

doi: 10.7690/bgzdh.2026.03.015

## 一种考虑成本的柔性作业车间分批调度方法

张 硕<sup>1,2</sup>, 庄存波<sup>1,2</sup>, 郭昊鑫<sup>1</sup>, 高庆霖<sup>1</sup>

(1. 北京理工大学机械与车辆学院, 北京 100081;

2. 北京理工大学唐山研究院河北省智能装配与检测技术重点实验室, 河北 唐山 063000)

**摘要:** 为解决现有柔性作业车间分批调度中批次划分决策与生产成本关系建模不足的问题, 提出一种融合 NSGA<sub>2</sub> 种群进化机制与模拟退火算法的多目标进化算法 NSGA<sub>2</sub>\_SA。设计一种 3 段式编码方案, 用于表示批次划分、机器选择和工序排序; 设计一种基于动态加工信息的自适应变异算子, 以提高搜索效率; 引入模拟退火算法, 以增强局部搜索能力。小、中、大 3 种规模实例的对比实验结果表明, 所提算法具备高效性和稳定性。

**关键词:** 柔性作业车间调度问题; 分批调度; 成本; NSGA<sub>2</sub>

**中图分类号:** TP301.6 **文献标志码:** A

## A Cost-consideration-based Scheduling Approach for Flexible Job-shop Scheduling Problem with Lot Streaming

Zhang Shuo<sup>1,2</sup>, Zhuang Cunbo<sup>1,2</sup>, Guo Haoxin<sup>1</sup>, Gao Qinglin<sup>1</sup>

(1. School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China;

2. Hebei Province Key Laboratory of Intelligent Assembly and Detection Technology, Tangshan Research Institute, Beijing Institute of Technology, Tangshan 063000, China)

**Abstract:** In order to solve the problem of insufficient modeling of the relationship between the batch division decision and the production cost in the existing flexible job shop batch scheduling, a multi-objective evolutionary algorithm NSGA<sub>2</sub>\_SA is proposed, which integrates the NSGA<sub>2</sub> population evolution mechanism and the simulated annealing algorithm. A three-segment coding scheme is designed to represent batch partition, machine selection and process sequencing; an adaptive mutation operator based on dynamic processing information is used to improve the search efficiency; a simulated annealing algorithm is introduced to enhance the local search ability. The experimental results of small, medium and large scale examples show that the proposed algorithm has high efficiency and stability.

**Keywords:** flexible job shop scheduling problem; batch scheduling; cost; NSGA<sub>2</sub>

### 0 引言

柔性作业车间调度问题 (flexible job-shop scheduling problem, FJSP) 作为当前车间调度研究的热门领域, 现有研究多聚焦于单件调度模式。在实际生产场景中, 企业为提升生产效率、降低单位制造成本, 普遍采用批量生产模式。同时, 消费市场的需求也正呈现出多品种、变批量、高灵活性的显著特征, 这对传统生产调度模式提出了更高要求。在这一背景下, 柔性作业车间分批调度问题 (flexible job-shop scheduling problem with lot streaming, FJSP-LS) 通过分批调度技术<sup>[1-2]</sup>将作业分成多个子批次, 允许子批次进行独立的生产与运输操作, 使得后续处理流程得以提前启动<sup>[3]</sup>, 从而缩短完工时间, 提高生产效率。

以电子装备装配车间为例, 常采用批处理机对多个工件进行同步处理。在传统串行批处理模式下,

单个批次的处理时长由该批次内所有工件加工时间的总和决定, 使得先进入批次的工件必须等待整批全部完成才能进入后续工序, 造成设备资源阶段性闲置与生产流程阻塞。而分批调度技术通过优化批次容量分配, 不仅能显著降低设备准备操作频次与物料流转消耗, 而且可使已完成处理的工件批次即时转入下游工序。这种调度模式形成的生产节奏, 既缩短了整体生产周期, 又实现了设备利用率与资源配置效率的平衡。

FJSP-LS 的批次划分通常依据 2 类标准<sup>[4]</sup>: 1) 根据作业子批次大小是否相等, 可分为均等与不均等子批; 2) 根据同一作业在不同批次间的前后处理操作是否保持一致, 可分为一致子批与可変子批。

在求解策略上, FJSP-LS 主要分为 3 类研究方向: 第 1 类策略同时优化批量划分与调度问题<sup>[5-6]</sup>, 该集成优化策略有助于获得全局最优解, 但随着批

收稿日期: 2024-11-01; 修回日期: 2024-12-06

基金项目: 国家自然科学基金资助项目 (52375479); 国家重点研发计划 (2024YFB3312700)

第一作者: 张 硕 (2002—), 男, 河北人, 硕士。

次数量的增加, 求解复杂度显著上升; 第 2 类策略将批量划分与排产过程分开处理<sup>[7-8]</sup>, 即先优化批量划分方案, 再将优化结果作为输入, 转换为传统的调度问题求解, 该策略降低了求解难度, 但可能导致整体最优性下降; 第 3 类策略假设批量划分方案已知, 仅优化作业调度<sup>[9-10]</sup>, 能有效简化问题求解, 但其优化效果高度依赖于先验知识。

由于 FJSP-LS 的问题规模及求解复杂性, 精确算法难以取得理想效果, 学者更多采用近似算法进行求解。例如, Bozek 等<sup>[11]</sup>设计了一种基于禁忌搜索的 2 阶段算法求解可变均等分批单目标 FJSP-LS, 但未量化分批相关的成本影响。刘雪红等<sup>[12]</sup>设计了一种改进的候鸟算法处理可变不均等分批多目标 FJSP-LS 问题, 虽考虑了准备时间和运输时间对完工时间的影响, 却未涉及其关联成本。Li 等<sup>[13]</sup>、闫富乾等<sup>[14]</sup>、黎增灿等<sup>[15]</sup>的研究分别利用改进人工蜂群、入侵杂草和灰狼算法求解一致不均等分批单目标 FJSP-LS 问题, 均聚焦于准备时间对完工时间的影响, 未考虑相关成本因素。Fan 等<sup>[16]</sup>结合可变邻域搜索与遗传算法求解一致不均等分批情况下的可重构 FJSP-LS 问题, 关注了设备重构对完工时间的影响, 同样未探讨重构及分批带来的成本。

上述调研可见, 现有研究在解决 FJSP-LS 问题上虽取得进展, 却普遍存在一个关键局限: 对生产成本的系统性考量不足。相关研究或实际生产中的非处理时间, 单纯视为影响完工时间的因素<sup>[12-16]</sup>或完全忽略其影响<sup>[11]</sup>, 未能充分建模和优化这些活动直接引发的关联成本, 如额外运输成本、准备成本等。文献[1]和[17]进一步指出: 批次规模不当会通过影响瓶颈工序、设备利用率、准备与运输频率等途径, 显著增加能耗或生产成本。此外, 实际生产中非处理时间普遍存在<sup>[18]</sup>, 不仅会影响生产效率, 而且是重要的生产成本。现有研究文献对此成本维度的建模与优化存在明显缺失。

为弥补上述研究空白, 笔者以最小化最大完工时间和最小化分批成本为目标, 建立了柔性作业车间分批调度模型, 提出一种融合 NSGA<sub>2</sub> 种群进化机制与模拟退火算法(simulated annealing algorithm, SA)的多目标进化算法 NSGA<sub>2</sub>\_SA。以 NSGA<sub>2</sub> 作为全局搜索框架, 模拟退火算法为局部搜索机制。

## 1 问题描述及数学模型

### 1.1 问题描述及符号含义

一批任务( $J=\{J_j|J_1, J_2, \dots, J_n, i=1, 2, \dots, n\}$ )在车

间内进行处理, 每个任务  $J_j$  包括的作业数量为  $B_j$ , 每项作业都有一组预先指定顺序的  $O_j$  工序。并且任务  $J_j$  相应的作业所需要的每个工序( $O_{j,h}$ )都可以至少在总数为  $m$  台的机器集中找到一个能够处理此工序的机器。每个任务  $J_j$  都可以根据作业数量  $B_j$  与最大分批数  $S_{\max,j}$  分批为多个子批, 每个子批的作业工序顺序继承所属任务的工序顺序。约束条件为:

- 1) 每个子批的每道工序最终只能在一台机器上处理, 并且一旦工序开始处理, 过程不能被中断。
- 2) 机器一次只能处理单一作业的单个子批。
- 3) 每个子批的处理顺序是固定的, 当子批的一个工序完成之后, 将转移到下一个工序。
- 4) 每个子批的处理时间由子批数量以及所选机器决定。
- 5) 当同一子批的 2 个相邻工序在不同的机器上进行处理时, 会存在运输时间(运输时间是已知的)。
- 6) 由于所选的处理机器不同, 不同的工序具有不同的准备时间, 如果属于同一任务的 2 道工序在同一机器上连续处理, 则准备时间为 0。

文中使用的符号含义如表 1 所示。

表 1 模型参数符号及说明

符号	描述
$J$	待处理任务集合
$j$	待处理任务
$O_j$	任务 $j$ 的工序数
$B_j$	任务 $j$ 的作业数量
$S_{\max,j}$	任务 $j$ 的最大分批数
$S_j$	任务 $j$ 所形成的实际分批数
$B_{j,p}$	任务 $j$ 第 $p$ 批次所包含的数量
$O_{j,h}$	任务 $j$ 的第 $h$ 个工序
$M$	处理机器集合
$i$	处理机器
$P_{i,j,h}$	任务 $j$ 中单个作业在机器 $i$ 上处理第 $h$ 个工序的时间
$P_{i,j,p,h}$	任务 $j$ 第 $p$ 批次的第 $h$ 个工序在机器 $i$ 上的处理时间
$ST_{i,j,h}$	工序 $O_{j,h}$ 在机器 $i$ 上处理的准备时间
$ST_{i,j,h,j',h'}$	在机器 $i$ 上处理当前工序 $O_{j,h}$ 所需的准备时间(前道工序为 $O_{j',h'}$ )
$T_{i,i'}$	机器 $i$ 到机器 $i'$ 的运输时间
$F_{j,p,h}$	任务 $j$ 的第 $p$ 批次的第 $h$ 个工序的完成时间
$R_{i,j,h}$	如果 $O_{j,h}$ 能够在机器 $i$ 上加工, 则 $R_{i,j,h}=1$ , 否则为 0
$D_{j,j'}$	当 $j \neq j'$ 时, $D_{j,j'}=1$ , 否则为 0
$x_{i,j,p,h}$	当任务 $j$ 的第 $p$ 批次第 $h$ 个工序在机器 $i$ 进行处理时, 取值为 1, 否则为 0
$I_{j,p,h}$	处理任务 $j$ 的第 $p$ 批次的第 $h$ 个工序的机器序号
$U_{j,p,h,j',p',h'}$	如果任务 $j$ 第 $p$ 批次的第 $h$ 道工序在任务 $j'$ 第 $p'$ 的第 $h'$ 道工序之前, 则为 1, 反之则为 0
$C_{\max}$	最大完工时间
LSC	分批成本
$Q$	很大的正整数

### 1.2 约束关系

笔者研究的调度问题的优化目标为最小化最大完工时间和最小化分批成本。在分批过程中, 随着批次数量的增加, 子批次的启动将伴随更多的准备

时间和运输时间，从而导致分批成本上升；因此，笔者将调度过程中所有子批的准备时间与运输时间进行加权求和，作为衡量分批成本的指标。

目标函数，表示最小化最大完工时间和最小化分批成本：

$$f = \{\min C_{\max}, \min LSC\} \quad (1)$$

最大完工时间以及分批成本的定义：

$$C_{\max} = \max(F_{j,p,h}) \forall (j,p,h); \quad (2)$$

$$LSC = \sum_{j=1}^n \sum_{p=1}^{S_j} \sum_{h=2}^{O_j} T_{j,p,h} + \sum_{i=1}^n \sum_{h=1}^{I_{j,p,h-1}} ST_{i,j,h,j',h'} \quad (3)$$

实际处理情况受限于处理能力：

$$x_{i,j,p,h} \leq R_{i,j,h}, \quad \forall (i,j,p,h) \quad (4)$$

确保每个批次的每个工序都被处理：

$$\sum_{M_j \in M} x_{i,j,p,h} = 1, \quad \forall (j,p,h) \quad (5)$$

对决策变量进行了一些限制：

$$x_{i,j,p,h} \in \{0, 1\} \quad (6)$$

确保批次数量的正确性：

$$\sum_{p=1}^{S_j} B_{j,p} = B_j; \quad (7)$$

$$1 \leq S_j \leq S_{\max,j} \quad (8)$$

对决策变量进行了一些限制：

$$B_{j,p} > 0 \quad (9)$$

定义了准备时间：

$$ST_{i,j,h,j',h'} = \begin{cases} ST_{i,j,h} & , O_{j',h'} \neq \emptyset, D_{j,j'} = 1 \\ 0 & , \text{其他} \end{cases} \quad (10)$$

定义了批次的处理时间：

$$P_{i,j,p,h} = P_{i,j,h} * B_{j,p}, \quad \forall (i,j,p,h) \quad (11)$$

防止在同一机器上同一时间处理多道工序：

$$F_{j,p,h} \geq F_{j',p',h'} + P_{i,j,p,h} + ST_{i,j,h} * D_{j,j'} - (2 - x_{i,j,p,h} - x_{i,j',p',h'} + U_{j,p,h,j',p',h'}) \Omega; \quad (12)$$

$$F_{j',p',h'} \geq F_{j,p,h} + P_{i,j',p',h'} + ST_{i,j',h'} * D_{j,j'} - (3 - x_{i,j,p,h} - x_{i,j',p',h'} - U_{j,p,h,j',p',h'}) \Omega \quad (13)$$

确保同一作业同一批次的先后处理顺序：

$$F_{j,p,h} \geq F_{j,p,h-1} + T_{i,j,p,h-1} + P_{i,j,p,h} * x_{i,j,p,h} \quad (14)$$

## 2 算法设计

### 2.1 算法流程框架

求解车间调度问题时常使用多目标进化算法，而其中 NSGA<sub>2</sub> 是多目标进化算法中最经典的算法之一。算法使用了更易实现的精英保留策略和临界层拥挤距离排序策略，以保留种群的多样性，但仍旧存在容易陷入局部最优等缺点<sup>[19]</sup>。笔者设计一种改进的多目标进化算法 NSGA<sub>2</sub>-SA，以 NSGA<sub>2</sub> 的种群进化机制作为算法的全局搜索框架，并将模拟退火算法嵌入到后代种群生成过程中，作为局部搜索框架，并依据 FJSP-LS 的特点对其后代种群生成机制提出了相应改进策略，最终得到算法的整体流程如图 1 所示。

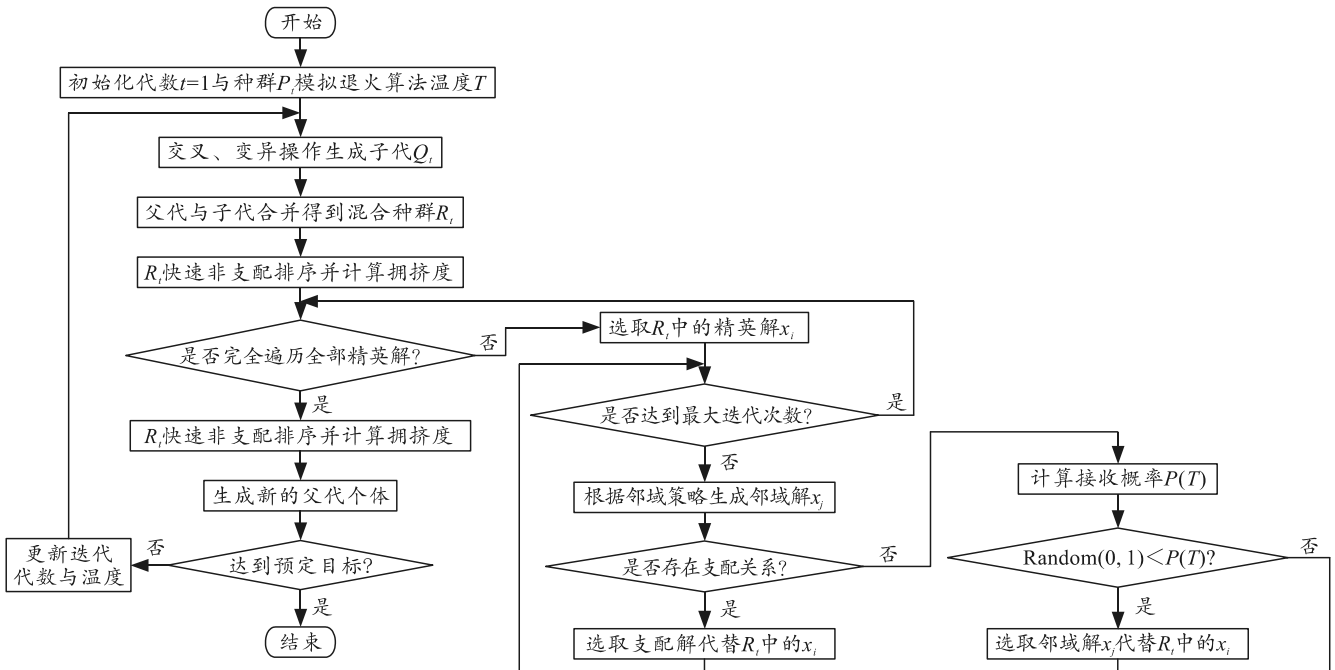


图 1 算法流程

### 2.2 编码方案和解码方案

#### 2.2.1 编码方案

笔者沿用了与文献[20]相同的批次编码方案，在限制最大子批数量的基础上，进一步优化批量大小和批次数目。同时，为了优化分批成本，在编码建模中引入了机器选择和工序排序。为此，笔者采用 LS+OS+MS 的 3 段式编码方案，在优化批次大小的同时，实现对机器选择和工序排序的同步优化，从而减少作业运输时间和设备准备时间，最终降低分批成本。

具体的编码方案，以表 2 为例。

表 2 某柔性作业车间实例

任务	工序	处理/准备时间			
		$M_1$	$M_2$	$M_3$	$M_4$
$J_1$	$O_{1,1}$	3/2	$J_1$	$O_{1,1}$	3/2
$J_1$	$O_{1,2}$	-	$J_1$	$O_{1,2}$	2/1
$J_2$	$O_{2,1}$	-	$J_2$	$O_{2,1}$	-
$J_2$	$O_{2,2}$	6/3	$J_2$	$O_{2,2}$	8/6
$J_2$	$O_{2,3}$	-	$J_2$	$O_{2,3}$	2/1
$J_3$	$O_{3,1}$	4/3	$J_3$	$O_{3,1}$	-
$J_3$	$O_{3,2}$	-	$J_3$	$O_{3,2}$	-

假设 3 个任务中每个作业的数量均为 10，且批次数量上限均为 4。对于某任务而言，若将 10 个作业分为 4 批，仅需在 LS 编码片段中添加 3 个游标编码，即可完成分批方案的表示；因此，LS 编码的长度为 3，其取值范围在 0~10。基于此，对于表 2 所示的实例，整个 LS 部分染色体的长度为 12，如图 2 所示。

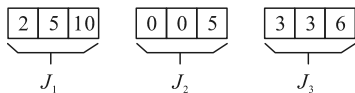


图 2 LS 染色体片段

对于表示任务  $J_1$  分批情况的编码片段来说，其游标的具体位置如图 3 所示。

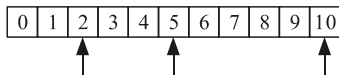


图 3 游标位置

表明任务  $J_1$  实际上被分为了数量分别为 2、3、5 的 3 个子批次。值得注意的是，在 LS 编码中引入 0 旨在保证编码方案的完备性如图 4 所示。

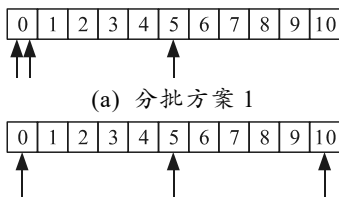


图 4 分批方案

图 4 的 2 个分批方案中，尽管两者均将当前任务分为数量为 5、5 的 2 个批次，但即便某个子批的数量为 0，其对应的 OS 和 MS 片段仍然在染色体中；因此，在后续解码过程中，这 2 个方案实际上是不同的分批方案。

在 OS 和 MS 编码片段中，需要根据批次数量上限对任务进行子批扩充，再按照每个子批的顺序，从前到后生成新的子批索引和工序序号。生成的部分子批如表 3 所示。

表 3 子批扩展表

任务	最大批次数	子批索引	工序
$J_1$	4	$B_1$	$O_{1,1}$
$J_1$	4	$B_1$	$O_{1,2}$
$J_1$	4	$B_2$	$O_{2,1}$
$J_1$	4	$B_2$	$O_{2,2}$
$J_1$	4	$B_3$	$O_{3,1}$
$J_1$	4	$B_3$	$O_{3,2}$
$J_1$	4	$B_4$	$O_{4,1}$
$J_1$	4	$B_4$	$O_{4,2}$
$J_2$	4	$B_5$	$O_{5,1}$
$J_2$	4	$B_5$	$O_{5,2}$
$J_2$	4	$B_5$	$O_{5,3}$
$J_2$	4	$B_6$	$O_{6,1}$
$J_2$	4	$B_6$	$O_{6,2}$
$J_2$	4	$B_6$	$O_{6,3}$
⋮	⋮	⋮	⋮

根据新生成的子批索引，接着对后续的 OS 和 MS 片段进行编码。

#### 2.2.2 解码方案

对 LS 染色体片段进行解码，根据每个任务的批次数上限值确定用于表达当前任务批次信息的 LS 基因片段。截取该片段后，可以解码出每个子批的实际批量大小。以图 2 为例，对于任务  $J_1$ ，只需截取 LS 染色体的前 3 个基因，即可解码出任务  $J_1$  的分批方案，该方案将任务分为 4 个子批，批量大小分别为 2、3、5 和 0，实际形成了 3 个子批。

对 MS 染色体片段进行解码，从左到右依次读取 MS 基因位的信息，并根据表 2 中的工序处理时间和 LS 解码得到的当前批次数量信息，计算出每个批次的处理时间。

成功解码 MS 染色体后，再从左到右读取 OS 染色体，以确定工序的顺序。并根据机器相邻工序的准备时间与子批相邻工序的运输时间对工序进行排序。排序方法为：

- 1) 如果工序  $O_{j,h}$  是作业  $J_j$  的第一个操作，同样也是待处理机器  $i$  上的第一个操作，则工序在机器准备完成之后立即进行处理。
- 2) 如果工序  $O_{j,h}$  不是作业  $J_j$  的第一个操作，却是待机器  $i$  上的第一个操作时，则一旦工序  $O_{j,h-1}$

完成, 工序  $O_{j,h}$  便会选择待处理机器准备好的时间与子批运输时间的最大值作为起始时间进行处理。

3) 当工序  $O_{j,h}$  不是作业  $J_j$  的第一个操作, 并且待处理机器  $i$  已经处理过其他工序时, 必须考虑待处理机器  $i$  上的空闲时间间隔是否满足插入的条件, 如图 5 为展示工序插入的情况。

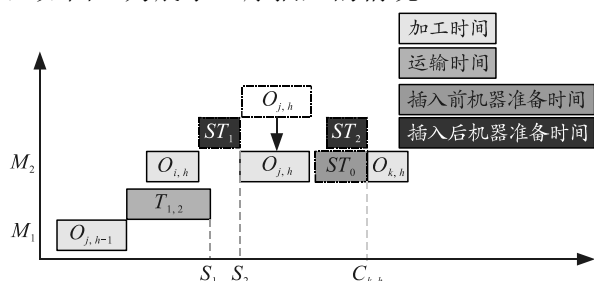


图 5 工序插入

### 2.3 改进策略

由于初始解的质量对算法的求解速度和求解质量有较大影响, 为了提高初始解的质量以及搜索能力, 笔者基于 FJSP-LS 的特点, 并结合染色体的 3 段式编码方式, 设计了种群初始化方法。

在 LS 部分, 为了保证初始解的质量, 一半的种群采用均等分批的策略生成初始化种群, 即每个子批的数量根据子批数量上限和当前任务的数量决定。对于无法整除的情况, 子批数量向下取整, 剩余的作业数量分配给最后一个子批。

为了保证解的多样性, 一半的种群采取随机分批策略, 即每个任务对应的 LS 编码在编码规则的限制下随机生成。

在 MS 部分, 为了保证解的多样性和质量, 同样采用了 2 种生成策略: 一半的种群采取随机生成策略, 在 MS 部分基因位上的数字根据工序可选机器集中的机器数量进行随机生成; 另一半的种群则采用全局选择策略, 具体执行步骤见文献[21]。

在 OS 部分, 采用随机生成策略, 即将与批次工序数量相对应的批次数序号随机插入染色体片段中, 以保证解的多样性。

变异操作可以通过改变个体的部分基因来增加解决方案的多样性, 从而使算法能够探索到解空间中的更多可行解。笔者针对 LS 和 MS 部分的变异策略进行了改进。

1) 在 LS 部分, 由于游标的位置决定了批次的大小, 在基因位可选范围内随机进行变异可能会降低变异操作的效率; 因此, 可以根据当前基因位的信息来指导变异操作, 从而更高效地探索解空间。具体实现步骤为:

① 获取当前基因位的取值  $L_1$ , 并根据  $L_1$  基因位能够取到的最大值  $L_{\max}$  确定变异长度  $L_{\text{shift}}$ 。

② 计算变异范围, 即  $[L_1 - L_{\text{shift}}, L_1 + L_{\text{shift}}]$  与  $[0, L_{\max}]$  的交集, 得到当前基因位的变异范围。

③ 在变异范围中随机选取一个数字  $L'_1$ , 作为当前基因位变异后的取值。

④ 对不合法染色体进行修正。

2) 在 MS 部分, 变异操作基于当前基因位所选机器的处理时间来决定变异。具体方法是, 确定当前工序的处理时间小于当前机器的所有机器序号, 随机选取一个机器作为 MS 变异后的取值。

3) 在 OS 部分, 则通过随机选择多个基因位位置进行调换来进行变异。

笔者采用一种具有较强引导性的变异策略, 这种策略通过在某些特定方向上进行变异加速了局部搜索, 但可能导致搜索陷入局部最优解。为了克服该问题, 引入了模拟退火算法<sup>[22]</sup>。该算法通过以一定概率接收较劣解, 并将其纳入搜索过程, 从而增强算法的局部搜索能力。

具体而言, 笔者设计的模拟退火算法通过从当前解出发, 以相等的概率分别生成 2 部分邻域解: 工序排序邻域解和机器选择邻域解。其中工序排序邻域解通过随机选取一对 OS 基因位调换其取值生成, 机器选择邻域解则通过随机选取一个 MS 基因位随机变异生成。在决定是否接受新邻域解时, 算法根据新邻域解与当前解的支配关系以及温度设定的接收概率来判断。接收规则如下:

① 当邻域解与原解存在 Pareto 支配关系时, 严格遵循支配关系选择非劣解。

② 当邻域解与原解互不支配时, 引入基于 Metropolis 准则的模拟退火机制, 其接受新邻域解的概率  $P_{\text{accept}}$ :

$$P_{\text{accept}} = e^{-\Delta_{\text{objective}}/T}; \quad (15)$$

$$T = T_0 * \alpha^t; \quad (16)$$

$$\Delta_{\text{objective}} = \sum |f_0[i] - f_1[i]|. \quad (17)$$

式中:  $\Delta_{\text{objective}}$  为目标函数差异值, 是邻域解与原解各目标差值的绝对值之和;  $T$  为迭代温度;  $T_0$  为初始温度;  $\alpha$  为冷却速率;  $t$  为迭代代数。

## 2.4 交叉策略与选择策略

### 2.4.1 交叉策略

在 LS 部分, 为了保证交叉后子代解的合法性, 笔者采用了均匀交叉方法。每个基因点独立决定是

否交换，既能增加种群的基因多样性，又能使父代信息的融合更加均衡；在 MS 部分，采用多点交叉策略，通过随机选择父代染色体中的多个位置，并在这些位置之间进行基因替换，以实现信息的交换与遗传多样性的增加；在 OS 部分，由于每个批次的工序处理顺序受到约束，传统的交叉方法容易导致生成非法子代。笔者采用了基于作业优先顺序的交叉 (POX) 方法来进行 OS 部分的染色体交叉，具体执行步骤见文献[21]。

### 2.4.2 选择策略

在选择操作上，笔者采用了 NSGA<sub>2</sub> 中基于快速非支配排序方法以及拥挤距离方法的后代种群选择规则，详细步骤与思路见文献[23]。

## 3 实验部分

### 3.1 计算机配置

所有实验代码均由 Python 3.7 编写实现，测试实例在配备 Intel(R) Core i5-12490F CPU 3.00 GHz 和 32.0 GB RAM 的计算机上运行。

### 3.2 测试实例

笔者在文献[18]提出的考虑准备时间以及运输时间的多个 FJSP 实例的基础之上，将单个作业扩展为多个作业。实例 1 包含 4 种不同类型的作业和 5 台机器；实例 2 包含 8 种不同类型的作业和 8 台机器；实例 3 包含 15 种不同类型的作业和 10 台机器，且每道工序可由多台机器处理。各实例中工序的处理时间、准备时间与运输时间详见文献[18]。在此设置基础上，每种作业的数量均扩展至 40。

### 3.3 算法参数设置

为了算法公平性，所有算法的参数设置均保持一致，如表 4 所示。算法的种群规模 NIND 设置为 100，最大迭代次数 MAXGEN 设置为 100。对于混合编码，均采用相同的 3 段式编码，将组批上限设置为 4 个批次，并将染色体编码交叉率设置为 0.8，变异概率设为 0.2。在改进策略中，模拟退火算法的初始温度为 100，冷却速度为 0.95，模拟退火局部搜索最大迭代代数设置为 3，精英种群比例为 0.2。

表 4 算法参数设置

NIND	MAXGEN	LotUb	$P_c$	$P_m$
100	100	4	0.8	0.2
$T_0$	$\alpha$	MaxIterNum	EliteRate	
100	0.95	3	0.2	

### 3.4 算法对比与分析

为验证 3 种改进策略的有效性，笔者以标准

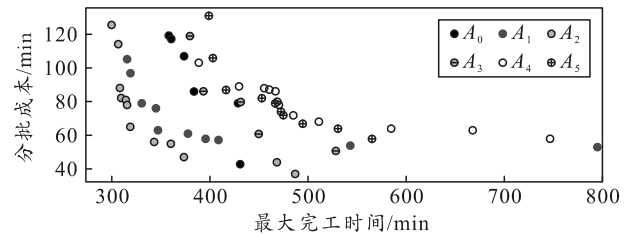
NSGA<sub>2</sub> 为基准算法开展对比实验。由于现有 FJSP-LS 研究中多目标种群进化算法的应用相对有限，为验证整体算法性能，将其与现有 FJSP 研究中使用较广的多目标进化算法 NSGA<sub>3</sub><sup>[24]</sup>、RVEA<sup>[25]</sup> 进行比较。其中 EM<sub>1</sub> 使用了种群初始化策略，EM<sub>2</sub> 使用了种群初始化策略和自适应变异算子策略，EM<sub>3</sub> 集成 3 种改进策略。NSGA<sub>3</sub> 是基于参考点对后代进行筛选的多目标遗传算法，RVEA 是一种针对高维问题提出的基于参考向量的进化算法。为了便于表示，对将各算法按表 5 对应规则编号为 A<sub>0</sub> 至 A<sub>5</sub>。为了保证结果的泛化性并降低随机性，每个算法均独立运行了 30 次。

表 5 对比算法

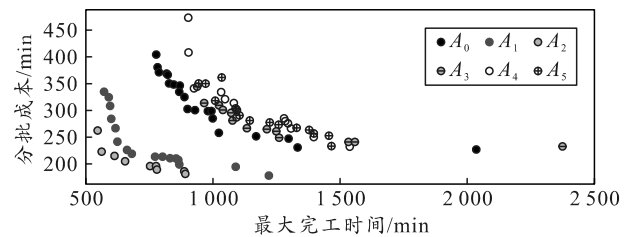
EM <sub>1</sub>	EM <sub>2</sub>	EM <sub>3</sub>	NSGA <sub>2</sub>	NSGA <sub>3</sub>	RVEA
A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>

#### 3.4.1 IPF 散点图

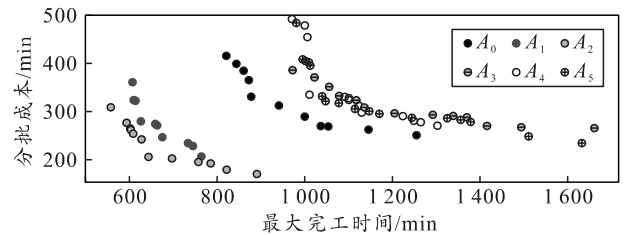
现对这 6 种算法在 3 个实例的散点图中专门分析逼近最优解的能力。具体操作为运行每种算法 30 次，并将最终得到的非支配解集进行并集操作，从中提取非支配解前沿 PF，以此作为算法逼近最优解能力的表现。如图 6 所示，展示了在 3 个实例中，6 种算法的 PF 散点图对比结果。



(a) 实例 1



(b) 实例 2



(c) 实例 3

图 6 6 种算法的 PF 散点图

从散点图中可以看出：3 种改进算法在逼近最

优解的能力上明显优于其他 3 种多目标优化算法。 $A_0$  到  $A_2$ ，算法逼近最优解的能力逐渐增强，进一步验证了所提改进策略的有效性。具体来看，在小规模算例实例 1 中，受限于解空间规模， $A_0$ 、 $A_3$  至  $A_5$  算法经多轮进化后表现趋同；相比之下，在更大规模、更高复杂度的实例 2 和实例 3 中，传统多目标算法( $A_3$ - $A_5$ )因搜索机制局限，难以高效探索庞大的解空间，导致收敛效率显著下降。融合改进策略的  $A_1$  与  $A_2$  算法，通过增强全局探索与局部开发能力，能够更高效地收敛至最优解区域。

### 3.4.2 IGD 指标趋势图

指标趋势是指随着迭代次数的增加，算法指标不断变化的过程。指标趋势图能够反映算法在每代迭代中的优化表现。针对 3 种规模的 3 组测试实例，笔者对 6 种算法分别运行 30 次，计算每代种群的反世代距离 (inverted generational distance, IGD) 取平均值，以此代表算法运行过程中的平均性能。如图 7 所示，展示了在 3 个实例中，6 种算法的 IGD 指标趋势图对比结果。

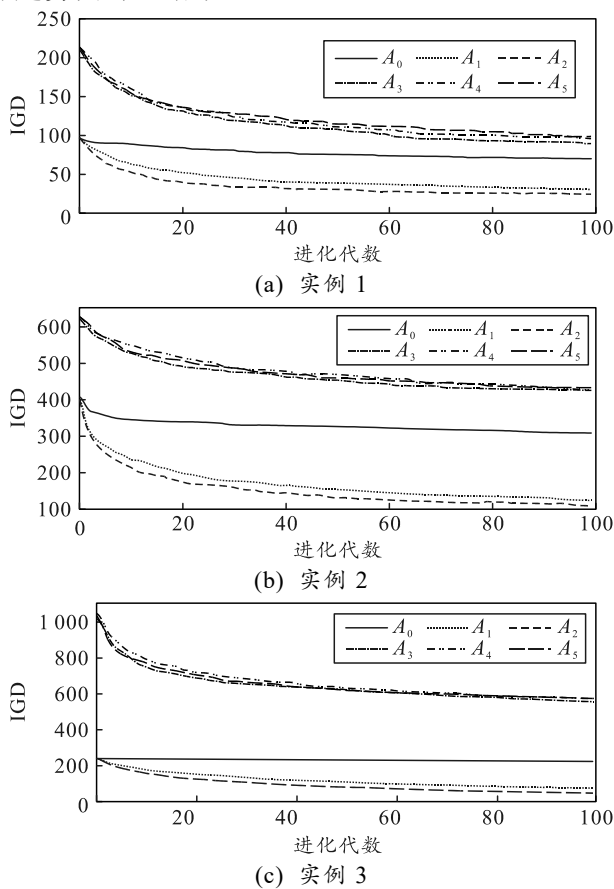


图 7 6 种算法的 IGD 指标趋势

从 IGD 指标趋势图中可以明显看出：3 种改进策略的有效性，得益于混合种群初始化策略，改进

算法  $A_0$  至  $A_2$  在第 0 代的 IGD 值已显著低于随机初始化种群的对比算法  $A_3$  至  $A_5$ ，证实了该策略在生成高质量初始解方面的有效性。基于高质量初始种群，算法  $A_0$ 、 $A_1$ 、 $A_2$  在实例 1 和 2 中均展现出优于对比算法的优化能力。在更大规模的实例 3 中，算法  $A_0$  出现了 IGD 值停滞现象，而  $A_1$ 、 $A_2$  仍保持显著下降趋势，这一差异表明固定参数的随机变异算子在复杂解空间中难以维持有效的全局探索，容易陷入局部最优；自适应变异机制显著提升了算法的探索效率和摆脱局部最优的能力。此外，在 3 个不同规模的算例中，融合 SA 局部搜索策略的  $A_2$  算法，其收敛速度较仅含自适应变异的  $A_1$  算法进一步提升。验证了所设计的 SA 局部搜索机制在加速算法收敛至高质量 Pareto 前沿方面的有效性。

### 3.4.3 IGD 指标箱线图

对 6 种算法分别独立运行 30 次，计算生成的最终非支配解集的 IGD 指标，并形成共 30 个指标数据。图 8 展示了 3 种不同规模的实例下，各算法的箱线图。

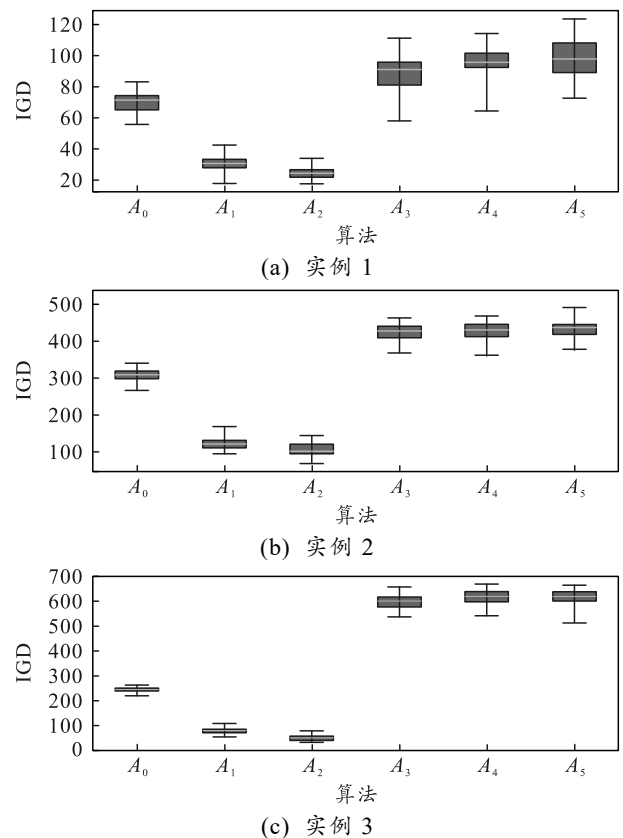


图 8 6 种算法的 IGD 指标箱线

从图中可以看出：3 种实例下， $A_0$ 、 $A_1$  和  $A_2$  的 IGD 中位数均小于其他算法，表明 3 种算法在解的质量上是最高的，同时  $A_0$ 、 $A_1$  和  $A_2$  相比其余算法

还有着较短的四分位数间距,表明算法具有不错的稳定性。

#### 4 结束语

笔者研究考虑成本的柔性作业车间分批调度问题,主要贡献为:

1) 针对传统柔性作业车间分批调度研究中对分批所需准备成本和运输成本建模不准确的问题,提出一种新的优化模型。该模型量化了与准备时间、运输时间相关的分批成本,以及与处理时间相关的生产效率,并建立了合理可行的数学模型。

2) 针对考虑成本的柔性作业车间分批调度问题,提出了 NSGA<sub>2</sub>\_SA 算法。该算法以 NSGA<sub>2</sub> 为全局搜索框架,SA 为局部搜索机制,针对种群初始化和变异策略提出了相应改进策略,从而提高了求解质量和效率。

3) 通过对扩展的小、中、大 3 种规模的测试实例进行实验,笔者将提出的改进算法与 NSGA<sub>2</sub>、NSGA<sub>3</sub> 和 RVEA 算法进行了对比,验证了改进策略在实际应用中的有效性。

未来的研究方向将聚焦于:精确建模批次上限,并深入探究其与最大完工时间及成本之间的量化关系;构建一个统一的优化框架,将批次上限、批量划分、工序排序和机器选择等问题同步优化,以提高整体调度效率。此外,为更好地契合企业实际生产需求,将进一步扩展准备时间模型,使其更精确地反映制造单元能力重构时间,从而更贴近真实生产环境。

#### 参考文献:

- [1] TIAN Z Q, JIANG X Y, TIAN G D, et al. Knowledge-Based Lot-Splitting Optimization Method for Flexible Job Shops Considering Energy Consumption[J]. Automation Science and Engineering, 2023, 21(3): 1-12.
- [2] 王海燕, 王万良, 黄凤立, 等. 分批优化调度问题综述及面向节能研究展望[J]. 计算机集成制造系统, 2017, 23(3): 542-556.
- [3] PAN Q K, RUIZ R. An estimation of distribution algorithm for lot-streaming flow shop problems with setup times[J]. Omega, 2012, 40(2): 166-180.
- [4] ZHANG B, PAN Q K, LEI L M, et al. A collaborative variable neighborhood descent algorithm for the hybrid flowshop scheduling problem with consistent sublots[J]. Applied Soft Computing, 2021, 106(Suppl C): 107305.
- [5] ZHANG B, PAN Q K, MENG L L, et al. A decomposition-based multi-objective evolutionary algorithm for hybrid flowshop rescheduling problem with consistent sublots[J]. Production Research, 2022, 61(3): 1013-1038.
- [6] YANG S L, XU Z G. The distributed assembly permutation flowshop scheduling problem with flexible assembly and batch delivery[J]. Production Research, 2021, 59(13): 4053-4071.
- [7] WANG S J, WANG X D, CHU F, et al. An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production[J]. Production Research, 2019, 58(8): 1-32.
- [8] WANG S S, KURZ M, MASON S J, et al. Two-stage hybrid flow shop batching and lot streaming with variable sublots and sequence-dependent setups[J]. Production Research, 2019, 57(22): 6893-6907.
- [9] MENG T, PAN Q K. A distributed heterogeneous permutation flowshop scheduling problem with lot-streaming and carryover sequence-dependent setup time[J]. Swarm and Evolutionary Computation, 2021, 60: 100804.
- [10] SANG H Y, GAO L, PAN Q K. Discrete Artificial Bee Colony Algorithm for Lot-streaming Flowshop with Total Flowtime Minimization[J]. Mechanical Engineering, 2012, 25(5): 990-1000.
- [11] BOZEK A, WERNER F. Flexible job shop scheduling with lot streaming and subplot size optimisation[J]. Production Research, 2018, 56(19): 6391-6411.
- [12] 刘雪红, 段程, 王磊. 基于改进候鸟算法的柔性作业车间分批调度问题[J]. 计算机集成制造系统, 2021, 27(11): 3185-3195.
- [13] LI X L, LU J S, YANG C X, et al. Research of Flexible Assembly Job-Shop Batch-Scheduling Problem Based on Improved Artificial Bee Colony[J]. Frontiers in Bioengineering and Biotechnology, 2022, 10: 909548.
- [14] 闫富乾, 陈浩杰, 丁国富, 等. 求解柔性分批调度问题的双层搜索框架入侵杂草算法[J]. 计算机集成制造系统, 2023, 29(2): 556-567.
- [15] 黎增灿, 丁林山, 管在林. 基于改进灰狼优化算法的柔性作业车间分批调度问题研究[J]. 现代制造工程, 2024(4): 13-25.
- [16] FAN J X, ZHANG C J, SHEN W M, et al. A matheuristic for flexible job shop scheduling problem with lot-streaming and machine reconfigurations[J]. Production Research, 2023, 61(19): 6565-6588.
- [17] 徐本柱, 费晓璐, 章兴玲. 柔性作业车间批量划分与并行调度优化[J]. 计算机集成制造系统, 2016, 22(8): 1953-1964.
- [18] ZHANG G H, HU Y F, SUN J H, et al. An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints[J]. Swarm and Evolutionary Computation, 2020, 54: 100664.
- [19] LIU Z W, LIU J H, ZHUANG C B, et al. Multi-objective complex product assembly scheduling problem considering parallel team and worker skills[J].

- Manufacturing Systems, 2022, 63(Suppl C): 454-470.
- [20] 白俊杰, 龚毅光, 王宁生, 等. 批量生产柔性作业车间优化调度研究[J]. 机械科学与技术, 2010(3): 293-298.
- [21] 高亮, 张国辉, 王晓娟. 柔性作业车间调度智能算法及其应用[M]. 武汉: 华中科技大学出版社, 2012.
- [22] KIRKPATRICK S, GELATT C D, VECCHI M P. Optimization by Simulated Annealing[J]. Computer Vision, 1983, 220(4598): 606-615.
- [23] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J].  
\*\*\*\*\*  
(上接第 67 页)
- [4] 林德福, 王辉, 王江, 等. 战术导弹自动驾驶仪设计与制导律分析[M]. 北京: 北京理工大学出版社, 2012: 49-59.
- [5] 张亚婷, 王伟. 基于开环穿越频率的伪攻角反馈自动驾驶仪设计[J]. 航天控制, 2015, 33(3): 19-28.
- [6] 周春成. 三回路自动驾驶仪结构分析与设计[J]. 工业控制计算机, 2017, 30(3): 60-62.  
\*\*\*\*\*  
(上接第 71 页)
- [3] 刘皓璐, 邵建伟, 王雪, 等. 基于数字孪生的配电自动化终端设备状态评价与故障预判[J]. 电网技术, 2022, 46(4): 1605-1613.
- [4] 杨玉莲, 齐林海, 王红, 等. 基于生成对抗和双重语义感知的配电网量测数据缺失重构[J]. 电力系统自动化, 2020, 44(18): 46-54.
- [5] 耿俊成, 郭志民, 李晓蕾, 等. 基于 LOF 和 SVM 的配电网线变关系数据校验方法[J]. 中国测试, 2021, 47(4): 49-54.
- [6] 李国栋, 王崢, 贾东梨, 等. 基于关联分析算法的配电网多级冗余数据校验与挖掘技术研究[J]. 电测与仪表, 2020, 57(15): 65-70.
- [7] 张文轩, 李京, 陈平, 等. 基于配电自动化终端的含 DG 配电网故障定位优化算法[J]. 水电能源科学, 2021, 39(7): 192-196.
- [8] 王鑫, 赵龙, 张淑娟, 等. 面向配电网大数据的自组织映射知识融合算法[J]. 合肥工业大学学报(自然科学版), 2022, 45(5): 620-624, 653.
- [9] 魏明奎, 叶葳, 沈靖, 等. 基于自组织特征神经网络和 Evolutionary Computation, 2002, 6(2): 182-197.
- [24] DEB K, JAIN H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints[J]. Evolutionary Computation, 2014, 18(4): 577-601.
- [25] CHENG R, JIN Y, OLHOFFER M, et al. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization[J]. Evolutionary Computation, 2016, 20(5): 773-791.  
\*\*\*\*\*  
[7] 钱杏芳, 林瑞雄, 赵亚男. 导弹飞行力学[M]. 北京: 北京理工大学出版社, 2008: 175-203.
- [8] GRAHAM D, LATHROP R C. The Synthesis of Optimum Transient Response Criteria and Standard Forms[J]. AIEE Trans, 1953(72): 273.
- [9] SCHULTZ W C, RIDEOUT C V. Control System Performance Measures: Past, Present and Future[J]. IRE Automatic Control, 1961, AC-6(1): 22-35.
- [10] ELLIS G. 控制系统设计指南[M]. 北京: 机械工业出版社, 2016: 111-112.  
\*\*\*\*\*  
最小二乘支持向量机的短期电力负荷预测方法[J]. 现代电力, 2021, 38(1): 17-23.
- [10] 刘达, 陈松灿. 基于高斯神经元的自组织映射网络研究[J]. 数据采集与处理, 2023, 38(1): 85-92.
- [11] 吴友婷, 杨洋, 梁湘三. 基于自组织映射的日本南部黑潮与黑潮延伸体的典型时空模态及其因果关系研究[J]. 海洋学报, 2022, 44(9): 38-54.
- [12] 王一旭, 王飞儿, 俞洁. 基于自组织映射与随机森林耦合模型的流域水质空间差异性评估[J]. 环境科学学报, 2020, 40(6): 2278-2285.
- [13] 林靖怡, 王尧, 李奎, 等. 基于自组织特征映射网络的电弧故障检测方法[J]. 电力自动化设备, 2020, 40(8): 210-219.
- [14] 王福琳, 杨超, 杜建, 等. 基于自组织映射网络的中老年人慢性病关联及共病模式可视化聚类分析[J]. 解放军医学杂志, 2022, 47(12): 1217-1225.
- [15] 张焕, 张庆, 于纪言. 卷积神经网络中激活函数的性质分析与改进[J]. 计算机仿真, 2022, 39(4): 328-334.
- [16] 王华鲜, 华容, 刘华平, 等. 无人机群多目标协同主动感知的自组织映射方法[J]. 智能系统学报, 2020, 15(3): 609-614.