构及N8邻域结构，进一步缩短工件完工时间、提高客户满意度。

2.4.1 N8邻域结构

现有的N5、N6、N7邻域结构是基于关键路径与关键块的工序扰动策略，在关键块内进行扰动来缩短完工时间，但将关键块内的工序移到关键块外可能缩短原有调度方案的完工时间。为此，N8邻域结构[27]考虑将关键块内的工序移到块外的情况，生成更多的邻域解，其有效性和工序移动后调度方案的可行性[28]已经通过实验验证。

2.4.2 CSI邻域结构

CSI邻域结构引入基于客户满意度的扰动操作，针对满意度低的工件非关键块工序进行扰动，进一步提高客户满意度。CSI邻域结构具体步骤如下：

1)计算所有工件的客户满意度，并选择客户满意度最低的2个工件。

2)这两个工件的工序中随机选择一道非关键块工序。

3)对该非关键块工序进行双点交换扰动操作，即将该非关键块工序与另一工件的某非关键块工序进行交换。

4)对该非关键块工序进行插入扰动操作，即将该非关键块工序插入到另一工件的某非关键块工序后。

5)对该非关键块工序进行邻域扰动操作，即在一定窗口长度内进行工序反转。

如图6所示，对图3所示的初始编码进行CSI邻域算子扰动操作。初始调度方案的完工时间为(39，56，70)，平均客户满意度为0.172，最小客户满意度为0.044，关键块为(O_{11})、(O_{12}-O_{21}-O_{22})、(O_{23}-O_{42})、(O_{43}-O_{44})，进行3种扰动操作后的调度方案如图7～图9所示。双点交换扰动操作后，调度方案的完工时间不变，但平均客户满意度提高至0.34，最小客户满意度提高至0.212。交叉扰动操作后，调度方案的完工时间、最小客户满意度不变，但平均客户满意度提高至0.334。邻域反转扰动操作后，调度方案完工时间缩短至(39，50，65)，平均客户满意度提高至0.344，最小客户满意度下降0.032。由此可见，CSI邻域结构能有效优化完工时间与客户满意度。

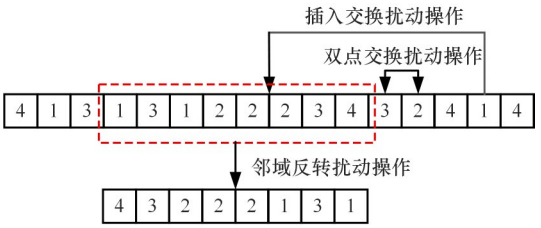


图6　CSI邻域结构

Fig.6　CSI neighborhood structure

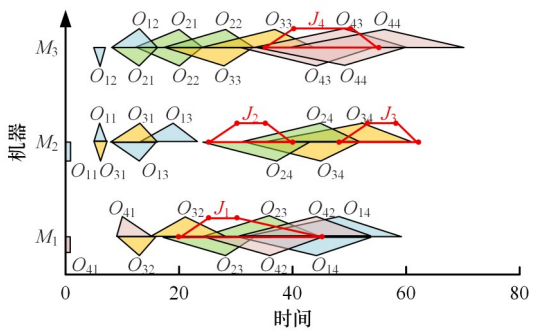


图7　双点交换扰动操作后的调度方案

Fig.7　Scheduling scheme after double-point exchange perturbation operation

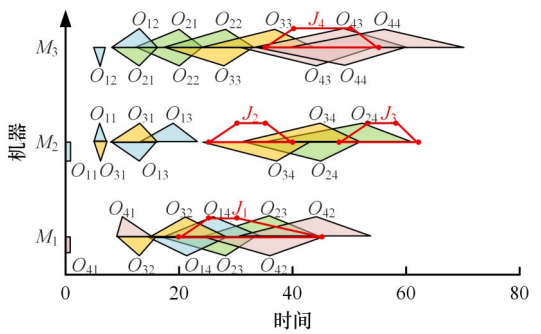


图8　插入扰动操作后的调度方案

Fig.8　Scheduling scheme after inserting disturbance operations

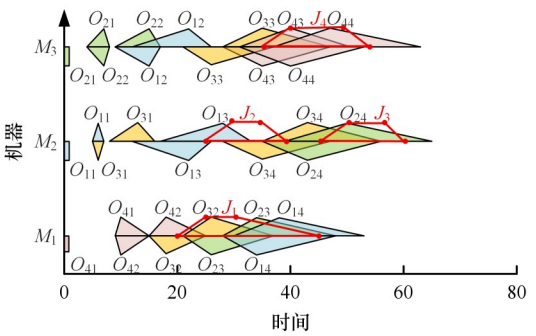


图9　邻域反转扰动操作后的调度方案

Fig.9　Scheduling scheme after neighborhood inversion perturbation operation

3　数值实验

3.1　实验设计

为评估所提模型及算法的有效性，本文选取FMK算例[28]与Remanu算例[29]进行验证。FMK算例包含有10组算例，最小规模算例的工件数为10，机器数为6，总工序数为55；最大规模算例的工件数为20，机器数为15，总工序数为240。Remanu算例包含8组算例，最小规模算例的工件数为5，机器数为4，总工序数为23；最大规模算例的工件数为20，机器数为15，总工序数为355。

所有实验均在配置为Intel i7-8565U CPU、8.00GB RAM的计算机上完成，编程语言为Python。所有算法均独立运行10次。

3.2　算法参数敏感性分析

算法参数的选择对其性能具有显著影响。通过前文算法描述可知，本文所提IMOEA的主要参数有总评估次数(种群数量与迭代次数的乘积)max F_e、局部搜索频率w、禁忌表长度T_l、邻域结构比例N_r，为探寻不同参数的取值变化对算法综合性能的影响，选取所有算例中处于中等规模的FMK06算例进行敏感性分析基于田口试验设计方法获取最优参数组合，将反转世代距离(IGD)

$$I(S,P^*)=\frac{1}{|P^*|}\sum_{\kappa=1}^{|P^*|}\min_{\tau\in\{1,2,\ldots,|S|\}}\sqrt{\sum_{h=1}^{H}(s_{\tau,h}-p_{\kappa,h})^2} \tag{15}$$

式中：S为算法求解解集；P^*为参考解集；$|P^*|$为参考点数目；τ为解集S中的索引；κ为参考解集P^*中参考点的索引；$s_{\tau,h}$为解集S中的第τ个解在第h个目标维度上的函数值；$p_{\kappa,h}$为解集P^*中的第κ个参考点在第h个目标维度上的函数值。

作为响应变量。IGD通过计算每个参考点到其最近解的距离均值来评价算法性能，值越小代表算法综合性能越好。

表2为一组四参数四水平$L_{16}(4^4)$正交试验表，选取中等规模的FMK06算例进行寻优参数，

在每个参数水平下独立运行10次，得到各组合下的平均响应值（表3）、平均响应值与等级（表4），各参数水平因子变化如图10所示。

表2 参数水平取值

Tab.2 Factors and levels of parameters

参数	水平			
	1	2	1	4
max F_e	25 000	37 500	50 000	62 500
T_l	5	10	15	20
w	5	10	15	20
N_r	0.3	0.4	0.5	0.6

表3 正交表与平均响应值

Tab.3 Orthogonal matrix and average response values

组合编码	水平				IGD
	1	2	3	4	
1	1	1	1	1	0.098 545
2	1	2	2	2	0.067 893
3	1	3	3	3	0.044 946
4	1	4	4	4	0.126 537
5	2	1	2	3	0.029 118
6	2	2	3	4	0.023 763
7	2	3	4	1	0.076 915
8	2	4	1	2	0.041 914
9	3	1	3	4	0.031 142
10	3	2	4	1	0.013 452
11	3	3	1	2	0.020 926
12	3	4	2	3	0.083 627
13	4	1	4	2	0.018 788
14	4	2	1	3	0.024 749
15	4	3	2	4	0.059 394
16	4	4	3	1	0.036 078

通过表4与图10可发现，总评估次数max F_e对算法性能影响最大，且max F_e过小会影响算法收敛速度与搜索广度。参数T_l对算法的影响位于第二位，T_l较小会限制算法搜索范围，导致算法难以跳出局部最优解，T_l过大会导致算法响应迟滞。参数w对算法性能的影响位于第三，决定了算法全局探索与局部搜索的平衡点，局部搜索频率很小会导致算法早熟收敛，反之会导致局部搜索不足，优质解无法充分扩展。参数N_r对算法的影响最小，均衡的邻域结构比例可保证邻域多样性与协同优化，提升算法稳定性。基于以上分析，参数max F_e、w、T_l、和N_r分别设为62 500、10、15和0.4。

表4 各参数平均响应值与等级

Tab.4 Average response values and ranks for each parameter

水平	max F_e	k	T_l	N_r
1	0.0845	0.0444	0.0465	0.0562
2	0.0429	0.0325	0.0600	0.0374
3	0.0373	0.0505	0.0340	0.0456
4	0.0348	0.0720	0.0589	0.0602
极差	0.0497	0.0396	0.0260	0.0228
排秩	1	2	3	4

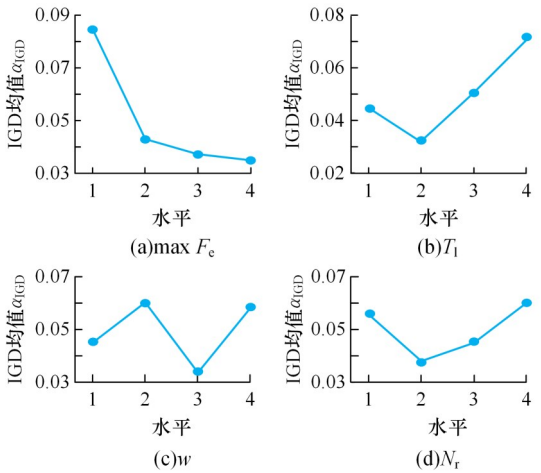


图10 四种参数水平因子变化趋势图

Fig.10 Four parameter level factor change trend chart

3.3 消融实验

为探究不同邻域算子对IMOEA搜索性能的影响，运用基于不同邻域算子的IMOEA对各基准算例进行求解。其中，IMOEA-1为无知识驱动的邻域搜索策略的IMOEA，IMOEA-2为基于N8邻域算子的知识驱动的邻域搜索策略的IMOEA，IMOEA-3为基于CSI邻域算子的知识驱动的邻域搜索策略的IMOEA。选取世代距离（GD）

$$G(S,P^*)=\frac{1}{|S|}\sum_{\tau=1}^{|S|}\min_{\kappa\in\{1,2,\cdots,|P^*|\}}\sqrt{\sum_{h=1}^{H}(s_{\tau,h}-p_{\kappa,h})^2} \quad (16)$$

式中：$|S|$为解集S中解的数量。

与IGD来评估算法性能。采用Friedman秩和检验分析消融实验结果，以判断邻域算子对IMOEA影响的显著性。表5、表6分别展示了所有算法独立运行后所得指标的均值与统计结果。

由表5、表6可得，IMOEA与3种变体算法（IMOEA-1/2/3）的算法性能有显著差异。从收敛性与分布性综合表现上看，IMOEA在所有算例上的GD与IGD的均值均优于IMOEA-1/2，只在算例Remanu02上劣于IMOEA-3；IMOEA-2/3在绝大多数基准算例上均优于IMOEA-1。这表明N8与CSI邻域算子可提高算法搜索能力。由不同规模算例的GD与IGD可得，问题规模较小时，IMOEA在绝大多数算例上的指标均优于IMOEA-1/2/3；IMOEA-2/3的IGD与GD上无显著差异。这是因为问题规模较小时，解空间的复杂度有限，N8与CSI邻域算子均能充分探索解空间并快速收敛至最优解附近的解空间，在提升

表5　IMOEA及其变体在不同测试算例上的统计结果

Tab.5　Statistical results of IMOEA and its variants on different test cases

算例	IGD				GD			
	IMOEA	IMOEA-1	IMOEA-2	IMOEA-3	IMOEA	IMOEA-1	IMOEA-2	IMOEA-3
FMK01	**1.2×10^{-2}**	3.65×10^{-2}	1.60×10^{-2}	3.20×10^{-2}	**6.83×10^{-3}**	2.96×10^{-2}	9.68×10^{-3}	2.18×10^{-2}
FMK02	**1.76×10^{-2}**	2.23×10^{-2}	2.20×10^{-2}	2.02×10^{-2}	**1.07×10^{-2}**	1.77×10^{-2}	1.12×10^{-2}	1.61×10^{-2}
FMK03	**1.69×10^{-2}**	3.19×10^{-2}	2.32×10^{-2}	2.56×10^{-2}	**1.20×10^{-2}**	2.14×10^{-2}	1.57×10^{-2}	1.76×10^{-2}
FMK04	**1.27×10^{-2}**	1.86×10^{-2}	1.60×10^{-2}	1.76×10^{-2}	**3.30×10^{-3}**	1.32×10^{-2}	9.30×10^{-3}	6.16×10^{-3}
FMK05	**8.77×10^{-3}**	1.62×10^{-2}	1.17×10^{-2}	1.50×10^{-2}	**6.58×10^{-3}**	1.09×10^{-2}	6.70×10^{-3}	9.87×10^{-3}
FMK06	**9.97×10^{-3}**	3.76×10^{-2}	2.28×10^{-2}	1.91×10^{-2}	**5.96×10^{-3}**	1.95×10^{-2}	1.22×10^{-2}	1.32×10^{-2}
FMK07	**4.84×10^{-3}**	1.08×10^{-2}	8.08×10^{-3}	1.20×10^{-2}	**1.42×10^{-3}**	7.84×10^{-3}	2.01×10^{-3}	7.81×10^{-3}
FMK08	**1.09×10^{-2}**	1.70×10^{-2}	1.47×10^{-2}	1.53×10^{-2}	**2.39×10^{-3}**	1.35×10^{-2}	5.68×10^{-3}	1.12×10^{-2}
FMK09	**2.32×10^{-2}**	4.27×10^{-2}	3.24×10^{-2}	4.11×10^{-2}	**1.12×10^{-2}**	5.25×10^{-2}	2.02×10^{-2}	2.29×10^{-2}
FMK10	**1.45×10^{-2}**	2.28×10^{-2}	2.08×10^{-2}	2.10×10^{-2}	**6.10×10^{-3}**	2.27×10^{-2}	1.89×10^{-2}	1.95×10^{-2}
Remanu01	**1.07×10^{-2}**	1.79×10^{-2}	1.70×10^{-2}	1.27×10^{-2}	**7.10×10^{-3}**	1.17×10^{-2}	7.96×10^{-3}	9.82×10^{-3}
Remanu02	**1.06×10^{-2}**	3.04×10^{-2}	1.62×10^{-2}	**6.91×10^{-3}**	4.28×10^{-3}	3.33×10^{-2}	1.13×10^{-2}	**2.19×10^{-3}**
Remanu03	**1.50×10^{-2}**	2.03×10^{-2}	1.57×10^{-2}	1.80×10^{-2}	**2.48×10^{-3}**	1.49×10^{-2}	8.59E-03	1.04×10^{-2}
Remanu04	**4.62×10^{-3}**	1.62×10^{-2}	1.49×10^{-2}	1.11×10^{-2}	**1.10×10^{-2}**	2.79×10^{-2}	1.49×10^{-2}	1.11×10^{-2}
Remanu05	**1.23×10^{-3}**	2.73×10^{-2}	2.12×10^{-2}	2.21×10^{-2}	**9.06×10^{-3}**	3.06×10^{-2}	2.32×10^{-2}	2.35×10^{-2}
Remanu06	**1.66×10^{-3}**	1.05×10^{-2}	5.73×10^{-3}	5.08×10^{-3}	**5.21×10^{-3}**	1.35×10^{-2}	9.08×10^{-3}	1.10×10^{-2}
Remanu07	**8.43×10^{-3}**	3.30×10^{-2}	1.95×10^{-2}	2.09×10^{-2}	**2.80×10^{-3}**	2.73×10^{-2}	9.95×10^{-3}	1.17×10^{-2}
Remanu08	**1.73×10^{-2}**	3.11×10^{-2}	2.79×10^{-2}	2.56×10^{-2}	**9.86×10^{-3}**	2.20×10^{-2}	1.55×10^{-2}	1.59×10^{-2}

表6　不同指标下的Friedman秩和检验（置信度为0.95）

Tab.6　Friedman rank sum test under different indicators（confidence level is as 0.95）

	IGD				GD			
算法	IMOEA	IMOEA-1	IMOEA-2	IMOEA-3	IMOEA	IMOEA-1	IMOEA-2	IMOEA-3
均值	1.05×10^{-2}	2.46×10^{-2}	1.81×10^{-2}	1.90×10^{-2}	6.57×10^{-3}	2.17×10^{-2}	1.18×10^{-2}	1.34×10^{-2}
Friedman	1.056	3.944	2.389	2.6111	1.056	4.0000	2.167	2.778
p-value	1.52×10^{-9}				6.84×10^{-10}			

算法收敛性的同时维持解集在目标空间中的均匀分布。处理中大规模算例时，IMOEA在所有算例的GD与IGD均优于IMOEA-1/2/3，而IMOEA-2在绝大多数算例上获得结果优于IMOEA-3。其原因在于IMOEA采用知识驱动的邻域搜索策略，N8邻域算子通过对关键路径上关键块工序进行调整，确保算法能探索更广泛的解空间，而CSI邻域算子针对优质解进行精细化搜索，在逼近真实的Pareto前沿的同时维持更好的分布性。相比之下，IMOEA-2缺少CSI邻域算子去深度探索优秀解周围的空间，导致其指标劣于IMOEA。IMOEA-3受限于CSI邻域算子的强局部性，在中大规模问题中易陷于局部最优且搜索效率低，导致其整体性能劣于IMOEA-2。

3.4　算法有效性分析

为进一步验证所提IMOEA的有效性，选取多目标粒子群优化（MOPSO）[30]算法、基于分解的多目标进化算法（MOEA/D）[25]、量子启发进化算法（QIEA）[31]、多目标教与学优化（MOTLBO）[32]算法、基于分解的聚类分层估计进化算法（MOEA/DCH）[25]进行对比。为进一步评价算法解集分布的均匀性，在GD与IGD的基础上引入间距指标（SP）：

$$P(S)=\sqrt{\frac{1}{|S|-1}\sum_{\tau=1}^{|S|}(\bar{d}-d_\tau)^2} \tag{17}$$

$$d_\tau=\min_{\omega\in\{1,2,\ldots,|S|\},\omega\neq\tau}\sqrt{\sum_{h=1}^{H}(s_{\tau,h}-s_{\omega,h})}$$

式中：ω为解集S内其他解的索引；d_τ为第τ个解到解集S中最近邻解的欧几里得距离；$\bar{d}$为d_τ的均值。

为公平起见，所有算法均设置相同的种群大小、迭代次数、编码解码策略。表7～表9给出了不同算法在各基准算例上的统计结果，各指标Friedman秩和检验结果如表10所示。图11、图12分别展示了IMOEA求解FMK02最优调度方案及帕累托前沿。

由表7～表10的统计结果可发现IMOEA与5种对比算法的性能有显著差异。从收敛性与分布性的综合表现上看，IMOEA在不同规模的基准算例上的GD与IGD均优于其他算法。从解集分布的均匀性上看，IMOEA在不同规模的基准

表7 不同算法在测试算例上的IGD

Tab.7 IGD of different algorithms on benchmark problems

算例	IMOEA	MOPSO	MOEA/D	QIEA	MOTLBO	MOEA/DCH
FMK01	**2.34×10^{-2}**	4.23×10^{-1}	2.14×10^{-1}	3.26×10^{-1}	3.89×10^{-1}	4.00×10^{-1}
FMK02	**2.12×10^{-2}**	5.30×10^{-1}	1.41×10^{-1}	1.87×10^{-1}	1.99×10^{-1}	2.52×10^{-1}
FMK03	**2.60×10^{-2}**	4.22×10^{-1}	1.78×10^{-1}	2.22×10^{-1}	2.17×10^{-1}	1.76×10^{-1}
FMK04	**2.72×10^{-2}**	4.98×10^{-1}	1.94×10^{-1}	2.84×10^{-1}	3.25×10^{-1}	2.16×10^{-1}
FMK05	**1.63×10^{-2}**	3.60×10^{-1}	1.31×10^{-1}	1.43×10^{-1}	1.77×10^{-1}	8.27×10^{-2}
FMK06	**1.84×10^{-2}**	5.38×10^{-1}	4.22×10^{-1}	5.04×10^{-1}	3.90×10^{-1}	1.36×10^{-1}
FMK07	**1.76×10^{-2}**	4.35×10^{-1}	1.93×10^{-1}	1.66×10^{-1}	3.52×10^{-1}	1.04×10^{-1}
FMK08	**1.97×10^{-2}**	4.08×10^{-1}	1.85×10^{-1}	1.89×10^{-1}	1.95×10^{-1}	2.37×10^{-1}
FMK09	**2.09×10^{-2}**	4.86×10^{-1}	2.13×10^{-1}	2.31×10^{-1}	3.84×10^{-1}	1.69×10^{-1}
FMK10	**3.18×10^{-2}**	4.43×10^{-1}	1.49×10^{-1}	1.84×10^{-1}	1.77×10^{-1}	1.57×10^{-1}
Remanu01	**2.65×10^{-2}**	3.98×10^{-1}	1.06×10^{-1}	1.56×10^{-1}	4.52×10^{-1}	4.98×10^{-1}
Remanu02	**1.28×10^{-2}**	2.69×10^{-1}	9.75×10^{-1}	1.07×10^{-1}	1.46×10^{-1}	1.52×10^{-1}
Remanu03	**2.56×10^{-2}**	4.10×10^{-1}	1.47×10^{-1}	1.71×10^{-1}	2.12×10^{-1}	1.96×10^{-1}
Remanu04	**2.00×10^{-2}**	3.72×10^{-1}	1.21×10^{-1}	1.33×10^{-1}	1.34×10^{-1}	1.85×10^{-1}
Remanu05	**2.83×10^{-2}**	3.40×10^{-1}	1.36×10^{-1}	1.55×10^{-1}	1.72×10^{-1}	1.55×10^{-1}
Remanu06	**2.00×10^{-2}**	4.17×10^{-1}	1.16×10^{-1}	1.56×10^{-1}	1.45×10^{-1}	1.41×10^{-1}
Remanu07	**3.43×10^{-2}**	4.77×10^{-1}	2.42×10^{-1}	2.26×10^{-1}	2.85×10^{-1}	2.05×10^{-1}
Remanu08	**4.29×10^{-2}**	4.85×10^{-1}	2.12×10^{-1}	2.67×10^{-1}	2.60×10^{-1}	2.81×10^{-1}

表8 不同算法在测试算例上的GD

Tab.8 GD of different algorithms on benchmark problems

算例	IMOEA	MOPSO	MOEA/D	QIEA	MOTLBO	MOEA/DCH
FMK01	**1.86×10^{-2}**	2.95×10^{-1}	2.21×10^{-1}	2.90×10^{-1}	3.19×10^{-1}	2.98×10^{-1}
FMK02	**1.99×10^{-2}**	2.45×10^{-1}	1.57×10^{-1}	1.96×10^{-1}	2.00×10^{-1}	1.95×10^{-1}
FMK03	**2.18×10^{-2}**	2.88×10^{-1}	1.88×10^{-1}	2.23×10^{-1}	2.08×10^{-1}	1.49×10^{-1}
FMK04	**2.47×10^{-2}**	3.17×10^{-1}	2.08×10^{-1}	2.83×10^{-1}	2.99×10^{-1}	1.81×10^{-1}
FMK05	**1.34×10^{-2}**	3.61×10^{-1}	1.62×10^{-1}	1.79×10^{-1}	2.24×10^{-1}	6.42×10^{-2}
FMK06	**1.07×10^{-2}**	3.86×10^{-1}	3.06×10^{-1}	3.29×10^{-1}	2.72×10^{-1}	1.97×10^{-1}
FMK07	**1.04×10^{-2}**	2.14×10^{-1}	1.07×10^{-1}	1.32×10^{-1}	8.70×10^{-2}	7.85×10^{-2}
FMK08	**1.67×10^{-2}**	3.54×10^{-1}	1.74×10^{-1}	2.00×10^{-1}	1.98×10^{-1}	1.69×10^{-2}
FMK09	**2.63×10^{-2}**	2.26×10^{-1}	2.31×10^{-1}	2.10×10^{-1}	2.77×10^{-1}	1.13×10^{-1}
FMK10	**2.99×10^{-2}**	3.29×10^{-1}	1.81×10^{-1}	2.04×10^{-1}	2.09×10^{-1}	1.29×10^{-1}
Remanu01	**1.95×10^{-2}**	1.83×10^{-1}	7.54×10^{-2}	1.37×10^{-1}	2.39×10^{-1}	3.04×10^{-1}
Remanu02	**1.04×10^{-2}**	2.53×10^{-1}	1.20×10^{-1}	1.32×10^{-1}	1.64×10^{-1}	1.64×10^{-1}
Remanu03	**2.18×10^{-2}**	2.81×10^{-1}	1.76×10^{-1}	1.99×10^{-1}	2.44×10^{-1}	1.68×10^{-1}
Remanu04	**1.52×10^{-2}**	2.10×10^{-1}	1.26×10^{-1}	1.33×10^{-1}	1.28×10^{-1}	1.53×10^{-1}
Remanu05	**2.50×10^{-2}**	2.70×10^{-1}	1.40×10^{-1}	1.59×10^{-1}	1.25×10^{-1}	1.38×10^{-1}
Remanu06	**1.07×10^{-2}**	2.21×10^{-1}	1.22×10^{-1}	1.45×10^{-1}	1.44×10^{-1}	1.26×10^{-1}
Remanu07	**2.49×10^{-2}**	2.84×10^{-1}	1.60×10^{-1}	1.75×10^{-1}	1.78×10^{-1}	1.55×10^{-1}
Remanu08	**3.71×10^{-2}**	2.58×10^{-1}	1.94×10^{-1}	1.96×10^{-1}	2.03×10^{-1}	1.69×10^{-1}

算法上的SP同样优于对比算法，验证了算法的有效性。IMOEA采用交叉与变异概率自适应机制，依据种群信息动态调整交叉与变异概率，实现对精英个体的低扰动保护与非精英个体的多样性增强，在维持种群收敛稳定性的同时优化全局探索效率。在此基础上，知识驱动的邻域搜索策略针对不同目标维度，在精英个体附近的解空间进行局部搜索，进一步提升算法性能。此外，IMOEA在不同算例规模上展现良好的鲁棒性。相较之下，随着算例规模增大，其余算法的GD与IGD普遍劣化，算法性能受问题规模影响显著。鲁棒排序及合理资源分配能在动态生产环境中有效提高客户满意度，避免物料、设备及工艺参数波动导致的计划偏离，保障订单及时交付。此外，考虑多维度客户满意度的调度方法有助于引导生产资源向关键客户订单优化配置，在确保生产效率

表 9　不同算法在测试算例上的 SP

Tab.9　SP of different algorithms on benchmark problems

算例	IMOEA	MOPSO	MOEA/D	QIEA	MOTLBO	MOEA/DCH
FMK01	$\mathbf{8.73\times10^{-3}}$	4.85×10^{-2}	2.79×10^{-2}	4.99×10^{-2}	4.83×10^{-2}	5.69×10^{-2}
FMK02	$\mathbf{9.19\times10^{-3}}$	2.97×10^{-2}	2.51×10^{-2}	2.79×10^{-2}	2.48×10^{-2}	4.48×10^{-2}
FMK03	$\mathbf{1.46\times10^{-2}}$	8.32×10^{-2}	2.26×10^{-2}	2.68×10^{-2}	2.63×10^{-2}	3.86×10^{-2}
FMK04	$\mathbf{8.11\times10^{-3}}$	6.09×10^{-2}	2.93×10^{-2}	3.17×10^{-2}	3.02×10^{-2}	4.07×10^{-2}
FMK05	$\mathbf{8.22\times10^{-3}}$	7.62×10^{-2}	1.96×10^{-2}	2.76×10^{-2}	2.45×10^{-2}	2.34×10^{-2}
FMK06	$\mathbf{6.34\times10^{-3}}$	4.37×10^{-2}	6.62×10^{-2}	2.36×10^{-2}	2.90×10^{-2}	1.00×10^{-2}
FMK07	$\mathbf{1.33\times10^{-2}}$	4.14×10^{-2}	2.46×10^{-2}	2.85×10^{-2}	2.56×10^{-2}	2.84×10^{-2}
FMK08	$\mathbf{1.17\times10^{-2}}$	4.08×10^{-2}	1.85×10^{-2}	1.89×10^{-2}	1.95×10^{-2}	5.37×10^{-2}
FMK09	$\mathbf{1.54\times10^{-2}}$	3.51×10^{-2}	2.61×10^{-2}	4.17×10^{-2}	2.62×10^{-2}	2.87×10^{-2}
FMK10	$\mathbf{1.48\times10^{-2}}$	9.27×10^{-2}	2.18×10^{-2}	2.27×10^{-2}	2.39×10^{-2}	2.91×10^{-1}
Remanu01	$\mathbf{1.23\times10^{-2}}$	8.18×10^{-2}	3.16×10^{-2}	4.54×10^{-2}	1.95×10^{-1}	2.78×10^{-2}
Remanu02	$\mathbf{9.64\times10^{-3}}$	4.56×10^{-2}	2.60×10^{-2}	2.34×10^{-2}	2.50×10^{-2}	3.11×10^{-2}
Remanu03	$\mathbf{1.60\times10^{-2}}$	6.37×10^{-2}	2.15×10^{-2}	3.39×10^{-2}	2.51×10^{-2}	2.85×10^{-2}
Remanu04	$\mathbf{1.52\times10^{-2}}$	4.65×10^{-2}	1.96×10^{-2}	2.32×10^{-2}	2.07×10^{-2}	3.98×10^{-2}
Remanu05	$\mathbf{1.49\times10^{-2}}$	5.91×10^{-2}	2.23×10^{-2}	2.15×10^{-2}	2.82×10^{-2}	3.63×10^{-2}
Remanu06	$\mathbf{1.86\times10^{-2}}$	6.38×10^{-2}	6.91×10^{-2}	7.11×10^{-2}	5.58×10^{-2}	5.06×10^{-2}
Remanu07	$\mathbf{1.99\times10^{-2}}$	5.64×10^{-2}	2.74×10^{-2}	4.46×10^{-2}	2.99×10^{-2}	3.76×10^{-2}
Remanu08	$\mathbf{1.98\times10^{-2}}$	3.29×10^{-2}	3.19×10^{-2}	4.28×10^{-2}	2.14×10^{-2}	3.81×10^{-2}

表 10　不同指标下的 Friedman 秩和检验（置信度为 0.95）

Tab.10　Friedman rank sum test under different indicators（confidence level is as 0.95）

指标	IGD			GD			SP		
算法	均值	Friedman	*p*-value	均值	Friedman	*p*-value	均值	Friedman	*p*-value
IMOEA	2.41×10^{-2}	1.000	4.68×10^{-6}	1.98×10^{-2}	1.000	6.42×10^{-11}	1.32×10^{-2}	1.000	7.12×10^{-11}
MOPSO	4.28×10^{-1}	5.889		2.76×10^{-1}	5.778		5.57×10^{-2}	5.500	
MOEA/D	1.78×10^{-1}	2.556		1.69×10^{-1}	2.333		2.59×10^{-2}	2.500	
QIEA	2.12×10^{-1}	3.750		1.96×10^{-1}	3.667		3.36×10^{-2}	3.000	
MOTLBO	2.56×10^{-1}	4.389		2.07×10^{-1}	4.222		3.77×10^{-2}	4.000	
MOEA/DCH	1.96×10^{-1}	3.417		1.55×10^{-1}	3.000		5.03×10^{-2}	4.000	

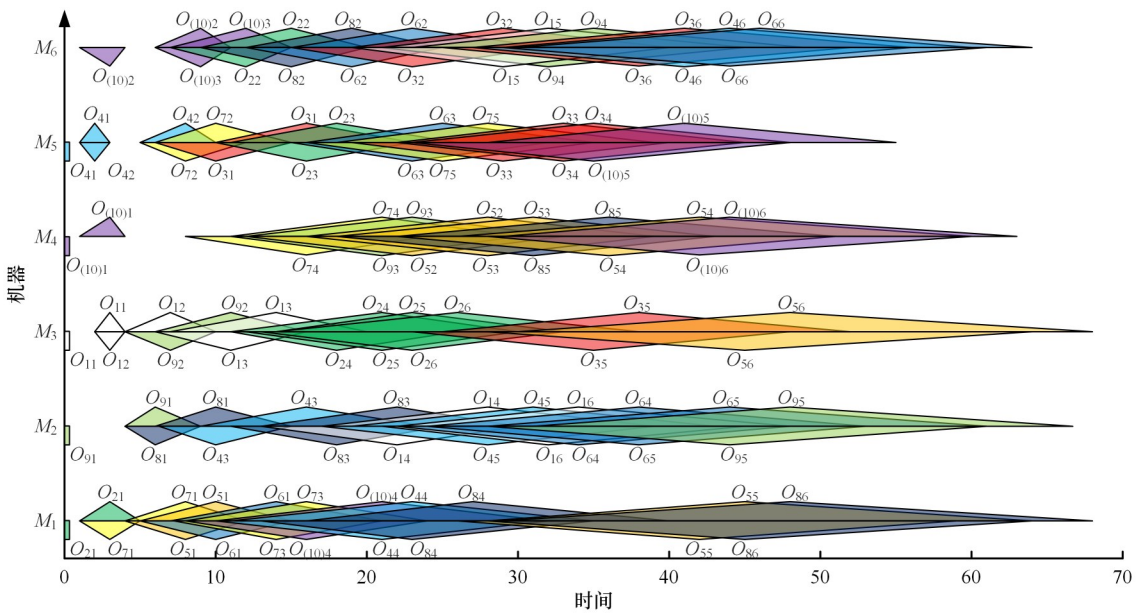


图 11　基于 IMOEA 方法求解 FMK02 的最优调度方案甘特图

Fig.11　Gantt chart of the optimal scheduling solution for FMK02 using IMOEA

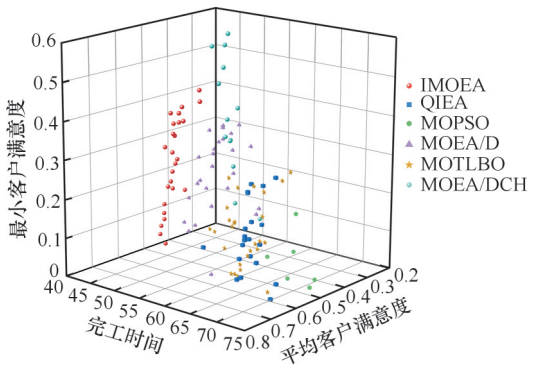


图12 不同算法求解FMK02的帕累托前沿图

Fig.12 Pareto fronts of different algorithms

的同时维护客户稳定性，增强企业在不确定的市场环境中的交付及竞争优势，实现可持续发展。

4 结论

本文针对考虑客户满意度的FFJSP构建了以最小化模糊完工时间、最大化平均客户满意度、最大化最小客户满意度为优化目标的FFJSP模型，并提出一种IMOEA进行求解。该算法采用交叉与变异自适应概率满足不同进化阶段的需求；针对问题特性设计了基于N8及CSI结构的知识驱动邻域搜索策略，进一步提升算法的局部搜索能力。最后通过多组基准算例对算法的有效性进行验证，结果表明：对于不同规模的实例数据，IMOEA的收敛性、分布性以及解集分布均匀性均显著优于MOPSO、MOEA/D、QIEA、MOTLBO、MOEA/DCH。

未来工作将重点围绕以下方面：①在模型构建方面，考虑动态事件与资源约束，建立考虑多维度客户满意度的FFJSP模型，构建动态需求驱动的模糊调度决策范式；②在算法求解方面，通过融合深度强化学习与模糊推理技术，开发具有在线自适应能力的混合智能算法，并建立数字孪生赋能的虚实交互验证平台。

参考文献：

[1] 祝正宇，郭具涛，吕佑龙，等. 面向柔性作业车间生产调度的深度强化学习方法[J]. 中国机械工程，2024，35(11)：2007-2014.
ZHU Zhengyu，GUO Jutao，LYU Youlong，et al. Deep Reinforcement Learning Method for Flexible Job Shop Scheduling[J]. China Mechanical Engineering，2024，35(11)：2007-2014.

[2] CHEN Zhaoming，ZOU Jinsong，WANG Wei. Digital Twin-oriented Collaborative Optimization of Fuzzy Flexible Job Shop Scheduling under Multiple Uncertainties[J]. Sādhanā，2023，48(2)：78.

[3] DAUZÈRE-PÉRÈS S，DING Junwen，SHEN Liji，et al. The Flexible Job Shop Scheduling Problem：a Review[J]. European Journal of Operational Research，2024，314(2)：409-432.

[4] JIMÉNEZ TOVAR M，ACEVEDO-CHEDID J，OSPINA-MATEUS H，et al. An Optimization Algorithm for the Multi-objective Flexible Fuzzy Job Shop Environment with Partial Flexibility Based on Adaptive Teaching - Learning Considering Fuzzy Processing Times[J]. Soft Computing，2024，28(2)：1459-1489.

[5] 李瑞，龚文引. 改进的基于分解的多目标进化算法求解双目标模糊柔性作业车间调度问题[J]. 控制理论与应用，2022，39(1)：31-40.
LI Rui，GONG Wenyin. An Improved Multi-objective Evolutionary Algorithm Based on Decomposition for Bi-objective Fuzzy Flexible Job-shop Scheduling Problem[J]. Control Theory & Applications，2022，39(1)：31-40.

[6] LIU Zhenggang，LIANG Xu，HOU Lingyan，et al. Multi-strategy Dynamic Evolution-based Improved MOEA/D Algorithm for Solving Multi-objective Fuzzy Flexible Job Shop Scheduling Problem[J]. IEEE Access，2023，11：54596-54606.

[7] DENG Libao，ZHU Yingjian，DI Yuanzhu，et al. Biased Bi-population Evolutionary Algorithm for Energy-efficient Fuzzy Flexible Job Shop Scheduling with Deteriorating Jobs[J]. Complex System Modeling and Simulation，2024，4(1)：15-32.

[8] LI Rui，GONG Wenyin，LU Chao，et al. A Learning-based Memetic Algorithm for Energy-efficient Flexible Job-shop Scheduling with Type-2 Fuzzy Processing Time[J]. IEEE Transactions on Evolutionary Computation，2023，27(3)：610-620.

[9] CHEN Xiaolong，LI Junqing，DU Yu. A Hybrid Evolutionary Immune Algorithm for Fuzzy Flexible Job Shop Scheduling Problem with Variable Processing Speeds[J]. Expert Systems with Applications，2023，233：120891.

[10] SUN Mengke，CAI Zongyan，ZHANG Haonan. A Teaching-learning-based Optimization with Feedback for L-R Fuzzy Flexible Assembly Job Shop Scheduling Problem with Batch Splitting[J]. Expert Systems with Applications，2023，224：120043.

[11] SOOFI P，YAZDANI M，AMIRI M，et al. Robust Fuzzy-stochastic Programming Model and Meta-heuristic Algorithms for Dual-resource Constrained Flexible Job-shop Scheduling Problem under Machine Breakdown[J]. IEEE Access，2021，9：155740-155762.

[12] WU Rui，TIAN Zheng，LI Xixing，et al. Improved Discrete Particle Swarm Optimization Algorithm for Solving Fuzzy Flexible Job Shop Machines and Automated Guided Vehicles Fusion Scheduling Problem[J]. Engineering Applications of Artificial Intelligence，2025，160：111951.

[13] ZHANG Cuilin，CHEN Jian，SANG Yaowen，et al. Integrated Ternary Scheduling and Execution Bottleneck Identification in Stochastic Job Shops[J]. Expert Systems with Applications，2026，296：129183.

[14] MA Jing，LI Yan. Solution to IPPS Problem under the Condition of Uncertain Delivery Time[C]//Smart Innovations in Communication and Computational Sciences. Singapore：Springer，2019：105-111.

[15] YANG M S，BA L，ZHENG H Y，et al. An Integrated System for Scheduling of Processing and Assembly Operations with Fuzzy Operation Time and Fuzzy Delivery Time[J]. Advances in Production Engineering & Management，2019，14(3)：367-378.

[16] WANG Ziqing，LIAO Wenzhu，ZHANG Yaping. Rescheduling Optimisation of Sustainable Multi-objective Fuzzy Flexible Job Shop under Uncertain Environment[J].

International Journal of Production Research，2024，62(24)：8904-8920.

[17] ZHU Zhenwei，ZHOU Xionghui. A Multi-objective Multi-micro-swarm Leadership Hierarchy-based Optimizer for Uncertain Flexible Job Shop Scheduling Problem with Job Precedence Constraints[J]. Expert Systems with Applications，2021，182：115214.

[18] 梁志珍，王晓佳. 双资源约束的柔性作业车间鲁棒调度方法[J]. 机械工程学报，2024，60(6)：114-126.
LIANG Zhizhen，WANG Xiaojia. Dual Resource Constraints Flexible Job Shop Robust Scheduling Method[J]. Journal of Mechanical Engineering，2024，60(6)：114-126.

[19] 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[J]. 华中科技大学学报(自然科学版)，2022，50(6)：81-88.
TANG Hongtao，LI Yue，WANG Lei. An Improved GWO Algorithm for Fuzzy Distributed Flexible Job Shop Scheduling Problem[J]. Journal of Huazhong University of Science and Technology (Nature Science Edition)，2022，50(6)：81-88.

[20] LI Junqing，LIU Zhengmin，LI Chengdong，et al. Improved Artificial Immune System Algorithm for Type-2 Fuzzy Flexible Job Shop Scheduling Problem[J]. IEEE Transactions on Fuzzy Systems，2021，29(11)：3234-3248.

[21] PAN Zixiao，LEI Deming，WANG Ling. A Bi-population Evolutionary Algorithm with Feedback for Energy-efficient Fuzzy Flexible Job Shop Scheduling[J]. IEEE Transactions on Systems，Man，and Cybernetics：Systems，2022，52(8)：5295-5307.

[22] LI Rui，GONG Wenyin，LU Chao. Self-adaptive Multi-objective Evolutionary Algorithm for Flexible Job Shop Scheduling with Fuzzy Processing Time[J]. Computers & Industrial Engineering，2022，168：108099.

[23] GNANAVELBABU A，CALDEIRA R H，VAIDYANATHAN T. A Simulation-based Modified Backtracking Search Algorithm for Multi-objective Stochastic Flexible Job Shop Scheduling Problem with Worker Flexibility [J]. Applied Soft Computing，2021，113：107960.

[24] FANG Jincheng，ZENG Afeng，ZHENG Shaofeng，et al. Improved Multiverse Optimization Algorithm for Fuzzy Flexible Job-shop Scheduling Problem[J]. IEEE Access，2023，11：48259-48275.

[25] ZHANG Xuwei，LIU Shixin，ZHAO Ziyan，et al. A Decomposition-based Evolutionary Algorithm with Clustering and Hierarchical Estimation for Multiobjective Fuzzy Flexible Jobshop Scheduling[J]. IEEE Transactions on Evolutionary Computation，2026，30(1)：2-15.

[26] 徐宜刚，陈勇，王宸，等. 改进NSGA-Ⅲ求解高维多目标绿色柔性作业车间调度问题[J]. 系统仿真学报，2024，36(10)：2314-2329.
XU Yigang，CHEN Yong，WANG Chen，et al. Improving NSGA-Ⅲ Algorithm for Solving High-dimensional Many-objective Green Flexible Job Shop Scheduling Problem[J]. Journal of System Simulation，2024，36(10)：2314-2329.

[27] XIE Jin，LI Xinyu，GAO Liang，et al. A New Neighbourhood Structure for Job Shop Scheduling Problems[J]. International Journal of Production Research，2023，61(7)：2147-2161.

[28] PALACIOS J J，PUENTE J，VELA C R，et al. Benchmarks for Fuzzy Job Shop Problems[J]. Information Sciences，2016，329：736-752.

[29] GAO Kai zhou，SUGANTHAN P N，PAN Quan ke，et al. An Improved Artificial Bee Colony Algorithm for Flexible Job-shop Scheduling Problem with Fuzzy Processing Time[J]. Expert Systems with Applications，2016，65：52-67.

[30] GU Xiaolin，HUANG Ming，LIANG Xu. A Discrete Particle Swarm Optimization Algorithm with Adaptive Inertia Weight for Solving Multiobjective Flexible Job-shop Scheduling Problem[J]. IEEE Access，2020，8：33125-33136.

[31] WU Xiuli，WU Shaomin. An Elitist Quantum-inspired Evolutionary Algorithm for the Flexible Job-shop Scheduling Problem [J]. Journal of Intelligent Manufacturing，2017，28(6)：1441-1457.

[32] LEI Deming，SU Bin. A Multi-class Teaching－Learning-based Optimization for Multi-objective Distributed Hybrid Flow Shop Scheduling[J]. Knowledge-Based Systems，2023，263：110252.

（编辑　张　洋）

作者简介：刘　设，女，1980年生，副教授、硕士研究生导师。研究方向为生产调度优化。发表论文18篇。E-mail：9858573@qq.com。田志强*（通信作者），男，1995年生，硕士研究生导师。研究方向为车间调度智能优化理论与方法。发表论文20余篇。E-mail：ZQ_Tian@sut.edu.cn。

本文引用格式：

刘设，李佳欣，田志强，等. 考虑客户满意度的模糊柔性作业车间调度优化方法[J]. 中国机械工程，2026，37(7)：1734-1745.

LIU She，LI Jiaxin，TIAN Zhiqiang，et al. Fuzzy Flexible Job-shop Scheduling Optimization Approach Considering Customer Satisfaction[J]. China Mechanical Engineering，2026，37(7)：1734-1745.