

基于改进遗传算法的异步并行拆卸序列规划

孙娴静^{1,2}, 唐秋华^{1,2}, 邓明星³

(武汉科技大学 1. 冶金装备及其控制教育部重点实验室; 2. 机械传动与制造工程湖北省重点实验室;
3. 汽车与交通工程学院, 湖北 武汉 430081)

摘要: 针对考虑优先关系约束和机器工作区域冲突约束的异步并行拆卸序列规划问题, 提出加入路径重连策略的遗传算法。通过定义零部件之间的与或优先关系, 对拆卸方向进行简化。且根据工作区域冲突约束, 避免零部件拆卸过程中产生工作区域冲突。在此基础上, 对种群进行编码解码, 并加入路径重连算子, 将遗传算法迭代得到的精英解进行路径重连以增强算法的局部搜索能力。最后通过多个实际案例证明算法的有效性。

关键词: 异步并行拆卸; 遗传算法; 路径重连

中图分类号: TH186

文献标志码: A

文章编号: 1007-7375(2022)04-0151-07

Asynchronous Parallel Disassembly Sequence Planning Based on Improved Genetic Algorithm

SUN Xianjing^{1,2}, TANG Qiuhua^{1,2}, DENG Mingxing³

(1. Key Laboratory of Metallurgical Equipment and Control of Ministry of Education;

2. Hubei Provincial Key Laboratory of Mechanical Transmission and Manufacturing Engineering;

3. School of Automobile and Traffic Engineering, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: Aiming at the problem of asynchronous parallel disassembly sequence planning considering the priority relationship constraint and the machine working area conflict constraint, a genetic algorithm with path reconnection strategy is proposed. The disassembly direction is simplified by defining the And/Or priority relationship between components. And via the working area conflict constraint, the working area conflict is avoided during the disassembly process of the parts. On this basis, the population is encoded and decoded, and a path reconnection operator is added to reconnect the elite solution obtained by the genetic algorithm iteration to enhance the local search ability of the algorithm. Finally, a number of actual cases prove the effectiveness of the algorithm.

Key words: asynchronous parallel disassembly; genetic algorithm (GA); path relinking

拆卸是将产品或装配体进行拆解, 获取目标零部件的过程。与焚烧、掩埋等其他报废产品处理方法相比, 通过拆卸来获取可直接重用或可修复的零部件, 实现资源的再利用, 故而拆卸具有显著的经济和环境优势, 在报废产品的再利用和再制造过程中具有重要的地位。

拆卸序列规划是指根据产品中零部件之间的几何结构、功能等一些预定的性能指标探索确定合理

可行的拆卸序列, 并从中选择最优的序列以指导产品的拆卸, 达到预期的拆卸目标。根据拆除零部件的逻辑顺序, 拆卸可以分为两大类: 顺序拆卸和并行拆卸。因此, 拆卸序列规划也可以分为顺序拆卸规划和并行拆卸规划。并行拆卸分为同步并行拆卸和异步并行拆卸。

到目前为止, 已经有很多学者对顺序拆卸规划问题进行了研究, 开发出许多模型和方法。但随着

收稿日期: 2020-12-10

基金项目: 国家自然科学基金资助项目(51875421)

作者简介: 孙娴静(1998—), 女, 湖北省人, 硕士研究生, 主要研究方向为拆卸序列规划。

通讯作者: 唐秋华(1970—), 女, 湖北省人, 教授, 博士, 主要研究方向为调度与优化。

E-mail: tangqiuhua@wust.edu.cn

技术的发展,对拆卸时效的要求逐渐提高,并行拆卸序列规划问题越来越受到学者的关注与研究。Zhang等^[1]针对性地建立拆卸混合图模型来解释4种不同的零部件之间的结构连接,然后提出协同拆卸层次树的生成方法,通过一种基于两用户定义变量的分支定界算法计算出并行拆卸序列规划的最小工作时间。随后,Zhang等^[2]基于模糊粗糙集,定义和推导了装配约束因子、拆卸优先因子、拆卸时间因子、拆卸工具因子和组合类型因子等5个拆卸因子,建立了并行拆卸的模糊-粗糙集映射模型,利用模糊-粗糙集映射模型生成最优的并行拆卸序列。Smith等^[3]利用模块化设计理论将待拆卸的零件转化为模块,并根据递归规则生成可行的拆卸序列,然后利用遗传算法寻找最优的并行拆卸序列。Ren等^[4]创建拆卸层次树形图来描述并行拆卸过程,并提出一种多目标离散人工蜂群算法,使求得的并行拆卸序列规划的拆卸时间最小化,利润最大化。

然而,以上文献主要针对同步并行拆卸序列规划,要求操作者同时开始拆卸操作,从而导致空闲时间多,作业效率低。因此,本文针对异步并行拆卸序列规划问题,提出改进遗传算法进行求解。

1 异步并行拆卸序列规划

异步并行拆卸是并行拆卸方式中的一种。异步并行拆卸是指允许操作者完成当前拆卸任务后,在不违背相关约束的情况下即时开始下一个拆卸任务,不必等待所有操作者都完成任务^[5]。在并行拆卸序列规划中,操作者开始执行拆卸操作的时间也尤为重要,会对产品拆卸的完成时间以及拆卸成本产生不小的影响。异步并行拆卸节省了部分操作者等待其他拆卸任务完成的空闲时间,减少了完成拆卸所需要的时间以及相关成本,更具有研究价值。

1.1 优先关系约束

优先关系表示产品零部件之间的拆卸先后顺序,根据逻辑关系对其进行划分,可以分为“与优先”关系和“或优先”关系。“与优先”关系指只有拆除所有紧前优先零部件,零件*i*才能被拆除。若同时有多个零部件对零件*i*具有优先关系,拆除其中某一个零部件,零件*i*即可拆卸,则零件*i*与其紧前优先零部件构成“或优先”关系^[6-7]。

图1为某产品的优先关系示意图。其中, $i = \{1, 2, \dots, 10\}$ 表示待拆卸任务,每个拆卸任务表示要拆

卸的零部件,其优先关系由优先级高的零件指向优先级低的零件。在图中,用有向点虚线边连接的任务2、3与任务1、8、9、10之间构成“或优先”关系。若零部件2被拆卸,零部件1、3、8、9、10均可被拆卸。其余有向实边接连的拆卸任务接触约束关系为“与优先”关系。

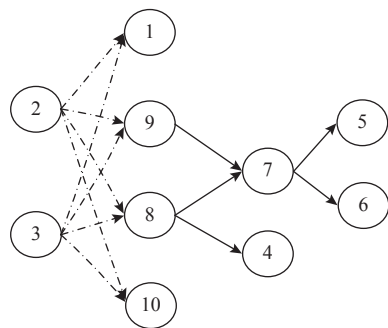


图1 某待拆卸产品的优先关系示意图

Figure 1 Schematic diagram of the priority relationship of a product to be disassembled

通常,在某一拆卸方向上的各零部件之间的优先约束关系总是严格对接的,当出现“或优先”关系则说明要想拆卸某一产品,至少有2种不同的拆卸方式,即至少有2种不同的拆卸方向。因此,在含有“或优先”关系的情况下,可以不用考虑产品的拆卸方向,在一定程度上简化了问题。

由此可以定义优先关系矩阵 $P = [p_{ij}]$ 来表示待拆卸产品的第*i*个拆卸任务与第*j*个拆卸任务之间的优先关系。

$$p_{ij} = \begin{cases} -1, & \text{任务} i \text{对任务} j \text{具有“或优先”关系;} \\ 1, & \text{任务} i \text{对任务} j \text{具有“与优先”关系;} \\ 0, & \text{其他情况。} \end{cases}$$

1.2 工作区域冲突约束

在异步并行拆卸序列规划中,通常是多个操作者共同拆卸一个产品,由于各零部件间存在复杂的装配关系,会存在当某个操作者拆卸某一零件时会占用另一个操作者的拆卸工作区域,使得另一个操作者的拆卸工作无法进行的问题。这时,2个操作者无法同时完成各自的拆卸任务,这种情况称为2个零部件之间存在工作区域冲突关系。

为了方便表达,可以像优先关系矩阵一样构造一个冲突矩阵 C 来描述。

$$C_{ij} = \begin{cases} 1, & \text{任务} i \text{对任务} j \text{存在工作区域冲突;} \\ 0, & \text{任务} i \text{对任务} j \text{不存在工作区域冲突。} \end{cases}$$

2 加入路径重连的改进遗传算法

2.1 编码

由于异步并行拆卸序列规划问题不仅涉及到每个任务之间的约束关系, 与操作者也相关, 因此采用双向量列表结构编码更加合适, 个体 ν 由拆卸任务向量 ν_1 和操作者向量 ν_2 组成。拆卸任务向量 $\nu_1 = \{i_1, i_2, \dots, i_k\}$, 向量中每个元素代表一个待拆卸任务, 由1到 K 的整数表示, K 为待拆卸产品的零部件总数。操作者向量 $\nu_2 = \{r_1, r_2, \dots, r_N\}$, 向量中每个元素代表执行向量 ν_1 中对应的拆卸任务的操作者, 由1到 N 的整数表示, N 为操作者数量。

在确定遗传算法的编码方式之后, 需要根据问题特点生成初始化种群, 以本文1.1节中提出的问题为例进行说明。拆卸任务向量 ν_1 只与待拆卸产品的各零部件有关。为方便解码, 需要在种群初始化时保证向量 ν_1 的可行性, 即严格遵循各零件之间的优先关系约束。由优先关系矩阵可知, 当优先关系矩阵 P 中第 j 列中的元素均为0时, 待拆卸任务 j 当前可拆卸, 不受其余零件的优先约束。将得到的可拆卸任务依次存入向量 ν_1 , 不断循环直至找出所有拆卸任务, 此时可得到拆卸任务向量 ν_1 , 操作者向量 ν_2 可以由操作者数量通过随机程序获得。初始化种群具体步骤如下。

步骤1 输入优先关系矩阵 P , 创建空向量 ν_1 和空集合 t_1 。

步骤2 若矩阵 P 为空矩阵, 则转到Step7, 否则执行Step3。

步骤3 找出矩阵 P 中元素全为0的列, 将列对应的拆卸任务存入集合 t_1 中。

步骤4 从集合 t_1 中随机选择一个待拆卸任务放入到向量 ν_1 最左端中。

步骤5 更新矩阵 P , 将已拆卸任务的行与列删除, 若删除的元素中包含“或优先”关系, 则同时对与之关联的其他“或优先”关系元素进行修改。

步骤6 将集合 t_1 中已拆卸任务删除, 返回Step2。

步骤7 根据操作者数量随机生成向量 ν_2 。

步骤8 输出向量 ν_1 和向量 ν_2 。

2.2 解码

在本文研究的异步并行拆卸序列规划问题中, 拆卸的目标为拆卸完成时间最小, 即最小化最大拆卸完成时间, 类似于JSP问题的最大完工时间。与

JSP问题不同, 在异步并行拆卸序列规划问题中还要考虑“与/或优先”关系和操作者的拆卸工作区域冲突问题。“或优先”关系存在不定向性, 在未开始拆卸之前, 多个具有“或优先”关系的待拆卸任务具有同样的拆卸优先级, 无法选择哪个任务先进行拆卸。

因此, 在解码开始之前, 要将拆卸任务之间的“或优先”关系解除。仍以1.1节中的问题为例, 通过编码, 该产品所有拆卸任务的拆卸顺序已经出来, 可以修改优先关系矩阵 P , 解除拆卸任务之间的“或优先”关系。假设个体编码 $\nu_1 = \{3, 9, 2, 10, 8, 4, 7, 1, 5, 6\}$, 则可以看作拆卸任务3对任务1、8、9、10具有“与优先”关系约束, 任务2不再对后续任务具有“或优先”关系约束。对应的, 优先关系矩阵可以修改为 $P_{21}, P_{28}, P_{29}, P_{2,10}$ 为0; $P_{31}, P_{38}, P_{39}, P_{3,10}$ 为1, 其余保持不变。

解码是根据种群个体编码计算出每个个体对应拆卸序列的最大完成时间。首先, 创建存放所有待拆卸任务开始拆卸时间的矩阵 st 和所有待拆卸零件结束时间的矩阵 ft 。按照编码 ν_1 的顺序对每个待拆卸零件进行拆卸, 将拆卸开始时间存入矩阵 st 中对应的位置, 拆卸结束时间存入矩阵 ft 中对应的位置。当遇到某一具有前序拆卸任务的拆卸任务时, 比较其前序任务的结束时间, 与所在并行序列的上一个任务结束时间, 取较大值为该任务的开始时间。当计算完时间后, 还需要判断当前任务是否存在在工作区域冲突, 若存在, 则对该个体进行惩罚, 若不存在则继续计算下一个任务。

判断零部件之间是否存在工作区域冲突主要有3个判断标准: 1) 两任务之间是否还存在优先关系; 2) 两任务中某一任务是否还有前序任务未执行; 3) 两个任务的执行时间是否重叠。若2个具有工作区域冲突关系的拆卸任务同时不满足以上3个条件, 则它们之间会产生工作区域冲突, 该解为不可行解。

2.3 遗传算子设计

遗传算法中的选择操作是对种群进行选择, 从而使优秀基因遗传到下一代。本文采用轮盘赌选择方法。

遗传算法中的交叉操作是对种群中被选择的两个个体的编码片段进行部分交叉, 从而产生两个新个体的过程。本文对向量 ν 采用优先保存交叉(precedence preservation crossover, PPX)方法^[8-9](图2)。

PPX可以将父代的优先关系传递给新的子代,从而保证子代的可行性;对向量 $\mathbf{v}_2$ 采用单点交叉方法,在向量 $\mathbf{v}_2$ 的长度范围内随机选择一个交叉点,将两个个体的向量 $\mathbf{v}_2$ 从交叉点处断开,并互换前半部分,即可获得两个新的子代 $\mathbf{v}_2$ 。

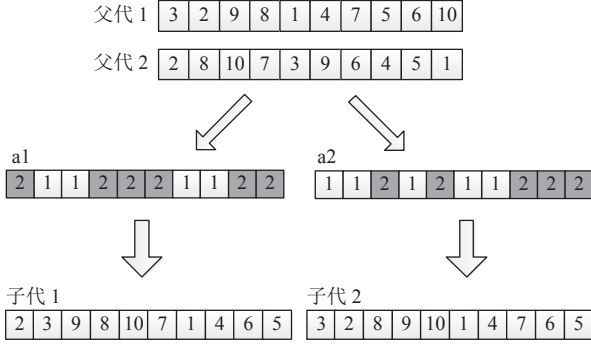


图 2 PPX操作示意图

Figure 2 Schematic diagram of PPX operation

遗传算法中的变异操作是以一定概率使个体中的基因发生变异以产生新个体的过程。若在向量 $\mathbf{v}_1$ 上进行变异操作,可能会产生不符合优先关系约束的不可行解。本文选择基于向量 $\mathbf{v}_2$ 进行一个简单突变:从向量 $\mathbf{v}_2$ 中随机选择一个变异点,将其值更换为其他并行序列。

2.4 路径重连算子设计

路径重连通过使用在搜索过程中得到的一组不同的高质量解对启发式算法实现进一步收敛和优化。在每个迭代中,路径重连应用于全局搜索阶段结束时找到的解决方案和从精英集中随机选择的解决方案,返回与这两个解决方案相似但可能更好的新解决方案^[10]。

2.4.1 路径距离计算

在算法开始前,定义以下参数^[11]。 M_o^s 为个体 S 中分配给拆卸任务 o 的操作者; L_M^s 为个体 S 中分配给操作者 M_o^s 的拆卸任务数; P_o^s 为 o 在操作者 M_o^s 上的位置。若两个个体中的拆卸任务 o 被分配给同一个操作者进行拆卸,则定义两个个体之间 o 的距离为

$$d_o(S_1, S_2) = |P_o^{S_1} - P_o^{S_2}|. \quad (1)$$

若两个个体中的拆卸任务 o 被分配给不同的操作者进行拆卸,定义两个个体之间 o 的距离为

$$d_o(S_1, S_2) = \min \left\{ P_o^{S_1} + P_o^{S_2}, \left(L_{M_o^{S_1}}^{S_1} - P_o^{S_1} \right) + \left(L_{M_o^{S_2}}^{S_2} - P_o^{S_2} \right) \right\}. \quad (2)$$

个体 S_1 和 S_2 之间的距离定义为

$$d(S_1, S_2) = \sum_{i=1}^n \sum_{j=1}^m d_{o_{ij}}(S_1, S_2). \quad (3)$$

2.4.2 重连策略

在遗传算法每次迭代完成后,选取出最优的部分个体组成精英集,并随机选择出两个个体,一个作为初始解 S_i ,一个作为引导解 S_g 进行路径重连。设 S_c 为当前解决方案(S_c 初始化为 S_i)。

首先,构造路径重连过程中得到的解集 N :对于 S_c 中的每个拆卸任务 o_i 依次进行判断,若 o_i 在解 S_c 和解 S_g 中分别位于不同操作者上,将 S_c 中的 o_i 移动到 S_g 中的操作者上(位置不变)。比较移动前后解 S_c 和 S_g 中 o_i 的距离,选择其中较短的存放入解集 N 中;若 o_i 在解 S_c 和解 S_g 中同一操作者的不同位置上,改变解 S_c 中 o_i 的位置,并将改变后的解存入解集 N 中。然后,删除解集 N 中的不可行解,以及到 S_g 的路径距离大于初始路径距离(解 S_c 到 S_g 的距离)的解,计算剩余解的拆卸完成时间。然后,对于 N 中剩下的解 S ,按拆卸完成时间大小进行升序排列,从中选择拆卸完成时间最小的解作为路径上的下一个 S_c 。若该解的最小完成时间优于初始解的拆卸完成时间,则同时将其存储在优解集合 $Path$ 中。重复以上步骤直到无法再得到更优的解。最后,返回优解集合 $Path$ 中完工时间最短的解 S_c 。

3 实验分析

3.1 案例说明

在1.1节的案例中,令操作者(并行序列)的数量为2,运行程序,结果如图3所示。

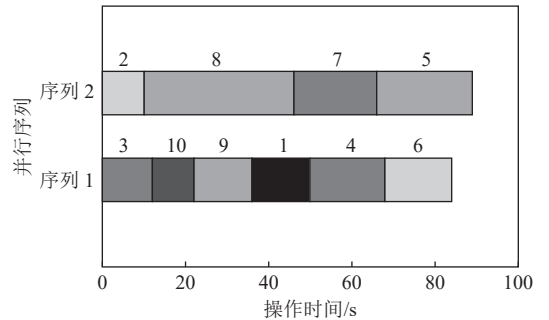


图 3 程序运行结果

Figure 3 Results on program running

图3中的柱形表示每个操作者进行操作的拆卸任务,各个任务从甘特图最左边开始依次被操作者进行拆卸,柱形上的数字对应的是正在拆卸的任务

的序号。

从图3中可以发现, 各操作者在零件拆卸过程中无空闲时间, 最大化利用了资源, 证明本文所述程序有效。

3.2 路径重连性能分析

3.2.1 参数校验

使用田口实验对精英解个数、种群大小、交叉概率、变异概率等参数进行校验, 结果如图4所示(信噪: 望小)。

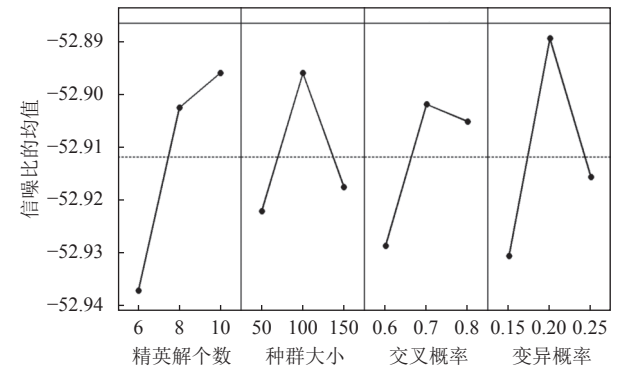


图4 平均完成时间的信噪比-主效应图

Figure 4 S/N ratio of average completion time-main effect diagram

信噪比是质量影响因子的主效应与误差效应的比值。一般地, 信噪比值越大, 其稳健性越好。由图4可知, 最佳参数组合: 精英解个数为10, 种群规模为100, 交叉概率为0.7, 变异概率为0.2。

3.2.2 路径重连有效性

为验证所提路径重连策略的性能, 令多个实际拆卸案例^[12-14]采用最佳参数组合进行运算。为了对多种路径重连方法进行区分, 将采用精英解与精英解之间进行路径重连的方法称为GA-PR1, 劣解向精英解进行路径重连的方法称为GA-PR2。固定上述算法的CPU运行时间 t 为 $N \times N \times 0.01$ s, 每个案例的3种算法均运行10次。表1展示了相同运行时间下各算法的相对分析误差(RPD)。

由表1可知, GA-PR1在最小RPD和平均RPD方面的表现均优于GA和GA-PR2。此外, 绘出3种算法在95%置信区间下的区间图, 如图5所示。结合图表可得, 在相同的CPU运行时间内, GA-PR1运行得到的结果更好且更加稳定。

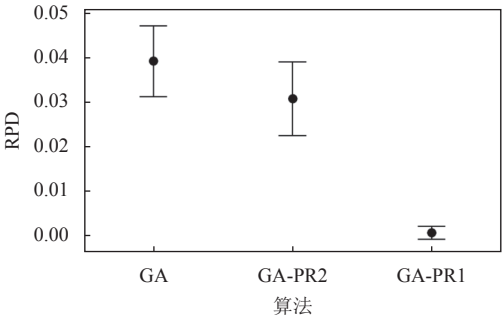
3.3 算法性能分析

为分析所提算法的综合性能, 将算法GA-PR1与参考文献[6]中提出的改进遗传算法IGA及改进离散蜜蜂算法MDBA^[15]进行对比实验。

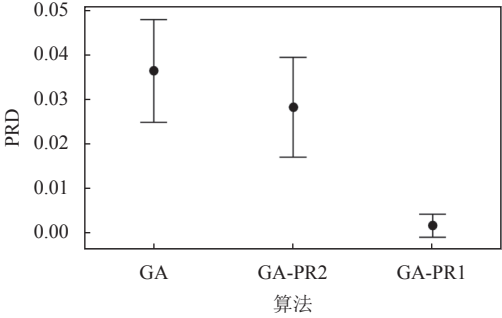
表1 GA、GA-PR2和GA-PR1的相对分析误差

Table 1 RPD of GA、GA-PR2 and GA-PR1

案例	M	GA		GA-PR2		GA-PR1	
		平均值	最小值	平均值	最小值	平均值	最小值
1	2	0.040	0.023	0.028	0.025	0.000	0.000
2	3	0.048	0.016	0.036	0.013	0.000	0.000
3	4	0.042	0.062	0.022	0.028	0.000	0.000
4	3	0.057	0.062	0.049	0.059	0.000	0.000
5	4	0.041	0.073	0.054	0.068	0.000	0.000
6	3	0.047	0.040	0.033	0.030	0.000	0.000
7	2	0.010	0.007	0.000	0.000	0.015	0.009
8	3	0.011	0.010	0.005	0.000	0.000	0.000
9	2	0.017	0.006	0.012	0.000	0.000	0.000
10	3	0.060	0.073	0.014	0.000	0.000	0.020
11	3	0.040	0.053	0.029	0.039	0.000	0.000
12	4	0.039	0.032	0.051	0.000	0.000	0.016
13	5	0.049	0.037	0.044	0.074	0.000	0.000
14	3	0.013	0.002	0.005	0.007	0.000	0.000
15	4	0.026	0.030	0.036	0.030	0.000	0.000
16	5	0.049	0.029	0.052	0.047	0.000	0.000
17	6	0.054	0.082	0.040	0.057	0.000	0.000
18	3	0.059	0.053	0.036	0.040	0.000	0.000
19	4	0.071	0.053	0.062	0.049	0.000	0.000
20	5	0.021	0.000	0.007	0.000	0.000	0.000
21	3	0.031	0.028	0.033	0.034	0.000	0.000



(a) 平均值



(b) 最小值

图5 3种算法(GA、GA-PR2和GA-PR1)在95%置信区间下的区间图

Figure 5 Interval graphs of the three algorithms (GA, GA-PR2 and GA-PR1) under the 95% confidence interval

通过田口实验确定各算法的最佳参数组合(表2),对21个案例在相同的CPU运行时间分别运行10次,结果如表3所示。

表 2 各算法参数校验结果

Table 2 Verification results of various algorithm parameters		
算法	参数	取值(范围)
IGA	种群大小	100
	交叉概率	0.8
	变异概率	0.2
MDBA	初始化侦察蜂数量	30
	最优侦察蜂数量	3
	较优侦察蜂数量	7
	最大迭代次数	100
	为最优侦查蜂派遣的觅食蜂数量	[3,10]
	为较优侦查蜂派遣的觅食蜂数量	[1,7]

表 3 IGA、MDBA和GA-PR1的相对分析误差

Table 3 RPD of the IGA、MDBA and GA-PR1							
案例	M	IGA		MDBA		GA-PR1	
		Ave	Min	Ave	Min	Ave	Min
1	2	0.050	0.030	0.023	0.011	0.000	0.000
2	3	0.078	0.084	0.034	0.000	0.000	0.003
3	4	0.072	0.052	0.000	0.000	0.012	0.014
4	3	0.040	0.036	0.051	0.035	0.000	0.000
5	4	0.030	0.044	0.045	0.063	0.000	0.000
6	3	0.045	0.025	0.071	0.045	0.000	0.000
7	2	0.019	0.000	0.009	0.047	0.000	0.021
8	3	0.006	0.000	0.000	0.000	0.010	0.000
9	2	0.009	0.003	0.021	0.009	0.000	0.000
10	3	0.032	0.008	0.045	0.004	0.000	0.000
11	3	0.009	0.013	0.051	0.053	0.000	0.000
12	4	0.014	0.033	0.034	0.000	0.000	0.033
13	5	0.026	0.056	0.046	0.056	0.000	0.000
14	3	0.015	0.002	0.017	0.011	0.000	0.000
15	4	0.013	0.018	0.037	0.027	0.000	0.000
16	5	0.058	0.047	0.054	0.029	0.000	0.000
17	6	0.046	0.037	0.050	0.025	0.000	0.000
18	3	0.065	0.047	0.039	0.016	0.000	0.000
19	4	0.079	0.039	0.022	0.000	0.000	0.000
20	5	0.024	0.000	0.000	0.000	0.002	0.000
21	3	0.018	0.021	0.021	0.019	0.000	0.000

根据表3绘制出3种算法在95%置信区间下的区间图,如图6所示。

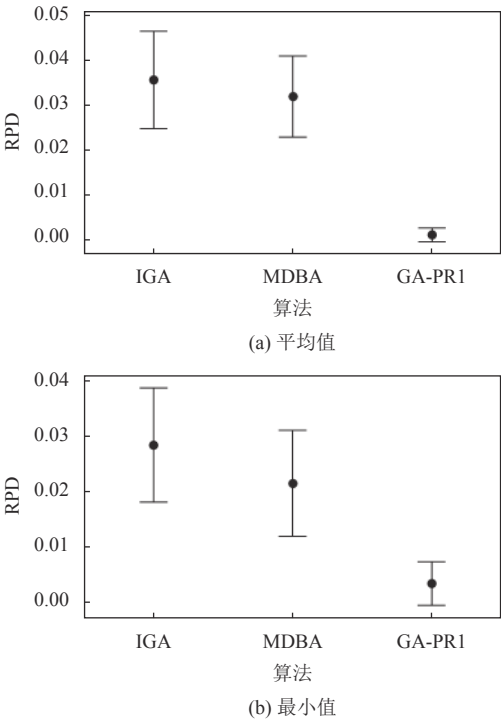


图 6 3种算法(IGA、MDBA和GA-PR1)在95%置信区间下的区间图
Figure 6 Interval graphs of the three algorithms (IGA, MDBA and GA-PR1) under the 95% confidence interval

从表3中可知,在平均值方面,相较于IGA,GA-PR1获得了20个较优解;相较于MDBA,GA-PR1获得了18个较优解。在最小值方面,相较于IGA,GA-PR1获得了17个较优解;相较于MDBA,GA-PR1获得了15个较优解。结合表3和图6可得,对比IGA和MDBA,所提出的GA-PR1具有更好的综合性能。

4 结论

本文采用加入路径重连的遗传算法来解决异步并行拆卸序列规划问题,并通过Matlab编程进行实现。所提出的方法具有以下特点。1)在模型中采用“或优先”关系来判断拆卸方向,对问题进行简化。2)采用路径重连策略,并通过实验证明其有效性。

虽然本文提出的方法已经通过案例进行验证,但在实际应用中依然存在些许不足。在实际拆卸中,往往不仅关注拆卸完成时间,更要对拆卸成本,资源消耗,拆卸利润等其他因素进行多方面权衡,寻找最佳方案,但本文所述方法仅考虑拆卸完成时间,略有不足。在以后的研究中,考虑更贴合

实际的具体需求, 提出更好的方法解决异步并行拆卸序列规划问题。

参考文献:

- [1] ZHANG X F, ZHANG S Y. Product cooperative disassembly sequence planning based on branch-and-bound algorithm[J]. *International Journal of Advanced Manufacturing Technology*, 2010, 51(9): 1139-1147.
- [2] ZHANG X F, YU G, HU Z Y, et al. Parallel disassembly sequence planning for complex products based on fuzzy-rough sets[J]. *International Journal of Advanced Manufacturing Technology*, 2014, 72(1-4): 231-239.
- [3] SMITH S, HUNG P Y. A novel selective parallel disassembly planning method for green design[J]. *Journal of Engineering Design*, 2015, 26(10-12): 283-301.
- [4] REN Y P, TIAN G D, ZHAO F, et al. Selective cooperative disassembly planning based on multi-objective discrete artificial bee colony algorithm[J]. *Engineering Applications of Artificial Intelligence*, 2017, 64: 415-431.
- [5] 邓明星, 陈方颖, 唐秋华, 等. 考虑多目标件的异步并行选择性拆卸序列[J]. *计算机集成制造系统*, 2020, 26(7): 1749-1755. DENG Mingxing, CHEN Fangying, TANG Qiuhua, et al. Asynchronous parallel selective disassembly sequence considering multi-targets[J]. *Computer Integrated Manufacturing Systems*, 2020, 26(7): 1749-1755.
- [6] REN Y P, ZHANG C Y, ZHAO F, et al. An asynchronous parallel disassembly planning based on genetic algorithm[J]. *European Journal of Operational Research*, 2018, 269(2): 647-660.
- [7] ALTEKIN F T, KANDILLER L, OZDEMIREL N E. Profit-oriented disassembly-line balancing[J]. *International Journal of Production Research*, 2008, 46(10): 2675-2693.
- [8] KONGAR E, GUPTA S M. Genetic algorithm for disassembly process planning[C/OL]//*Proceedings Volume 4569, Environmentally Conscious Manufacturing II*, 2002 (2002-02-11). <https://doi.org/10.1117/12.455264>.
- [9] KHEDER M, TRIGUI M, AIFAOU N. Disassembly sequence planning based on a genetic algorithm[J]. *Journal of Mechanical Engineering Science*, 2015, 229(12): 2281-2290.
- [10] MARZO R G, RIBEIRO C C. A GRASP with path - relinking and restarts heuristic for the prize-collecting generalized minimum spanning tree problem[J]. *International Transactions in Operational Research*, 2020, 27(3): 1419-1446.
- [11] DING J W, LYU Z P, LI C M, et al. A two-individual based evolutionary algorithm for the flexible job shop scheduling problem[C/OL]//*Proceedings of the AAAI Conference on Artificial Intelligence*, 2019(2019-07-17). <https://doi.org/10.1609/aaai.v33i01.33012262>.
- [12] 任亚平. 废旧产品拆解序列规划问题建模与优化研究[D]. 武汉: 华中科技大学, 2019.
REN Yaping. Research on the modeling and optimization of disassembly sequence planning problem for used products [D]. Wuhan: Huazhong University of Science & Technology, 2019.
- [13] GO T F, WAHAB D A, RAHMAN M N A, et al. Genetically optimised disassembly sequence for automotive component reuse[J]. *Expert Systems with Applications*, 2012, 39(5): 5409-5417.
- [14] WANG K P, LI X Y, GAO L. A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem[J]. *Computers & Industrial Engineering*, 2019, 130: 634-649.
- [15] XU W J, TANG Q, LIU J Y, et al. Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing[J]. *Robotics and Computer-Integrated Manufacturing*, 2020, 62: 101860.

(责任编辑: 张广珍)