DOI: 10.3901/JME.2018.03.125

多目标不完全拆卸线平衡问题的建模与优化^{*}

李六柯 张则强 朱立夏 邹宾森
(西南交通大学机械工程学院 成都 610031)

摘要: 针对实际生产中在满足约束条件下仅考虑拆卸需求零件和危害零件的特点,以工作站数目、空闲时间均衡指标和拆卸成本为优化目标,构建了不完全拆卸线平衡问题多目标模型。基于解的离散性和优化目标的多重性,提出一种 Pareto 解集思想的变邻域-粒子群融合算法。该算法通过建立拆卸任务和粒子群迭代搜索的对应关系,将变邻域搜索作为局部搜索策略,同时引入 Pareto 解集思想、拥挤距离机制处理多目标问题,以保证求解结果的多样性;通过 Hyper-volume 指标解决了多目标优化难以评价算法收敛性能及 Pareto 解集优劣等问题。采用所提算法求解不同规模完全拆卸线平衡问题测试算例,其中不同搜索深度的对比试验表明了变动搜索深度能很好地兼顾求解质量和求解效率,不同算法的对比试验表明了所提算法的优越性。最后,将所提模型与求解方法应用至某打印机不完全拆卸线的设计中。

关键词: 拆卸线平衡问题; 不完全拆卸; 变邻域搜索; 粒子群算法; Hyper-volume 指标

中图分类号: TH165; TP301

Modeling and Optimizing for Multi-objective Partial Disassembly Line Balancing Problem

LI Liuke ZHANG Zeqiang ZHU Lixia ZOU Binsen

(School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031)

Abstract: Aiming at the disassembly characteristics of only considering the parts in demand and the hazardous parts in the actual production, a multi-objective mathematical model of the partial disassembly line balancing problem is constructed including the workstation number, smooth rate and disassembly cost three optimization objectives. And a Pareto based particle swarm algorithm incorporated with variable neighborhood is proposed considering the discrete nature of the solution and multiple optimization objectives. The mapping relationship between the disassembly tasks and the particle swarm iterative search is structured and the variable neighborhood search method is incorporated as the local search strategy in the proposed algorithm. To deal with the multiple objectives for the problem, the Pareto solution set and the crowding distance evaluation mechanism is introduced to ensure the diversity of the solution results. In addition, the Hyper-volume indicator is used to overcome the difficulty in evaluating the algorithm convergence performance and the Pareto muster in solving the multi-objective optimization problems. The superiority of the proposed algorithm is verified by different scale test cases of complete disassembly line balancing problem, and the contrast experiments show that the change of search depth of the proposed algorithm owns excellent performance the quality and computation efficiency. Finally, the proposed mathematical model and the presented algorithm is applied to the partial disassembly line design of a certain type of printer.

Key words: disassembly line balancing problem; partial disassembly; variable neighborhood search; particle swarm optimization algorithm; Hyper-volume indicators

0 前言

近年来,我国以汽车、电器电子产品为主的机电产品报废数量庞大,此外随着人民生活品质的提高,机电产品更新换代愈加频繁,可预见其报废量

仍将持续增长。由于废旧机电产品兼具资源性和环境危害性,国家出台了相关的法律法规及补贴政策来约束和引导企业承担产品废弃后的回收和资源化责任。产品回收过程离不开拆卸操作,通过拆卸废旧产品分离出需求零件和危害零件,其中需求零件指具有较高回收价值的零件,这类零件可通过再制造技术使其恢复到使用状态,故经济价值较高,拆卸需求零件可为企业带来较高的经济效益;危害零件指在拆卸过程中对环境有害的零件,这类零件经

^{*} 国家自然科学基金(51205328, 51405403)、教育部人文社会科学研究青年基金(12YJCZH296)和四川省应用基础研究计划项目(2014JY0232)资助项目。20170219 收到初稿,20170627 收到修改稿

标准化拆卸处理能最大限度地减少对环境的污染,符合绿色制造对环境保护的要求。单个产品拆卸或单元拆卸较之于拆卸线,缺乏严格的拆卸步骤和拆卸标准,容易造成二次损坏且工作效率低下,而拆卸线是废旧产品拆卸组织方式的必然趋势,可实现大规模自动化高效拆卸。流水线生产方式下工作站作业时间的均衡性严重制约拆卸效率,引起了国内外学者的广泛关注。

自 2001 年正式提出拆卸线平衡问题(Disassembly line balancing problem, DLBP)^[1]以来,启发式算法^[1-3]、数学规划方法^[4]和智能算法^[5-11]相继被用于求解该问题。DLBP 是多目标优化问题,涉及工作站数目、空闲时间均衡指标、需求指标、危害指标和拆卸成本等多个目标。在早期研究中,多采用考虑线性加权的单目标启发式规则生成一个可行拆卸序列,但由于各目标量纲不同,加权值难以确定等原因,求解结果不能很好地平衡各目标且具有不确定性。为提高求解质量,提出了能够求得确定最优解的数学规划方法,但与启发式算法类似同样只考虑单目标,加之 DLBP 是 NP-完全问题^[10],随着问题规模的增大,数学规划方法遍历的解空间呈指数增长,不适合求解大规模问题。目前,智能算法因其通用性好和全局寻优能力强等优点,在求解 DLBP 中得到广泛应用,包括禁忌搜索算法^[5]、蚁群算法^[6]、粒子群算法^[7]、人工蜂群算法^[8]和变邻域搜索算法^[9]等,上述算法采用字典顺序法对多目标进行处理,根据假定的重要程度依次优化各目标,均可求得问题的单个较优解,其实质仍为单目标优化。文献[11]提出了一种多目标蚁群算法,引入 Pareto 解集思想,在决策者偏好未知情况下提供多种备选方案,为求解多目标问题提供了一种新思路。

前述研究均是针对拆卸废旧机电产品全部零件的完全拆卸模型^[1-12],然而实际拆卸由于需求零件和危害零件需要后续处理,必须执行拆卸操作,而将其余零部件通过破碎机、自动分拣机分离不同材料类型的颗粒物,不必进行完全拆卸作业,这大大减少了拆卸工序。因此,文献[13]采用 AND/OR 关系图定义拆卸过程,建立混合整数规划模型用于求解以利润为导向的不完全拆卸线平衡问题(Partial disassembly line balancing problem, PDLBP)。但该文献的目标仅考虑了拆卸成本因素,不能反映各工作站的负荷均衡状况,在求解方法方面,精确方法仅能求解小规模问题,难以应对中大规模问题,因此限制了方法的可应用性。参考现有文献,尚未发现有以零件优先关系图定义拆卸过程的不完全拆卸研究。

粒子群算法(Particle swarm optimization, PSO)^[13]是一种新型群智能算法,其优势在于操作简单且全局搜索能力强。最早在连续函数优化上取得了显著效果,随后在旅行商问题^[14]、装配序列优化问题^[15]和调度问题^[16]等典型的组合优化问题中得到成功应用。文献[7, 12]直接将连续 PSO 算法用于求解多目标 DLBP,采用字典顺序法依次优化各目标,其实质仍为单目标优化,求解结果单一。采用随机数组进行编码,解码方法是将数组升序排列后的顺序作为问题的解,该解码过程依赖其他基因值来确定,当父代产生子代后,子代不能继承父代的优良性能,这种编码方式不具备 Lamarckian 性^[17]。

针对上述粒子群算法在求解多目标 DLBP 时的不足,提出一种基于变邻域-粒子群融合的离散多目标改进粒子群算法(Multi-objective improved particle swarm optimization, MIPSO),采用基于任务编号的编码方式,针对 PDLBP 离散性、强约束性等特点,重新定义速度、位置及相关操作^[14]以满足优先关系约束;将用于调整各任务在拆卸序列中所处位置的局部搜索采用变邻域搜索方式,以适应算法迭代前期未知搜索空间大,而后期局部搜索空间重复的性质;各粒子通过趋近局部最优解和全局最优解来引导种群向最优解靠拢,同时采用精英保留策略记录种群在寻优过程中所得最优解;考虑到 PDLBP 是多目标优化问题,引入 Pareto 解集思想和拥挤距离机制^[18],获得分布较为均匀的 Pareto 最优解集,并采用 Hyper-volume 指标^[19]加以评价。

1 PDLBP 数学模型

1.1 问题描述

传统的拆卸线平衡问题多以完全拆卸模型为主,要求拆卸废旧机电产品的全部零件,同时综合拆卸线开启的工作站数目最少、各工作站的负荷均衡、尽早拆卸需求零件和危害零件等四个目标,将拆卸任务合理分配到各工作站中。然而,为提高拆卸收益,同时降低拆卸后零部件对环境的危害,在拆卸过程中一些具有较高回收价值的零部件和对环境有危害的零部件必须进行拆卸作业,而那些经济价值较低或无经济价值的零部件以及对环境无危害的零部件则不需经过拆卸作业,可以直接粉碎加工进入回收阶段,故而不完全拆卸应运而生。相较于完全拆卸,不完全拆卸操作工序更少,降低了拆卸成本,同时兼顾了回收效益和环保理念,因而不完全拆卸更符合企业实际拆卸情况。

某废旧机电产品由 n 个零件组成,定义每个零

件对应一项拆卸任务, 根据各零件间的物理约束, 绘制零件优先关系图, 确定拆卸优先关系矩阵 A 和零件类别矩阵 B

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

式中 A ——拆卸优先关系矩阵;

n ——拆卸任务总数;

a_{pq} ——任务 p 和任务 q 的优先关系, 若任务 p 为任务 q 的紧前任务, 则 $a_{pq}=1$, 否则 $a_{pq}=0$, 其中 $p=1, 2, \cdots, n$ 、 $q=1, 2, \cdots, n$ 。

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{M1} & b_{M2} & \cdots & b_{Mn} \end{pmatrix}$$

式中 B ——零件类别矩阵;

M ——零件类型总数;

n ——拆卸任务总数;

b_{ji} ——拆卸任务 i 与零件类型 j 的关系, 若执行拆卸任务 i 可得第 j 类零件, 该零件若为需求零件则 $b_{ji}=1$, 若为危害零件则 $b_{ji}=-1$, 否则 $b_{ji}=0$, 其中 $j=1, 2, \cdots, M$ 、 $i=1, 2, \cdots, n$ 。

已知节拍时间上限 $T_{\max}$ 、任务 i 的作业时间 t_i 、任务 i 的拆卸成本 U_i 、第 j 类零件的需求 D_j 和第 j 类零件有无危害 Y_j (若有危害 $Y_j=1$, 否则 $Y_j=0$)。在满足优先关系约束和节拍约束前提下, 将需求零件和危害零件对应的拆卸任务分配到最多 $K(K=n)$ 个工作站中。为简化问题作如下假设: ① 拆卸单一产品; ② 待拆卸产品的数量无限; ③ 零件不可再分; ④ 待拆卸产品零件数量齐全; ⑤ 标准化拆卸作业, 拆卸时间已知; ⑥ 拆卸线采用直线型布局; ⑦ 不考虑零部件在拆卸线上的流转时间; ⑧ 忽略其他突发状况。

1.2 数学模型

决策变量:

$$x_{ik} = \begin{cases} 1 & \text{任务 } i \text{ 分配到第 } k \text{ 个工作站} \\ 0 & \text{否则} \end{cases}$$

$$S_k = \begin{cases} 1 & \text{工作站 } k \text{ 开启} \\ 0 & \text{否则} \end{cases}$$

目标函数

$$F = \min[F_1, F_2, F_3] \quad (1)$$

$$F_1 = \sum_{k=1}^K S_k \quad (2)$$

$$F_2 = \sum_{k=1}^K \left(S_k \cdot T_c - \sum_{i=1}^n t_i \cdot x_{ik} \right)^2 \quad (3)$$

$$F_3 = \sum_{k=1}^K S_k \cdot T_c \cdot (\max\{U_i \mid i \in (i \mid x_{ik} = 1)\}) \quad (4)$$

约束条件

$$\sum_{i=1}^n t_i \cdot x_{ik} \leq T_{\max} \cdot S_k \quad \forall k \quad (5)$$

$$\sum_{k=1}^K x_{ik} \leq 1 \quad \forall i \quad (6)$$

$$x_{ik} \leq \sum_{h=1}^k x_{lh} \quad \forall k, \forall i, \forall l \in \{l \mid a_{li} = 1\} \quad (7)$$

$$S_{k-1} \geq S_k \quad \forall k \in \{2, 3, \cdots, K\} \quad (8)$$

$$\sum_{i \in E} \sum_{k=1}^K x_{ik} \geq D_j \quad \forall j, E \in \{i \mid b_{ji} = 1\} \quad (9)$$

$$\sum_{k=1}^K x_{ik} = Y_j \quad \forall j, \forall i \in \{i \mid b_{ji} = -1\} \quad (10)$$

$$T_c = \max \left(\sum_{i=1}^n t_i \cdot x_{ik}, \forall k \right) \quad (11)$$

引入两个决策变量: $S_k(k=1, 2, \cdots, K)$ 表示工作站 k 是否开启, 若开启, $S_k=1$, 否则 $S_k=0$; x_{ik} 表示任务 i 分配到工作站 k , 若成立 $x_{ik}=1$, 否则 $x_{ik}=0$ 。已知优先关系矩阵、零件类别矩阵、节拍时间上限、零件拆卸时间、拆卸成本、各类零件的危害和需求。

优化目标式(1)表示最小化三个目标, 其中目标 1: 式(2)计算工作站数目 F_1 , 开启新的工作站意味着需要投入更多的设备、人员、场地等资源, 因此要求工作站数目最小; 目标 2: 式(3)计算空闲时间均衡指标 F_2 , 当工作站的作业时间不一致时, 较长的工作站作业时间会引起下游工作站处于等待状态, 同时上游工作站处于堵塞状态, 使得各工作站的负荷不均衡, 不符合流水线生产方式的“一个流”要求, 而各工作站最理想的状态是作业时间均为节拍时间, 此时各工作站的空闲时间均为 0, 生产进度严格按照节拍时间组织生产; 目标 3: 式(4)计算拆卸成本 F_3 , 从经济角度出发要求拆卸成本最小, 为评价拆卸线的经济效益提供了依据。

约束条件说明: 式(5)为节拍时间约束, 保证开启的工作站内拆卸作业时间之和不超过节拍时间上限; 式(6)为任务分配约束, 保证每个拆卸任务只能

分配到一个工作站,允许有不参与分配的任务;式(7)为优先关系约束,保证各拆卸任务的前序任务优先分配;式(8)为工位约束,保证开启的工作站内分配有拆卸任务;式(9)为需求约束,保证各类零件至少满足需求;式(10)为危害约束,保证危害零件必须拆除;式(11)确定节拍时间 T_c 。

本文将完全拆卸模型扩展到不完全拆卸模型,使之更加符合生产实际。针对企业只拆卸需求零件和危害零件的特点,在节拍时间不确定下增加了需求约束和危害约束,并对原有完全拆卸线平衡问题模型作了相应修改以表达不完全拆卸线平衡问题的约束。完全拆卸线平衡问题是不完全拆卸线平衡问题的一个特例,在约束条件式(6)只取等号,保证每个任务能且仅能分配到一个工作站,这时需求约束式(9)、危害约束式(10)均能满足约束条件,该模型退化为完全拆卸线平衡问题模型。从模型中可以看出,不完全拆卸线平衡问题模型较完全拆卸线平衡问题模型约束更多、更为复杂,求解难度加大。

2 求解 PDLBP 的 MIPSO

粒子群算法通过模拟鸟类觅食行为,建立了速度-位置更新模型,最早用于求解连续函数优化问题,有收敛速度快,精度高等优点。近年来,粒子群算法的应用扩展到了离散优化问题^[14-16],其难点在于编码的离散化处理以及保证算法各操作所得解的可行性。

2.1 离散化

本文根据 PDLBP 解的离散性、强约束性等特点,提出了一种基于变邻域-粒子群融合的离散 MIPSO。其速度、位置更新公式与连续粒子群算法类似见式(12)、(13),重新定义速度、位置及相关操作。

$$\mathbf{v}_{t+1} = w \times \mathbf{v}_t \oplus c_1 \times (\mathbf{x}_t - \mathbf{p}_t) \oplus c_2 \times (\mathbf{x}_t - \mathbf{g}_t) \quad (12)$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + w \times \mathbf{v}_{t+1} \quad (13)$$

式中, $\mathbf{x}_t/\mathbf{x}_{t+1}$ 和 $\mathbf{v}_t/\mathbf{v}_{t+1}$ 分别表示第 $t/t+1$ 阶段(迭代次数)粒子的位置和速度; $\mathbf{p}_t$ 是粒子 t 阶段的局部最优解; $\mathbf{g}_t$ 是粒子 t 阶段的全局最优解; w 是惯性权值; c_1 、 c_2 是加速因子。

(1) 位置: 假设废旧产品包含 n 个零件,解的形式为一组满足约束条件的 n 维矢量 $\mathbf{X}(x_1, x_2, \dots, x_n)$, 各维元素为零件编号,同一编号不允许出现两次。

(2) 速度: 定义 $I(x_i, x_j)$ 表示交换解矢量 $\mathbf{X}(x_1, x_2, \dots, x_n)$ 中第 i 维和第 j 维的元素,只有满足优先

关系约束的交换对才能交换各自元素,称为有效交换对记为 Γ^+ , 否则为无效交换对记为 Γ 。由一系列有效交换对构成的列矢量称为速度 $\mathbf{v}(\Gamma_1^+, \Gamma_2^+, \dots)^T$ 。

(3) 加法操作: 包括速度加法和位置加法,分别记为“ $\oplus$ ”、“ $+$ ”。在式(12)速度更新公式中,迭代开始时无初始速度, $(t+1)$ 阶段速度为 t 阶段速度、粒子局部速度和粒子全局速度三部分的速度加法操作,速度加法是速度与速度之间的加法操作。结合 PDLBP 的强约束性, t 阶段速度 $\mathbf{v}_t$ 中交换对 $\Gamma_t^+(t)$ 对于 t 阶段的粒子是有效交换对,但对于 $(t+1)$ 阶段的粒子可能是无效交换对,故不能直接作为 $(t+1)$ 阶段的初始速度 $\mathbf{v}_{t+1}$ 中的交换对 $\Gamma_0^+(t+1)$, 而必须从 t 阶段的初始速度中剔除对于 $(t+1)$ 阶段粒子的无效交换对,假设 $(t+1)$ 阶段粒子所有有效交换对为 $\Gamma^+(t+1)$, 则 $\Gamma_0^+(t+1) = \Gamma_t^+(t) \cap \Gamma^+(t+1)$ 。在式(13)位置更新公式中,位置加法是位置与速度之间的加法操作,根据有效交换对 $\Gamma_t^+(t+1)$ 交换 t 阶段粒子位置 $\mathbf{X}(t)$ 中对应维数的元素。

(4) 减法操作: 两位置间的减法操作结果为速度,记为“ $-$ ”。比较两位置对应维数上元素是否相同,同一维数上不同元素构成交换对,从所有交换对中剔除无效交换对即为减法操作所求速度,例如: t 阶段粒子位置和局部最优位置的减法操作,结果为 t 阶段粒子的局部最优速度; t 阶段粒子的位置和全局最优位置的减法操作,结果为 t 阶段粒子的全局最优速度。

(5) 乘法操作: 实数与速度之间的乘法操作结果为速度,记为“ $\times$ ”,例如 $\mathbf{v}_2 = c \times \mathbf{v}_1$ 。具体操作如下: (i). 统计速度 $\mathbf{v}_1$ 包含的交换对个数 $|\mathbf{v}_1|$, 乘以该实数 c , 向下取整 $|\mathbf{v}_2| = \lfloor (c \times |\mathbf{v}_1|) \rfloor$; (ii). 若 $0 < c < 1$, 即 $|\mathbf{v}_2| < |\mathbf{v}_1|$, 从速度 $\mathbf{v}_1$ 的交换对中随机挑选 $|\mathbf{v}_2|$ 个交换对组成速度 $\mathbf{v}_2$, 否则从全部有效交换对中随机挑选 $(|\mathbf{v}_2| - |\mathbf{v}_1|)$ 个交换对与 $\mathbf{v}_1$ 合并组成 $\mathbf{v}_2$ 。

2.2 变邻域搜索

迭代开始时将粒子目标值作为各粒子的局部最优值、全局最优值。在粒子跟随局部、全局最优值的过程中,将该粒子的邻域最优值作为粒子局部最优值以更新局部最优速度和位置,将粒子群的最优值作为粒子全局最优值以更新全局最优速度和位置,按照式(12)、(13)更新粒子速度和位置。邻域搜索的步骤如下。

(1) 输入初始粒子的位置即初始拆卸序列 $\mathbf{X}$, 搜索深度 r 。

(2) 随机选择一个任务作为邻域出发点,找出其紧前、紧后任务,确定可交换任务集 V_1 。

(3) 判断可交换任务集 V_1 是否为空,若为空,

转至步骤(2), 否则转至步骤(4)。

(4) 从 V_1 中随机选择一个任务, 找出其紧前、紧后任务, 确定该任务的可交换任务集 V_2 。

(5) 判断邻域出发点是否在可交换任务集 V_2 中, 若存在, 将上述两任务交换, 记邻域搜索的深度前进一步, 转至步骤(6), 否则转至步骤(4)。

(6) 将新粒子作为初始粒子, 循环步骤(2)~(5), 直至达到搜索深度 r 。

为了扩大算法迭代前期粒子群邻域搜索的范围, 同时考虑算法迭代后期粒子有更大概率追随全局最优值, 局部搜索采用变邻域搜索方式: 随着迭代次数的增加, 减小搜索深度 r 。初始搜索深度 $r=r_e=\lceil n/2 \rceil$, 终止搜索深度 $r_0=2$, 间隔 $\lceil MG/(r_e-r_0) \rceil$ 次循环, 即迭代次数大于 $\left(\frac{r_e-r+1}{r_e-r_0} \cdot MG\right)$ 时, 则令 $r=r-1$, 直至 $r=r_0$ 保持该搜索深度不变, 其中 MG 表示算法最大迭代次数。

2.3 多目标处理方法

多目标问题的各子目标间可能存在冲突、量纲不一致等情况, 无法进行比较。因此, 多目标问题不同于单目标问题, 不能简单地判断解的优劣。本文结合 Pareto 解集思想处理多目标问题, 对于 Z 个子目标的多目标最小化优化问题, 其中任意两个解 X_1 、 X_2 满足式(14), 则称 X_1 支配 X_2 , X_1 对应的目标值优于 X_2 对应的目标值。不被可行决策空间其它解支配的解构成 Pareto 最优解集, 其目标值的集合称为 Pareto 最优前沿。

$$\begin{cases} \forall i \in \{1, 2, \dots, Z\}, f_i(X_1) \leq f_i(X_2) \\ \exists j \in \{1, 2, \dots, Z\}, f_j(X_1) < f_j(X_2) \end{cases} \quad (14)$$

粒子初始化随机产生粒子的位置, 将该位置作为粒子局部最优位置、全局最优位置的初始状态。粒子通过邻域搜索在附近搜寻相邻解, 并计算这些相邻解的目标值。结合 Pareto 解集思想, 记录粒子经历的非劣位置, 随机选择一个位置作为其局部最优位置。整合全部位置保留非劣解, 所构成的 Pareto 最优解集作为现阶段的外部档案, 每个粒子随机追随一个全局最优位置。

设置外部档案容量 N_0 , 当非劣解个数大于 N_0 时算法效率下降, 需要对外部档案中的非劣解进行筛选。引入 NSGA-II 拥挤距离机制^[18]快速标定非劣解排序, 对每一个子目标 f_m 按升序排列, 其中 $m \in \{1, 2, \dots, Z\}$, 令 f_m 的两个边界解子拥挤距离为 $l_1^m = l_{N_0}^m = 1$, 其余解在 f_m 的子拥挤距离定义如下

$$l_i^m = \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{N_0} - f_m^1} \quad (15)$$

$$L_i = \sum_{m=1}^Z l_i^m \quad (16)$$

式中, $i \in \{2, 3, \dots, (N_0-1)\}$, N_0 为非劣解个数, l_i^m 表示第 i 个解在第 m 个子目标上的子拥挤距离, L_i 表示第 i 个解的拥挤距离。

2.4 Pareto 解集评价

Hyper-volume 指标^[19]用于评价 Pareto 最优解集的质量, 简称 H 指标。该指标具有单调递增性, 因此, 当算法循环一定代数后, H 指标趋于平稳, 可认为算法在此时收敛。H 指标表示 Pareto 最优解集相对参考点所支配目标空间的超体积, 选取各目标的上界作为参考点。以图 1 中双目标为例, H 指标等于非劣解目标值 O_1 、 O_2 、 O_3 、 O_4 与参考点 O 围成的阴影面积, 该阴影面积扩展到多维称作超体积, 超体积越大, Pareto 最优解集的质量越好。

图 1 双目标时的 H 指标计算示意图

2.5 算法流程

求解多目标不完全拆卸线平衡问题的 MIPS0 流程。

(1) 输入问题信息: 优先关系矩阵 A 、总任务数 n 、零件类别矩阵 B 、节拍时间上限 $T_{\max}$ 、拆卸时间 t 、拆卸成本 U 、各类零件的危害 Y 和需求 D 。

(2) 设置算法参数: 种群规模 N 、最大迭代次数 MG 、初始搜索深度 $r_e = \lceil n/2 \rceil$ 、终止搜索深度 $r_0=2$ 、外部档案容量 N_0 、加速因子 c_1 , c_2 、惯性权重值 w 。

(3) 根据优先关系矩阵 A 随机生成初始种群 POP , 令外部档案 $Q = \emptyset$ 。

(4) 计算目标值, 更新局部最优位置 p_0 和目标值 $F(p_0)$, 筛选非劣解更新外部档案 Q 。

(5) 迭代计数器 $i=1$, 搜索深度 $r=r_e$ 。

(6) 种群 POP 中各粒子执行搜索深度为 r 的邻域搜索, 更新其局部最优位置 p_i 和目标值 $F(p_i)$, 筛选非劣解更新外部档案 Q 。

(7) 统计外部档案 Q 的非劣解个数 N_Q , 若 $N_Q > N_0$, 则根据式(15)、(16)计算各非劣解的拥挤距离 L , 筛选外部档案 Q , 保留 N_0 个拥挤距离大的非劣解, 执行下一步骤, 否则直接执行下一步骤。

(8) 为每个粒子随机从外部档案 Q 中选定一个全局最优位置 g_i 和目标值 $F(g_i)$ 。

(9) 计算初始速度、局部速度和全局速度, 根据式(12)、(13)更新粒子速度和位置, 计算更新位置的目标值, 随机选择一个非劣解作为粒子新位置以更新种群 POP , 筛选非劣解更新外部档案 Q , 计算外部档案 Q 的 H 指标。

(10) 每迭代 $\lceil MG / (r_e - r_0) \rceil$ 次, 搜索深度 $r = r - 1$ 。

(11) 判断 $i < MG$, 若成立, 令迭代计数器 $i = i + 1$, 执行步骤(6); 若不成立则终止算法, 输出外部档案 Q 作为最优拆卸方案。

MIPSO 的流程如图 2 所示。

图2 MIPSO 算法流程图

3 算法验证

采用 Matlab R2010b 开发 MIPSO 程序, 在 Intel(R) Pentium(R) CPU, 主频 2.90 GHz 双核, 内存 2 GB 计算机环境下运行程序。鉴于不完全拆卸线平衡问题尚无测试算例, 而完全拆卸模型是不完全拆卸模型的一个特例, 因此本文通过求解文献[3]、[11]研究的 P10、P52 完全拆卸线平衡问题验证算法的有效性, 其中对小规模 P10 问题设计不同的搜索深度, 探讨邻域搜索中搜索深度 r 参数的设置, 同时与新启发式算法(New heuristic, NH)^[2]、Greed/2-Opt^[3]、禁忌搜索算法(Tabu search algorithm,

TS)^[5]、蚁群算法(Ant colony optimization, ACO)^[6]、人工蜂群算法(Artificial bee colony algorithm, ABC)^[8]的计算结果进行对比, 验证 MIPSO 求解小规模问题的有效性。针对大规模 P52 问题设置邻域搜索深度, 对比多目标蚁群算法(Multi-objective ant colony optimization, MACO)^[11]的计算结果, 验证 MIPSO 求解大规模问题时的性能。

3.1 小规模算例

P10 问题以工作站数目 f_1 、空闲时间均衡指标 f_2 、危害指标 f_3 和需求指标 f_4 为优化目标, 详细参数及计算公式参见文献[3]。其中工作站数目 f_1 、空闲时间均衡指标 f_2 与所建模型的目标 F_1 、 F_2 含义相同, 此外新模型还考虑了实际生产中企业更加关心的拆卸成本 F_3 , 但是为方便与 NH、Greed/2-Opt、TS、ACO、ABC 五种算法进行比较, 因此, 引入文献[3]中的 4 个目标来验证本文所提 MIPSO 的求解性能。为测试邻域搜索深度 r 对算法性能的影响, 分别取固定搜索深度 $r = \{2, 3, 4, 5\}$ 和变动搜索深度 $r = 5d$ (初始搜索深度为 5); 为兼顾求解质量和计算时间, 设置种群规模 $N = 20$, 迭代次数 $MG = 40$; 因需要同时优化 4 个目标, 为求得侧重点不同的多样性解, 令外部档案容量 $N_0 = 10$; 为平衡局部最优位置与全局最优位置对粒子下一阶段位置的影响, 加速因子 c_1 、 c_2 选择(0, 1)的随机值、惯性权值 w 的取值参考文献[20], 算法运行 10 次的结果见表 1, 其中均值最优结果以粗体标示, 未达标 H_e 值用下划线标示。 MG_e 表示算法收敛到最大 H 指标 H_e 值的迭代次数, $T(s)$ 表示算法运行时间。P10 问题各目标最差值为 $f_1 = 10$, $f_2 = 5\ 887$, $f_3 = 10$, $f_4 = 16\ 945$, 为便于计算将参考点设置为(10, 6 000, 10, 17 000)。

对于小规模 P10 问题, 保持算法其它参数不变, 在五种搜索深度下重复运行 10 次, 总共包含 50 组试验。分析表 1 数据可知: 搜索深度 $r = 2$ 第 1、2 组试验, 搜索深度 $r = 3$ 第 10 组试验未能找到最优解, 余下 47 组试验均找到了最优解, 寻优率高达 94%; 对比分析四种固定搜索深度均值结果, 随着搜索深度的增加, 算法运行时间加长同时收敛速度加快, 因而可以通过增加迭代次数提高算法在较小固定搜索深度下收敛到最优值的概率; 变动搜索深度 $r = 5d$ 达到收敛的平均迭代次数为 8.3 次, 平均运行时间仅比固定搜索深度 $r = 4$ 长 0.01 s, 比固定搜索深度 $r = 5$ 短 0.09 s, 因而采用变动搜索深度可以综合算法运行时间和收敛速度, 达到改进算法的目的, 算法收敛时的结果见表 2 中方案 6~12。

表 1 不同搜索深度下 MIPSO 求解 P10 问题的结果

	$r=2$			$r=3$			$r=4$			$r=5$			$r=5d$		
	MG_e	$H_e/10^9$	T/s	MG_e	$H_e/10^9$	T/s	MG_e	$H_e/10^9$	T/s	MG_e	$H_e/10^9$	T/s	MG_e	$H_e/10^9$	T/s
1	15	<u>1.966 2</u>	4.71	8	1.966 4	4.95	12	1.966 4	5.09	12	1.966 4	5.23	12	1.966 4	5.10
2	22	<u>1.966 2</u>	4.70	17	1.966 4	4.70	17	1.966 4	4.80	26	1.966 4	4.79	8	1.966 4	4.91
3	24	1.966 4	4.43	9	1.966 4	4.76	11	1.966 4	4.83	5	1.966 4	4.88	8	1.966 4	4.79
4	20	1.966 4	4.46	29	1.966 4	4.71	10	1.966 4	4.81	9	1.966 4	4.87	9	1.966 4	4.80
5	34	1.966 4	4.43	8	1.966 4	4.82	15	1.966 4	4.78	18	1.966 4	4.92	7	1.966 4	4.78
6	25	1.966 4	4.56	19	1.966 4	4.78	7	1.966 4	4.80	5	1.966 4	4.97	7	1.966 4	4.91
7	36	1.966 4	4.43	7	1.966 4	4.74	6	1.966 4	4.79	6	1.966 4	4.97	8	1.966 4	4.78
8	13	1.966 4	4.57	20	1.966 4	4.58	33	1.966 4	4.81	8	1.966 4	4.89	12	1.966 4	4.79
9	9	1.966 4	4.58	29	1.966 4	4.61	8	1.966 4	4.94	9	1.966 4	4.92	6	1.966 4	4.78
10	10	1.966 4	4.41	10	<u>1.966 2</u>	4.63	13	1.966 4	4.75	15	1.966 4	4.97	6	1.966 4	4.82
均值	20.8	<u>1.966 34</u>	4.53	15.6	<u>1.966 38</u>	4.73	13.2	1.966 4	4.84	11.3	1.966 4	4.94	8.3	1.966 4	4.85

表 2 P10 问题各算法的分配方案

序号	算法	分配方案	f_1	f_2	f_3	f_4
1	NH	[(6,5)→(7,10)→(9,1)→(4)→(8)→(2,3)]	6	1143	3	7 935
2	Greed/2-Opt	[(6,5)→(7,9)→(4,1)→(8)→(10,2,3)]	5	219	3	7 575
3	TS	[(6,1,10)→(5)→(7,4)→(8)→(9,2,3)]	5	341	5	9 605
4	ACO	[(10,5)→(6,7)→(4,9)→(8)→(1,2,3)]	5	211	4	10 090
5	ABC	[(6,1,10)→(5)→(7,4)→(8)→(9,2,3)]	5	341	5	9 605
6		[(6,4)→(10,5)→(7,9)→(8)→(1,2,3)]	5	211	5	8 885
7		[(10,5)→(6,7)→(9,4)→(8)→(1,2,3)]	5	211	4	9 730
8		[(6,5)→(9,7)→(4,1)→(8)→(10,2,3)]	5	219	4	7 510
9	MIPSO	[(6,5)→(7,9)→(4,1)→(8)→(10,2,3)]	5	219	3	7 575
10		[(6,4)→(10,5)→(9,7)→(8)→(1,2,3)]	5	211	6	8 820
11		[(6,9)→(5,10)→(7,4)→(8)→(1,2,3)]	5	241	5	7 445
12		[(6,9)→(5)→(7,10)→(4,1)→(8)→(2,3)]	6	975	4	7 150

表 2 对比了 NH、Greed/2-Opt、TS、ACO、ABC 和 MIPSO 六种算法的结果: 最小工作站数目 $f_1=5$, 开启 5 个工作站, 最小空闲时间均衡指标 $f_2=211$, 最小危害指标 $f_3=3$, 最小需求指标 $f_4=7150$, 各目标最优值用粗体标示; 前五种算法均将多目标问题处理为单目标问题, 只能得到单一解, 而本文所提 MIPSO 能得到 7 个互不占优的非劣解, 解的多样性增多; 对比前五种算法的解, 其中 ABC 的解 5 和 TS 的解 3 相同, Greed/2-Opt 的解 2 支配解 1、解 3 和解 5, ACO 的解 4 与解 2 互不占优, 因此 Greed/2-Opt 和 ACO 的解为非劣解; MIPSO 的解 7 与 ACO 的解 4 比较, $f_4^7 < f_4^4$, 其他三个目标相同, 所以解 7 支配解 4, 与 Greed/2-Opt 的解 2 互不占优; 解 9 与 Greed/2-Opt 的解 2 相同; 余下 5 个解与解 2、解 4 互不占优。综上所述, MIPSO 所求结果具有多样性, 求解质量优于现有五种算法, 表明了所提算法在求解小规模问题时有效。

3.2 大规模算例

为验证所提算法在求解大规模问题时的性能, 以文献[11]中 52 项拆卸任务的高速电子套结机 P52

问题为研究对象, 出于可比性考虑, 仍以平均闲置率 F_{Idle} 、平滑率 F_{Smooth} 、拆卸成本 F_{Cost} 为目标。为统一最小化优化方向, 令不平滑率 $F_{NSmooth}=1-F_{Smooth}$, 具体公式详见文献[11], 通过与文献[11]中 MACO 的结果对比, 以验证所提 MIPSO 求解大规模问题时的性能。其中, P52 问题各目标最差值为 $F_{Idle}=1$, $F_{Smooth}=1$, $F_{Cost}=547.56$ 元, 为便于计算选择(1, 1, 600)作为参考点。设置算法参数: 种群规模 $N=60$, 在对小规模问题测试时变动搜索深度优势明显, 因而设置 $r=26d$ (初始搜索深度为 26), 算法其他参数与 3.1 节保持一致。为设置合理的迭代次数, 先对 MIPSO 作预计算, 令迭代次数 $MG=5\ 000$, 对应的 H 指标迭代过程和结果如图 3 所示。其中图 3a H 指标呈单调递增趋势, 在迭代前期 H 指标跳跃式增长, 在第 240 次迭代后, 出现第一个短暂的平稳期, 第 500 次迭代后出现较长的平稳期, 其后最大的跳跃出现在第 2 291 次迭代, 但 H 指标增幅仅为 0.208%, 此后直至算法终止 H 指标保持平稳, 表明所提算法具有较好的收敛性; 图 3b 中方框表示 60 个随机粒子组成的初始种群, 其分散性强, 目标值

较差,随着迭代次数的增加,粒子逐步趋近 Pareto 前沿,圆圈表示终止种群,即算法终止后 60 个初始粒子所处的位置,终止种群分布较集中,目标值较优;三角形表示算法运行过程中粒子经历的最佳位置。结合图 3a、3b 可知,所提 MIPSO 可求出分散性强、目标值较优的多样性解,为决策者提供侧重点不同的选择方案。

权衡求解结果和计算时间,选取处于较长平稳期的迭代次数 $MG=700$,算法重复运行 10 次。每次运行 MIPSO 所得 H_e 及非劣解个数 N_e 见表 3,其

中第 8 次运行所得 Pareto 解集的 H_e 最大,第 1、5、8、10 次运行所得非劣解个数最多。综合 10 次所得非劣解如图 4 所示,算法运行 10 次的结果均在 Pareto 前沿附近,其中 F_{Idle} 均为 0.057 9,开启 7 个工作站,而文献[11]中 MACO 求得 6 个解的闲置率为 0.057 9,开启 7 个工作站,同时有 4 个解的闲置率为 0.175 7,开启 8 个工作站; F_{Smooth} 指标的范围在 79.13%~99.99%,较 MACO 结果更具分散性; F_{Cost} 指标的范围在 126.624~146.436 元,均小于 MACO 的解。

图3 预计算迭代过程及结果图

表3 MIPSO 运行 10 次最大 H 指标及非劣解个数

	1	2	3	4	5	6	7	8	9	10
H_e	445.418	445.423	444.941	444.144	444.814	444.155	445.243	445.710	444.664	444.585
N_e	10	8	<u>4</u>	9	10	8	9	10	6	10

图4 MIPSO 运行 10 次的 Pareto 前沿

选择上述 MIPSO 运行 10 次中 H_e 值最小的第 4 次运行的结果与文献[11]中 MACO 的结果进行对比。算法第 4 次运行结果见表 4,求得 9 个非劣解,闲置率 F_{Idle} 均为 0.0579,开启 7 个工作站;平滑率 F_{Smooth} 的范围为 91.79%~99.95%;拆卸成本 F_{Cost} 的范围为 128.268~140.772 元。MIPSO 解 2 的平滑率最大 $F_{Smooth}=0.999 5$,对应的拆卸成本

$F_{Cost}=140.772$ 元,MACO 最大平滑率为 0.974 7,对应的拆卸成本为 173.388 元,平滑率增加 2.54%,拆卸成本减少 18.81%,而从 MIPSO 求解的结果可知,平滑率与拆卸成本互相冲突,一个目标值的增加需要牺牲另一个目标,由此可知,文献[11]中 MACO 的求解性能仍有较大的提升空间。MIPSO 解 5 的平滑率最小 $F_{Smooth}=0.917 9$,对应的拆卸成本 $F_{Cost}=128.268$ 元,与 MACO 最大平滑率的解比较,平滑率减少 5.83%,拆卸成本减少 26.02%,两者互不占优。MIPSO 解 4 的平滑率次小 $F_{Smooth}=0.970 2$,对应的拆卸成本 $F_{Cost}=128.640$ 元,平滑率最小的解 5 与之比较,拆卸成本降低 0.29%,代价是平滑率减少 5.39%。综上所述,第四次运行结果中解 4、5 与 MACO 的解互不占优,余下 7 个解其平滑率 $F_{Smooth}>0.974 7$,对应的拆卸成本 $F_{Cost}<150.546$ 元,均能支配 MACO 的解。就算法性能而言,所提 MIPSO 在求得比 MACO 更具多样性结果的同时解的质量大幅提升。

表 4 MIPSO 第 4 次运行结果

序号	分配方案	F_{Idle}	F_{Smooth}	$F_{Cost}/元$
1	[(29,21,36,28,33,18)→(4,32,37,3,31,42,14,15,11,47,46)→(1,25,12,45)→(2,35,27,52,49,23,51,34,43)→(26,30,9,39)→(22,19,40,16,50,41,48,38,7,17,44,24,8,13)→(10,6,20,5)]	0.057 9	0.998 1	129.906
2	[(4,36,47,32,21,28,3,46)→(2,31,29,45,25,14,33,52)→(49,1,12,30)→(51,9,15,26,37)→(18,40,11,42,34,23,38,27,19)→(22,35,16,24,7,8,50,43,13,10,44,48,41)→(6,17,20,39,5)]	0.057 9	0.999 5	140.772
3	[(37,36,33,3,42,21)→(29,47,14,46,44,28,39,4,2,32,43,31)→(1,9,26)→(12,30,25,52)→(18,49,11,23,15,27,19,35,34)→(22,51,7,16,38,17,24,8,40,13,10,41,50)→(6,20,48,45,5)]	0.057 9	0.989 8	128.730
4	[(29,36,33,3,42,21)→(37,47,14,46,44,28,39,4,32,2,43,31)→(1,9,26)→(12,30,25,52)→(18,49,11,23,15,27,19,35,34)→(22,51,7,16,38,17,24,8,40,13,10,41,50)→(6,20,48,45,5)]	0.057 9	0.970 2	128.640
5	[(33,36,18,3,42,21)→(4,32,28,37,16,47,29,46,31,15,14,52)→(26,1,9)→(12,11,30,25)→(23,43,39,35,49,51,38,27,40,34)→(24,41,2,22,44,7,17,8,13,10,50,19)→(6,20,5,48,45)]	0.057 9	0.917 9	128.268
6	[(36,21,42,33,37,18)→(29,4,28,47,31,3,15,14,11,32,46)→(30,1,43,12)→(26,23,25,27)→(35,52,38,2,19,49,44,34,45,22,24)→(17,8,39,7,13,16,51,20,10,6)→(5,9,40,41,50,48)]	0.057 9	0.998 3	135.642
7	[(29,4,36,28,18,33,47)→(37,32,14,15,21,31,11,42,3,46)→(1,12,25,45)→(2,34,27,49,16,35,19,51,23)→(9,30,26,39)→(38,22,40,50,52,44,7,48,41,43,24,17,8,13)→(20,10,6,5)]	0.057 9	0.998 2	132.396
8	[(37,33,42,21,18,36)→(4,32,47,29,46,31,28,15,14,3,44)→(1,16,12,25)→(9,30,39,26)→(34,49,51,35,2,52,40,11,27,23)→(22,7,19,38,45,43,24,50,41,48,17,8,13)→(20,10,6,5)]	0.057 9	0.997 0	129.750
9	[(36,21,42,33,37,3)→(29,4,28,47,31,18,15,14,11,32,46)→(30,1,43,12)→(26,23,25,27)→(35,52,38,2,19,49,44,34,45,22,24)→(17,8,39,7,13,16,51,20,10,6)→(5,9,40,41,50,48)]	0.057 9	0.998 6	135.804

4 实例应用

现以某型号打印机拆卸线为研究对象, 分析所提模型和算法的应用情况。该拆卸线零件类别矩阵 B 为单位阵, 需求 D 、危害 Y 、拆卸时间 $t(s)$ 、单位时间拆卸成本 $U(元/s)$ 及其紧前任务 h 等拆卸信息见表 5, 其中拆卸时间是经多次拆卸计时取平均值测定。综合考虑零件拆卸时间和拆卸约束条件, 给定节拍时间上限 $T_{\max}=150 s$, 算法参数设置同第 3.2 节一致, 同时优化工作站数目 F_1 、空闲时间均衡指标 F_2 和拆卸成本 F_3 , 其优化过程 H 指标变化曲线如图 5 所示, 优化结果见表 6。

图 5 PDLBP 迭代过程 H 指标变化曲线

观察图 5 H 指标变化曲线可知: H 指标在前 20 次迭代过程中跳跃式增长, 在 60 次迭代后出现了较长的稳定期, 在 200 次迭代后 H 指标收敛到 $8.824 5 \times 10^7$, 兼顾求解质量和计算时间, 可认为算法收敛。MIPSO 求得 3 个非劣解的 Pareto 最优解集见表 6, 工作站数目 F_1 均为 4, 空闲时间均衡指标 F_2 的结果为 0~18, 拆卸成本 F_3 的结果范围在 5.394 0~5.497 8 元; 方案 1 节拍时间 $T_c=147 s$, 有 8 个未拆卸零件, 空闲时间均衡指标最小 $F_2=0$ 达到理想状态; 方案 2 节拍时间 $T_c=148 s$, 有 7 个未拆卸零件, 与其它两个方案比较互不占优; 方案 3 节拍时间 $T_c=145 s$, 有 8 个未拆卸零件, 拆卸成本最小 $F_3=5.394 0$ 元; 三种方案均有未拆卸零件, 从而减少了不必要的拆卸任务, 同时节拍时间因拆卸任务改变而进行自适应调整, 其中方案 3 节拍时间最小, 单位时间内理论拆卸产量最大。分析三个优化目标的结果可知: 随着空闲时间均衡指标 F_2 的降低, 即各工作站的负荷越来越均衡, 拆卸成本 F_3 随之增加, 说明空闲时间均衡指标与拆卸成本这两个目标存在博弈关系, 不能同时达到最优。因而决策者在选择方案时必须明确偏好, 若优先考虑拆卸成本最低, 则选择方案 3; 若优先考虑空闲时间均衡指标最小, 则选择方案 1; 若要求兼顾拆卸成本低和空闲时间均衡指标小, 则选择方案 2。

表 5 拆卸任务相关信息表

序号	零件名称	<i>D</i>	<i>Y</i>	<i>t</i>	<i>U</i>	<i>h</i>	序号	零件名称	<i>D</i>	<i>Y</i>	<i>t</i>	<i>U</i>	<i>h</i>
1	电源线	0	0	3	0.002 9	0	29	拾纸组件	0	0	15	0.005 3	25,26,27,28
2	硒鼓	0	0	3	0.002 9	0	30	带排线 2	0	0	2	0.002 7	11
3	搓纸轮	1	0	2	0.002 7	0	31	防尘罩螺钉	0	0	40	0.010 3	11
4	分页器螺钉	0	0	16	0.005 5	0	32	齿轮防尘罩	0	0	3	0.002 9	30,31
5	分页器	0	0	5	0.003 3	4	33	继电器螺钉	0	0	8	0.003 9	11
6	进纸盘	0	0	40	0.010 3	0	34	继电器	1	0	2	0.002 7	33
7	前面板	0	0	18	0.005 9	9,11	35	格式化板螺钉	0	0	24	0.007 1	11
8	左侧板螺钉	0	0	8	0.003 9	0	36	格式化板	1	0	2	0.002 7	35
9	左侧板	0	0	33	0.008 9	8	37	传动齿轮组	0	0	15	0.005 3	32
10	右侧板螺钉	0	0	8	0.003 9	0	38	弹簧组	0	0	2	0.002 7	32
11	右侧板	0	0	13	0.004 9	10	39	ECU 罩螺钉	0	0	32	0.008 7	7
12	拉杆	0	0	5	0.003 3	0	40	ECU 外罩	0	0	36	0.009 5	39
13	顶盖螺钉	0	0	16	0.005 5	0	41	基座螺钉	0	0	32	0.008 7	0
14	顶盖	0	0	10	0.004 3	9,11,12	42	基座	0	0	4	0.003 1	27,41
15	组排线	0	0	18	0.005 9	9	43	发动机螺钉	0	0	16	0.005 5	40
16	加热组件螺钉	0	0	24	0.007 1	14	44	发动机	1	0	3	0.002 9	43,48
17	加热组件	1	0	4	0.003 1	16	45	右内侧板螺钉	0	0	8	0.003 9	11
18	后盖板螺钉	0	0	8	0.003 9	0	46	右内侧板	0	0	2	0.002 7	43,45
19	后盖板	0	0	2	0.002 7	18	47	激电板螺钉	0	0	8	0.003 9	40
20	电源板螺钉 1	0	0	16	0.005 5	9	48	激电板排线 1	0	0	2	0.002 7	40
21	电源板螺钉 2	0	0	32	0.008 7	9	49	激电板排线 2	0	0	2	0.002 7	40
22	电源板	0	1	15	0.005 3	15,20,21, 24,25,28	50	激光器电路板	0	1	10	0.004 3	24,30,47, 48,49
23	塑料挡片	0	1	6	0.003 5	15	51	激光器螺钉	0	0	32	0.008 7	50
24	带排线 1	0	0	2	0.002 7	9	52	激光器带排线	0	0	2	0.002 7	40
25	地线	0	0	2	0.002 7	9	53	激光器	0	1	2	0.002 7	49,51,52
26	组件螺钉 1	0	0	16	0.005 5	19	54	左内侧板螺钉	0	0	8	0.003 9	22
27	组件螺钉 2	0	0	32	0.008 7	0	55	左内侧板	0	0	2	0.002 7	54
28	拾纸组件排线	0	0	4	0.003 1	9							

表 6 MIPSO 求解不完全拆卸线平衡问题 P55 的结果

序号	分配方案	未拆卸零件	<i>F</i> ₁	<i>F</i> ₂	<i>F</i> ₃ /元	<i>T</i> _c /s
1	[(12,8,27,9,10,1,3,4,13,28,25,15)→(11,24,41,7,45,20,35,36,39)→(21,40,5,33,48,30,34,23,22,49,2,14,16)→(6,18,47,42,19,50,51,52,26,43,53,44,17)]	46,29,31,32,38,37,54,55	4	0	5.497 8	147
2	[(12,2,3,8,10,1,13,9,11,28,30,15,27)→(4,45,7,41,20,35,39,36)→(21,40,5,33,34,25,48,23,6,49,24,14)→(16,22,18,42,47,19,50,51,52,26,43,46,44,17,53)]	29,31,32,38,37,54,55	4	2	5.416 8	148
3	[(2,8,6,12,10,27,13,9)→(11,28,25,15,35,4,41,7,45,30,1,36)→(39,21,40,5,33,23,20,34,47)→(43,49,24,14,16,22,18,48,42,19,50,51,52,53,3,46,44,17)]	26,29,31,32,54,38,37,55	4	18	5.394 0	145

5 结论

(1) 建立了以工作站数目、空闲时间均衡指标和拆卸成本为目标的不完全拆卸线平衡问题多目标模型，在完全拆卸线平衡问题模型的基础上，放松任务分配约束以允许某些任务不参与分配，同时增加需求约束和危害约束以保证拆卸全部需求零件和危害零件。

(2) 针对以往将连续 PSO 算法直接用于离散多目标拆卸线平衡问题的求解以及将多目标处理成单

目标进行计算等不足，设计了一种基于变邻域-粒子群融合的离散多目标改进粒子群算法。采用基于任务的编码方式，重新定义速度、位置及相关操作；不考虑决策者的偏好程度，引入 Pareto 解集思想、拥挤距离决策机制处理多目标问题，保证求解结果的多样性；同时借助 H 指标解决多目标问题难以评价算法收敛性能及 Pareto 解集优劣等问题。

(3) 采用变邻域搜索实现粒子的局部寻优，保证算法迭代前期粒子在较大的解空间进行搜索，在算法迭代后期粒子有更大概率追随全局最优解。

(4) 鉴于 PDLBP 尚无测试算例，采用所提

MIPSO 求解完全拆卸线平衡问题的 P10、P52 问题, 计算结果表明所提 MIPSO 优于其他对比算法。通过测试不同搜索深度, 验证了变动搜索深度的优越性。采用预计算分析 H 指标的变化曲线, 并权衡求解质量和计算时间, 为算法参数的选择提供了一种新思路。结合企业生产实际, 给出了所提模型和算法在某打印机拆卸线设计的具体应用。

参 考 文 献

- [1] GUNGOR A, GUPTA S M, POCHAMPALLY K, et al. Complications in disassembly line balancing[C]// Intelligent Systems and Smart Manufacturing. International Society for Optics and Photonics, 2001: 289-298.
- [2] AVIKAL S, JAIN R, YADAV H C, et al. A new heuristic for disassembly line balancing problems with AND/OR precedence relations[C]// Proceedings of the Second International Conference on Soft Computing for Problem Solving, December 28-30, 2012. Springer India, 2014: 519-525.
- [3] MCGOVERN S M, GUPTA S M. 2-opt heuristic for the disassembly line balancing problem[C]// Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing III, 2004: 71-84.
- [4] 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 Transactions, 2009, 41(10): 866-881.
- [5] KALAYCI C B, GUPTA S M. A tabu search algorithm for balancing a sequence-dependent disassembly line[J]. Production Planning & Control, 2013, 25(25): 149-160.
- [6] MCGOVERN S M, GUPTA S M. Ant colony optimization for disassembly sequencing with multiple objectives[J]. International Journal of Advanced Manufacturing Technology, 2006, 30(5): 481-496.
- [7] KALAYCI C B, GUPTA S M. A particle swarm optimization algorithm for solving disassembly line balancing problem[C]// Proceedings for the Northeast Region Decision Sciences Institute, 2012: 347-357.
- [8] 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.
- [9] KALAYCI C B, POLAT O, GUPTA S M. A variable neighborhood search algorithm for disassembly lines[J]. Journal of Manufacturing Technology Management, 2015, 26(2): 182-194.
- [10] MCGOVERN S M, GUPTA S M. A balancing method and genetic algorithm for disassembly line balancing[J]. European Journal of Operational Research, 2007, 179(3): 692-708.
- [11] 丁力平, 谭建荣, 冯毅雄, 等. 基于 Pareto 蚁群算法的拆卸线平衡多目标优化[J]. 计算机集成制造系统, 2009, 15(7): 1406-1413.
- [12] 方群, 张则强, 李明. 面向多目标拆卸线平衡问题的一种改进粒子群优化算法[J]. 现代制造工程, 2016, (4): 8-15.
- [13] FANG Qun, ZHANG Zeqiang, LI Ming. A developed PSO with mutation operators for multi-objective disassembly line balancing problems[J]. Modern Manufacturing Engineering, 2016, (4): 8-15.
- [14] ALTEKIN F T, KANDILLER L, OZDEMIREL N E. Profit-oriented disassembly-line balancing[J]. International Journal of Production Research, 2008, 46(10): 2675-2693.
- [15] CLERC M. Discrete particle swarm optimization, illustrated by the traveling salesman problem[M]. Berlin Heidelberg: Springer, 2004.
- [16] 王丰产, 孙有朝, 李娜. 多工位装配序列粒子群优化算法[J]. 机械工程学报, 2012, 48(9): 155-162.
- [17] WANG Fengchan, SUN Youchao, LI Na. Multi station assembly sequence planning based on particle swarm optimization algorithm[J]. Journal of Mechanical Engineering, 2012, 48(9): 155-162.
- [18] 侯媛彬, 薛斐, 郑茂全, 等. 基于多层编码的遗传-粒子群融合算法流水线优化控制[J]. 机械工程学报, 2015, 51(9): 159-164.
- [19] HOU Yuanbin, XUE Fei, ZHENG Maoquan, et al. Optimization of flow-shop control by using genetic-particle swarm algorithm of multilayer-coded[J]. Journal of Mechanical Engineering, 2015, 51(9): 159-164.
- [20] CHENG R, GEN M, TSUJIMURA Y. A tutorial survey of job-shop scheduling problems using genetic algorithms—I. representation[J]. Computers & Industrial Engineering, 1996, 30(4): 983-997.
- [21] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-194.

182-197.

- [19] BADER J, ZITZLER E. HypE: An algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary Computation, 2011, 19(1): 45-76.
- [20] TAHERKHANI M, SAFABAKHSH R. A novel stability-based adaptive inertia weight for particle swarm optimization[J]. Applied Soft Computing, 2016, 38: 281-295.

作者简介: 李六柯, 男, 1993 年出生, 硕士研究生。主要研究方向为生产线平衡与智能优化。

E-mail: bk20111413@my.swjtu.edu.cn

张则强(通信作者), 男, 1978 年出生, 博士, 教授, 博士研究生导师。主要研究方向为制造系统与智能优化。

E-mail: zzq_22@163.com

朱立夏, 女, 1993 年出生, 博士研究生。主要研究方向为生产线平衡与智能优化、设施布局优化。

邹宾森, 男, 1990 年出生, 硕士研究生。主要研究方向为生产线平衡与智能优化。

第 7 届上银优秀机械博士论文奖——佳作奖

高 Q 值硅微机械陀螺系统特性分析与面向精度的设计方法研究

作者: 申强

毕业学校: 西北工业大学

指导教师: 苑伟政

微机械陀螺是测量刚体角运动的基础核心元器件, 是代表精密仪器制造水平的高端复杂测量装置。陀螺精度是制约其发展与应用的关键, 而品质因数(Q)是影响硅微机械陀螺精度及其系统性能的重要因素, 对陀螺的机械耦合以及电耦合误差, 系统零位、噪声以及闭环电路控制能力有着重要影响。本文以 Z 轴硅微机械陀螺为基础, 结合机械单元和电路单元, 从系统角度研究了高 Q 值硅微机械陀螺特性、高 Q 值对陀螺精度的作用机理以及以陀螺精度为目标的系统设计方法、系统控制电路设计等问题。主要结果如下所述。

1. 分析了高 Q 值硅微机械陀螺系统特性。获得了高 Q 值作用下陀螺带宽与有效品质因数的反比例关系以及系统零位输出的“交流”分量表达式; 系统地建立了陀螺结构热传递模型, 为分析、提高陀螺温度特性以及热启动时间提供坚实的理论基础。

2. 研究了高 Q 值影响硅微机械陀螺精度的作用机理。确认了高 Q 值降低阻尼不平衡分量是零位漂移减小的关键因素, 高 Q 值降低驱动电压从而提高中心电容的取值裕度是系统信噪比提高的重要原因; 在此基础上, 建立了由 8 个误差源组成的陀螺零位模型, 解决了零位分量贡献值难以确认的问题, 该模型对认识影响陀螺精度的本质因素具有重大理论和实用价值。

3. 建立了系统级、器件级以及结构级参数之间的关联, 阐明了陀螺 Q 值、带宽、中心电容、接口电路噪声是系统设计的关键参数, 提出了以关键参数为支撑引导陀螺系统精度的设计流向, 解决了面向精度的陀螺系统设计问题; 进一步通过增加敏感模态振荡抑制梳齿电极实现敏感模态静电力闭环控制, 该电路省略了复杂的调制、解调单元, 极大地简化了控制电路复杂度。

4. 验证了陀螺系统设计、分析方法的可行性。通过实验测试对比, 改进后硅微机械陀螺信噪比提高到改进前的 16.3 倍, 与理论设计值的误差仅为 10.4%, 陀螺精度达到 $2.1^{\circ}/h$, 测试结果表明陀螺系统设计分析方法是有效的, 陀螺样机能够满足战术级精度需求。