

# 强化学习引导的产品变更路径多目标差分进化算法

宋贤芳, 杨扬, 张勇<sup>†</sup>, 郑瑞钊

(中国矿业大学 信息与控制工程学院, 江苏 徐州 221116)

**摘要:** 由于零部件之间复杂的关联关系, 产品设计变更效应的传播在所难免. 为降低产品设计变更带来的风险, 以变更对产品综合性能影响、变更经济成本和变更工期作为优化目标, 本文提出了一种强化学习引导的产品变更路径多目标差分进化算法. 首先, 建立问题的复杂产品网络模型, 揭示产品零件变更的传播机制; 接着, 引入变更传播强度, 间接评价零件变更对产品综合性能的影响; 同时, 考虑变更经济成本和变更工期指标, 建立产品设计变更传播路径的多目标优化模型; 进一步, 利用双深度Q-网络帮助种群在不同阶段选择适合的进化策略, 提出一种强化学习引导的差分进化算法, 简称为DDQN-DE, 并通过上述算法求解最佳的产品设计变更传播路径. 最后, 以创维公司某型号电视机的设计变更问题为例, 并与已有算法进行对比, 实验验证了所提算法的有效性.

**关键词:** 多目标优化; 设计变更; 差分进化; 强化学习

**引用格式:** 宋贤芳, 杨扬, 张勇, 等. 强化学习引导的产品变更路径多目标差分进化算法. 控制理论与应用, 2025, 42(1): 109 – 117

DOI: 10.7641/CTA.2024.30149

## Reinforcement learning guided multi-objective differential evolutionary algorithm for product change paths

SONG Xian-fang, YANG Yang, ZHANG Yong<sup>†</sup>, ZHENG Rui-zhao

(School of Information and Control Engineering, China University of Mining and Technology, Xuzhou Jiangsu 221116, China)

**Abstract:** Due to the complex interrelationships between components, the propagation of product design change effect is inevitable. To reduce the risk associated with product design changes, this paper proposes a multi-objective differential evolution algorithm guided by reinforcement learning to optimize the impact of changes on the product performance, economic cost, and change duration. Firstly, a complex product network model is established to reveal the propagation mechanism of component changes. Then, the change propagation intensity is introduced to indirectly evaluate the impact of component changes on the product performance. Meanwhile, considering the economic cost and change duration of design changes, a multi-objective model for optimizing design change propagation paths is established. Furthermore, utilizing dual deep Q-networks to assist populations in selecting appropriate evolutionary strategies at different stages, introducing a reinforcement learning-guided differential evolution algorithm, abbreviated as DDQN-DE, and using the above-mentioned algorithm to determine the optimal propagation path for product design changes. Finally, taking the design change problem of a TV product from skyworth company as an example and comparing with existing algorithms, the validity of the proposed algorithm is verified through experiments.

**Key words:** multiobjective optimization; design change; differential evolution; reinforcement learning

**Citation:** SONG Xian-fang, YANG Yang, ZHANG Yong, et al. Reinforcement learning guided multi-objective differential evolutionary algorithm for product change paths. *Control Theory & Applications*, 2025, 42(1): 109 – 117

## 1 引言

设计变更在复杂产品研发过程中不可避免<sup>[1]</sup>. 由于客户需求、供应链中断、法律法规变化等原因, 现实生产经常需要重新设计整个产品或其关键零部件<sup>[2]</sup>.

由于产品零部件之间复杂的关联关系, 一个零部件的设计变更可能导致其他零部件发生连锁变更, 从而引发变更传播<sup>[3]</sup>. 如果不能有效控制产品设计变更传播, 势必对产品综合性能、成本和研发工期产生负面影响

收稿日期: 2023-03-21; 录用日期: 2024-02-29.

<sup>†</sup>通信作者. E-mail: yongzh401@126.com.

本文责任编辑: 王凌.

国家自然科学基金项目(62203446), 江苏高校“青蓝工程”项目资助.

Supported by the National Natural Science Foundation of China (62203446) and the Qing Lan Project of Jiangsu University.

响<sup>[4]</sup>。因此,研究变更传播路径优化问题,可以帮助决策者选择最佳的变更方案,减小零部件变更对产品综合性能、成本和研发工期等的负面影响。

研究控制产品设计问题时,首先需要建立一个合理的产品零件关联模型。Yang等<sup>[5]</sup>使用设计结构矩阵(design structure matrix, DSM)定量表示产品组件间的关联关系,完成了产品零部件关系的可视化。针对DSM仅适用于某些特定产品实例的缺点,Diagne等<sup>[6]</sup>提出了扩展概念设计语义矩阵,可以为DSM提供额外的信息。李玉鹏等<sup>[7]</sup>充分考虑产品变更时零部件的变更传播流向,构建了面向复杂产品网络结构的有向加权网络。这些工作为有效分析产品变更传播机制奠定了良好的基础,但是,他们都未考虑与用户体验相关的产品综合性能指标,使得构建的复杂产品网络模型难以满足实际需求。

在建立产品零件关联模型后,需要定义一个合适的指标来评估设计变更对产品综合性能的影响。Dong等<sup>[8]</sup>根据网络变更率、变更放大率和变更放大节点率指标,定义了一种基于变更工作量的综合评价指标。该指标独立于专家经验,为产品设计变更问题提供了一种更为清晰和客观的变更影响评估方法。Ma等<sup>[9]</sup>同时考虑变更传播概率、设计裕度、节点度和长程连接信息,定义了一种综合评价指标,用于评估产品变更传播强度(change propagation intensity, CPI)。上述指标可以从不同角度评价设计变更对产品网络模型的影响,然而,由于已有复杂产品网络模型的自身局限性,这些指标并不能反映产品设计变更对综合性能的影响。因此,目前针对零件变更对产品综合性能影响的揭示和量化研究成果仍较少。

此外,为了有效控制产品变更传播的范围,还需要寻找最优的产品设计变更传播路径。少数学者尝试将具有全局搜索能力的进化算法用于搜索最优传播路径。针对复杂产品设计变更传播路径优化问题,骆建芬等<sup>[10]</sup>建立了设计变更传播路径问题的多目标优化模型,并采用带精英策略的非支配排序遗传算法II(non-dominated sorting genetic algorithm II, NSGA-II)算法求解上述模型。针对多源设计变更,为了最小化CPI、变更风险成本和传播路径长度,Ren等<sup>[11]</sup>建立了设计变更传播路径问题的多目标优化模型,并给出了一种基于NSGA-II的求解算法。上述算法可以为设计师提供多个变更方案。然而上述算法还存在以下不足:一方面,由于复杂产品网络模型自身局限性,算法在寻找最优路径时未考虑变更对产品综合性能的影响;另一方面,由于采用了传统的多目标进化算法,已有算法依然存在收敛速度慢、易陷入局部最优等缺点。

差分进化(differential evolution, DE)算法是由Storn等<sup>[12]</sup>提出的一种基于种群的进化算法,其具有较强

的鲁棒性、搜索能力,且易于实现。曹蓓等<sup>[13]</sup>设计具有修复功能的参数自适应策略,开发了改进DE算法。Tan等<sup>[14]</sup>通过适应值曲面来指导变异策略的选择,给出了具有自适应变异算子的DE算法。Wu等<sup>[15]</sup>设计修复策略和删除策略,提出了快速有效的改进DE算法。上述改进的DE算法均提高了算法的竞争力,但是,由于只考虑算法本身,仍然很容易陷入局部最优。

强化学习(reinforcement learning, RL)<sup>[16]</sup>是机器学习中的一种著名范式和方法论。DE的进化过程可以看作是一个有限马尔可夫决策过程<sup>[17]</sup>,其中种群信息相当于环境状态,DE策略相当于动作。鉴于此,针对多模态多目标优化问题, Li等<sup>[18]</sup>提出了一种基于RL的DE算法,通过Q-learning算法动态调整群体的进化方向。针对结构优化问题, Huynh等<sup>[19]</sup>提出了一种基于RL的参数控制方法,通过Q-learning算法在每次搜索迭代时自适应调整算法的控制参数。上述方法均通过搜索Q值表来计算值函数,当状态连续或状态集很大时,查找和存储Q值表代价相对较大。为此, Tan等<sup>[20]</sup>提出了一种基于深度Q-网络(deep Q-network, DQN)的DEDQN算法,采用深度RL方法实现进化过程中DE策略的自适应选择。然而, DQN通过执行当前时刻最大Q值对应的动作来优化策略,容易过高估计某个动作的价值。进一步,学者们提出了双深度Q-网络(double deep Q-network, DDQN)<sup>[21]</sup>,将动作选择和动作价值评估分开,降低了过优估计的风险。因此,本文利用DDQN来选择种群在不同阶段的变异策略。

针对已有工作的不足和实际需求,研究一种融合RL和DE的产品设计变更传播路径多目标优选方法。同时考虑变更传播对产品综合性能影响、变更经济成本和变更工期的最小化,以及问题的约束特性,将复杂产品设计变更传播路径优选问题建模为一个约束多目标优化模型,并设计一种RL引导的DE算法求解该模型。本文的主要贡献如下:

1) 在考虑最小化变更经济成本和变更工期等常用指标的情况下,利用变更传播强度评价变更对产品综合性能的影响;同时考虑问题的多约束特性,建立产品设计变更传播路径的约束多目标优化模型。相比已有的优化模型,一方面,产品综合性能评价指标的引入能够确保变更方案的可行性和实用性;另一方面,在变更传播时,考虑更多的实际约束能够确保变更方案符合实际需求。

2) 利用DDQN自主切换不同进化策略,提出一种RL引导的多目标DE算法,相比已有产品设计变更传播路径求解算法,一方面该算法具有较强的全局搜索能力,不容易陷入局部最优;另一方面,由于能够使种群在不同阶段选择适合的进化搜索策略,进一步提升种群的寻优能力。

3) 定义一种基于超体积测度的奖励函数, 相比已有的奖励函数, 由于考虑了种群进化的多样性和收敛性, 能够获得分布均匀以及收敛的Pareto最优解集。

本文的其余部分安排如下: 第2节介绍复杂产品网络模型和路径优选问题的多目标优化模型; 第3节详细介绍所提算法; 第4节通过对比实验分析验证所提算法的有效性; 最后, 第5节对本文进行总结, 并展望了未来工作。

## 2 复杂产品网络模型与问题描述

### 2.1 复杂产品网络模型

假设某产品由 $n$ 个零件组成, 单个零件使用节点 $v_i$ 表示, 两两零件之间的关联关系和关联强度分别使用边 $e_{ij}$ 和权重 $w_{ij}$ 表示, 那么该产品可以表示为复杂网络 $G = (V, E, W)$ . 其中:  $V = \{v_i | i = 1, 2, \dots, n\}$ 表示节点集合,  $E = \{e_{ij} | i, j = 1, 2, \dots, n\}$ 表示节点间边集合,  $W = \{w_{ij} | i, j = 1, 2, \dots, n\}$ 表示权重集合. 零件之间的关联强度, 在复杂网络模型构建中至关重要. 本节使用区间直觉模糊集 (interval-valued intuitionistic fuzzy sets, IVIFS)<sup>[22]</sup>确定节点之间的关联强度。

首先, 根据表1以及专家经验对节点之间的关联强度进行评估, 确定节点之间的综合关联强度区间 $\tilde{w}_{ij}$ . 文献[22]考虑了节点间的功能联系、节点间的物理结构关系、用户参与程度, 以及产品的可持续性和适应性. 类似地, 本文也从上述4个方面来计算两个节点之间的关联强度值, 具体公式如下:

$$\tilde{w}_{ij} = h_s R_{ij}^s + h_f R_{ij}^f + h_{cs} R_{ij}^{cs} + h_a R_{ij}^a, \quad (1)$$

式中:  $R_{ij}^s, R_{ij}^f$ 分别为节点之间的物理结构相关强度、功能相关强度;  $R_{ij}^{cs}$ 为设计这些节点时用户的参与程度;  $R_{ij}^a$ 为产品的可持续性和适应性.  $h_s, h_f, h_{cs}$ 和 $h_a$ 分别为上述4种因素的权重. 以功能相关强度 $R_{ij}^f$ 为例, 其相应计算公式<sup>[22]</sup>如下:

$$R_{ij}^f = \begin{cases} ([\underline{\mu}_{ij}^f, \bar{\mu}_{ij}^f], [\underline{v}_{ij}^f, \bar{v}_{ij}^f]), & i \neq j, \\ 1, & i = j, \end{cases} \quad (2)$$

式中 $[\underline{\mu}_{ij}^f, \bar{\mu}_{ij}^f], [\underline{v}_{ij}^f, \bar{v}_{ij}^f]$ 分别表示节点 $v_i$ 和 $v_j$ 的功能相关程度区间和非相关程度区间。

表1 相关强度的评估区间

Table 1 Assessment interval of the relevant intensity

相关强度	区间值
非常强	[0.90, 0.95], [0.02, 0.05]
强	[0.70, 0.75], [0.20, 0.35]
正常	[0.50, 0.55], [0.40, 0.45]
弱	[0.20, 0.25], [0.70, 0.75]
非常弱	[0.02, 0.05], [0.90, 0.95]
无	[0, 0], [1, 1]

基于上述分析, 可以获得不同节点间的关联强度区间, 随后采用去模糊化可以得到节点 $v_i$ 和 $v_j$ 间的具体关联强度值 $w_{ij}$ 。

### 2.2 变更传播模型

为了评估节点变更的传播影响, 需要描述节点之间的变更传播机制. 为此, 通过引入变更传播概率和设计裕度概念, 建立产品设计变更的传播模型。

#### 1) 变更传播概率.

变更传播概率 $P_{ij}$ 表示变更从节点传播到节点的概率<sup>[23]</sup>. 具体地, 基于产品设计变更数据库, 通过计算如下条件概率来获取:

$$P_{ij} = P(v_j | v_i) = \frac{P(v_i \cap v_j)}{P(v_i)} = \frac{P_{ji} P(v_j)}{P(v_i)}, \quad (3)$$

式中:  $P(v_i)$ 为节点 $v_i$ 发生变更的概率;  $P(v_i \cap v_j)$ 为两个节点同时发生变更的概率。

#### 2) 设计裕度.

设计裕度<sup>[24]</sup>是在设计产品时设计者为关键零件留出的设计余量. 当节点 $v_i$ 的设计变更影响 $c_j$ 高于预设的设计裕度 $\Delta c_j$ 时, 需要对此节点进行重新设计。

根据变更传播概率以及设计裕度等概念, 可以计算出节点 $v_j$ 的设计变更影响 $c_j$ , 即

$$c_j = P_{ij} \times c_{i-1,i}, \quad (4)$$

式中:  $c_{i-1,i}$ 表示从节点 $v_i$ 的上游节点传播到节点 $v_i$ 时的变更传播影响; 初始变更节点的设计影响 $c_0$ 根据专家经验给出。

### 2.3 性能指标

#### 2.3.1 变更对产品综合性能的影响

变更传播强度CPI是设计变更传播路径的关键因素<sup>[9]</sup>. 本节从变更传播指数、聚集系数和组件重要性方面对CPI进行量化, 并通过CPI指标间接描述出变更对产品综合性能的影响。

1) 变更传播指数. 变更传播指数可以确定节点的属性<sup>[25]</sup>, 即吸收者/携带者/扩散者. 具体地, 变更传播指数 $K_i$ 的计算公式如下:

$$K_i = \frac{K_{out}^i - K_{in}^i}{K_{out}^i + K_{in}^i}, \quad (5)$$

式中:  $K_{out}^i$ 代表节点 $v_i$ 的出度, 即节点 $v_i$ 变更影响到的其他节点的个数;  $K_{in}^i$ 代表节点 $v_i$ 的入度, 即变更会影响到节点 $v_i$ 的其他节点的个数。

2) 聚集系数. 聚集系数被用来描述网络中节点之间的整合程度. 在加权网络中, 为客观描述节点之间的整合程度需要充分考虑节点之间边的权重值, Onnela等<sup>[26]</sup>将加权聚集系数 $C_i^w$ 定义为

$$C_i^w = \frac{2}{d_i(d_i - 1)} \sum_{j,h} \frac{(w_{ij} w_{jh} w_{hi})^{\frac{1}{3}}}{w_{\max}}, \quad i \neq j \neq h, \quad (6)$$

式中:  $d_i$ 表示节点 $v_i$ 的度,即产品网络中与节点 $v_i$ 存在关联关系的节点个数;  $w_{ij}$ ,  $w_{jh}$ 和 $w_{hi}$ 分别表示节点 $v_i$ ,  $v_j$ 和 $v_h$ 之间的三边权重,其取值通过等式(1)–(2)计算获得;  $w_{\max}$ 是产品网络中最大的连边权重值。

3) PageRank (PR) 算法. PageRank 算法是用来评估网页重要程度的性能指标<sup>[27]</sup>,它将网页之间的连接视为节点之间的连接.由于产品零件之间的关系和网页之间的关系存在相似性,因此采用PageRank算法来评估复杂产品网络中节点的重要性.节点 $v_i$ 的重要性 $PR_i$ 计算公式如下:

$$PR_i = \frac{1-q}{n} + q \sum_{v_j \in DS_i} \frac{PR_j}{K_j^{\text{out}}}, \quad (7)$$

式中:节点 $v_j$ 是节点 $v_i$ 的下游节点;  $DS_i$ 是节点 $v_i$ 的下游节点集合;  $q \in (0, 1)$ 是阻尼因子,一般设置为0.85;  $n$ 是节点总数.

进一步,借鉴文献[11]和文献[28]的思想对上述指标进行处理.首先,根据该指标在整体评价中的相对重要程度为其分配一个权重;接着,使用线性加权求和法计算这些指标的加权和,进而评估面向多个指标的综合效果.用于评估一条变更路径 (change path, CP)的总变更传播强度CPI值如下:

$$CPI = \sum_{v_i, v_j \in CP} (w_1 K_i + w_2 C_i^w + w_3 PR_i) \times p_{ij}, \quad (8)$$

式中:  $w_1$ ,  $w_2$ 和 $w_3$ 分别表示变更传播指数、加权聚集系数和组件自身重要性的权重;  $v_i$ ,  $v_j$ 分别表示当前正在变更的节点和下一个需要变更的节点;  $CP = \{cp_0, cp_1, \dots, cp_{i-1}, cp_i, cp_{i+1}, \dots, cp_m | m < n\}$ 表示一条变更传播路径.其中:  $cp_0$ 表示初始变更节点;  $cp_m$ 表示路径的终止节点.

### 2.3.2 变更经济成本和变更工期

设计变更不仅会增加产品设计的经济成本,而且会花费一定的工期.为此,除了变更对产品性能影响之外,还需要考虑变更经济成本和变更工期指标.假设每个零件均具有固定的经济成本和时间成本,那么,一条变更传播路径的变更经济成本和工期描述为

$$\text{Cost}(CP) = \sum_{v_i \in CP} \text{cost}_i, \quad (9)$$

$$\text{Time}(CP) = \sum_{v_i \in CP} \text{time}_i, \quad (10)$$

式中:节点 $v_i$ 表示当前正在发生变更的节点;  $\text{cost}_i$ ,  $\text{time}_i$ 分别表示节点 $v_i$ 的变更经济成本和工期.

## 2.4 问题约束

在产品的设计过程中,受各种因素的限制,一个可行的变更传播路径必须满足诸多的实际约束.为了符合日益增多的实际需求,与已有约束多目标优化模型不

同,本节全面地考虑如下5个实际约束:

**约束1** 节点 $v_i$ 不能改变,即当前变更节点 $cp_i \neq v_i$ .例如,由于技术封锁导致供应链中断或其他不可抗力,受影响的零件无法找到合适的替换零件.

**约束2** 节点 $v_i$ 必须改变,即当前变更节点 $cp_i = v_i$ .例如,现在使用的一个零件无法获得而重新设计的成本又超出预期.

**约束3** 当一个节点 $v_i$ 改变时,其下游节点 $v_j$ 必须改变,即如果 $cp_i = v_i$ ,且 $v_j \in ds_i$ ,则 $cp_j = v_j$ .例如,一个节点与其下游节点密切相关,其下游节点是其重要的辅助节点.

**约束4** 若一个节点 $v_i$ 保持不变,则其下游节点 $v_j$ 不能改变,即如果 $cp_i \neq v_i$ ,且 $v_j \in ds_i$ ,则 $cp_j \neq v_j$ .例如,下游节点 $v_j$ 是节点 $v_i$ 的必不可少的辅助节点,其变化主要由节点 $v_i$ 决定.

**约束5** 当节点 $v_i$ 的设计变更影响 $c_i < \Delta c_i$ 时,可以忽略其变更影响,变更设计结束.

## 2.5 优化问题描述

本文假设如下:首先,将初始变更节点均视为单一节点,不考虑多个节点同时发生变更的情况;其次,在设计变更过程中,不考虑中途追加变更需求的情况.这些假设能够有效减少问题的复杂度,使本文研究更专注于问题求解.同时考虑最小化CPI、变更经济成本、变更工期等3个目标,可将产品变更传播路径优化问题建模为如下多目标优化模型:

$$\begin{aligned} \min F(CP) = & \\ & \{CPI(CP), \text{Cost}(CP), \text{Time}(CP)\} \\ \text{s.t. Constraints from 2.4 Section,} \\ & CP = \{cp_0, cp_1, \dots, \\ & \quad cp_{i-1}, cp_i, cp_{i+1}, \dots, cp_m | m < n\}, \quad (11) \end{aligned}$$

式中:  $v_i, v_j \in CP$ ,  $CPI(\cdot)$ ,  $\text{Cost}(\cdot)$ ,  $\text{Time}(\cdot)$ 分别由式(8)–(9)和(10)构成.

## 3 双深度Q-网络引导的多目标差分进化算法

为了克服现有DE算法搜索机制单一且不能适应问题变化的不足, Tan等尝试将DQN<sup>[29]</sup>和DE算法相结合,提出了基于DQN的DE算法DEDQN<sup>[20]</sup>,但是它依然存在如下缺点: 1) 该算法使用DQN选择DE的进化策略,会使智能体陷入过度估计的困境,导致其选择的动作不一定是最优动作; 2) 该算法将奖励函数设置为种群进化效率.在处理多目标优化问题时,由于未考虑种群进化的多样性和收敛性,导致其很难获得分布均匀以及收敛的Pareto最优解集.

鉴于此,本节利用DDQN帮助进化算法自主切换不同进化策略,定义一种基于超体积测度的奖励函数,

研究一种RL引导的DE算法. 此外, 基于产品设计变更路径优化问题的特点, 本文采用整数编码策略.

### 3.1 双深度Q-网络

在RL中, 智能体在执行动作 $a_t$ 后会获得一个即时奖励值 $r_t$ . 为了累积奖励的最大化, 需要估计动作对于未来的影响, 基于此, 文献[21]使用动作价值函数选择动作 $a_t$ . 动作价值函数 $Q(s_t, a_t)$ 由即时奖励 $r_t$ 和下一时刻的状态 $s_{t+1}$ 的最大动作价值构成, 表达为

$$Q(s_t, a_t) = r_t + \gamma \max_a Q(s_{t+1}, a), \quad (12)$$

式中 $\gamma$ 为折扣因子, 用于平衡当前收益和未来收益.

DQN会使智能体陷入过度估计的困境. 鉴于此, 学者们提出了DDQN, 其神经网络由目标Q-网络和主Q-网络两个神经网络组成, 但主要任务不同, 主Q-网络负责选择下一个状态下最优的动作, 目标Q-网络负责计算动作的价值. 这种设计有利于克服DQN中Q值函数的过度估计问题, 进一步提高模型的稳定性和学习效率<sup>[21]</sup>. 在训练过程中, 通过时间差分法不断更新 $Q(s_t, a_t; \theta_t)$ , 即

$$Y_t^{\text{DDQN}} \equiv r_t + \gamma Q(s_{t+1}, \arg \max_a Q(s_{t+1}, a; \theta_t); \theta_t^-), \quad (13)$$

$$Q(s_t, a_t; \theta_t) \leftarrow Q(s_t, a_t; \theta_t) + \alpha(Y_t^{\text{DDQN}} - Q(s_t, a_t; \theta_t)), \quad (14)$$

式中:  $Y_t^{\text{DDQN}}$ 是DDQN的目标函数;  $\alpha$ 是学习率, 表示新信息覆盖旧信息的比率;  $\theta_t$ 和 $\theta_t^-$ 分别是目标Q-网络和主Q-网络在当前时刻的参数. DDQN在初始训练时, 随机设置主Q-网络参数 $\theta_t$ . 在训练过程中, 通过损失函数 $L_\theta$ 不断更新参数 $\theta_t$ , 即

$$L_\theta = E[(Y_t^{\text{DDQN}} - Q(s_t, a; \theta_t))^2], \quad (15)$$

式中 $E(\cdot)$ 是平均期望.

### 3.2 DDQN引导的进化策略(动作)选择机制

本节使用DDQN来帮助DE提升性能. 在进化过程中, DDQN与当前种群不断进行交互, 根据种群当前的状态选择合适的DE策略, 并获取奖励值. 通过实时选择适合当前种群的进化策略, 能够充分发挥DE在不同阶段的搜索特性.

#### 1) 状态特征.

状态特征的表达在一定程度上会影响智能体和环境的交互效果. 以最小化问题为例, 当第 $i$ 个个体的某一个目标函数值变小时, 该个体的目标函数值之和也具有减小的趋势, 因此, 使用个体的目标之和来刻画种群状态的好坏. 本文将种群的状态特征 $s_t$ 定义为

$$s_t = \{F_1, \dots, F_i, \dots, F_{NP}\}, \quad (16)$$

式中:  $F_i = \sum_{j=1}^M f_i^j$ ,  $NP$ 是种群的大小,  $M$ 是目标函数

个数,  $f_i^j$ 是第 $i$ 个个体的第 $j$ 个目标函数值. 然而, 实际设计工作中, 不同的目标具有不同的单位和取值范围. 为了解决上述问题, 利用最小-最大规范化方法处理这些指标.

#### 2) 动作空间.

DE算法的性能主要由DE策略决定, 因此选择合适的DE化策略尤其重要. 为充分发挥不同DE策略的性能, 在不同阶段采取不同的进化策略. 本文使用如下3种不同的DE策略, 包括DE/rand/1, DE/current to rand/1以及进化策略DE/best/2. 由此, DDQN的动作空间 $A$ 可以表示为

$$A = \{\text{DE/rand/1}, \text{DE/best/2}, \text{DE/current to rand/1}\}. \quad (17)$$

在传统DE算法中, 个体执行DE策略后, 会出现非整数和缺失的元素, 无法直接应用于产品设计变更传播路径问题中. 因此, 本节通过定义算术运算符, 提出了一种基于整数编码且的DE策略. 以DE/rand/1为例, 具体定义如下:

$$H_{i,G} = X_{r_1,G} \oplus F \otimes (X_{r_2,G} \ominus X_{r_3,G}), \quad (18)$$

式中:  $H_{i,G}$ 表示在 $G$ 次迭代时, 生成的第 $i$ 个新个体;  $X_{r_1,G}$ ,  $X_{r_2,G}$ ,  $X_{r_3,G}$ 表示在 $G$ 次迭代时, 从种群中随机选择的3个个体.  $\ominus$ 的执行规则为: 首先, 创建差异信息存储列表lis; 接着, 获取主导个体, 即个体 $X_{r_2,G}$ ,  $X_{r_3,G}$ 中的非支配解作为主导个体, 另一个则作为辅助个体, 如果个体之间相互支配则从个体 $X_{r_2,G}$ ,  $X_{r_3,G}$ 中随机选择主导个体; 最后, 将主导个体和辅助个体进行比较, 找出两者的差异分量, 将分量存储到lis中, 并记录每个分量在主导个体中的位置.  $\oplus$ 和 $\otimes$ 的执行规则为: 首先, 根据中的元素个数生成相应数量的随机数; 接着, 将依次生成的rand与缩放因子 $F$ 进行比较, 如果 $\text{rand} > F$ 时, 将lis中的分量替换掉个体 $X_{r_1,G}$ 中相应位置的分量, 从而生成新个体 $H_{i,G}$ .

#### 3) 奖励函数定义.

在RL中, 奖励函数主要用于评价采取动作的好坏. 超体积测度(hyper-volume, HV)<sup>[30]</sup>是多目标优化算法中常用的一种两元性能评价指标. 在无需问题真实Pareto最优解集的情况下, HV可以同时评价一个非支配解集的收敛性和多样性. 鉴于此, 本文使用HV改进率作为种群进化能力的度量指标, 并由此计算DDQN-DE的奖励函数值. 具体地, 奖励函数 $r_t$ 设置如下:

$$r_t = (\text{HV}_t - \text{HV}_{t-1}) / \text{HV}_{t-1}, \quad (19)$$

$$\text{HV} = \delta \left( \bigcup_{i=1}^{|S|} \text{hv}_i \right), \quad (20)$$

式中:  $\text{HV}_t$ 是 $t$ 时刻种群的HV值;  $\delta$ 表示Lebesgue测度, 用来测量体积;  $|S|$ 表示非支配解集的数目;  $\text{hv}_i$ 表示参照点与当前种群中第 $i$ 个个体构成的超体积.

#### 4) 动作选择.

DDQN根据状态 $s_t$ 来选择动作 $a_t$ ,同时获取相应的奖励值 $r_t$ .如果 $r_t > 0$ ,那么当DDQN遇见相似种群状态时,应该选择相同的动作 $a_t$ .但是,由于动作 $a_t$ 可能仅偏向于提高种群的多样性或收敛性,DDQN连续使用该动作会使算法收敛速度过慢或陷入局部最优.因此,一般采用 $\varepsilon$ -greedy策略进行动作的选择,以较小概率 $\varepsilon$ 随机选择其他动作. $\varepsilon$ -greedy策略的实现公式<sup>[20]</sup>为

$$a_t = \begin{cases} \text{random}(A), & \varepsilon > \text{rand}, \\ \arg \max_a Q(s_t, a; \theta_t), & \text{其他}, \end{cases} \quad (21)$$

式中:  $\text{rand} \in (0, 1)$ 是随机生成的数; $\text{random}(A)$ 表示从动作空间 $A$ 中随机选择DE策略.

## 4 实验分析

### 4.1 测试问题与实验环境

以某型号创维电视的设计变更问题为例,验证文中所提算法的有效性.首先,根据产品设计变更数据库获得了产品的变更传播概率矩阵,如图1所示.接着,通过企业设计工程师的评估信息构建创维电视网络模型如图2所示.进一步,根据企业历史变更信息,选取5个易变更的零件作为初始变更节点进行实验,如表2所示.为了保证实验的公平性和合理性,所有算法均在Python 3.6, AMD Ryzen 55600H CPU, 16.00 GB RAM环境中重复运行20次.所提算法种群大小设置为200,迭代次数设置为100,根据文献[31–32]的建议,将交叉概率设置为0.1,缩放因子设置为0.7;根据参考文献[33–35]并结合本文的研究问题,将DDQN网络的参数设置如下: $\alpha = 0.01$ ,  $\gamma = 0.9$ ,  $f = 5$ ,更新阈值  $\text{Thresh} = 10$ .初始设计变更影响均设置为0.4.

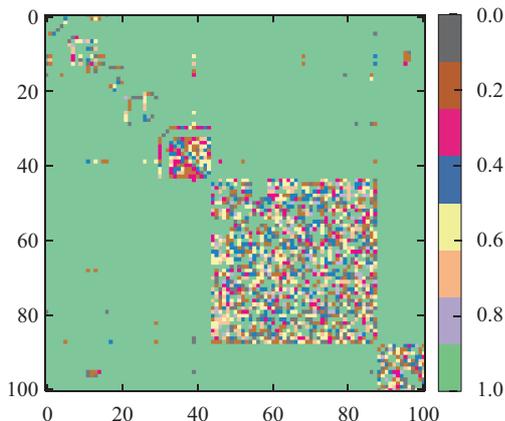


图1 创维电视变更传播概率矩阵

Fig. 1 Skyworth TV change propagation probability matrix

### 4.2 所提算法有效性分析

本节将所提算法DDQN-DE与5种典型的求解产品变更传播路径的算法进行比较,验证所提算法在解决多目标问题和求解方案方面的有效性和优越性.其

中,NSGA-II,基于分解的多目标进化算法(multi objective evolutionary algorithm based on decomposition, MOEA/D)和DEDQN是典型的多目标进化优化算法.Improved ACO以及最小风险路径方法(least risky change propagation path algorithm, LRCPP)均是求解产品设计变更路径时比较具有代表性的算法.Improved ACO算法是比较经典的进化优化算法,如文献[7]、文献[9]均使用上述算法求解路径问题;LRCPP是传统算法中比较经典的解决算法,如文献[36]使用其作为对比算法.上述5种对比算法及其关键参数设置,如下:

1) 非支配排序遗传算法-II(NSGA-II)<sup>[37]</sup>.设置种群大小为200,迭代次数为100,交叉概率为0.8,变异概率为0.1.

2) 基于分解的多目标进化算法(MOEA/D)<sup>[38]</sup>.设置种群大小为200,迭代次数为100,交叉概率为0.8,变异概率为0.1,聚合方法使用切比雪夫距离.

3) 基于深度强化学习和混合变异的差分进化算法(DEDQN)<sup>[20]</sup>.设置种群大小为200,迭代次数为100,初始交叉概率以及缩放因子均设置为0.5,训练回合数为100,  $\alpha = 0.01$ ,  $\gamma = 0.9$ ,  $f = 5$ .

4) 改进蚁群算法(improved ACO)<sup>[7]</sup>.设置种群大小为200,迭代次数为100,信息素重要程度因子为0.5,启发函数重要程度因子为1,挥发速度为0.1.

5) 渐进式变化传播的最小风险路径方法(LRCPP)<sup>[39]</sup>.

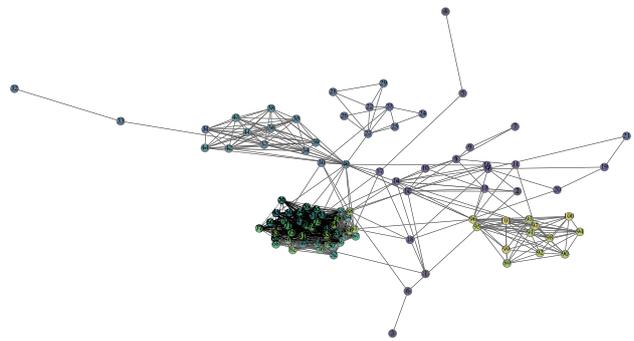


图2 创维电视网络模型

Fig. 2 Skyworth TV network model

表2 初始变更节点

Table 2 Initial change nodes

节点	零件名称
1	电子线
18	机壳组件
36	插件肖特基二极管
85	直流-直流转换器
97	脉冲宽度调制控制集成电路

本节将DDQN-DE算法分别与3种多目标进化算

法NSGA-II, MOEA/D以及DEDQN进行对比. 为了验证多目标优化算法的有效性, 采用HV指标来评估算法的分布性和收敛性. 为了将NSGA-II和MOEA/D应用到产品设计变更路径求解问题中, 采用文献[11]提出的交叉和变异策略作为算法的更新策略.

实验结果如表3所示, “+”和“\*”分别表示DDQN-DE所取得的实验结果显著优于和近似于对比算法, 可以看出: 1) 对于表2中所有的初始变更节点, 算法DD-

QN-DE所获得的平均HV值均优于NSGA-II, MOEA/DD以及DEDQN; 2) 由T检验结果可知, DDQN-DE算法所得的HV值明显优于NSGA-II, MOEA/D以及DEDQN; 3) 由标准差分析可知, 对于节点1, DDQN-DE算法的标准差略高于算法DEDQN; 对于节点36以及85, DDQN-DE算法的标准差略高于算法NSGA-II. 然而, 对于其余变更节点, DDQN-DE算法的标准差均优于所有对比算法, 算法稳定性更好.

表 3 DDQN-DE算法和多目标优化算法的HV值(平均值/标准差)

Table 3 HV values obtained by the DDQN-DE algorithm and multi-objective optimization algorithm(mean/std)

算法/节点	节点1	节点18	节点36	节点85	节点97
DDQN-DE	<b>0.908/0.021</b>	<b>0.894/0.018</b>	<b>0.873/0.036</b>	<b>0.824/0.017</b>	<b>0.856/0.016</b>
NSGA-II	0.871/0.023(+)	0.864/0.025(+)	0.863/ <b>0.018</b> (+)	0.792/ <b>0.016</b> (+)	0.837/0.016(+)
MOEA/D	0.824/0.042(+)	0.848/0.042(+)	0.778/0.055(+)	0.789/0.020(+)	0.774/0.024(+)
DEDQN	0.887/ <b>0.016</b> (+)	0.885/0.022(*)	0.856/0.039(+)	0.802/0.027(+)	0.843/0.023(+)

进一步, 本节将DDQN-DE算法分别与Improved-ACO算法和LRCPP算法进行对比, 验证算法在求解变更方案方面的优越性. 在实验中, 选择零件1(电子线)作为初始变更节点; 将3个优化目标值归一化, 作为对比算法的优化目标值. 图3和图4中的红色节点分别是通过Improved ACO和LRCPP算法获得的最优变更传播路径. 图5展示了通过DDQN-DE算法获得的Pareto最优解集.

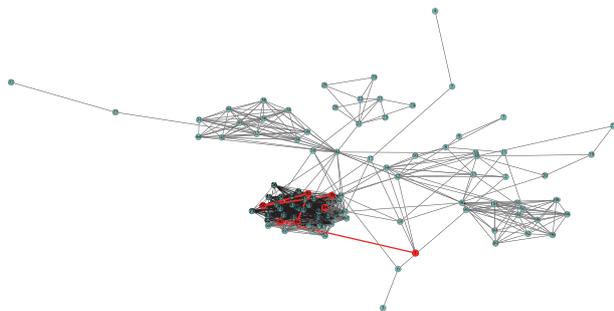


图 3 Improved ACO算法的最优变更路径

Fig. 3 The optimal change path of Improved ACO algorithm

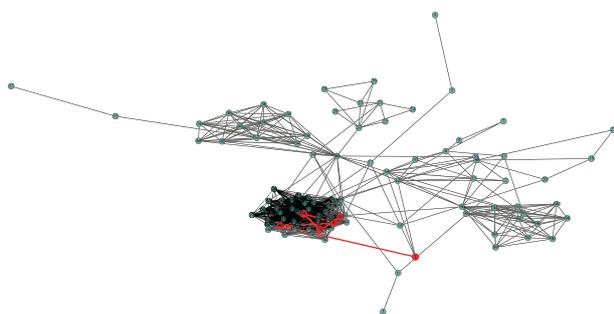


图 4 LRCPP算法的最优变更路径

Fig. 4 The optimal change path of LRCPP algorithm

具体路径信息如表4所示, 其中P1-P3是从图5的

Pareto最优解集中选取的3个变更方案; P4和P5分别是通过Improved ACO和LRCPP算法获得的变更方案. 从实验结果可以看出: 1) 相比于Improved ACO和LRCPP算法, DDQN-DE算法能够同时获得多个非劣解. 2) DDQN-DE算法获得的方案P1-P3各有优缺点, 如方案P1的CPI值比P2, P3的CPI值小, 即变更对产品综合性能影响更小; 方案P2的变更经济成本优于P1和P3; 方案P3的变更工期优于P1, P2. 3) 在DDQN-DE算法所得非劣解中, 皆存在一个解支配Improved ACO和LRCPP算法得到的最优变更传播路径. 例如, DDQN-DE算法所得解P1支配Improved ACO的解P4, DDQN-DE算法所得解P2支配LRCPP的解P5. 综上, DDQN-DE算法在求解变更方案方面优于Improved ACO和LRCPP算法, 它可以提供多个优秀的变更方案供设计者进行选择.

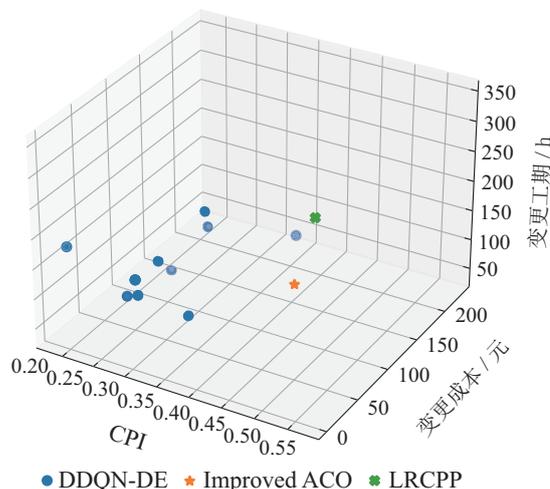


图 5 DDQN-DE算法的Pareto最优解集

Fig. 5 Pareto optimal solution set for DDQN-DE algorithm

表4 设计变更方案

Table 4 Design change programs

算法	编号	变更路径	变更成本/元	变更工期/时	CPI
DDQN-DE	P1	1-80-88-58	12.30	175.0	0.18
	P2	1-80-59-63	4.25	165.0	0.28
	P3	1-17-40-15	228.45	42.0	0.32
Improved ACO	P4	1-80-46-65-60-58-84	12.67	225.0	0.53
LRCPP	P5	1-80-61-74-79-86	7.80	345.0	0.57

## 5 结论

针对产品设计变更传播路径求解问题, 本文提出了一种RL引导的产品变更路径多目标DE算法, 即DDQN-DE算法. 本文给出了复杂产品网络模型, 引入了变更传播模型, 建立了设计变更传播路径问题的约束多目标优化模型, 并且提出了DDQN引导的DE算法. 为了评估DDQN-DE算法的有效性, 在某型号创维电视的设计变更问题上进行了实验, 实验结果表明, 所提算法的综合性能优于多目标进化算法NSGA-II, MOEA/D, DEDQN, 相比于Improved ACO和LRCPP算法, 所提算法可以提供多个优秀的变更方案供设计者进行选择.

尽管该研究取得了较好的结果, 但它仍有一些局限性. 所提算法主要面向单变更源的产品设计问题. 在实际工程中多个零件同时发生变更的现象普遍存在, 因此, 多源变更设计是未来需要解决的关键问题之一.

## 参考文献:

- [1] YIN L, SUN Q, XU Y, et al. Risk analysis of engineering change for distributed product design. *Journal of Computing and Information Science in Engineering*, 2020, 21(4): 1 – 38.
- [2] ZHANG N, LI M, REN H, et al. A network-based four-phase routing approach of multisource design change propagation on complex products. *Kybernetes*, 2021, 50(11): 3058 – 3081.
- [3] TANG Dunbing, XU Ronghua, TANG Jicheng, et al. Analysis of engineering change impacts based on design structure matrix. *Journal of Mechanical Engineering*, 2010, 46(1): 154 – 161. (唐敦兵, 徐荣华, 唐吉成, 等. 基于设计结构矩阵的工程变更影响分析. *机械工程学报*, 2010, 46(1): 154 – 161.)
- [4] JONKERS R K, SHAHROUDI K E. A design change, knowledge, and project management flight simulator for product and project success. *IEEE Systems Journal*, 2021, 15(1): 1130 – 1139.
- [5] YANG Q, KHERBACHI S, HONG Y S, et al. Identifying and managing coordination complexity in global product development project. *International Journal of Project Management*, 2015, 33(7): 1464 – 1475.
- [6] DIAGNE S, COULIBALY A, DE BEUVRON F D. Complex product modeling based on a multi-solution extended conceptual design demantic matrix for behavioral performance assessment. *Computers in Industry*, 2016, 75: 101 – 115.
- [7] LI Yupeng, LI Mengze, WANG Zhaotong. Multi-source design change propagation path optimization for complex product based on weighted and directed network model. *Journal of Mechanical Engineering*, 2019, 55(6): 213 – 222. (李玉鹏, 李孟泽, 王召同. 基于有向加权网络模型的复杂产品多源设计变更传播路径优化. *机械工程学报*, 2019, 55(6): 213 – 222.)
- [8] DONG C C, YANG Y, CHEN Q, et al. A complex network-based response method for changes in customer requirements for design processes of complex mechanical products. *Expert Systems with Applications*, 2022, 199: 117124.
- [9] MA S H, JIANG Z L, LIU W P. Evaluation of a design property network-based change propagation routing approach for mechanical product development. *Advanced Engineering Informatics*, 2016, 30(4): 633 – 642.
- [10] LUO Jianfen, LI Mengze, LI Ting, et al. Optimization of design-change propagation of complex products based on the multi-objective genetic algorithm. *Journal of Machine Design*, 2020, 37(10): 66 – 72. (骆建芬, 李孟泽, 李婷, 等. 基于多目标遗传算法的复杂产品设计变更传播优化. *机械设计*, 2020, 37(10): 66 – 72.)
- [11] REN H B, TING L, LI Y P, et al. Multi-source design change propagation path optimisation based on the multi-view complex network model. *Journal of Engineering Design*, 2021, 32(1): 28 – 60.
- [12] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.
- [13] CAO Bei, CHEN Qingda, DING Jinliang. Operational optimization of fluid catalytic cracking reaction-regeneration system with multi-yield constraints using improved differential evolution algorithm. *Control Theory and Applications*, 2019, 36(8): 1207 – 1216. (曹蓓, 陈庆达, 丁进良. 多收率约束的催化裂化反再系统改进差分进化操作优化. *控制理论与应用*, 2019, 36(8): 1207 – 1216.)
- [14] TAN Z P, LI K S, WANG Y. Differential evolution with adaptive mutation strategy based on fitness landscape analysis. *Information Sciences*, 2021, 549: 142 – 163.
- [15] WU P, XU L, D'ARIANO A, et al. Novel formulations and improved differential evolution algorithm for optimal lane reservation with task merging. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(11): 21329 – 21344.
- [16] SUTTON R S, BARTO A G. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2018.
- [17] BELLMAN R. A Markovian decision process. *Journal of Mathematics and Mechanics*, 1957, 6(5): 679 – 684.
- [18] LI Z H, SHI L, YUE C T, et al. Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems. *Swarm and Evolutionary Computation*, 2019, 49: 234 – 244.
- [19] HUYNH T N, DO D T T, LEE J. Q-learning-based parameter control for differential evolution for structural optimization. *Applied Soft Computing*, 2021, 107: 107464.
- [20] TAN Z P, LI K S. Differential evolution with mixed mutation strategy based on deep reinforcement learning. *Applied Soft Computing*, 2021, 111: 107678.

- [21] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning. *The 30th AAAI Conference on Artificial Intelligence*. Phoenix, Arizona USA: AAAI, 2016: 2049 – 2100.
- [22] ZHANG N, YANG Y, ZHENG Y, et al. Module partition of complex mechanical products based on weighted complex networks. *Journal of Intelligent Manufacturing*, 2019, 30(4): 1973 – 1998.
- [23] KOH E C Y, CALDWELL N H M, CLARKSON P J. A method to assess the effects of engineering change propagation. *Research in Engineering Design*, 2012, 23(4): 329 – 351.
- [24] JEANG A. Combined parameter and tolerance design optimization with quality and cost. *International Journal of Production Research*, 2001, 39(5): 923 – 952.
- [25] GIFFIN M, DE WECK O, BOUNOVA G, et al. Change propagation analysis in complex technical systems. *Journal of Mechanical Design*, 2009, 131(8): 081001.
- [26] ONNELA J P, SARMAKI J, KERTESZ J, et al. Intensity and coherence of motifs in weighted complex networks. *Physical Review E*, 2005, 71(6): 065103.
- [27] BRIN S, PAGE L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 1998, 30(1/7): 107 – 117.
- [28] LI T, ZHANG N, REN H, et al. Design change propagation routing in the modular product. *Advanced Engineering Informatics*, 2022, 54: 101784.
- [29] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning. *Nature*, 2015, 518(7540): 529 – 533.
- [30] GUERREIOR A, FONSECA C, PAQUETE L. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys*, 2021, 54(6): 1 – 42.
- [31] RONKKONEN J. *Continuous multimodal global optimization with differential evolution-based methods*. Lappeenranta, Finland: Lappeenranta University of Technology, 2009.
- [32] CHENG Qiang, GAO Yuanjie, CHU Hongyan, et al. Flexible job shop scheduling of machining based on multi-objective differential evolution algorithm. *Journal of Beijing University of Technology*, 2023, 49(3): 335 – 345.  
(程强, 高元杰, 初红艳, 等. 基于多目标差分进化算法的机加工柔性作业车间调度. 北京工业大学学报, 2023, 49(3): 335 – 345.)
- [33] LI T Y, MENG Y, TANG L X. Scheduling of continuous annealing with a multi-objective differential evolution algorithm based on deep reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 2023, DOI: 10.1109/TASE.2023.3244331.
- [34] CHEN Chaoyi, LU Juan, CHEN Kai, et al. Research on analytical model and DDQN-SVR prediction model of turning surface roughness. *Journal of Mechanical Engineering*, 2021, 57(13): 262 – 272.  
(陈超逸, 鲁娟, 陈楷, 等. 车削表面粗糙度解析模型与DDQN-SVR预测模型研究. 机械工程学报, 2021, 57(13): 262 – 272.)
- [35] XIONG Zhihua, CHEN Hao, WANG Changsheng, et al. Assembly task allocation of human-robot collaboration based on deep reinforcement learning. *Computer Integrated Manufacturing Systems*, 2023, 29(3): 789 – 800.  
(熊志华, 陈昊, 王长生, 等. 基于深度强化学习的人机协作组装任务分配. 计算机集成制造系统, 2023, 29(3): 789 – 800.)
- [36] ZHENG R Z, ZHANG Y, SUN X Y, et al. Multi-objective particle swarm optimization of complex product change plan considering service performance. *CAAI Transactions on Intelligence Technology*, 2023, 8(3): 1058 – 1076.
- [37] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [38] ZHANG Q F, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712 – 731.
- [39] ULLAH I, TANG D B, WANG Q, et al. Least risky change propagation path analysis in product design process. *Systems Engineering*, 2017, 20(4): 379 – 391.

#### 作者简介:

**宋贤芳** 博士, 目前研究方向为智能优化理论及其应用、进化特征选择等, E-mail: songxf0614@126.com;

**杨扬** 硕士研究生, 目前研究方向为智能优化、产品设计变更等, E-mail: yangyang103702@126.com;

**张勇** 教授, 博士生导师, 目前研究方向为智能优化、数据驱动等, E-mail: yongzh401@126.com;

**郑瑞钊** 博士研究生, 目前研究方向为智能优化、产品设计变更等, E-mail: zhengruizhao998@gmail.com.