

一种混合人工蜂群算法求解顺序相依拆卸线平衡问题

刘佳¹, 王书伟^{2†}

(1. 电子科技大学 经济与管理学院, 成都 611731; 2. 西南交通大学 经济管理学院, 成都 610031)

摘要: 拆卸线平衡问题直接影响回收再制造成本. 为此, 构建了最小工作站开启数量、最短总拆卸时间、均衡工作站空闲时间、尽早拆卸有危害和高需求零部件的多目标顺序相依拆卸线平衡问题优化模型, 提出一种混合人工蜂群算法. 所提出算法在观察蜂跟随阶段采用分阶段选择评价法, 以便更好地区分蜜源; 在侦查蜂开采阶段构建基于全局学习的搜索机制, 以提高开采能力. 蜜蜂寻优过程中设计了简化变邻域搜索策略, 提高了寻优效率. 对比实验结果验证了模型的有效性和算法的优越性.

关键词: 顺序相依拆卸线平衡问题; 多目标优化; 混合人工蜂群算法; 简化变邻域搜索

中图分类号: TH165; TP18 **文献标志码:** A

A hybrid artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem

LIU Jia¹, WANG Shu-wei^{2†}

(1. School of Management & Economics, University of Electronic Science and Technology of China, Chengdu 611731, China; 2. School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: Disassembly line balancing problems directly affect the reverse/remanufacturing cost. A multi-objective sequence-dependent disassembly line balancing problem(SDDLBP) optimization model is established for minimizing the number of opened workstations, minimizing the total disassembly time, distributing the idle times across the opened workstations evenly, and processing the hazardous components and high-demand parts early. Then a hybrid artificial bee colony(HABC) algorithm is proposed to solve this problem. In the onlooker bee phase, a multi-stage evaluation method is presented to distinguish food sources better. In the scout bee phase, a search strategy based on the global learning mechanism is constructed to improve the exploitation ability of the scouts. When the bees search for new food sources, the reduced variable neighborhood search(RVNS) strategy is used to promote the search efficiency. Finally, computational results demonstrate the effectiveness of the proposed model and algorithm.

Keywords: sequence-dependent disassembly line balancing problem; multi-objective optimization; hybrid artificial bee colony algorithm; reduced variable neighborhood search

0 引言

科技与经济的快速发展使电子产品消费与废弃量与日俱增, 回收再利用废旧电子产品中有价值和有害的零部件, 可有效节约资源的同时降低环境污染, 促进经济与环境的协调发展. 拆卸作为回收再制造过程最关键环节之一, 其效率直接影响再制造成本^[1], 因此拆卸线平衡问题(DLBP)逐渐受到人们的重视.

Gupta等^[2]提出了DLBP, 并建立了简单DLBP数学模型. 随后, 研究者主要采用精确数学方法和元启

发式算法对DLBP进行优化. 文献[3-6]使用基于不同策略的传统数学规划方法求解DLBP, 优点是求解精度高, 但DLBP是NP完全问题^[7], 精确数学方法不适用于求解大规模DLBP. 因此, McGovern等^[8]应用遗传算法优化DLBP, 目标为最小化工作站数量和工作站负载均衡. Prakash等^[9]提出了基于约束的模拟退火算法, 确定拆卸顺序以最小化产品库存水平. 基于Pareto理论, 研究者^[10-11]分别通过细菌觅食算法和蚁群算法求解多目标DLBP. 李明等^[12]建立以精益生产为准则的多目标U型DLBP模型, 并通过人工蜂群

收稿日期: 2017-01-02; 修回日期: 2017-05-05.

责任编辑: 孙秋野.

作者简介: 刘佳(1985—), 女, 讲师, 博士生, 从事多目标优化、算法分析的研究; 王书伟(1985—), 男, 讲师, 博士, 从事供应链管理、智能算法的研究.

†通讯作者. E-mail: wangshuwei@my.swjtu.edu.cn

算法求解. 考虑到拆卸时间不确定, Kalayci等^[13]引入三角模糊隶属度函数, 并提出了一种多目标模糊人工蜂群算法.

上述对DLBP的研究仅考虑任务间的先后关系约束, 实际上, 无先后关系约束的两个任务之间可能存在相互干扰, 会因拆卸顺序的不同导致任务拆卸时间不确定, 最终影响拆卸线平衡. 该类问题被定义为顺序相依问题, 最早由Scholl等^[14]在装配线平衡问题(ALBP)中提出, 并构建了顺序相依装配线平衡问题模型(SDALBP). 贪婪随机自适应搜索法^[15]和混合遗传算法^[16]被用于求解单目标SDALBP. 综合考虑最小工作站数量、最小总装配费用和平滑指数, Hamta等^[17]提出将粒子群与变邻域搜索算法相结合求解多目标SDALBP. 针对平行装配线上的多产品混流装配, Akpinar等^[18]建立了平行混合型SDALBP数学模型, 并提出混合蚁群-遗传算法. 与装配线情况相似, 产品拆卸过程中同样存在顺序相依问题, Kalayci等^[19]将该类问题引入DLBP中, 并构建了以最小化工作站数量, 均衡工作站空闲时间, 尽早拆卸有危害和高需求零部件为多目标的顺序相依DLBP数学模型. 随后, Kalayci等^[20-23]分别使用混合遗传算法、禁忌算法、变邻域算法和人工蜂群算法求解该模型. 目前, 针对顺序相依DLBP的研究较少, 所构建的求解问题数学模型均未考虑产品任务总拆卸时间, 容易造成在拆卸过程中为了均衡工作站空闲时间而选择受干扰时间长的任务优先拆卸, 导致产品总拆卸时间增加, 操作工人和机器设备工作负荷增大. 此外, Kalayci等认为, 在解决顺序相依拆卸线平衡问题模型(SDDLBP)时, 目标函数均衡工作站空闲时间可兼顾最小化工作站数量, 此观点并非具有普适性, 具体证明见1.3节. 鉴于此, 本文构建以工作站数量最少、总拆卸任务时间最短、均衡工作站空闲时间、尽早拆卸有危害和高需求零部件为多目标的SDDLBP, 并依据各目标函数在实际拆卸过程中的重要性, 提出一种基于字典排序的混合人工蜂群算法(HABC). HABC算法将人工蜂群算法(ABC)与简化变邻域搜索算法(RVNS)相融合, 集成了ABC算法良好的觅食分工行为和自组织能力以及RVNS算法局部搜索能力强的优点, 在求解SDDLBP中表现出了良好性能.

1 问题描述与数学模型

1.1 问题描述

SDDLBP考虑无先后关系约束任务间的相互干扰, 以图1包含6个零件的产品拆卸为例, 圆圈

代表任务编号, 实线箭头表示任务先后关系, 虚线箭头表示任务间相互干扰, 带下划线数字为拆卸时间, 斜体数字为干扰时间. 由图1可知, 拆卸过程中任务2和任务3无先后关系约束, 但二者存在相互干扰. 如果任务2先于任务3拆卸, 任务3将阻碍任务2以最便捷的方式操作, 导致任务2实际拆卸时间为6s, 增量为1s; 反之, 若任务3在任务2之前拆卸, 任务3实际拆卸时间增至14s. 本文研究的SDDLBP就是在满足产品拆卸先后关系和工作站节拍约束前提下, 考虑无先后关系约束任务之间的相互干扰, 实现多目标优化.

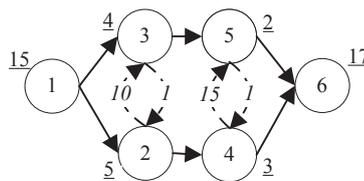


图1 产品任务先后关系

1.2 模型描述

由 N 个零件组成的待拆产品, 每个零件对应一项拆卸任务, $i, j = 1, 2, \dots, N$ 表示任务(零件)编号. 任务 i 的拆卸时间为 t_i , 若任务 i 先于任务 j 拆卸且 j 对 i 产生阻碍, 则任务 i 的实际拆卸时间为 $t'_i = t_i + \sum_{j=1}^N sd_{ji}$, 其中 sd_{ji} 为任务 j 对 i 的阻碍产生的拆卸时间增量. 工作站数量为 M , 节拍时间为 CT , 工作站 $k(k = 1, 2, \dots, M)$ 的作业时间为 ST_k , 空闲时间为 IT_k . $DPM = [P_{ij}]_{N \times N}$ 为拆卸任务先后关系矩阵, 若任务 i 为 j 的前驱任务, 则 $P_{ij} = 1$, 否则 $P_{ij} = 0$. PS_l 为拆卸任务序列第 l 个位置对应的零件编号.

1.3 模型构建

本文考虑无先后关系约束任务间的相互影响, 按照实际拆卸过程中各目标重要程度由高到低, 从最小化工作站数量、最短总拆卸任务时间、均衡工作站空闲时间、尽早拆卸有危害和高需求零部件5个方面建立多目标SDDLBP优化模型.

目标函数:

1) 最小化开启工作站数量 NWS , 降低人力、物力等资源投入成本, 即

$$\min f_1 = \sum_{k=1}^M z_k. \tag{1}$$

其中 z_k 表示工作站 k 是否开启, 若开启, 则 $z_k = 1$, 否则, $z_k = 0$.

2) 最短总拆卸任务时间 DT , 降低操作工人和机器设备作业负荷, 即

$$\min f_2 = \sum_{k=1}^M ST_k. \tag{2}$$

3) 均衡工作站空闲时间即平滑指数SI, 保证各工作站作业负荷相对均衡, 即

$$\min f_3 = \sum_{k=1}^M (z_k \times IT_k^2). \quad (3)$$

4) 尽早拆除有害零部件H, 以降低对人体健康和环境的危害, 即

$$\min f_4 = \sum_{l=1}^N (l \times h_{PS_l}). \quad (4)$$

其中 h_{PS_l} 表示编号为 PS_l 的零件的危害性, 若有危害, 则 $h_{PS_l} = 1$, 否则 $h_{PS_l} = 0$.

5) 尽早拆除高需求零部件D, 使经济效益最大化, 即

$$\min f_5 = \sum_{l=1}^N (l \times d_{PS_l}). \quad (5)$$

其中 d_{PS_l} 表示编号为 PS_l 的零件需求量.

Kalayci 等^[19-23] 指出, 在 SDDLBP 中, 平滑指数 f_3 可兼顾最小化工作站数量 f_1 , 该观点在求解 SDDLBP 时并非具有普适性. 证明如下: 以图1产品拆卸为例, 拆卸序列 $S_1\{1, 2, 3, 4, 5, 6\}$ 和 $S_2\{1, 3, 2, 5, 4, 6\}$ 符合产品任务先后关系, 为该产品两个可行拆卸序列. 拆卸过程因任务间相互干扰, 造成序列 S_1 的总拆卸时间增加量为 $SD_1 = sd_{32} + sd_{54} = 1 + 1 = 2s$, S_2 的总干扰时间为 $SD_2 = sd_{23} + sd_{45} = 25s$. 如图2所示, 在满足工作站节拍时间 $CT = 20s$ 条件下, 拆卸序列 S_1 和 S_2 开启工作站数量分别为 $NWS_1 = 3, NWS_2 = 4$, 平滑指数分别为 $SI_1 = 5^2 + 4^2 + 3^2 = 50, SI_2 = 5^2 + 1^2 + 0^2 + 3^2 = 35$. 虽然平滑指数 $SI_2 < SI_1$, 但由于序列 S_2 拆卸过程中任务间干扰时间长, 导致开启工作站数 $NWS_2 > NWS_1$. 因此, 在求解 SDDLBP 时, 受干扰时间影响, 平滑指数 f_3 不能兼顾最小化工作站数量 f_1 这一目标.

拆卸序列 S_1	1	2	3	4	5	6
拆卸时间	(15)	(5)	(4)	(3)	(2)	(17)
干扰时间		(+1)	(+1)			
空闲时间	(5)		(4)			(3)
拆卸序列 S_2	1	3	2	5	4	6
拆卸时间	(15)	(4)	(5)	(2)	(3)	(17)
干扰时间		(+10)		(+15)		
空闲时间	(5)		(1)		(0)	(3)
工作站	1	2	3	4		

图2 拆卸序列 S_1 和 S_2 任务分配

约束条件:

1) 每个任务仅被分配到一个工作站, 即

$$\sum_{k=1}^M x_{ik} = 1, \quad i = 1, 2, \dots, N. \quad (6)$$

其中 x_{ik} 表示任务 i 是否分配到工作站 k 上, 分配则

$x_{ik} = 1$, 否则 $x_{ik} = 0$.

2) 任务分配需满足拆卸先后关系, 即

$$x_{ik} \leq \sum_{j=1}^M x_{jk}, \quad \forall P_{ij} = 1, \quad i, j = 1, 2, \dots, N. \quad (7)$$

3) 工作站总拆卸时间不超过节拍时间, 即

$$ST_k \leq CT, \quad k = 1, 2, \dots, M. \quad (8)$$

2 混合人工蜂群算法

受自然界蜜蜂种群觅食行为和分工机制启发, Karaboga^[24] 提出了人工蜂群算法(ABC). 人工蜂群由雇佣蜂、观察蜂和侦查蜂3种角色蜜蜂构成, 其内部良好的分工协作机制, 使算法表现出较强的空间搜索能力, 但标准ABC算法中雇佣蜂和观察蜂在单一邻域内“探索”(exploration)蜜源, 侦查蜂在整个解空间内随机“开采”(exploitation)蜜源, 导致算法搜索效率较低. 为了解决上述问题, 本文提出一种混合人工蜂群算法求解SDDLBP, 该算法在蜜蜂搜索蜜源过程中设计了简化变邻域搜索策略, 以提高寻优效率.

2.1 编码与解码

SDDLBP 是一个离散的组合优化问题, 问题的解为产品的一个可行拆卸序列. 本文采用整数排列编码方式, 数字代表拆卸任务编号, 从左向右的顺序代表任务的拆卸顺序. 以图1产品拆卸为例, 整数排列 $\{1, 3, 2, 5, 4, 6\}$ 满足图1产品任务先后关系, 为该产品的一个可行拆卸序列, 其表示首先拆卸任务1, 然后按照排列顺序从左向右依次拆卸, 最后拆卸任务6.

解码过程即考虑无先后关系约束任务间的相互干扰, 满足工作站作业时间不超过节拍时间, 按照整数排列顺序依次将任务分配至工作站. 设 $CT = 20s$, 以可行解 $\{1, 3, 2, 5, 4, 6\}$ 的解码为例. 首先开启工作站1并分配任务1, 此时工作站1作业时间 $ST_1 = 15s$, 空闲时间 $IT_1 = 20 - 15 = 5s$, 随后分配任务3, 由于受任务2的干扰, 任务3的实际拆卸时间 $t'_3 = t_3 + sd_{23} = 4 + 10 = 14s > 5s$, 超出工作站1当前空闲时间, 因而需开启工作站2, 并将任务3分配至工作站2中, 以此类推直至所有任务分配完毕, 解码结果如图2中拆卸序列 S_2 所示.

2.2 多目标评价方法

SDDLBP 多目标之间存在相互冲突, 针对实际拆卸过程中各目标重要程度不同的特点, 按其重要性由高到低排序处理, 获得的(近似)最优解更符合企业需求^[19-23]. 因此, 本文采用字典排序法评价解的优劣, 即按照目标函数 $f_1 \sim f_5$ 的先后顺序, 将最小化工作站数量以减少人力、物力等成本的投入作为首要评价

目标;然后依次考虑工人、机器设备的作业负荷,平滑指数以及危害指标;最后考虑市场需求指标. 例如,若解 x_1 在 f_1 上优于解 x_2 , 则解 x_1 优于 x_2 , 若两个解 f_1 的值相同, 则比较 f_2 的值, 以此类推.

2.3 初始化阶段

初始化阶段包括参数初始化和蜜源初始化. HABC算法包含3个控制参数: 初始可行解数量(SN)角色控制转换阈值(uptlimit)和算法终止迭代时间(maxTime), 其中SN等于雇佣蜂和观察蜂的数量, uptlimit控制雇佣蜂向侦察蜂的转换. HABC算法中每个蜜源对应一个可行解, 即一个可行拆卸序列, 用 n 维向量 $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ 表示. 为了保证初始蜜源多样性, 扩大初始搜索空间, 本文采用面向工作站的随机生成法构造初始解, 使用整数排列编码表示. 考虑无先后关系约束任务间的相互干扰, 在满足产品拆卸先后关系和节拍约束条件下生成可分配任务集 A^* , 从中随机选取任务分配至工作站, 直至所有任务分配完毕. 具体步骤如下:

Step 1: 开启工作站 $k, k = 1$.

Step 2: 生成可选任务集 C^* . 如果任务 i 未分配, 且无前驱任务或其前驱任务已分配, 则该任务为可选任务.

Step 3: 生成可分配任务集 A^* . 从 C^* 中选取实际拆卸时间小于或等于当前工作站空闲时间的任务构成 A^* . 若 A^* 为空, 则需开启下一工作站 $k, k = k + 1$, 重复Step 3.

Step 4: 从 A^* 中随机选取一个任务分配至当前工作站.

Step 5: 重复Step 2~Step 4, 直至 C^* 为空.

2.4 雇佣蜂引领阶段

蜜源初始化后, 雇佣蜂对当前蜜源进行局部探索, 贪婪选择质量更高的蜜源, 记录其位置和适应值信息, 并以舞蹈的方式将蜜源信息传递给观察蜂, 引领观察蜂对蜜源进一步探索. 标准ABC算法中, 雇佣蜂对当前蜜源进行单一邻域局部搜索, 搜索空间小且效率较低. 本文在雇佣蜂引领阶段设计了简化变邻域搜索(RVNS)策略, 在扩大搜索范围的同时降低搜索时间, 以提高雇佣蜂的探索能力.

RVNS通过系统变换邻域结构扩大搜索空间, 并在局部搜索过程中采用“一步改进”法, 以提高搜索效率^[25]. 通过对P-中值问题求解, 对比结果表明RVNS与快速交换(FI)启发式方法求解效果相当, 但在平均求解时间方面, RVNS为FI的1/18, RVNS特别适用于局部耗时的大规模计算问题^[26]. 邻域结构

是RVNS算法的核心内容之一, 其在满足问题约束条件前提下, 以当前解为基础通过一定操作实现解序列的有效转换. 本文针对SDDLBP特点, 在雇佣蜂探索蜜源阶段, 设计了交换、插入、逆序和多次插入4种邻域结构, 如图3(a)~图3(d)所示. 雇佣蜂对当前蜜源进行邻域搜索时, 首先对解 x 进行扰动, 即在当前邻域范围内随机选择一点 x' , 然后采用“一步改进”法进行局部搜索, 即从 x' 的邻域集 $N_k(x')$ 中选择一个新解 x'' 进行比较. 若 x'' 优于 x , 则更新当前解 $x = x''$, 并返回第一个邻域结构重新搜索; 否则, 跳至下一邻域结构继续搜索. 具体实现过程如图4所示.

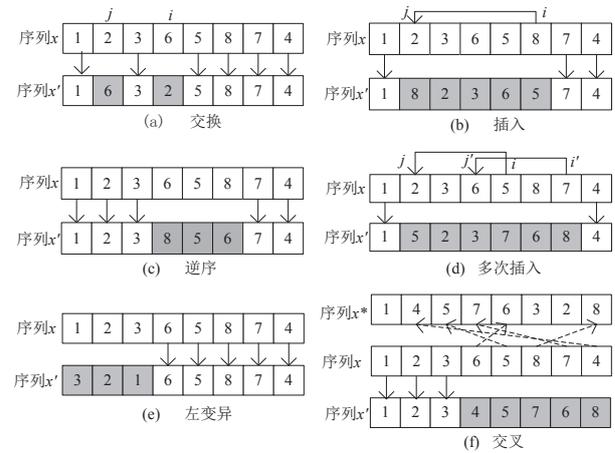


图3 邻域结构

2.5 观察蜂跟随阶段

观察蜂根据雇佣蜂传递的信息以轮盘赌的方式按照 $P_i = f_i / \sum_{i=1}^{SN} f_i$ 选择雇佣蜂跟随, 携带蜜源质量越高的雇佣蜂被选择跟随的可能性越大. 根据人工蜂群算法工作原理, 随着迭代次数的增加, 雇佣蜂逐渐向最优蜜源靠拢, 其所携带蜜源的质量差异也越来越小. 文献[23]采用平滑指数 f_3 作为观察蜂选择蜜源的适应值评价指标, 但至算法迭代中期, 大部分蜜源的平滑指数已趋于一致, 无法有效区分蜜源质量, 使算法寻优速度下降. 为了解决这一问题, 使观察蜂在整个迭代过程中能够根据最有效的适应值评价函数选择蜜源深度探索. 本文依据目标函数的重要性程度, 提出一种分阶段选择评价法, 具体过程为: 在迭代初期, 将目标函数 f_1 作为观察蜂选择雇佣蜂跟随的适应值评价函数; 迭代过程中判断种群中各蜜源的 f_1 值是否趋于一致, 若一致, 则表明目标函数 f_1 已经不能有效区分蜜源优劣, 此时选择目标函数 f_2 作为下一迭代阶段的适应值评价函数, 以此类推. 当观察蜂选定某雇佣蜂跟随后, 将在该雇佣蜂携带蜜源位置采用与其相同的RVNS搜索策略进一步深度探索蜜源, 从而加快算法寻优速度.

2.6 侦查蜂开采阶段

蜜源经过 $uplimit$ 次探索后仍未改进将被放弃,此时雇佣蜂转变为侦查蜂开采新蜜源. 在标准 ABC 算法中, 侦查蜂将在整个解空间范围内随机搜索, 盲目性较大. 此外, 被放弃的解通常是经过多次迭代获取的局部较(最)优解, 采用随机搜索产生一个比当前解质量更优的解的概率较低. 因此, 本文提出一种基于全局学习机制的侦查蜂搜索策略, 即在当前全局最好解基础上, 采用由左变异和交叉(图3(e)和图3(f))组成邻域结构集的 RVNS 搜索策略进行开采, 生成的新序列既保留当前全局最好解的部分优秀序列片段, 又有新序列片段的注入, 可有效降低侦查蜂搜索盲目性, 提高开采到蜜源的质量.

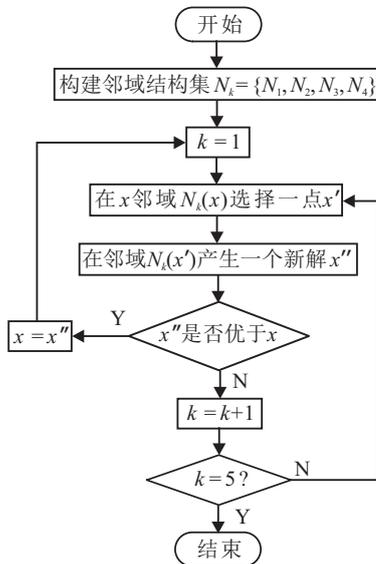


图4 RVNS搜索过程

2.7 HABC算法流程

HABC算法初始化蜜源后, 雇佣蜂、观察蜂和侦查蜂分工协作更新蜜源, 直到满足停止迭代条件, 算法寻优过程如下所示.

Step 1: 参数初始化 SN , $uplimit$, $maxTime$.

Step 2: 采用面向工作站的随机生成法构建初始蜜源 $x_i (i = 1, 2, \dots, SN)$, 并记录最好蜜源 x_{best} .

Step 3: 雇佣蜂引领阶段, 每一个雇佣蜂重复以下子步骤:

Step 3.1: 在当前蜜源 x_i 上应用交换、插入、逆序和多次插入组成的 RVNS 策略产生新解 x'_i ;

Step 3.2: 评价新解 x'_i 的质量 $f(x'_i)$;

Step 3.3: 更新当前蜜源并记录蜜源未改进次数 $limit$, 若 x'_i 优于 x_i , 则 $x_i = x'_i$, $limit = 0$, 否则 $limit = limit + 1$.

Step 4: 观察蜂跟随阶段, 每一个观察蜂重复以下子步骤:

Step 4.1: 采用分阶段选择评价法选择雇佣蜂跟随;

Step 4.2: 同 Step 3.1;

Step 4.3: 同 Step 3.2;

Step 4.4: 同 Step 3.3.

Step 5: 侦查蜂开采阶段, 若蜜源 x_i 未改进次数 $limit$ 超过 $uplimit$, 则重复以下子步骤:

Step 5.1: 在当前最好蜜源 x_{best} 上应用由左变异和交叉组成的 RVNS 策略产生新解 x''_{best} ;

Step 5.2: 评价新解 x''_{best} 的质量 $f(x''_{best})$;

Step 5.3: 更新蜜源, 若 x''_{best} 优于 x_i , 则 $x_i = x''_{best}$, $limit = 0$.

Step 6: 更新最优蜜源 x_{best} .

Step 7: 若达到终止条件, 则输出 x_{best} ; 否则, 跳至 Step 3 继续搜索更优蜜源.

3 实验结果与分析

HABC算法采用 C++ 编码, 在 Core i5-4288U, 4 G 内存电脑上运行, 具体参数设置为 $SN = 25$, $uplimit = 8$, $maxTime = 25$ s. 目前已发表文献中对 SDDLBP 的研究仅有小、中规模两个算例, 分别为包含 10 个零件的 P10 算例和包含 25 个零件的 P25 手机拆卸案例, 具体数据指标参见文献[19]. 鉴于此, 本文对文献[13]中包含 47 个零部件的笔记本电脑拆卸案例进行修改, 设计了 SDDLBP 大规模 P47 算例, 以充分验证本文所提出算法的有效性. 假设任务 3/4、9/10、11/12、20/22、23/24、34/35、37/43 之间存在相互干扰, 时间增量为 $sd_{3,4} = 3$, $sd_{4,3} = 5$, $sd_{9,10} = 7$, $sd_{10,9} = 1$, $sd_{11,12} = 3$, $sd_{12,11} = 1$, $sd_{20,22} = 6$, $sd_{22,20} = 2$, $sd_{23,24} = 12$, $sd_{24,23} = 5$, $sd_{34,35} = 5$, $sd_{35,34} = 2$, $sd_{37,43} = 8$, $sd_{43,37} = 25$, 其他数据请参见对应文献.

为了验证 HABC 算法求解性能, 本文利用 HABC 算法、标准 ABC 和 VNS 算法对 P10, P25 和 P47 三种规模算例进行求解. 为确保结果的可靠性, 每个算例运行 100 次, 计算其均值和标准差, 对比结果如表 1 所示. 对于小规模算例 P10, 3 种元启发式算法都能找到最优解且标准差为 0, 最优解为 $f_1 = 5$, $f_2 = 177$, $f_3 = 119$, $f_4 = 5$, $f_5 = 8305$, 其所对应的拆卸序列任务在工作站上的分配如图 5 中的 S_1 所示. 随着任务数的增多, 解空间规模呈几何级增加, 求解 P 规模更大的 SDDLBP 的最优解愈加困难, 因此, 元启发式算法只能在指定时间内寻得问题的满意解即近似最优解, 从 P25 和 P47 算例求解结果可以看出, ABC、VNS 和 HABC 算法对目标函数 f_1 和 f_2 的求解结果一致, 但对于目标函数 f_3 , HABC 算法获得最优解的均值和

表 1 VNS、ABC和HABC算法求解结果对比

目标函数	方法	P10(maxTime = 5 s)		P25(maxTime = 25 s)		P47(maxTime = 50 s)	
		均值	标准差	均值	标准差	均值	标准差
f_1	VNS	5.00	0.00	10.00	0.00	5.00	0.00
	ABC	5.00	0.00	10.00	0.00	5.00	0.00
	HABC	5.00	0.00	10.00	0.00	5.00	0.00
f_2	VNS	177.00	0.00	163.00	0.00	878.00	0.00
	ABC	177.00	0.00	163.00	0.00	878.00	0.00
	HABC	177.00	0.00	163.00	0.00	878.00	0.00
f_3	VNS	119.00	0.00	35.28	9.39	3,022.42	407.69
	ABC	119.00	0.00	35.34	7.51	3,009.82	426.35
	HABC	119.00	0.00	35.00	0.00	2,993.60	233.46
f_4	VNS	5.00	0.00	80.79	8.16	367.72	163.49
	ABC	5.00	0.00	80.81	10.17	349.42	161.22
	HABC	5.00	0.00	81.31	3.76	334.78	168.67
f_5	VNS	8 305.00	0.00	936.63	67.45	2 852.44	842.07
	ABC	8 305.00	0.00	937.55	73.35	2 790.22	817.23
	HABC	8 305.00	0.00	938.02	50.06	2 715.58	887.31

标准差均优于标准ABC和VNS算法,在求解大规模P47算例时,优势则更加明显,充分体现了本文提出的HABC算法的可行性与优越性。

图5为采用是否考虑总拆卸时间DT的两种模型对P10算例进行求解所获得的最优拆卸序列 S_1 和 S_2 的解码结果对比。可以看出:两模型获得最优解在工作站开启数量NWS上相同;在总拆卸时间DT方面,本文所提出模型获得的最优拆卸序列 S_1 ,因任务间相互干扰造成拆卸时间总增加量为 $SD_1 = sd_{56} + sd_{96} + sd_{54} + sd_{14} + sd_{23} = 2 + 1 + 1 + 2 + 2 = 8s$,总拆卸时间 $DT_1 = \sum_{i=1}^{10} t_i + SD_1 = 169 + 8 = 177s$;拆卸序列 S_2 的 $SD_2 = sd_{56} + sd_{96} + sd_{41} + sd_{45} + sd_{32} = 2 + 1 + 4 + 4 + 3 = 14s$, $DT_2 = 169 + 14 = 183s$;在平滑指数SI方面, $SI_1 = 3^2 + 3^2 + 7^2 + 4^2 + 6^2 = 119$, $SI_2 = 5^2 + 3^2 + 4^2 + 4^2 + 1^2 = 69$ 。 $DT_2 > DT_1$, $SI_2 < SI_1$,归其原因在于,拆卸过程中, S_2 存在选择“蹩脚”(受拆卸干扰时间长)任务优先拆卸的现象,即选择任务1和任务5先于任务4拆卸,任务2先于任务3拆卸,导致序列总干扰时间增加,工作站空闲时间减少,以此达到最小化平滑指数的目标。

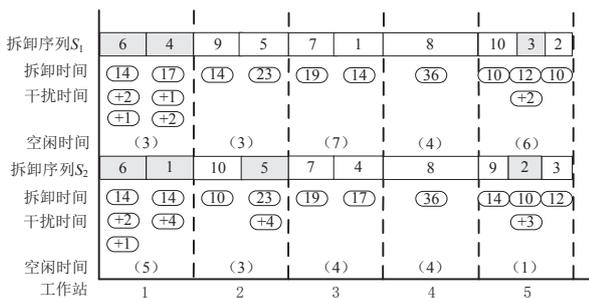


图 5 P10算例在两种模型获得最优拆卸序列解码

需要注意的是,对比拆卸序列 S_1 和 S_2 的作业

时间,可以看出:采用不考虑总拆卸时间DT模型,获得的最优拆卸序列 S_2 的任务总干扰时间接近本文所提出模型获得的最优拆卸序列 S_1 的2倍($SD_1 = 8s$, $SD_2 = 14s$),这不仅增加了操作工人的工作量,同时也加大了机器设备的作业负荷与能耗。实际上,工人应以最便捷的方式拆卸零部件,从而提高工作效率。此外,对P25和P47等中大规模产品拆卸,伴随产品零部件的增多,拆卸过程中任务间相互干扰也将随之增加,若不考虑最小化实际总拆卸时间DT,则会导致工人和机器设备的作业负荷不合理增加,与现实情况不符。因此,针对SDDLBP特点,在满足最小工作站数量NWS前提下,考虑最短总拆卸时间DT这一目标,可以有效避免在拆卸过程中选择“蹩脚”的任务优先拆卸,验证了本文所提出模型的合理性。

4 结 论

本文针对拆卸过程中无先后关系约束任务间可能存在相互干扰,会因为拆卸顺序不同导致拆卸时间不确定的SDDLBP进行研究,建立了多目标SDDLBP优化模型。在最小化工作站开启数量、均衡工作站空闲时间、尽早拆卸有危害和高需求零部件多目标基础上,考虑了最短总拆卸时间,以减少因任务间相互干扰导致实际总拆卸时间增加,并提出一种基于字典排序的混合人工蜂群算法。通过模型对比,得出考虑总拆卸时间DT可有效减少工人和机器设备的作业负荷不合理增加,证明了本文所提出模型的合理性及有效性。此外,通过对小、中、大3种规模SDDLBP算例求解,验证了HABC算法求解性能明显优于标准ABC和VNS算法,并随着问题规模的增大优势更为明显,验证了本文所提出HABC算法的可行性及优越性。

参考文献(References)

- [1] Ilgin M A, Gupta S M. Environmentally conscious manufacturing and product recovery(ECMPRO): A review of the state of the art[J]. *J of Environmental Management*, 2010, 91(3): 563-591.
- [2] Gupta S M, Gungor A. Product recovery using a disassembly line: challenges and solution[C]. *Proc of the 2001 IEEE Int Symposium on Electronics and the Environment*. Denver, 2001: 36-40.
- [3] Koc A, Sabuncuoglu I, Erel E. Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph[J]. *IIE Trans*, 2009, 41(10): 866-881.
- [4] Itekin F T, Akkan C. Task-failure-driven rebalancing of disassembly lines[J]. *Int J of Production Research*, 2012, 50(18): 4955-4976.
- [5] Bentaha M L, Battaia O, Dolgui A. A sample average approximation method for disassembly line balancing problem under uncertainty[J]. *Computers & Operations Research*, 2014, 51(3): 111-122.
- [6] Altekin F T, Kandiller L, Ozdemirel N E. Profit-oriented disassembly-line balancing[J]. *Int J of Production Research*, 2008, 46(10): 2675-2693.
- [7] MCGovern S M, Gupta S M. Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem[J]. *Int J of Production Research*, 2007, 45(18/19): 4485-4511.
- [8] MCGovern S M, Gupta S M. A balancing method and genetic algorithm for disassembly line balancing[J]. *European J of Operational Research*, 2007, 179(3): 692-708.
- [9] Prakash P, Ceglarek D, Tiwari M K. Constraint-based simulated annealing(CBSA) approach to solve the disassembly scheduling problem[J]. *The Int J of Advanced Manufacturing Technology*, 2012, 60(9): 1125-1137.
- [10] 胡扬, 张则强, 汪开普, 等. 多目标拆卸线平衡问题的 Pareto 细菌觅食算法[J]. *计算机应用研究*, 2016, 33(11): 3265-3269.
(Hu Y, Zhang Z Q, Wang K P, et al. Pareto based bacteria foraging optimization for multi-objective disassembly line balancing problem[J]. *Computer Application Research*, 2016, 33(11): 3265-3269.)
- [11] Ding L P, Feng Y X, Tan J R, et al. A new multi-objective ant colony algorithm for solving the disassembly line balancing problem[J]. *Int J of Advanced Manufacturing Technology*, 2010, 48(5): 761-771.
- [12] 李明, 张则强, 胡扬. U型布局的拆卸线平衡问题及其求解算法研究[J]. *现代制造工程*, 2015(7): 7-12.
(Li M, Zhang Z Q, Hu Y. Research on U-shaped disassemblyline balancing problem and solving algorithm[J]. *Modern Manufacturing Engineering*, 2015(7): 7-12.)
- [13] Kalayci C B, Hancilar A, Gungor A, et al. Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm[J]. *J of Manufacturing Systems*, 2015, 37: 672-682.
- [14] Scholl A, Boysen N, Flidner M. The sequence-dependent assembly line balancing problem[J]. *OR Spectrum*, 2008, 30(3): 579-609.
- [15] Andrés C, Miralles C, Pastor R. Balancing and scheduling tasks in assembly lines with sequence-dependent setup times[J]. *European J of Operational Research*, 2008, 187(3): 1212-1223.
- [16] Yolmeh A, Kianfar F. An efficient hybrid genetic algorithm to solve assembly line balancing problem with sequence-dependent setup times[J]. *Computers & Industrial Engineering*, 2012, 62(4): 936-945.
- [17] Hamta N, Ghomi S M T F, Jolai F, et al. A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect[J]. *Int J of Production Economics*, 2013, 141(1): 99-111.
- [18] Akpinar S, Bayhan G M, Baykasoglu A. Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks[J]. *Applied Soft Computing*, 2013, 13(1): 574-589.
- [19] Kalayci C B, Gupta S M. Ant colony optimization for sequence-dependent disassembly line balancing problem[J]. *J of Manufacturing Technology Management*, 2013, 24(3): 413-427.
- [20] Kalayci C B, Polat O, Gupta S M. A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem[J]. *Annals of Operations Research*, 2016, 242(2): 321-354.
- [21] Kalayci C B, Gupta S M. A tabu search algorithm for balancing a sequence-dependent disassembly line[J]. *Production Planning & Control*, 2014, 25(2): 149-160.
- [22] Kalayci C B, Polat O, Gupta S M. A variable neighbourhood search algorithm for disassembly lines[J]. *J of Manufacturing Technology Management*, 2015, 26(2): 182-194.
- [23] Kalayci C B, Gupta S M. Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem[J]. *Expert Systems with Applications*, 2013, 40(18): 7231-7241.
- [24] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Erciyes university, 2005.
- [25] Mladenovic N, Hansen P. Variable neighborhood search[J]. *Computers & Operations Research*, 1997, 24(11): 1097-1100.
- [26] Hansen P, Mladenovic N. Variable neighborhood search: Principles and applications[J]. *European J of Operational Research*, 2001, 130(3): 449-467.