

# 离散鲸鱼算法求解拆解序列规划问题<sup>①</sup>



顾嘉豪

(三峡大学 电气与新能源学院, 宜昌 443002)

通信作者: 顾嘉豪, E-mail: g1564780326@126.com

**摘要:** 工业产品的回收再制造有利于降低生产成本和保护环境, 而制定优秀的产品拆解序列规划提高拆解效率、降低回收成本为其关键. 针对回收设备在实际拆解中的因素, 建立一种基于离散鲸鱼算法 (DWOA) 的拆解序列规划模型. 该模型目标函数以位置改变为代价作为新的评价指标, 利用分层组合的方法快速生成初始群体. 离散鲸鱼算法具有优先保护约束交叉机制、启发式变异、优秀的全局和局部搜索能力. 以回收上橡皮板和空气围带进行对比实验, 结果表明在相同时间下, 离散鲸鱼算法的算法稳定性、寻优能力、收敛速度都要优于其他算法.

**关键词:** 拆解序列规划; 离散鲸鱼算法; 遗传算法; 启发式变异; 产品回收

引用格式: 顾嘉豪. 离散鲸鱼算法求解拆解序列规划问题. 计算机系统应用. <http://www.c-s-a.org.cn/1003-3254/8863.html>

## Disassembly Sequence Planning Using Discrete Whale Optimization Algorithm

GU Jia-Hao

(College of Electrical Engineering & New Energy, China Three Gorges University, Yichang 443002, China)

**Abstract:** Recycling and remanufacturing industrial products is conducive to reducing production costs and protecting the environment. It is very important to make excellent equipment disassembly sequence planning to improve disassembly efficiency and reduce recovery costs. For the factors of equipment recycling in actual disassembly, a disassembly sequence planning model based on a discrete whale optimization algorithm (DWOA) is proposed in this study. The objective function of the model employs the position change cost as the new evaluation indicator and adopts the stratified combination method to rapidly generate the initial population. DWOA features the precedence preservative crossover mechanism, heuristic mutation, and better global and local search ability. Comparative experiments are conducted with recycled upper rubber plate and standstill seal to test the feasibility of the proposed algorithm. The experimental results show that at the same time, the algorithm stability, optimization ability, and convergence speed of DWOA are better than those of other algorithms.

**Key words:** disassembly sequence planning; discrete whale optimization algorithm (DWOA); genetic algorithm; heuristic mutation; product recycling

随着工业的快速发展, 大量的工业制造废弃物不仅污染环境, 占用土地面积, 而且浪费资源. 因此, 为解决上述问题需要对工业废旧产品进行高效回收、再制造以及再利用, 这对环境保护以及资源的有效利用具有重要实际意义<sup>[1]</sup>. 而拆解是实现废旧产品回收再利用

的前提, 拆解序列规划的意义在于找到设备最佳的拆卸顺序, 以提高效率, 降低拆解成本<sup>[2]</sup>. 本研究结合智能算法, 设计了一个高效拆解序列规划模型, 为产品回收作业提供指导.

设备的拆解是一个很复杂的过程, 在对设备拆解

① 收稿时间: 2022-04-23; 修改时间: 2022-05-22; 采用时间: 2022-05-28; csa 在线出版时间: 2022-08-12

过程进行建模时需要考虑不可忽视的实际因素,将实际的物理模型转化为数学模型,并且还要建立目标函数以及借助智能算法才能做出可行且较优的决策<sup>[3]</sup>.在过去的研究中,设备组件的表达通常会用到图论中的有向图模型、无向图模型、与或图模型和 Petri 网模型<sup>[4]</sup>.

图像模型无法被计算机识别并加以计算,因此需要将建立好的图像模型用数学的形式表达出来.在拆解序列规划问题中,常用数学矩阵来进一步描述设备零部件之间的关系,例如干扰矩阵、接触矩阵和连接矩阵等<sup>[5]</sup>.连接矩阵表示零部件之间的连接类型,干扰矩阵和接触矩阵分别表示零部件之间的空间干扰和接触关系,常用来从设备中获取拆解信息.

在对设备进行数字化表示后,需要建立目标函数并结合算法来求解最佳拆解序列.大量研究表明,拆解序列规划问题属于复杂的完全非确定性多项式问题<sup>[6]</sup>,像这类问题可以使用穷举法对结果进行逐个检验,可以得出最终答案.但是,穷举法的计算时间随问题的复杂程度会呈指数增长因而变得不可适用.随后,基于种群的智能算法被广泛地应用于废弃 (EOL) 产品<sup>[7]</sup>或设备维修的拆解序列规划中.

杨书仪等<sup>[8]</sup>将遗传算法与模拟退火算法进行结合,平衡了算法的全局搜索和局部搜索能力,为混凝土泵车分动箱的拆解提供了一个更加高效的途径.张雷等<sup>[9]</sup>提出了一种并行拆解的规划模型,其并行拆解相对于串行拆解更加适应工厂的生产模式,最后利用传统的人工蜂群算法来制定优秀的拆解序列.郭洪飞等<sup>[10]</sup>考虑了多种拆解因素,并利用综合评价方法对拆解序列进行了更加科学的评价.

Kongar 等<sup>[11]</sup>提出将群智能优化算法遗传算法应用到产品的拆解回收利用中,为后续的拆解序列规划研究提供了新思路.Xu 等<sup>[12]</sup>使用了改进的人工蜂群算法求解了人机协作拆解的工业产品再制造问题,最大限度地减少了拆解成本和难度.Tseng 等<sup>[13]</sup>通过扁虫的再生特性提出了一种扁虫算法,基于再生机制的扁虫算法在进行算法求解时不容易陷入局部最优.Zhu 等<sup>[14]</sup>提出了一种基于零部件约束矩阵的方法来构建产品拆解的信息模型,并使用了混合蛙跳算法来求解拆解序列规划问题.

以上算法都属于经典的元启发式算法,这些算法通过模拟生物行为或者物理现象来解决问题,凭借其

概念简单且易于实现的优点,元启发式算法在工程应用领域越来越受欢迎.除以上算法之外,近些年才提出的鲸鱼算法<sup>[15]</sup>具有收敛速度快、全局搜索能力强等优点.在以往的研究中,设备的拆解序列规划问题不同于一般的连续空间优化问题<sup>[16]</sup>,而是离散的空间优化问题.鲸鱼算法最早提出来用于解决连续的优化问题,因此如何改进鲸鱼算法解决离散优化问题成为了本研究的关键.因此,本研究提出了一种新的离散鲸鱼算法并结合改进的遗传算法变异机制以平衡全局和局部搜索能力并将其运用到了设备拆解序列规划问题中,以求解出最优的拆解零件顺序,减少拆解成本和提高拆解效率.

## 1 拆解序列规划模型

拆解序列规划的目标是找到拆解代价或拆解时间最小的拆解序列,本节将给出约束关系模型、目标函数和评价指标为拆解序列规划问题建立数学模型.

### 1.1 约束关系

图 1 为某一简单设备零部件的有向图,图中标号代表各个零部件,图中箭头表示拆解流向.箭头前的零件必须比箭头后的零件要先拆,例如节点 3 必须要优先于节点 5 拆解.

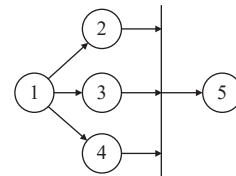


图 1 简单拆解有向图模型

两个拆解节点之间的约束关系转化为数学模型可以由约束矩阵  $C = [c_{ij}]_{N \times N}$  表示,其中  $i, j$  为拆解节点,既设备零部件,且  $i, j$  都不超过设备零件总数,  $c_{ij}$  的取值由式 (1) 表示:

$$c_{ij} = \begin{cases} 1, & \text{如果节点 } i \text{ 必须在节点 } j \text{ 之前拆除} \\ 0, & \text{其他} \end{cases} \quad (1)$$

### 1.2 目标函数

给出目标函数之前须首先定义一维拆解序列数组  $ds$ ,它表示完成设备拆除项目所需要的零件拆解顺序.在建立目标函数时,拆卸工作负荷的相同部分已被移除,例如准备必要的材料,拆卸操作,物理消耗等.由于设备的回收属于完全拆解,当所有零件都要被拆除时,

单个零件的拆解时间最后相加并被抵消. 因此, 本研究采用惩罚代价的形式对拆解过程中影响最终拆解效率的部分进行考虑. 目标函数由式 (2) 给出:

$$\min F = \sum_{k=0}^{n-2} (T_{ds(k),ds(k+1)} + P_{ds(k),ds(k+1)} + D_{ds(k),ds(k+1)}) \quad (2)$$

其中,  $k$  为拆解序列中的零件位置,  $ds(k)$  表示第  $k$  个零件的编号,  $T_{ds(k),ds(k+1)}$  表示第  $k$  个零件与第  $k+1$  个零件之间工具改变的代价, 同理将工具改变代价、位置改变代价  $P_{ds(k),ds(k+1)}$  和方向改变代价  $D_{ds(k),ds(k+1)}$  相加得到最终的目标评价函数. 最终得出的函数值越小, 既拆解代价越小, 所对应的拆解序列就越好.

### 1.3 评价指标

#### 1.3.1 工具改变

设备零部件的拆解离不开工具. 由于需要拆解的零件类型很多, 拆解工作人员难以避免地需要频繁更换手中的工具来拆卸设备上的零件, 这将十分耗时耗力. 因此将工具的改变作为拆解序列的评价指标很有必要. 一般来讲, 设备上的某一个零件对应着一种拆解工具, 拆解不同零件时, 工具会随之更换. 在此用矩阵  $T_{ij} = [t_{ij}]_{N \times N}$  表示拆解零件节点  $i$  和拆解零件节点  $j$  之间的工具改变代价, 由式 (3) 表示为:

$$t_{ij} = \begin{cases} 1, & \text{两节点拆解所需工具不同、且节点 } j \text{ 拆解工具为小型工具} \\ 2, & \text{两节点拆解工具不同、且节点 } j \text{ 拆解工具为大型工具} \\ 0, & \text{两节点拆解工具相同} \end{cases} \quad (3)$$

其中, 节点  $i$  是在节点  $j$  前拆解的, 所以当下一步拆解的零件  $j$  需要大型工具时比较耗时. 比如吊机, 它的启动和调用都很费时. 而小型工具如梅花扳手, 六角扳手, 由于这些工具易于获取, 工具改变代价自然会小一些. 而当前后两个拆解零件节点一样时, 就没必要更换工具了, 代价自然为 0.

#### 1.3.2 位置改变

在拆解大型设备时, 零件之间的距离较远, 工作人员来回地走动也是一种低效率的表现, 因此需要将零件之间的距离作为拆解序列的评价指标. 然而, 在施工现场很难精确测量设备各个零件的空间位置距离, 此时可以借助三维软件 Unity 3D. 成功导入到 Unity 3D 的设备模型会自动添加一个 Transform 组件, 在该组件中有一个三元变量 position, 它表征模型零件在三维空

间的位置, 例如零件  $i$  的 position 为  $(x_i, y_i, z_i)$ , 零件  $j$  的 position 为  $(x_j, y_j, z_j)$ , 则可用矩阵  $P_{ij} = [p_{ij}]_{N \times N}$  来表示, 其公式如式 (4) 所示:

$$p_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (4)$$

#### 1.3.3 方向改变

除了零件的位置以外, 在三维直角坐标系中, 零件的拆解方向也起着很关键的作用, 其通常可以用  $\{+x, -x, +y, -y, +z, -z\}$  来定义, 用式 (5) 表示:

$$d_{ij} = \begin{cases} 0, & \text{节点 } i \text{ 与 } j \text{ 的拆解方向一致} \\ 1, & \text{拆解方向改变了 } 90^\circ \\ 2, & \text{拆解方向改变了 } 180^\circ \end{cases} \quad (5)$$

因此可以用矩阵  $D_{ij} = [d_{ij}]_{N \times N}$  表征设备零件拆解方向改变的代价.

## 2 离散鲸鱼算法

标准的鲸鱼算法在早期阶段进行全局搜索, 在后期进行局部搜索. 鲸鱼算法具有收敛速度快, 全局搜索能力强的优点, 然而传统的鲸鱼算法不能有效地跳出局部最优并且容易陷入其中. 与此同时, 遗传算法的变异机制可以打破当前的最优解, 跳出局部最优, 提高局部搜索能力. 受此启发, 提出的离散鲸鱼算法并结合可行解生成器、优先保护约束交叉机制、启发式变异以提高算法的搜索性能.

### 2.1 可行解快速生成方法

在生成初始种群时, 需要得到一组满足约束关系的拆解序列可行解. 然而随着设备节点数量的增多, 如果采用随机排列组合的方式, 会产生很多不满足约束关系的非可行解以及少许可行解. 此时采用排除非可行解并留下可行解的方式会很费时且效率不高. 为此本研究采用了如图 2 所示的分层组合的方式对图 1 中的简单设备进行完全搜索, 来完成可行解即初始群体个体的生成.

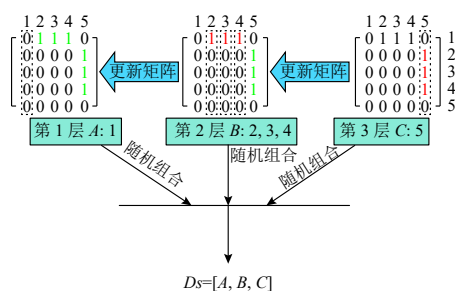


图 2 分层组合法

可行解快速生成的具体操作步骤如下。

1) 选定要拆解的节点为 5, 第 5 列检索到 1 的节点有 2, 3, 4, 意味着 2, 3, 4 节点必须优先于 5 拆解, 于是先把 5 放在最后一层 C 中。

2) 紧接着检索 2, 3, 4 列, 发现它们都指向 1, 于是将 2, 3, 4 放在 B 层。

3) 检索第 1 列, 发现没有约束, 将节点 1 放在 A 层停止检索。

4) 为了得到不同的初始种群个体序列, 将这 3 层内的节点随机排序, 最终将 A、B、C 三组数据依次放置到拆解序列数组  $ds$  中。对于复杂设备, 如此重复操作选择其他要拆解的节点, 就可以快速的得到满足约束关系的拆解序列初始种群。

### 2.2 优先保护约束交叉机制

在过去的研究中, 遗传算法里的交叉步骤指选取两个优秀的父本进行交叉运算, 有几率生成更加优秀的子代。然而遗传算法中的部分匹配交叉是从父代随机选择两个个体  $P1, P2$  和两个点并将两点之间部分提取出来, 分别放在子代相同的位置。这样子确实可以生成新子代但无法确定子代个体里面的所有节点是否还满足约束关系, 因此还要再进行筛选判断, 问题会变得很复杂。于是, 一种优先保护约束交叉机制被提出来解决这个问题, 如图 3 所示。

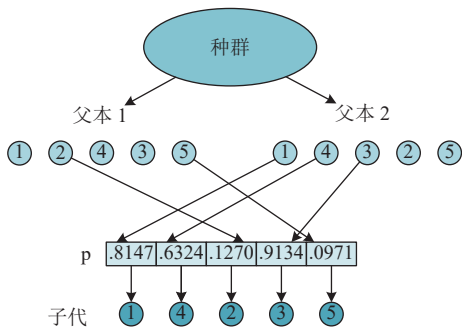


图 3 优先保护约束交叉

从生成的初始种群中抽出两个父本, 分别为父本 1=[1, 2, 4, 3, 5], 父本 2=[1, 4, 3, 2, 5]。它们当中的节点都满足节点间的约束关系, 给出 0 到 1 间的随机变量  $p$ , 如果  $p$  大于 0.5, 则从父本 2 选择节点; 如果  $p$  小于 0.5, 则从父本 1 选择节点。优先保护约束交叉机制子代生成的操作步骤如下。

1) 当  $p=0.8147 > 0.5$  和  $p=0.623 > 0.5$  时, 从父本 2 中选择最开始的节点 1、4 存入子代中;

2) 当  $p=0.1270 < 0.5$  时, 从父本 1 中开始选择, 删除

已经存在的 1、4 节点。父本 1 中只剩下 2, 3, 5 节点, 因此按顺序选择节点 2 并存入子代。

3)  $p=0.9134 > 0.5$  时, 删去父本 2 中的 1, 4, 2 后剩下 3, 5, 于是将 3 存入子代; 当  $p=0.0971 < 0.5$ , 选择父本 1 中的 5 存入子代, 最后得出满足约束关系的子代=[1, 4, 2, 3, 5]。

### 2.3 离散鲸鱼更新机制

在优先保护约束交叉机制中, 生成子代的两个以上才的父本后才有交叉的意义。而在离散鲸鱼算法中, 需要确定 3 个父代个体。它们分别是当前迭代中种群适应度值最好, 既拆解代价最小的拆解序列个体  $X^*$ , 随机选择的个体  $X_{rand}$ , 将种群按照拆解代价由小到大排列的前  $B$  个元胞数组  $G$ 。最终给出如式 (6) 所示的离散鲸鱼算法:

$$X(t+1) = \begin{cases} f[X^*(t)], & \text{if } |A| < 1, p < 0.5 \\ f[X_{rand}(t)], & \text{if } |A| \geq 1, p < 0.5 \\ f[G(t)], & \text{if } p \geq 0.5 \end{cases} \quad (6)$$

其中,  $X(t+1)$  指的就是当前迭代的下一代, 即子代个体,  $f(\cdot)$  是指采用优先保护约束交叉机制从父本中选择节点并生成子代的拆解序列。其中元胞数组  $G$  的长度是由  $B$  决定的,  $B$  由式 (7) 表示:

$$B = \left\lceil \left[ \left( \frac{N_p}{2} - 1 \right) \times \left( 1 - \frac{t}{t_{max}} \right) + 1 \right] \right\rceil \quad (7)$$

其中,  $N_p$  是种群大小,  $t_{max}$  是迭代的最大值,  $\lceil \cdot \rceil$  是小数全部舍去的取整。  $B$  是随着迭代次数的增加不断减小到 1 的一个变量, 用来模拟鲸鱼前期搜索, 后期捕食的特性。

### 2.4 启发式变异

通过交叉可以产生新的优秀个体, 但是交叉随机性太强, 可能迭代很久才会出现一个更加优秀的个体。因此, 为了增强算法局部搜索能力, 通常采用遗传算法中的变异机制, 但是传统的遗传变异会改变拆解序列中节点的值, 这将使得拆解序列不符合约束关系而被舍去。

因此针对设备拆解的拆解序列优化的变异不应产生新节点, 只能是对拆解节点的位置变异。一般来讲, 改变位置的方法可以是随机选择两个节点并判断交换后是否满足约束条件再进行交换, 也可以是选出一个节点再在满足约束关系的情况下插入序列中的其他位

置. 以上两种方法都可以在不破化拆解序列可行性的情况下产生新的变异可行解, 但由于这两种方法随机性太强, 产生的新序列其适应度可能更好也可能更差, 因此提出一个启发式变异方法, 寻找最优的插入位置可以加强算法的局部收敛能力.

如图 4 所示, 以简单设备为例. 如果选择节点 2 作为变异节点, 则检索 2 的行和列, 并且将约束节点 2 的节点 1 和 5 找到分别作为要插入的起点与终点. 然后将节点 2 插入如图 4 所示的箭头指示的位置, 并计算每个位置更改的拆卸代价. 如果拆卸代价较短, 则保存节点 2 的新位置以获得优异的后代. 启发式变异旨在增强算法局部搜索能力, 对于不同的拆解序列, 优先保护约束交叉机制子代生成的步骤如下.

- 1) 设定变异次数  $Nm$  为种群数量的一半.
- 2) 从相对工作量较小的拆解序列集合中随机选择一个拆解序列  $seq$  作为变异个体.
- 3) 从选择的变异个体  $seq$  中随机选择第  $Mp$  个节点作为变异节点, 然后查找该节点可以重新插入的位置范围, 为了减少判断次数, 首先, 检查约束矩阵中第  $seq(Mp)$  列, 记录数据等于 1 所对应列的节点编号作为起始节点集合  $StartNode$ , 检查约束矩阵第  $seq(Mp)$  行, 记录数据等于 1 所对应行的节点编号作为终止节点集合  $EndNode$ ; 然后检查拆解序列  $seq$ , 则可插入的位置即在  $StartNode$  集合中节点后和  $EndNode$  集合中节点前都为可插入位置.
- 4) 计算所有可插入位置对应的工作量的值, 并保留工作量最小的个体用于更新种群.

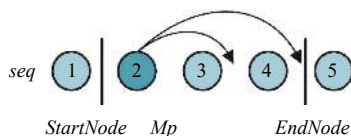


图 4 启发式变异

### 2.5 离散鲸鱼算法模型

综上所述, 结合启发式变异的离散鲸鱼算法整体的流程如图 5 所示.

## 3 实验分析

### 3.1 实验数据集

本实验使用文献 [17] 中的水轮机主轴密封的零部件信息、约束矩阵以及各设备的拆解代价数据. 针对

回收上橡皮板和空气围带两个项目设计实验. 紧接着就是利用这个两个项目来测试各个算法的性能, 在测试实验中, 每个算法都执行 100 次, 实验结果中记录种群大小  $Np$ , 迭代次数  $t$ , 规划得到的拆解序列结果中代价最小值 MIN, 标准差 STD, 平均运行时间  $T$ , 以及代价最小拆解序列所占比例 ROM. 运行环境为: 测试软件: Matlab 2016; 编码语言: C 语言; 处理器: 英特尔酷睿 i5, 6300hq×4, 主频 2.3 GHz; 内存: 16 GB.

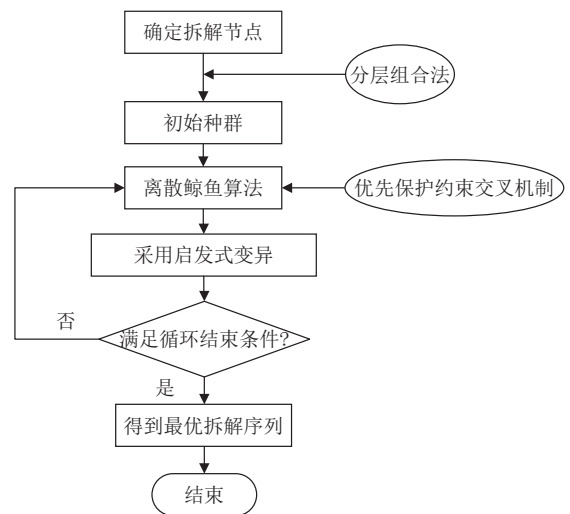


图 5 离散鲸鱼算法模型

### 3.2 实验结果

实验采用 Konger 等<sup>[11]</sup> 提出的遗传算法 GA, Yeh<sup>[18]</sup> 提出的 SSO 算法和 Xia 等<sup>[19]</sup> 提出的 STLBO 算法与 DWOA 算法进行比对, 根据结果观察 DWOA 的表现. 其实验对比结果如表 1 所示.

表 1 实验结果

Case	$Np$	$t$	Algorithm	MIN	STD	ROM (%)	T(s)
1	40	100	GA	57.0	3.325	—	<b>0.6992</b>
	40	100	STLBO	47.1	2.488	—	5.1911
	40	100	SSO	55.8	2.119	—	0.7995
	40	100	DWOA	<b>42.1</b>	<b>0.431</b>	<b>47</b>	7.9786
2	60	100	GA	84.1	3.925	—	16.3265
	60	100	STLBO	75.2	3.206	—	1.5520
	60	100	SSO	88.1	2.013	—	<b>1.5148</b>
	60	100	DWOA	<b>66.5</b>	<b>0.501</b>	<b>38</b>	10.3265

表 1 中“—”代表算法没有找到拆解代价最小的拆解序列. 根据实验可知, 将每个算法运行 100 次后, 在项目 1 与项目 2 中, DWOA 相对于其他算法搜索到了最小拆解代价 42.1 和 66.5, 标准差 STD 分别为 0.431 和 0.501, 说明鲸鱼算法的稳定性要比 GA, SSO, STLBO 要强. 但是项目 1 中 DWOA 算法的运行时间 7.9786 s

要大于其他算法,说明算法性能得到提升的同时也增加了计算量。

由4种算法获得的最小拆卸代价图像迭代数据如图6所示,项目1中,DWOA经历了12次迭代后就取到了最小拆卸代价42.1,这表明DWOA可以比其他算法更早地找到最优值.两个项目中只有DWOA搜索到了最小的拆卸代价,而其他算法则没有,证明所提DWOA具有更强的稳定性和更快的收敛速度.项目1最终得到最小拆卸代价对应的拆卸序列为:密封水箱上盖板螺栓→密封水箱上盖板→上压板螺栓→上环板密封环连接螺栓→上环板→上压板→上密封橡皮板.项目2最终得到最小拆卸代价对应的拆卸序列为:上密封橡皮板→下压板螺栓→下压板→下密封橡皮板→底座瓣螺栓→水箱底座→空气围带。

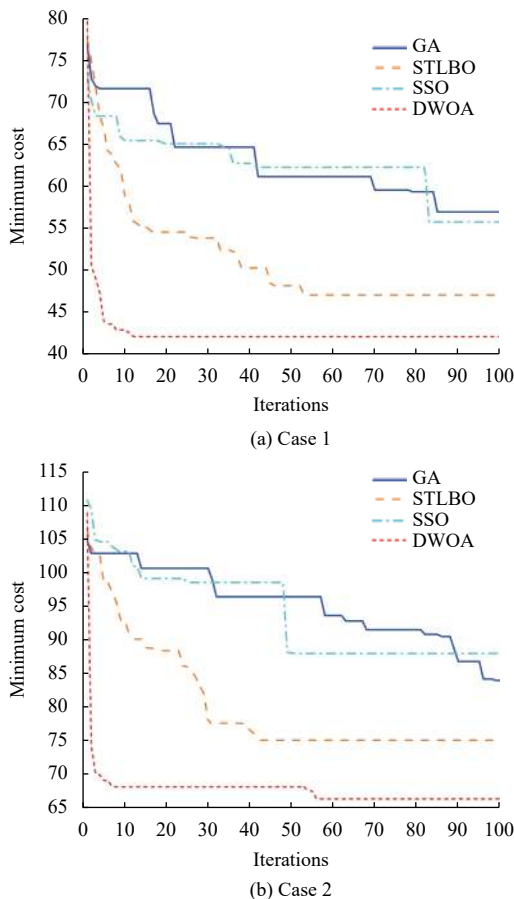


图6 不同算法的寻优曲线对比

#### 4 结论与展望

本文提出了一种新的离散鲸鱼算法(DWOA)来解决设备回收拆卸序列规划问题.该算法的改进之处在

于利用了分层组合的方法快速生成种群初始群体以及结合了启发式变异以及优先保护约束交叉机制.最后以上橡皮板和空气围带为对象进行对比实验,验证了它的稳定性、寻优能力、收敛速度要比其他算法强.接下来的研究工作是借助Unity3D软件并结合算法制作一款软件以增强拆解过程可视化表达也是一个有价值的研究方向。

#### 参考文献

- 任亚平. 废旧产品拆解序列规划问题建模与优化研究 [博士学位论文]. 武汉: 华中科技大学, 2019.
- Guo XW, Zhou MC, Abusorrah A, *et al.* Disassembly sequence planning: A survey. *IEEE/CAA Journal of Automatica Sinica*, 2021, 8(7): 1308–1324. [doi: 10.1109/JAS.2020.1003515]
- Zhou ZD, Liu JY, Pham DT, *et al.* Disassembly sequence planning: Recent developments and future trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2019, 233(5): 1450–1471. [doi: 10.1177/0954405418789975]
- Ong SK, Chang MML, Nee AYC. Product disassembly sequence planning: State-of-the-art, challenges, opportunities and future directions. *International Journal of Production Research*, 2021, 59(11): 3493–3508. [doi: 10.1080/00207543.2020.1868598]
- Zhang WL, Ma MX, Li HY, *et al.* Generating interference matrices for automatic assembly sequence planning. *The International Journal of Advanced Manufacturing Technology*, 2017, 90(1–4): 1187–1201. [doi: 10.1007/s00170-016-9410-x]
- 李佰霖. 面向水电站设备检修的虚拟仿真及自动规划方法研究 [博士学位论文]. 武汉: 华中科技大学, 2020.
- Favi C, Marconi M, Germani M, *et al.* A design for disassembly tool oriented to mechatronic product demanufacturing and recycling. *Advanced Engineering Informatics*, 2019, 39: 62–79. [doi: 10.1016/j.aei.2018.11.008]
- 杨书仪, 李京京, 文泽军, 等. 混凝土泵车分动箱拆解序列规划. *机械设计与研究*, 2014, 30(5): 106–109, 124. [doi: 10.13952/j.cnki.jofmdr.2014.0151]
- 张雷, 彭宏伟, 卞本阳, 等. 复杂产品并行拆解建模及规划方法研究. *中国机械工程*, 2014, 25(7): 937–943. [doi: 10.3969/j.issn.1004-132X.2014.07.016]
- 郭洪飞, 陈志彬, 任亚平, 等. 基于零件回收综合评价的废旧产品拆解序列与拆解深度集成决策研究. *机械工程学报*, 2022, 58(4): 258–268.

- 11 Kongar E, Gupta SM. Disassembly sequencing using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 2006, 30(5): 497–506.
- 12 Xu WJ, Tang Q, Liu JY, *et al.* Disassembly sequence planning using discrete Bees algorithm for human-robot collaboration in remanufacturing. *Robotics and Computer-Integrated Manufacturing*, 2020, 62: 101860. [doi: [10.1016/j.rcim.2019.101860](https://doi.org/10.1016/j.rcim.2019.101860)]
- 13 Tseng HE, Huang YM, Chang CC, *et al.* Disassembly sequence planning using a Flatworm algorithm. *Journal of Manufacturing Systems*, 2020, 57: 416–428. [doi: [10.1016/j.jmsy.2020.10.014](https://doi.org/10.1016/j.jmsy.2020.10.014)]
- 14 Zhu JF, Xu ZG, Su KY, *et al.* Asynchronous parallel disassembly sequence planning for multi-manipulator based on improved shuffled frog leaping algorithm. *SN Applied Sciences*, 2020, 2(5): 906. [doi: [10.1007/s42452-020-2680-9](https://doi.org/10.1007/s42452-020-2680-9)]
- 15 Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, 95: 51–67. [doi: [10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008)]
- 16 Rana N, Latiff MSA, Abdulhamid SM, *et al.* Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments. *Neural Computing and Applications*, 2020, 32(20): 16245–16277. [doi: [10.1007/s00521-020-04849-z](https://doi.org/10.1007/s00521-020-04849-z)]
- 17 Li BL, Li CS, Cui XL, *et al.* A disassembly sequence planning method with team-based genetic algorithm for equipment maintenance in hydropower station. *IEEE Access*, 2020, 8: 47538–47555. [doi: [10.1109/ACCESS.2020.2979247](https://doi.org/10.1109/ACCESS.2020.2979247)]
- 18 Yeh WC. Simplified swarm optimization in disassembly sequencing problems with learning effects. *Computers & Operations Research*, 2012, 39(9): 2168–2177.
- 19 Xia K, Gao L, Li WD, *et al.* Disassembly sequence planning using a simplified teaching-learning-based optimization algorithm. *Advanced Engineering Informatics*, 2014, 28(4): 518–527. [doi: [10.1016/j.aei.2014.07.006](https://doi.org/10.1016/j.aei.2014.07.006)]

(校对责编: 牛欣悦)