

基于强化学习与遗传算法的机器人并行拆解序列规划方法

汪开普¹, 马晓艺¹, 卢超², 殷旅江^{3*}, 李新宇⁴

(1. 武汉理工大学机电工程学院, 湖北武汉 430070; 2. 中国地质大学(武汉)计算机学院, 湖北武汉 430078;
3. 湖北汽车工业学院经济管理学院, 湖北十堰 442002; 4. 华中科技大学机械科学与工程学院, 湖北武汉 430074)

摘要:在拆解序列规划问题中,为了提高拆解效率、降低拆解能耗,引入了机器人并行拆解模式,构建了机器人并行拆解序列规划模型,并设计了基于强化学习的遗传算法。为了验证模型的正确性,构造了混合整数线性规划模型。算法构造了基于目标导向的编解码策略,以提高初始解的质量;采用Q学习来选择算法迭代过程中的最佳交叉策略和变异策略,以增强算法的自适应能力。在一个34项任务的发动机拆解案例中,通过与四种经典多目标算法对比,验证了所提算法的优越性;分析所得拆解方案,结果表明机器人并行拆解模式可以有效缩短完工时间,并降低拆解能耗。

关键词:拆解序列规划;机器人并行拆解;混合整数线性规划模型;遗传算法;强化学习

中图分类号:TP18;TP301.6 **文献标志码:**A

文章编号:1001-2486(2025)02-024-11



论
文
拓
展

Robotic parallel disassembly sequence planning method based on reinforcement learning and genetic algorithm

WANG Kaipu¹, MA Xiaoyi¹, LU Chao², YIN Lijiang^{3*}, LI Xinyu⁴

(1. School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China;

2. School of Computer Science, China University of Geosciences(Wuhan), Wuhan 430078, China;

3. School of Economics and Management, Hubei University of Automotive Technology, Shiyan 442002, China;

4. School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: To improve the disassembly efficiency and reduce disassembly energy consumption, the robotic parallel disassembly mode was introduced in the disassembly sequence planning problem, a robotic parallel disassembly sequence planning model was constructed, and a genetic algorithm based on reinforcement learning was designed. To verify the correctness of the model, a mixed integer linear programming model was constructed. In the algorithm, a goal-oriented encoding and decoding strategy was constructed to improve the quality of the initial solution. Q-learning was used to select the best crossover and mutation strategies in the iteration process to enhance the algorithm's adaptability. Finally, in an engine disassembly case with 34 tasks, the superiority of the proposed algorithm was verified by comparing with four classic multi-objective algorithms. The analysis of the disassembly schemes shows that the robotic parallel disassembly mode can effectively shorten the completion time and reduce disassembly energy consumption.

Keywords: disassembly sequence planning; robotic parallel disassembly; mixed integer linear programming model; genetic algorithm; reinforcement learning

拆解是装备维修与再制造、退役装备回收过程中的关键环节^[1]。拆解可得到有价值的零部

件,不仅可以节约资源、减少废弃物填埋量,还能带来一定的经济效益^[2]。为了提高拆解效率并

收稿日期:2024-01-31

基金项目:国家自然科学基金资助项目(52305552);湖北省自然科学基金资助项目(2023AFB138);武汉市知识创新专项曙光计划资助项目(2023020201020322);中央高校基本科研业务费专项资金资助项目(WUT233104001);武汉理工大学自主创新研究基金资助项目(104972024KFYjc0040)

第一作者:汪开普(1991—),男,湖北黄冈人,副研究员,博士,硕士生导师,E-mail:wangkaipu@whut.edu.cn

*通信作者:殷旅江(1980—),男,湖北十堰人,教授,博士,硕士生导师,E-mail:yinlvjiang@126.com

引用格式:汪开普,马晓艺,卢超,等.基于强化学习与遗传算法的机器人并行拆解序列规划方法[J].国防科技大学学报,2025,47(2):24-34.

Citation: WANG K P, MA X Y, LU C, et al. Robotic parallel disassembly sequence planning method based on reinforcement learning and genetic algorithm[J]. Journal of National University of Defense Technology, 2025, 47(2): 24-34.

降低拆解能耗,需要合理地规划拆解任务顺序,这类问题被称为拆解序列规划(disassembly sequence planning, DSP)问题。

常见的拆解方式有顺序拆解和并行拆解两种^[2]。顺序拆解中同一时刻只能进行一项拆解操作,而并行拆解在同一时刻允许同时拆解多个零部件。显然,并行拆解能够缩短产品的拆解完工时间^[3]。为了降低手工拆解的事故率,同时考虑机器人的灵活性,未来机器人拆解将成为手工拆解的替代方案^[4]。机器人并行拆解模式可以进一步提高拆解效率,同时需要考虑拆解过程中工具的更换^[5],本文将对机器人并行拆解序列规划(robotic parallel DSP, RPDSP)问题展开研究。

为了评价拆解效率,通常将拆解完工时间作为 RPDSP 的主要评价指标之一。此外,绿色制造要求减少拆解过程中的能耗。文献[6]将拆解能耗定义为基本拆解能耗和拆解任务转换能耗;文献[7]考虑回收产品能够节约制造的能耗。拆解能耗主要受机器人的功率、待机功率和工具转换等因素的影响。因此,RPDSP 的优化目标包括拆解完工时间和拆解能耗。

DSP 的优化方法主要包括精确方法和元启发式方法。文献[8]建立了以利润为导向的 DSP 模型,采用精确求解器进行求解。精确方法不适用于大规模问题,而元启发式方法能够在合理时间内找到近优解,如遗传算法^[9-11]、分散搜索算法^[6]、量子免疫克隆算法^[12]、人工蜂群算法^[7, 13]等均用于优化 DSP。上述算法在迭代过程中的策略是固定的,寻优能力有限,而多策略算法性能更优,如基于概率的交叉变异策略^[14-15]、基于组合的交叉变异策略^[16]用于求解多种组合优化问题。

以 Q 学习为代表的强化学习与元启发式算法相结合,能够利用 Q 表记录算法迭代中的状态变化,进而动态地改变寻优策略,提高多策略算法的自适应能力,得到更优的解^[17],如自适应遗传算法^[18]、基于 Q 学习的人工蜂群算法^[19]等成功应用于车间调度问题。鉴于这类算法的优越性能以及 RPDSP 问题的复杂性,本文将结合问题特征,设计基于强化学习的遗传算法,通过 Q 学习动态决策遗传算法策略。

综上,本文考虑将机器人并行拆解方式引入拆解序列规划问题中,建立最小化完工时间和能耗的多目标机器人并行拆解序列规划模型,并设计基于强化学习的遗传算法,通过工程案例来验

证所提模型和算法的性能。

1 机器人并行拆解序列规划模型

1.1 问题描述

RPDSP 是在已知机器人数量、拆解任务时间、优先关系、拆解工具等拆解信息的前提下,合理地将拆解任务分配到机器人中,尽可能缩短拆解完工时间,降低拆解能耗。

该问题的主要特点如下:

1) 机器人并行作业:充分利用机器人操作的灵活性,将多个机器人布置在产品的不同方位同时并行作业,如图 1 所示,4 个机器人可以同时作业,从而极大地缩短产品的下线时间。

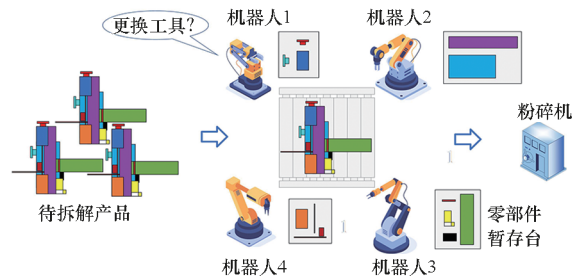


图 1 机器人并行拆解序列规划示意图

Fig. 1 Schematic diagram of robotic parallel disassembly sequence planning

2) 作业时间组成复杂:不同任务在不同机器人中的作业时间不同,拆解完工时间取所有机器人的最大完工时间。每个机器人的完工时间中包含拆解作业时间、工具更换时间以及待机时间。需要指出的是,若机器人作业开始时刻之前存在待机时间,且机器人末尾存在工具更换时间,则可将工具更换时间前移至作业开始时刻之前,以缩短机器人完工时间。

3) 拆解能耗组成复杂:机器人在不同工况下均会产生对应的能耗,机器人的能耗包含拆解作业能耗、工具更换能耗以及待机能耗。

4) 拆解约束多且复杂:在规划拆解任务序列时需要考虑任务之间的优先关系,即优先约束;多个任务可以在不同机器人中同时作业,则需要考虑任务的时间先后约束;此外,任务的位置决定了任务之间是否存在工具更换,即任务位置约束和工具更换约束。

该问题的难点在于如何合理地安排不同任务在不同机器人中的作业时间窗,同时满足作业时间约束、优先约束、任务位置约束、工具更换约束等多个拆解约束,并优化完工时间、拆解能耗等评价指标。

1.2 数学模型

本文数学模型符号如表 1 所示,决策变量如表 2 所示。

表 1 数学模型符号

Tab. 1 Symbols in the mathematical model

符号	含义
i, j	拆解任务编号,拆解任务集合为 I ,且 $i, j \in I$,最大拆解任务数为 $ I $
m	机器人编号,机器人集合为 M ,最大机器人数量为 $ M $
t_{im}	任务 i 在机器人 m 中的拆解时间
o_i	任务 i 的工具类型,工具类型总数为 O
k	任务在机器人中的位置,位置集合为 K ,最大位置为 $ K $;设集合 $K' = \{1, \dots, K - 1\}$
t_b	拆解工具更换时间
p_{ij}	任务之间的优先关系属性:若任务 i 是任务 j 的直接紧前任务,则 $p_{ij} = 1$;否则 $p_{ij} = 0$
e_m^d	机器人 m 拆解作业时的单位时间能耗
e_m^c	机器人 m 更换工具时的单位时间能耗
e_m^s	机器人 m 待机时的单位时间能耗
ψ	极大数

表 2 数学模型决策变量

Tab. 2 Decision variables in the mathematical model

变量	含义
T_C	所有机器人的最大完工时间(非负变量)
x_{im}	1,任务 i 被分配到机器人 m ;否则为 0
y_m	1,机器人 m 启用;否则为 0
z_{imk}	1,任务 i 被分配到机器人 m 的第 k 位置处;否则为 0
a_{im}	1,任务 i 是机器人 m 中的最后一个拆解任务;否则为 0
b_m	1,机器人 m 中首尾任务的拆解工具不同;否则为 0
d_{mk}	1,机器人 m 的第 k 位置与第 $k + 1$ 位置处任务的拆解工具不同;否则为 0
t_i^f	任务 i 的完工时刻(非负变量)
T_m^s	机器人 m 的开始作业时刻(非负变量)

机器人并行拆解序列规划问题的数学模型如下所示。

综合优化目标:最小化完工时间和拆解能耗,其表达式为

$$\min F = (f_1, f_2) \quad (1)$$

子目标 1:所有机器人的最大完工时间,即

$$f_1 = T_C \quad (2)$$

子目标 2:所有机器人的拆解能耗总和,即

$$f_2 = \sum_{m \in M} \sum_{i \in I} e_m^d x_{im} t_{im} + \sum_{m \in M} e_m^c t_b \left(\sum_{k \in K'} d_{mk} + b_m \right) + \sum_{m \in M} e_m^s \left(T_C - \sum_{i \in I} x_{im} t_{im} \right) \quad (3)$$

约束条件主要包括拆解模式约束、作业时间约束、优先约束、任务位置约束、工具更换约束,具体约束条件如下:

约束 1:采用完全拆解模式,即所有任务都被拆解,其表达式为

$$\sum_{m \in M} x_{im} = 1 \quad \forall i \in I \quad (4)$$

约束 2:机器人 m 开始作业时刻约束,即

$$T_m^s = \sum_{i \in I} z_{iml} (t_i^f - t_{im}) \quad \forall m \in M \quad (5)$$

线性化 $z_{iml} t_i^f$,令 $\pi_{im} = z_{iml} t_i^f$,则新的约束 2 如式(6)所示:

$$T_m^s = \sum_{i \in I} (\pi_{im} - z_{iml} t_{im}) \quad \forall m \in M \quad (6)$$

对 π_{im} 进行约束,如式(7)~(8)所示:

$$\pi_{im} \leq t_i^f \quad i \in I, \forall m \in M \quad (7)$$

$$t_i^f - \psi(1 - z_{iml}) \leq \pi_{im} \leq \psi \cdot z_{iml} \quad \forall i \in I, \forall m \in M \quad (8)$$

约束 3:单个机器人的完工时间约束,即

$$t_i^f \geq x_{im} t_{im} \quad \forall i \in I, \forall m \in M \quad (9)$$

约束 4:拆解序列的完工时间约束,即

$$x_{im} t_i^f + \max(0, b_m t_b - T_m^s) \leq T_C \quad \forall i \in I, \forall m \in M \quad (10)$$

线性化式(10),引入 0-1 变量 $\tau_{im}^f = x_{im} t_i^f$ 和非负变量 $\tau_m^b = \max(0, b_m t_b - T_m^s)$,新的约束 4 如式(11)所示:

$$\tau_{im}^f + \tau_m^b \leq T_C \quad \forall i \in I, \forall m \in M \quad (11)$$

对 τ_{im}^f 进行约束,如式(12)~(13)所示:

$$\tau_{im}^f \leq t_i^f \quad \forall i \in I, \forall m \in M \quad (12)$$

$$t_i^f - \psi(1 - x_{im}) \leq \tau_{im}^f \leq \psi \cdot x_{im} \quad \forall i \in I, \forall m \in M \quad (13)$$

引入辅助 0-1 变量 θ_m 对 τ_m^b 进行约束,如式(14)~(17)所示:

$$\tau_m^b \geq b_m t_b - T_m^s \quad \forall m \in M \quad (14)$$

$$\tau_m^b \leq \psi \cdot \theta_m \quad \forall m \in M \quad (15)$$

$$\tau_m^b \leq b_m t_b - T_m^s + \psi(1 - \theta_m) \quad \forall m \in M \quad (16)$$

$$\tau_m^b \leq t_b \quad \forall m \in M \quad (17)$$

约束 5:任务的拆解时刻存在优先关系,任务的时间窗约束可表示为

$$t_j^f - t_i^f + \psi(3 - \sum_{m \in M} x_{im} - \sum_{m \in M} x_{jm} - p_{ij}) \geq \sum_{m \in M} x_{jm} t_{jm} \quad (18)$$

$$\forall i, j \in I, i \neq j$$

约束 6:机器人中的相邻任务约束,即

$$t_j^f - t_i^f + \psi(2 - z_{imk} - z_{jm, k+1}) \geq t_{jm} + d_{mk} t_b \quad (19)$$

$$\forall i, j \in I, i \neq j; \forall m \in M; \forall k \in K'$$

约束 7:机器人中任务位置的优先约束,即

$$\sum_{k \in K} k \cdot z_{imk} - |K|(3 - x_{im} - x_{jm} - p_{ij}) \leq \sum_{k \in K} k \cdot z_{jmk} \quad \forall i, j \in I, i \neq j; \forall m \in M \quad (20)$$

约束 8:位置约束,机器人中的任务都被分配了位置,其表达式为

$$\sum_{k \in K} z_{imk} = x_{im} \quad \forall i \in I, \forall m \in M \quad (21)$$

约束 9:位置约束,机器人中的位置不一定被分配了任务,其表达式为

$$\sum_{i \in I} z_{imk} \leq 1 \quad \forall m \in M, \forall k \in K \quad (22)$$

约束 10:位置约束,在机器人中按位置顺序分配任务,其表达式为

$$\sum_{i \in I} z_{imk} \geq \sum_{j \in I} z_{jm, k+1} \quad \forall m \in M, \forall k \in K' \quad (23)$$

约束 11:机器人启用与任务之间的关系,即

$$y_m \leq \sum_{i \in I} x_{im} \leq |I| y_m \quad \forall m \in M \quad (24)$$

约束 12:机器人中相邻任务的工具类型改变约束可表示为

$$\sum_{i \in I} o_i z_{imk} - \sum_{j \in I} o_j z_{jm, k+1} \leq O \cdot d_{mk} + O \cdot (2 - \sum_{i \in I} z_{imk} - \sum_{j \in I} z_{jm, k+1}) \quad \forall m \in M, \forall k \in K' \quad (25)$$

$$\sum_{j \in I} o_j z_{jm, k+1} - \sum_{i \in I} o_i z_{imk} \leq O \cdot d_{mk} + O \cdot (2 - \sum_{i \in I} z_{imk} - \sum_{j \in I} z_{jm, k+1}) \quad \forall m \in M, \forall k \in K' \quad (26)$$

$$2d_{mk} \leq \sum_{i \in I} z_{imk} + \sum_{j \in I} z_{jm, k+1} \quad \forall m \in M, \forall k \in K' \quad (27)$$

$$d_{mk} \leq \max(\sum_{i \in I} o_i z_{imk} - \sum_{j \in I} o_j z_{jm, k+1}, \sum_{j \in I} o_j z_{jm, k+1} - \sum_{i \in I} o_i z_{imk}) \quad \forall m \in M, \forall k \in K' \quad (28)$$

线性化式(28),引入非负变量 β_{mk} 和 0-1 变量 α_{mk} ,式(28)可改写为式(29)~(33):

$$\beta_{mk} \geq \sum_{i \in I} o_i z_{imk} - \sum_{j \in I} o_j z_{jm, k+1} \quad \forall m \in M, \forall k \in K' \quad (29)$$

$$\beta_{mk} \geq \sum_{j \in I} o_j z_{jm, k+1} - \sum_{i \in I} o_i z_{imk} \quad \forall m \in M, \forall k \in K' \quad (30)$$

$$\beta_{mk} \leq \sum_{i \in I} o_i z_{imk} - \sum_{j \in I} o_j z_{jm, k+1} + 2O \cdot \alpha_{mk} \quad (31)$$

$$\forall m \in M, \forall k \in K'$$

$$\beta_{mk} \leq \sum_{j \in I} o_j z_{jm, k+1} - \sum_{i \in I} o_i z_{imk} + 2O(1 - \alpha_{mk}) \quad (32)$$

$$\forall m \in M, \forall k \in K'$$

$$d_{mk} \leq \beta_{mk} \leq 0 \quad \forall m \in M, \forall k \in K' \quad (33)$$

约束 13:机器人中最后位置处的任务约束,机器人启用时存在最后位置任务,可表示为

$$\sum_{i \in I} a_{im} = y_m \quad \forall m \in M \quad (34)$$

约束 14:机器人中任意任务不一定处于最后位置处,其表达式为

$$a_{im} \leq \sum_{k \in K} z_{imk} \quad \forall i \in I, \forall m \in M \quad (35)$$

约束 15:机器人中任务处于最后位置时的约束可表示为

$$\sum_{k \in K} k \cdot z_{imk} \geq \sum_{j \in I} \sum_{k \in K} z_{jmk} - |K|(2 - a_{im} - x_{im}) \quad (36)$$

$$\forall i \in I, \forall m \in M$$

约束 16:机器人中首尾位置处工具类型改变约束可表示为

$$\sum_{i \in I} o_i a_{im} - \sum_{j \in I} o_j z_{jm1} \leq O \cdot b_m + O(2 - \sum_{i \in I} a_{im} - \sum_{j \in I} z_{jm1}) \quad (37)$$

$$\forall m \in M$$

$$\sum_{j \in I} o_j z_{jm1} - \sum_{i \in I} o_i a_{im} \leq O \cdot b_m + O(2 - \sum_{i \in I} a_{im} - \sum_{j \in I} z_{jm1}) \quad \forall m \in M \quad (38)$$

$$b_m \leq y_m \quad \forall m \in M \quad (39)$$

$$b_m \leq \max(\sum_{i \in I} o_i a_{im} - \sum_{j \in I} o_j z_{jm1}, \sum_{j \in I} o_j z_{jm1} - \sum_{i \in I} o_i a_{im}) \quad \forall m \in M \quad (40)$$

线性化式(40),引入非负变量 ξ_m 和 0-1 变量 ζ_m ,式(40)可改写为式(41)~(45):

$$\xi_m \geq \sum_{i \in I} o_i a_{im} - \sum_{j \in I} o_j z_{jm1} \quad \forall m \in M \quad (41)$$

$$\xi_m \geq \sum_{j \in I} o_j z_{jm1} - \sum_{i \in I} o_i a_{im} \quad \forall m \in M \quad (42)$$

$$\xi_m \leq \sum_{i \in I} o_i a_{im} - \sum_{j \in I} o_j z_{jm1} + 2O \cdot \zeta_m \quad \forall m \in M \quad (43)$$

$$\xi_m \leq \sum_{j \in I} o_j z_{jm1} - \sum_{i \in I} o_i a_{im} + 2O(1 - \zeta_m) \quad \forall m \in M \quad (44)$$

$$b_m \leq \xi_m \leq 0 \quad \forall m \in M \quad (45)$$

综上,机器人并行拆解序列规划问题的混合整数线性规划模型可表示为:

优化目标:式(1)~(3)。

约束条件:式(4)、式(6)~(9)、式(11)~(27)、式(29)~(39)、式(41)~(45)。

1.3 模型验证与示例

采用精确求解器 CPLEX 对数学模型进行验证,并通过拆解方案对问题特征进行说明。如图 2 所示的 11 项拆解任务案例中,存在 3 种拆解工具,分别用 O_1 、 O_2 、 O_3 来表示。提供 4 种不同类型的机器人,不同机器人拆解同一任务的拆解时间不同,如表 3 所示。

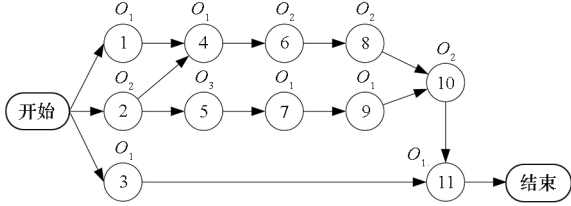


图 2 11 项拆解任务优先关系

Fig. 2 Precedence relationship of 11 disassembly tasks

模型中的参数分别设置为: $t_b = 2$ s, $e_m^d = [0.3$ kW, 0.25 kW, 0.35 kW, 0.4 kW], $e_m^c = 0.8 \cdot e_m^d$ kW, $e_m^s = 0.1 \cdot e_m^d$ kW, $\psi = 122$ 。分别以两个子目标为优化目标进行计算,即求解两个混合整数线性规划模型,优化结果如表 4 所示。结果方案的甘特图如图 3 所示,图中红色框表示工具更换时间。

表 3 机器人在 11 项任务上的拆解时间

Tab. 3 Disassembly time of robots on 11 tasks

机器人编号	任务										
	1	2	3	4	5	6	7	8	9	10	11
1	8	9	15	12	10	8	12	10	12	10	14
2	10	8	17	11	10	8	11	11	13	11	12
3	8	7	16	10	9	11	7	9	12	12	13
4	9	8	12	10	8	9	10	12	12	10	12

表 4 11 项任务的拆解案例结果

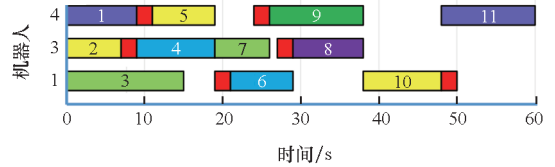
Tab. 4 Results of the disassembly cases with 11 tasks

子目标	结果值	差距值/%	运行时间/s
f_1	60	1.67	77.78
f_2	39.105	0.62	69.95

在图 3(a) 所示的完工时间最小的拆解方案中,启用了 3 个机器人,分别为机器人 1、3、4。完工时间主要受优先关系、工具更换以及机器人拆解时间的影响。该方案的完工时间为 60 s,可以从中提取出关键拆解路径:2→5→7→9→10→11。

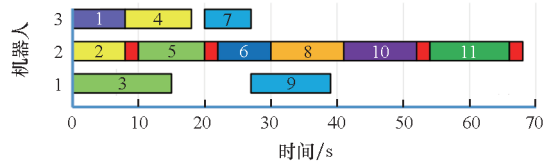
任务 2 和 5 的作业时间差主要是工具更换时间,任务 5、7、9、10、11 的时间窗由优先关系决定。因此,优化完工时间需要找到关键拆解路径。

在图 3(b) 所示的能耗最小拆解方案中同样启用了 3 个机器人,分别为机器人 1、2、3。机器人 2 的作业时间最长,其原因是拆解能耗主要由机器人的拆解作业功率与拆解时间来决定,机器人 2 的功率最小,让功率小的机器人多承担拆解任务可以降低总能耗。因此,在优化能耗时应尽可能给功率小的机器人多分配任务。



(a) 完工时间最小的拆解方案

(a) Disassembly scheme with the minimum completion time



(b) 能耗最小的拆解方案

(b) Disassembly scheme with the minimum energy consumption

图 3 11 项任务的拆解方案甘特图

Fig. 3 Gantt chart of the disassembly scheme with 11 tasks

本节精确方法只适用于求解小规模问题,针对大规模多目标优化问题,考虑采用元启发式算法进行求解。遗传算法作为经典的元启发式算法,与强化学习结合后,在组合优化问题中表现出优越的性能,因此下一节将设计基于强化学习的遗传算法。

2 基于强化学习的遗传算法

遗传算法主要包括初始化、交叉、变异等算子,其中交叉和变异存在多种操作方式,强化学习可以在算法迭代过程中根据算法环境动态地决策遗传算法的操作算子,以增强算法的寻优能力。基于强化学习的遗传算法框架如图 4 所示。

2.1 编码策略

编码序列包含拆解任务序列和机器人分配序列。拆解任务序列采用整数编码,基于任务优先关系来确定拆解任务顺序。机器人分配序列由拆解任务对应的机器人编号组成。在编码阶段可以采用如下所示的启发式规则。

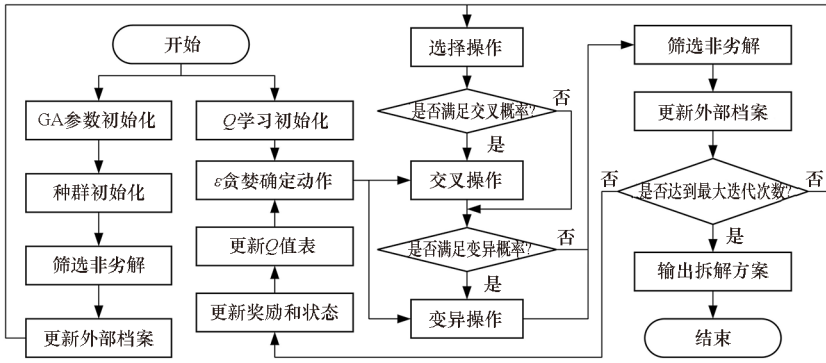


图 4 基于强化学习的遗传算法框架

Fig. 4 Framework of genetic algorithm based on reinforcement learning

规则 1:最小完工时间与关键拆解路径相关,因此在编码任务时应尽可能优先分配有优先约束的任务及其直接紧前任务,且在分配机器人时应使关键路径上任务的拆解时间尽可能短,从而缩短拆解完工时间。

规则 2:拆解能耗主要和机器人功率与拆解时间之积相关,因此机器人功率与拆解时间之积较小的机器人应该多分配任务,反之则应该少分配任务,以降低总的拆解能耗。

需要指出的是,规则 1 中强调的是优先分配存在优先约束的任务,而无优先约束的任务应尽可能分配到机器人的空闲时间窗中,避免任务分配时因优先约束而存在过多的空闲时间。具体编码过程如算法 1 所示。

算法 1 编码策略

Alg. 1 Encoding strategy

输入: 任务数 $|I|$, 优先矩阵 P , 种群数 N , 拆解时间 T
 输出: 编码序列 P_{op}

1. **while** $|P_{op}| < N$ **do**
2. **for** $k = 1$ to $|I|$ **do**
3. 根据优先矩阵和规则 1 挑选任务 i_k
4. 根据拆解时间和规则 2 挑选机器人 m_k
5. 更新优先矩阵 P
6. **end for**
7. 得到可行编码序列 $P_{op_n} = \{i_k; m_k\}$
8. **end while**
9. **return** $P_{op} = \{P_{op_n}\}$

2.2 解码策略

解码时需要确定任务的作业时间窗,分配任务时需要考虑优先约束和工具更换时间,并且尽可能使任务之间无空闲时间,从而确保完工时间最短、能耗最小。解码策略如算法 2 所示。

算法 2 解码策略

Alg. 2 Decoding strategy

输入: 编码序列, 任务数 $|I|$, 优先矩阵 P , 种群规模 N , 任务拆解时间 T , 拆解工具 O
 输出: 解码序列

1. **while** $|P_{op}| < N$ **do**
2. **for** $k = 1$ to $|I|$ **do**
3. 根据矩阵 P 确定位置 k 处任务的所有紧前任务的最大完工时间 T_{max}^p
4. 确定机器人 m_k 中的最大完工时间 T_{max}^m
5. 根据工具 O 信息确定当前任务与机器人 m_k 中的末位任务是否存在工具更换时间 t_b
6. 当前任务的开始时刻为 $\max(T_{max}^p, T_{max}^m + t_b)$
7. 确定任务的完工时刻 $t_m^k + \max(T_{max}^p, T_{max}^m + t_b)$
8. **end for**
9. 确定每个机器人的任务开始时刻 T_m^s 与最后一项任务的完成时刻 T_m^f
10. 确定每个机器人的最后一项任务与第一项任务是否存在工具更换时间 t_b
11. 机器人 m 中完工时间 $T_m = T_m^f + \max(0, t_b - T_m^s)$
12. 拆解序列的最大完工时间为 $T_C = \max(T_m)$
13. **end while**
14. **return** 所有任务的作业时间窗以及最大完工时间

2.3 选择算子

因 RPDSO 是多目标优化问题,在进行拆解方案选择时无法直接比较适应度值的大小,而是通过多目标评价指标超体积 (hypervolume, HV) 来评价拆解序列的质量,记为 h , 设参考点为 $r^* = (r_1^*, r_2^*)^T$, 则 h 值为染色体 x 的目标值到参考点 r^* 之间的超体积,其表达式为

$$h(x) = [(r_1^* - f_1(x))/r_1^*] \times [(r_2^* - f_2(x))/r_2^*] \quad (46)$$

可以计算染色体 x_i 被选择的概率 $P(x_i)$ 为

$$P(x_i) = 0.1 + 0.9h(x_i) / \sum_{i=1}^N h(x_i) \quad (47)$$

2.4 交叉算子

受优先约束的制约,交叉过程中需确保结果的可行性,为此设计了以映射交叉为主的 5 种交叉方式,包括单点交叉、双点内交叉、双点外交叉、多点交叉、随机交叉,如图 5 所示。前四种交叉方式均采用映射交叉,父代 1 中需要交叉的序列通

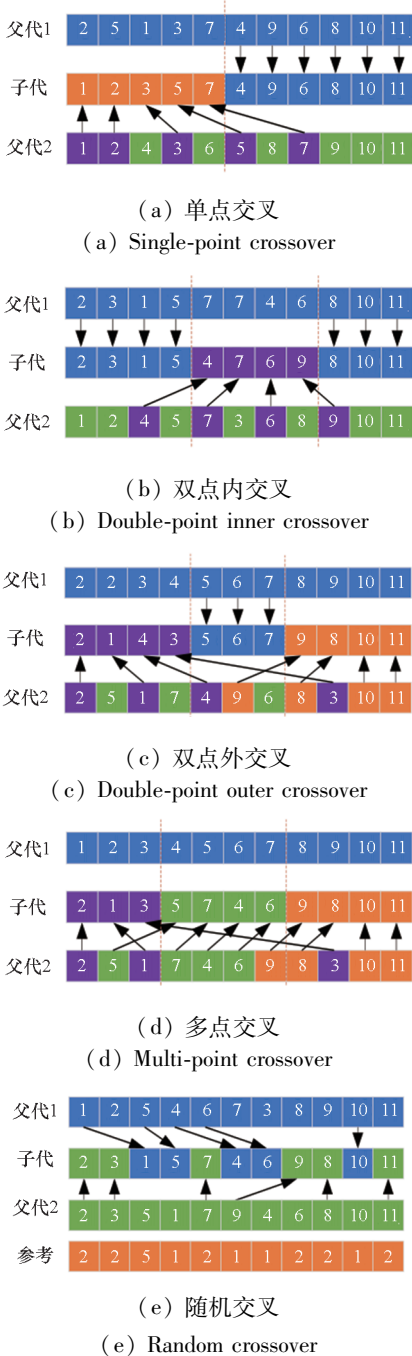


图 5 交叉算子

Fig. 5 Crossover operator

过父代 2 映射得到,这些序列满足优先约束,则组成的子代同样满足优先约束。随机交叉中按照参考序列中的顺序依次选择两个父代中的个体,组成新的子代同样满足优先约束。

2.5 变异算子

同样地,受优先约束制约,变异操作也需要考虑结果的可行性。通常有交换变异和插入变异两种方式,如图 6 所示。在确定变异点后,先找到该任务的直接紧前任务和紧后任务,则该任务可以变动的范围在距离其最近的紧前任务和紧后任务之间。若选择的两个变异点互为彼此可变动区间中的任务,则可执行交换变异。

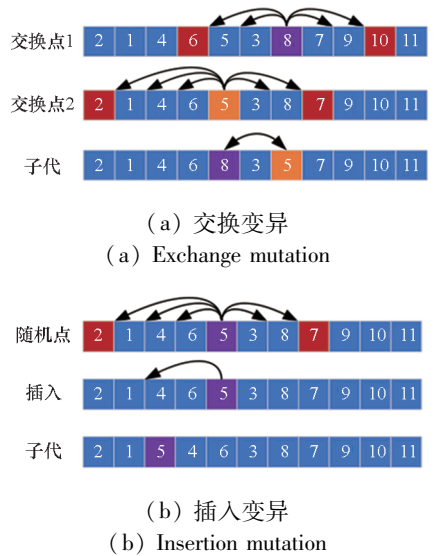


图 6 变异算子

Fig. 6 Mutation operator

2.6 强化学习策略

本文用到的强化学习是 Q 学习,主要包括状态(s)、动作(a)、奖励值(r)和 Q 值表等要素。在基于 Q 学习的多目标遗传算法 (multi-objective genetic algorithm based on Q learning, MOGA-QL) 中, Q 学习根据遗传算法的状态 s 来执行最佳的动作 a ,即执行最佳的交叉变异算子^[20],同时获取奖励值 r ,并更新 Q 值表,得到新的状态 s' ,以此循环直至算法结束。

在评价算法的状态(s^*)时,主要考虑算法在迭代过程中种群的平均适应度(f^*)、种群的多样性(d^*)和最佳个体的适应度值(m^*),其表达式如式(48)~(51)所示。

$$f^* = \sum_{i=1}^N h(x_i^g) / \sum_{i=1}^N h(x_i^1) \quad (48)$$

$$d^* = \left[\sum_{i=1}^N |h(x_i^g) - h(x_i^1)| / N \right] / \left[\sum_{i=1}^N |h(x_i^1) - h(x_i^1)| / N \right] \quad (49)$$

$$m^* = \max_{1 \leq i \leq N} h(x_i^g) / \max_{1 \leq i \leq N} h(x_i^1) \quad (50)$$

$$s^* = w_1 f^* + w_2 d^* + w_3 m^* \quad (51)$$

式中: $h(x_i^1)$ 表示第一次迭代后个体 x_i 的超体积值; g 表示迭代次数; w_1 、 w_2 、 w_3 为权重系数,分别取0.35、0.35和0.3。 $s^* \in [0, 1]$,将其分为20个区间,区间间距为0.05, s^* 落于不同区间表示算法处于不同的状态。

遗传算法中有5种交叉方式和2种变异方式,交叉与变异组合共10种动作。奖励值为算法迭代过程中种群超体积的增量,其表达式为

$$r = \sum_{i=1}^N h(x_i^g) - \sum_{i=1}^N h(x_i^{g-1}) \quad (52)$$

综上所述可知, Q 学习中存在20种状态和10种动作,则 Q 值表的维度为 20×10 。初始 Q 值表值为0,迭代过程中其更新公式^[21]为

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_a Q(s', a)] \quad (53)$$

式中, α 表示学习率, γ 表示折扣因子。

2.7 算法伪代码

MOGA-QL的伪代码如算法3所示。

算法3 MOGA-QL 伪代码

Alg.3 Pseudocode of MOGA-QL

输入:拆解特征数据,算法参数,包括种群数 N ,最大迭代次数 G ,交叉概率 p_c ,变异概率 p_m , Q 学习的学习率 α 、折扣因子 γ

输出:拆解方案

1. Q 值表置为0,外部档案置空
2. 算法初始化,生成 N 个个体,计算个体适应度值,根据Pareto支配关系更新外部档案
3. **for** $g=1$ to G **do** //迭代过程
4. 根据 ε 贪婪规则确定动作 a ,即交叉方式和变异方式
5. **for** $n=1$ to N **do** //种群个体寻优过程
6. 执行选择操作
7. **if** $rand < p_c$ **then**
8. 执行交叉操作
9. **end if**
10. **if** $rand < p_m$ **then**
11. 执行变异操作
12. **end if**
13. **end for**
14. 筛选非劣解,更新外部档案
15. 计算奖励值,更新状态 s 和 Q 值表
16. 更新贪婪概率 $\varepsilon = 1 - g/G$
17. **end for**
18. **return** 外部档案中的拆解方案

3 应用案例分析

以文献[22]中的发动机拆解为例,分析所提模型与算法的应用性能。该发动机含34项拆解任务,任务之间的优先关系如图7所示,共5种机器人,机器人的功率分别为0.9 kW、1.0 kW、1.2 kW、1.25 kW、0.75 kW,共5种类型的拆解工具,工具更换时间为3 s。

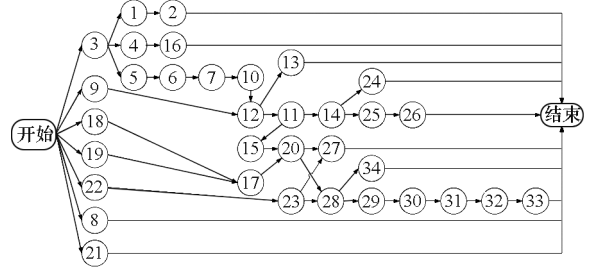


图7 发动机拆解任务的优先关系

Fig.7 Precedence relationships of engine disassembly tasks

3.1 算法对比

引入4种经典多目标算法与MOGA-QL进行对比,包括非支配排序遗传算法II(non-dominated sorting genetic algorithm II, NSGA-II)、强度Pareto进化算法2(strength Pareto evolutionary algorithm 2, SPEA2)、多目标粒子群优化(multi-objective particle swarm optimization, MOPSO)和多目标人工蜂群(multi-objective artificial bee colony, MOABC)算法。算法的种群数为100,最大运行时间为300 s;交叉和变异概率分别设为0.8和0.2;MOPSO中的加速度系数分别为1.5和1.5,权重系数为0.5;MOABC中的Limit设为100;MOGA-QL中的其他参数设置为 $\alpha=0.1$ 和 $\gamma=0.4$ 。

每种算法分别独立运行20次,将所有的结果进行非支配解筛选,将其目标值作为真实Pareto前沿。对结果进行归一化处理,并采用超体积、反世代距离(inverted generational distance, IGD)和广泛性(spread)评价算法的非支配解集。5种算法在3种指标上的均值和95%的置信区间如图8所示。

对比可知:在HV和IGD指标上,MOGA-QL的均值优于4种对比算法,MOGA-QL的置信区间跨度明显小于NSGA-II和SPEA2,且与对比算法的置信区间无重叠;在spread指标上MOGA-QL的均值优于MOPSO和MOABC,置信区间跨度与NSGA-II和SPEA2相差不大。对比结果表明MOGA-QL的收敛性和分散性优于对比算法。取每种算法取得最佳HV指标时算法的迭代数据,

绘制如图 9 所示的迭代图。

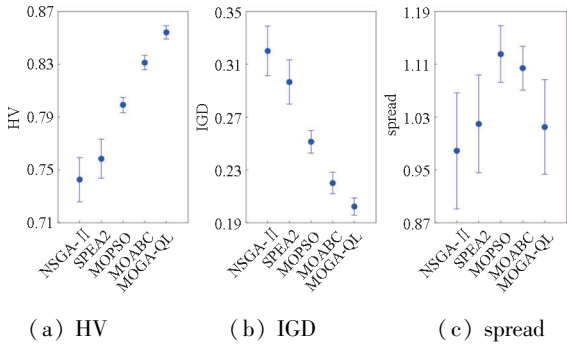


图 8 算法在评价指标上的均值和 95% 置信区间
Fig. 8 Mean values and 95% confidence intervals of algorithms on evaluation indicators

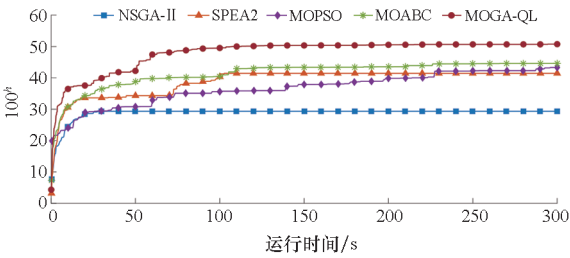


图 9 算法在 HV 指标上的迭代图
Fig. 9 Iteration diagram of algorithms on HV indicator

5 种算法取得最大 HV 值分别为 0.733 4、0.808 8、0.818 2、0.824 6、0.852 7, MOGA-QL 的收敛性优于 4 种对比算法, 5 种算法收敛性能从优到劣的排列顺序是 MOGA-QL > MOABC > MOPSO > SPEA2 > NSGA-II。绘制 5 种算法结果的 Pareto 前沿, 如图 10 所示。MOGA-QL 的 Pareto 前沿更靠近真实 Pareto 前沿, 且 MOGA-QL 取得了已知最优子目标值, 图中标记出了这两个方案 A 和 B。

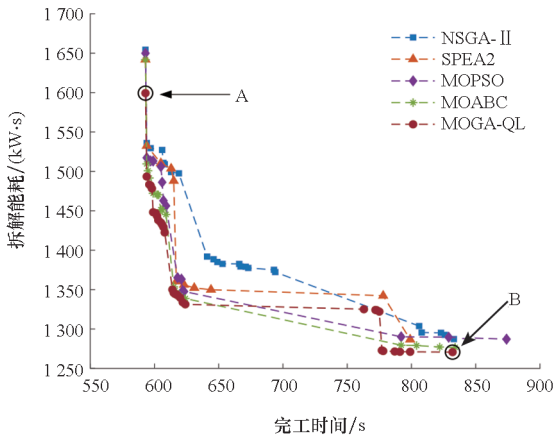


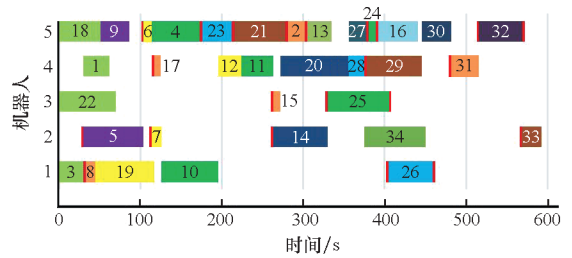
图 10 算法的 Pareto 前沿
Fig. 10 Pareto front of the algorithms

在对比算法中, NSGA-II、SPEA2 与所提 MOGA-QL 均采用了交叉、变异算子, 所不同的是

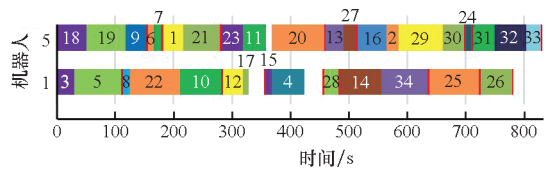
NSGA-II 和 SPEA2 采用的是确定型算子, 即在算法迭代过程中交叉与变异策略不变, 而所提 MOGA-QL 有 5 种交叉算子和 2 种变异算子, 通过 Q 学习动态决策算法迭代过程中的交叉与变异策略, 增强了算法的寻优能力。在 MOPSO 与 MOABC 中需要将连续性算子离散化, 且同样采用确定型算子, 其性能也比所提 MOGA-QL 差。

3.2 拆解方案分析

在图 10 给出的所有方案中: 方案 A 的完工时间最小(593 s), 拆解能耗为 1 599.29 kW·s; 方案 B 的拆解能耗最小, 为 1 270.65 kW·s, 完工时间为 832 s。观察图 10 可知, MOGA-QL 求得的 31 个拆解方案均为可选优质方案, 不同方案在两个子目标上各有侧重, 任何一个方案都无法同时使两个子目标最小。绘制方案 A 和 B 的甘特图, 如图 11 所示。图中红色表示工具更换时间, 空白部分表示待机时间。



(a) 拆解方案 A
(a) Disassembly scheme A



(b) 拆解方案 B
(b) Disassembly scheme B

图 11 拆解方案甘特图

Fig. 11 Gantt chart of disassembly scheme

方案 A 启用了 5 个机器人, 共产生 22 次工具更换, 工具更换时间为 66 s, 空闲时间为 1 536 s。在第 2 个机器人中, 因任务 5 与任务 33 的工具类型不同, 即机器人的首尾任务存在工具更换时间 3 s, 故将该工具更换时间提至任务 5 之前, 从而缩短第 2 个机器人的完工时间。该方案的关键路径是 3→5→6→7→10→12→11→15→20→28→29→30→31→32→33。该路径主要受优先约束制约, 无法并行拆解, 完工时间主要由该路径决定, 其他任务之间优先约束较少, 可以并行拆解。

方案 B 中只启用了机器人 1 和 5,拆解作业时间分别为 700 s 和 798 s,共产生 16 次工具更换,工具更换时间为 48 s,空闲时间为 166 s,远小于方案 A 中的空闲时间。机器人总能耗主要由拆解作业能耗和工具更换能耗组成,待机能耗相对较小。机器人 1 和 5 的功率小于另外 3 个机器人,故选用了这两个机器人,从而降低总的机器人拆解能耗。若机器人之间的任务分配不均,则会增大完工时间,进而导致机器人的空闲时间增加,过长的待机时间同样会产生可观的待机能耗。因此,需要在不同功率的机器人与拆解任务分配之间寻找平衡。

4 结论

针对装备维修与再制造以及退役装备回收的拆解规划问题,本文引入了机器人并行拆解模式,构建了机器人并行拆解序列规划模型,考虑了优先约束和拆解工具更换对拆解时间的影响,从完工时间和拆解能耗两个维度衡量拆解序列的性能。基于拆解特征,设计了基于强化学习的遗传算法,通过 Q 学习来决策算法迭代过程中的最佳交叉变异算子,提升了算法结果的质量。在发动机拆解案例中,验证了所提算法的性能优于四种经典多目标算法,且所提算法能够得到在完工时间和拆解能耗上各有侧重的多种拆解方案。

未来研究中可以考虑采用深度强化学习算法^[23]来指导拆解序列的规划,进一步增强算法对动态环境的适应性。

参考文献 (References)

- [1] ALLAGUI A, BELHADJ I, PLATEAUX R, et al. Reinforcement learning for disassembly sequence planning optimization [J]. *Computers in Industry*, 2023, 151: 103992.
- [2] ONG S K, CHANG M M L, NEE A Y C. Product disassembly sequence planning: state-of-the-art, challenges, opportunities and future directions [J]. *International Journal of Production Research*, 2021, 59(11): 3493–3508.
- [3] PISTOLESI F, LAZZERINI B. TeMA: a tensorial memetic algorithm for many-objective parallel disassembly sequence planning in product refurbishment [J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(6): 3743–3753.
- [4] LIU J Y, XU Z L, XIONG H, et al. Digital twin-driven robotic disassembly sequence dynamic planning under uncertain missing condition [J]. *IEEE Transactions on Industrial Informatics*, 2023, 19(12): 11846–11855.
- [5] WANG K P, GAO L, LI X Y, et al. Energy-efficient robotic parallel disassembly sequence planning for end-of-life products [J]. *IEEE Transactions on Automation Science and Engineering*, 2022, 19(2): 1277–1285.
- [6] GUO X W, ZHOU M C, LIU S X, et al. Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints [J]. *IEEE Transactions on Cybernetics*, 2020, 50(7): 3307–3317.
- [7] REN Y P, JIN H Y, ZHAO F, et al. A multiobjective disassembly planning for value recovery and energy conservation from end-of-life products [J]. *IEEE Transactions on Automation Science and Engineering*, 2021, 18(2): 791–803.
- [8] BENTAHA M L, VOISIN A, MARANGÉ P. A decision tool for disassembly process planning under end-of-life product quality [J]. *International Journal of Production Economics*, 2020, 219: 386–401.
- [9] REN Y P, MENG L L, ZHANG C Y, et al. An efficient metaheuristics for a sequence-dependent disassembly planning [J]. *Journal of Cleaner Production*, 2020, 245: 118644.
- [10] WANG K P, GUO J, DU B G, et al. A novel MILP model and an improved genetic algorithm for disassembly line balancing and sequence planning with partial destructive mode [J]. *Computers & Industrial Engineering*, 2023, 186: 109704.
- [11] ZHANG X S, FU A P, ZHAN C S, et al. Selective disassembly sequence planning under uncertainty using trapezoidal fuzzy numbers: a novel hybrid metaheuristic algorithm [J]. *Engineering Applications of Artificial Intelligence*, 2024, 128: 107459.
- [12] FENG Y X, CUI K Y, HONG Z X, et al. Disassembly sequence planning of product structure with an improved QICA considering expert consensus for remanufacturing [J]. *IEEE Transactions on Industrial Informatics*, 2023, 19(5): 7201–7213.
- [13] TIAN G D, REN Y P, FENG Y X, et al. Modeling and planning for dual-objective selective disassembly using and/or graph and discrete artificial bee colony [J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(4): 2456–2468.
- [14] GUAN Y X, CHEN Y N, GAN Z X, et al. Hybrid flow-shop scheduling in collaborative manufacturing with a multi-crossover-operator genetic algorithm [J]. *Journal of Industrial Information Integration*, 2023, 36: 100514.
- [15] GUTIÉRREZ-AGUIRRE P, CONTRERAS-BOLTON C. A multioperator genetic algorithm for the traveling salesman problem with job-times [J]. *Expert Systems with Applications*, 2024, 240: 122472.
- [16] CONTRERAS-BOLTON C, GATICA G, BARRA C R, et al. A multi-operator genetic algorithm for the generalized minimum spanning tree problem [J]. *Expert Systems with Applications*, 2016, 50: 1–8.
- [17] 张晓雄, 丁松, 李明浩, 等. 强化学习在多阶段装备组合规划问题中的应用 [J]. *国防科技大学学报*, 2021, 43(5): 127–136.
ZHANG X X, DING S, LI M H, et al. Application of reinforcement learning in multi-period weapon portfolio planning problems [J]. *Journal of National University of Defense Technology*, 2021, 43(5): 127–136. (in Chinese)
- [18] CHEN R H, YANG B, LI S, et al. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem [J]. *Computers & Industrial Engineering*, 2020, 149: 106778.

- [19] WANG J, TANG H T, LEI D M. A Q -learning artificial bee colony for distributed assembly flow shop scheduling with factory eligibility, transportation capacity and setup time[J]. *Engineering Applications of Artificial Intelligence*, 2023, 123: 106230.
- [20] 郭洪飞, 陆鑫宇, 任亚平, 等. 基于强化学习的群体进化算法求解双边多目标同步并行拆解线平衡问题[J]. *机械工程学报*, 2023, 59(7): 355–366.
GUO H F, LU X Y, REN Y P, et al. Reinforcement learning-based swarm evolutionary algorithm to solve two-sided multi-objective synchronous parallel disassembly line balancing problem[J]. *Journal of Mechanical Engineering*, 2023, 59(7): 355–366. (in Chinese)
- [21] 朱建文, 赵长见, 李小平, 等. 多约束强化学习最优智能滑翔制导方法[J]. *国防科技大学学报*, 2022, 44(4): 116–124.
- ZHU J W, ZHAO C J, LI X P, et al. Multi constraint optimal intelligent gliding guidance via reinforcement learning[J]. *Journal of National University of Defense Technology*, 2022, 44(4): 116–124. (in Chinese)
- [22] GO T F, WAHAB D A, AB RAHMAN M N, et al. Genetically optimised disassembly sequence for automotive component reuse[J]. *Expert Systems with Applications*, 2012, 39(5): 5409–5417.
- [23] LIU W, WANG R, ZHANG T, et al. Hybridization of evolutionary algorithm and deep reinforcement learning for multiobjective orienteering optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2023, 27(5): 1260–1274.