基于改进社会工程算法的拆解线平衡多目标优化设计

田广东[1,2,3]，张志峰[2]，张诚[3]，张乐乐[4]
（1.山东交通学院 运输车辆检测、诊断与维修技术交通行业重点实验室，济南 250357；
2.浙江大学 流体动力与机电系统国家重点实验室，杭州 310027；3.山东大学 机械工程学院，
济南 250061；4.交通运输部管理干部学院 综合运输教研部，北京 101601）

摘要：目的 为了提高拆卸效率，减少拆卸成本和拆卸能耗，提出一种基于社会工程算法的优化方法。**方法** 首先根据社会工程算法的原理设计了编码和解码方式，然后引入交换子和交换序列的概念，对算法中的训练和再训练、发现攻击和响应攻击等步骤进行设计，最后选取一个高速电子套结机作为实验对象进行了验证。**结果** 与其他多目标优化算法相比，具有良好的寻优能力。**结论** 提出的算法有效地解决了拆解线平衡问题，与其他算法相比具有更好的性能。

关键词：拆解线平衡；社会工程算法；绿色设计；绿色制造
中图分类号：TB472 **文献标识码：**A **文章编号：**1001-3563(2022)12-0052-07
DOI：10.19554/j.cnki.1001-3563.2022.12.006

Multi-objective Optimization of Disassembly Line Balancing Based on Improved Social Engineering Optimizer

TIAN Guang-dong[1,2,3], *ZHANG Zhi-feng*[2], *ZHANG Cheng*[3], *ZHANG Le-le*[4]
(1.Transport Industry Key Laboratory of Vehicle Detection, Diagnosis and Maintenance Technology, ShanDong JiaoTong University, Jinan 250357, China; 2.State Key Laboratory of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou 310027, China; 3.School of Mechanical Engineering, Shandong University, Jinan 250061, China; 4.Transport Management Institute Ministry of Transport of the P. R. China, Ministry of Comprehensive Transport Education and Research, Beijing 101601, China)

ABSTRACT: This paper aims to improve disassembly efficiency and reduce disassembly cost and disassembly energy consumption, and an optimization method based on a social engineering optimizer is proposed. Firstly, in this paper, the encoding and decoding method is designed according to the principle of social engineering optimizer. And then the concepts of swap sub and swap sequence are introduced. Next, the steps of training and retraining, discovering attack and responding to attack in the algorithm are designed. Finally, a high-speed electronic tacking machine is selected as the experimental objective for validation. This method has good optimization capacity compared with other multi-objective algorithms. The proposed algorithm effectively addresses the disassembly line balancing problem and has better performance compared with other algorithms.

KEY WORDS: disassembly line balancing; social engineering optimizer; green design; green manufacturing

收稿日期：2022–01–14
基金项目：浙江大学流体动力与机电系统国家重点实验室开放基金（GZKF-202012），山东交通学院运输车辆检测、诊断与维修技术交通行业重点实验室开放基金（JTZL2101），国家自然科学基金（52075303）
作者简介：田广东（1980—），男，博士，教授，主要研究方向为再制造和绿色制造、绿色物流和运输、汽车的智能检查和维修、决策和智能优化等。
通信作者：张诚（1996—），男，硕士生，主攻决策和智能优化、再制造和绿色制造等。

随着制造业的不断发展，大量报废（End–Life，EOL）产品带来的资源再利用和潜在的环境污染问题亟待解决。如何高效、经济地处理 EOL 产品已成为一个研究热点。EOL 产品的拆卸是回收或维护的重要组成部分[1]，可以减少环境污染，促进资源回收。拆卸序列规划（Disassembly Sequence Planning，DSP）旨在生成零件或子装配的拆卸顺序[2]，以满足各种拆卸要求，例如拆卸成本和收益。然而，随着报废产品数量的大量增加，针对单个产品的拆卸序列规划方法已不适用，由此产生了拆解线平衡问题（Disassembly Line Balancing Problem，DLBP）的概念[3]。拆解线平衡问题旨在简化拆卸活动，使每个工作站消耗的总拆卸时间大致相同，并接近周期时间。

DLBP 问题在提出后逐渐引起了人们的重视。以往对 DLBP 的研究主要集中在线性规划方法和启发式算法上。Altekin 等[4]指出回收产品的可变性可能导致某些拆卸任务的操作失败，并提出了一种基于混合整数规划的方法来解决问题。Riggs 等[5]考虑到 EOL 产品的各种状态所引起的 DLBP 问题的复杂性，采用联合优先图方法处理各种 EOL 状态下的 DLBP 问题。Ilgin 等[6]开发了一种线性物理规划方法，用于处理混合模型 DLBP 目标中的不确定性。Altekin[7]开发了 2 个二阶锥规划模型和 5 个分段线性混合整数规划模型，减少随机拆卸流水线平衡问题的工作站数量，并对这些模型的求解结果进行了比较和分析。Tuncel 等[8]指出，回收产品的质量和结构具有高度的不确定性，不同质量和结构的回收产品的拆卸方法也不同。基于这些考虑，他们提出了一种基于蒙特卡罗的强化学习技术来处理拆解线平衡问题中的随机动力学问题。以线性规划为主的方法能够得到精确解。随着拆解产品零件数量的增加，算法的复杂度以指数级增加，对中大规模的 DLBP 问题实际不可行。元启发式方法在求解多目标 DLBP，特别是求解大规模多目标 DLBP 时非常有效。它们可以在较短的时间内获得接近最优解甚至最优解，因此，越来越多的元启发式算法（如粒子群算法[9]、人工蜂群算法[10]、禁忌搜索算法[11]、变邻域搜索算法[12]）已先后用于求解多目标 DLBP 问题。Kalayci 等[13]提出了模糊 DLBP，将拆卸次数假设为三角模糊数，并采用一种混合离散人工蜂群算法求解该问题。张彪等[14]提出了多目标模糊拆解线平衡问题，利用三角模糊数表征不确定性下拆解线的拆卸次数，并采用改进的人工鱼群算法求解该问题。郑斐峰等[15]首先提出了不给定任务时间概率分布的随机 DLBP 的无分布模型。Bentaha 等[16]针对随机 DLBP 开发了 2 种以拆卸收益最大化和工作量平衡为目标的随机规划模型，并提出了一种结合 L–shaped 算法和蒙特卡罗抽样技术的精确求解方法来求解模型。Ren 等[17]考虑拆卸对环境的影响提出了一种基于 2–Opt 算法的三相方法来解决多目标 DLBP，并与 Avikal 等[18-19]提出的 2 种启发式方法进行了对比。Wang 等[20]提出了一种多目标离散花授粉算法来解决多目标随机双边部分拆解线平衡问题。

上述研究成果促进了智能算法在拆解线平衡问题中的应用。然而，根据 No Free Lunch 理论[21]，上述方法不能解决所有的优化问题。基于当前或新的优化问题的新算法有可能优于解决当前的元启发式算法，因此，仍有必要尝试寻找新的优化算法来解决 DLBP 问题。Fathollahi–Fard 等[22]提出的社会工程算法（Social Engineering Optimizer，SEO）是受社会工程理论概念的启发，有 4 个主要步骤和 3 个待定参数，比传统方法更简单、更智能。该方法首先生成 2 个随机的解决方案，即攻击者和防御者。根据社会工程技术的规则，攻击者获取防御者的信息以达到预期的目标（得到最优值），防御者则进行躲避。当攻击者达到目标时，它会从当前位置寻找下一个受害者（一个新的防御者）。目前还没有研究将 SEO 算法应用于 DLBP 问题。

结合以上文献，提出了一种改进的 SEO 算法。与原来的 SEO 算法不同，使用交换序列而不是拆卸序列作为待优化的个体。为了将原算法改进为能够求解离散优化问题，在算法的攻击阶段定义了 3 个序列操作算子来取代原本的数值计算。

1　拆解线平衡数学模型

针对拆解线平衡问题的特点，建立如下多目标拆解线平衡优化模型[25]：

$$F_{\mathrm{Idle}}=\frac{\sum_{n=1}^{N}\left(T_{\mathrm{b}}-\sum_{m=1}^{M}t_m\cdot x_{mn}\right)}{\sum_{n=1}^{N}T_{\mathrm{b}}} \tag{1}$$

$$F_{\mathrm{Smooth}}=-\frac{1}{\ln N}\left[\frac{T_{\mathrm{b}}-\sum_{m=1}^{M}t_m\cdot x_{mn}}{\sum_{n=1}^{N}\left(T_{\mathrm{b}}-\sum_{m=1}^{M}t_m\cdot x_{mn}\right)}\cdot\ln\left(\frac{T_{\mathrm{b}}-\sum_{m=1}^{M}t_m\cdot x_{mn}}{\sum_{n=1}^{N}\left(T_{\mathrm{b}}-\sum_{m=1}^{M}t_m\cdot x_{mn}\right)}\right)\right] \tag{2}$$

$$F_{\mathrm{Cost}}=\sum_{n=1}^{N}T_{\mathrm{b}}\cdot\max\left\{U_i\mid i\in\left\{i\mid x_{in}=1\right\}\right\}+N\cdot S \tag{3}$$

$$f_{1\min}=F_{\mathrm{Idle}} \tag{4}$$

$$f_{2\min}=1-F_{\mathrm{Smooth}} \tag{5}$$

$$f_{3\min}=F_{\mathrm{Cost}} \tag{6}$$

$$\text{约束：}\ \frac{\sum_{m=1}^{M}t_m}{T_{\mathrm{b}}}\leqslant N\leqslant M \tag{7}$$

$$\sum_{m=1}^{M}t_m\cdot x_{mn}\leqslant T_{\mathrm{b}}\quad n=1,2,\cdots,N \tag{8}$$

$$\sum_{n=1}^{N}x_{mn}=1\quad m=1,2,\cdots,M \tag{9}$$

$$\sum_{\mu=1}^{N}\mu\cdot x_{i\mu}\leqslant\sum_{v=1}^{N}v\cdot x_{jv}\quad\forall\boldsymbol{C}(i,j)=1 \tag{10}$$

式中：M 为拆卸任务总数；N 为工作站总数；T_b 为拆卸节拍，即工作站的工作时间上限；t_m 为第 m 个拆卸任务的操作时间；x_{mn} 表示拆卸任务是否被分配到工作站，若任务 m 被分配到工作站 n，x_{mn}=1；U_i 为第 i 个拆卸任务的单位时间成本；S 为开启一个工作站的固定运行成本；$\boldsymbol{C}$ 为拆卸优先矩阵，若零件 i 必须在零件 j 之前拆卸，$\boldsymbol{C}(i, j)$=1。式（1）表示拆解线的闲置率；式（2）表示拆卸任务安排的平滑度，即各工作站的闲置时间差异；式（3）表示各工作站的单位时间拆卸成本和所有工作站的运行成本之和。

目标函数中，式（4）表示最小化拆解线的闲置率；式（5）表示最小化各工作站的闲置时间差异；式（6）表示最小化拆卸成本。由于式（4）—（6）的量级差异过大，因此生成 Pareto 前沿前要先进行归一化。

约束条件中，式（7）表示开启的工作站数不小于理论需要的工作站数，且不大于拆卸任务总数；式（8）表示每个工作站执行拆卸任务的时间不得大于拆卸节拍；式（9）表示每个拆卸任务只能执行一次；式（10）表示拆卸任务的执行必须满足优先约束条件。

2 改进的社会工程算法

社会工程算法的出现是受到社会工程理论的启发。社会工程（Social Engineering, SE）是指通过某种技术获取人们的个人信息，诱导或强迫人们满足社会工程师要求的行为[23]。改进的社会工程算法引入了交换子和交换序列对其进行离散化改造。

2.1 交换子和交换序列

假设由 n 个零件组成的机构的拆卸序列为 $S = (a_i), i = 1, 2, \cdots, n$。令 a_j 和 a_k 为 S 中的两步拆卸操作，定义 $O_S(a_j, a_k)$为交换子，表示交换拆卸操作 a_j 和 a_k。交换序列为多个交换子组成的序列。由 l 个交换子形成的交换序列表示为 $S_S = (O_{S1}, O_{S2}, \cdots, O_{Sl})$。当交换序列作用于拆卸序列时，交换序列中的所有交换子依次作用于拆卸序列。为了使 SEO 能够解决离散优化问题，提出 3 个操作算子[24]。

1）加法算子 ⊕。设 S_{S1} 和 S_{S2} 是长度均为 n 的交换序列，加法算子的具体操作是将 2 个交换序列叠加在一起，表示为 $S_S = S_{S1} \oplus S_{S2}$。$S_S$ 的长度为 $2n$。

2）乱序算子 ⊕。假设 S_S 是长度为 n 的交换序列，β[0,1]是 SEO 的输入参数。设 μ= ceil[$n \times U(0, 1)$], v = ceil[$n \times \sin(\beta)$]，乱序算子的具体操作为重新排列拆卸序列中的第 μ 位到第 $\mu + v$ 位，若 $\mu + v \geqslant n$，则重新排列第 μ 位到最后一位，表示为 $S_S' = S_S$ [$U(0, 1) \times \sin(\beta)$]。应特别注意，重新排序拆卸序列里第 μ 位到 $\mu + v$ 位以外的其他拆卸任务的操作表示为 $S_S' = S_S \oplus [1 - U(0, 1) \times \sin(\beta)]$。

3）平均算子◎。设 S_{S1} 和 S_{S2} 是长度均为 n 的交换序列，平均算子的具体操作为对 S_{S1} 和 S_{S2} 的每一位按照相同的概率随机选择一个组成新的交换序列，表示为 $S_S = S_{S1}$ ◎ S_{S2}。S_S 的长度为 n。

2.2 生成可行解

社会工程算法通过模拟社会工程的步骤来实现搜索。社会工程算法分为 4 个主要步骤，第 1 步是初始化。社会工程算法需要生成 2 个初始的可行解，分别称为攻击者和防御者。结合 SEO 的基本步骤，可行解的生成方法如下：

1）根据约束关系生成优先矩阵 $\boldsymbol{C}$。

2）寻找矩阵中全为 0 的行，随机选择其中一行，将该行代表的零件拆卸；将矩阵中该行对应的列全部置 0，并移除这一行。

3）重复上述步骤，直到优先矩阵为空，生成可行的拆卸序列。

4）生成 3 条拆卸序列，按照适应度排序，将适应度居中的作为基本序列，根据文献[24]给出的方法，生成 2 条交换序列，即为可行解。将适应度最好的拆卸序列生成的交换序列作为攻击者，另一个交换序列为防御者，见图 1。

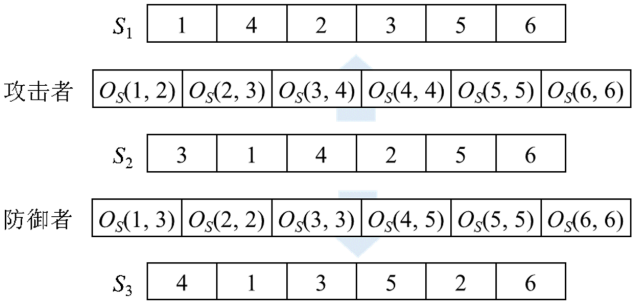

图 1 生成攻击者和防御者

Fig.1 Attackers and defenders generation

2.3 训练和再训练

如图 2 所示，设交换序列的长度为 n，预先设置 1 到 n 之间的一个随机整数 α。然后随机选择攻击者中的 α 个交换子，替换防御者中相应的位置，生成 α

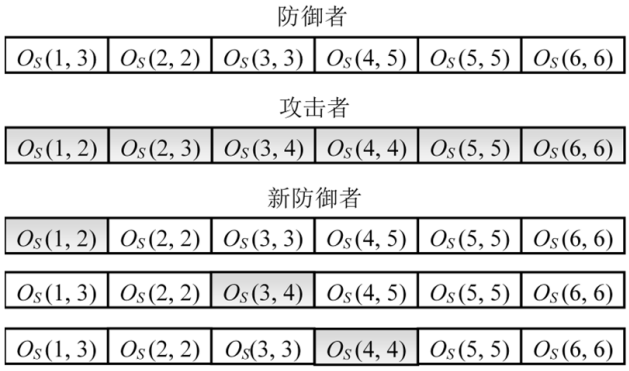

图 2 训练和再训练

Fig.2 Training and retraining

个新解。如果新解中的最优解比防御者好，则用该解代替防御者。

2.4　发现攻击

为了发现攻击，社会工程算法定义了 4 种攻击方式，分别是获取信息（Obtaining）、钓鱼（Phishing）、调虎离山（Diversion Theft）和假托（Pretext）。这一步的操作为从 4 种攻击方式中随机选择一种方式。在攻击操作中只用到一个参数 β 作为搜索时的输入变量。4 种攻击方式如下。其中，防御者表示为 p_{dnew} 和 p_{dold}，攻击者表示为 p_a。

$$p_{\mathrm{dinew}}=\{p_{\mathrm{dold}}\otimes[1-\sin\beta\times U(0,1)]\}\oplus\{(p_{\mathrm{dold}}\odot p_{\mathrm{a}})\otimes[\sin\beta\times U(0,1)]\} \tag{11}$$

$$p^{1}_{\mathrm{dnew}}=\{p_{\mathrm{a}}\otimes[1-\sin\beta\times U(0,1)]\}\oplus\{(p_{\mathrm{dold}}\odot p_{\mathrm{a}})\otimes[\sin\beta\times U(0,1)]\} \tag{12}$$

$$p^{2}_{\mathrm{dnew}}=\{p_{\mathrm{dold}}\otimes[1-\cos(\beta)\times U(0,1)]\}\oplus\{(p_{\mathrm{dold}}\odot p_{\mathrm{a}})\otimes[\cos(\beta)\times U(0,1)]\} \tag{13}$$

$$p_{\mathrm{dnew}}=\{p_{\mathrm{dold}}\otimes[1-\sin\beta\times U(0,1)]\}\oplus\left\{\left[\left(p_{\mathrm{dold}}\odot\left(p_{\mathrm{a}}\otimes(U(0,1)\times\cos(\beta))\right)\right)\right]\otimes[\sin\beta\times U(0,1)]\right\} \tag{14}$$

$$p_{\mathrm{dnew}}=\{[p_{\mathrm{dold}}\otimes(\cos(\beta)\times U(0,1))]\otimes[1-\sin\beta\times U(0,1)]\}\oplus\left\{\left[\left(p_{\mathrm{dold}}\otimes(U(0,1)\times\cos(\beta))\right)\odot p_{\mathrm{a}}\right]\otimes[\sin\beta\times U(0,1)]\right\}n \tag{15}$$

2.5　验证可拆卸性

通过上述步骤生成的新防御者需要验证其可拆卸性。从序列的第 1 个位置开始，如果该位置代表的拆卸任务可以执行，则检查下一个任务的可执行性；如果无法执行，则随机选择一个可执行的任务代替，然后检查下一个任务的可执行性。重复上述操作，直到最后一个任务满足拆卸要求为止。

2.6　响应攻击

令 k 为攻击次数。每次攻击后，评估新防御者的适应度并与旧防御者进行比较，然后选择适应度更好的作为防御者。如果防御者的适应度优于攻击者，则两者的身份将互换。当攻击持续 k 次后，攻击停止，再生成新的防御者。重复上述步骤，直到达到最大迭代次数或算法停止。

2.7　算法流程

改进的 SEO 算法流程如下所示。

Step1：输入问题信息：优先矩阵 $\boldsymbol{C}$，拆卸任务信息，拆卸任务 i 的执行时间 t_i 和单位时间成本 U_i，工作站的固定运行成本 S，拆卸节拍 T_{b}。

Step2：设置算法参数：最大迭代次数 I_{max}，攻击次数 k，再训练率 α，攻击参数 β。

Step3：设置迭代次数 $gen=1$，生成最初的攻击者和防御者。

Step4：对防御者进行训练和再训练，生成 α 个新解，选择最优的作为防御者。

Step5：设置攻击次数 $a=1$。

Step6：从 4 种攻击方法中随机选择一种生成新防御者，若新防御者的适应度更好，替换原防御者。

Step7：检查拆卸序列的可拆卸性，修正拆卸序列。

Step8：判断防御者是否优于攻击者，若优于，则防御者成为新的攻击者，否则不做改变。

Step9：判断攻击次数是否达到上限值，若达到，则停止攻击，算法转至 Step10；否则转至 Step6。

Step10：生成新的防御者。

Step11：判断算法终止条件。若 $gen=I_{\max}$ 或满足预设终止条件，算法终止，输出攻击者；否则转至 Step4。

3　算法验证

为验证算法的有效性以及将文中所提模型和算法推广于实例应用,基于处理器为 Intel(R) Core(TM) i5–7400 CPU @ 3.00GHz，8.00GB 内存的计算机，Win10 系统下采用 Matlab R2018a 开发了所提算法的实验程序。

以文献[25]中的高速电子套结机为例，确定最大迭代次数 $I_{\max}$=500，攻击次数 k=30，再训练率 α=0.5，攻击参数 β=5π/18，拆卸节拍 T_{b}=600，运行一次的 Pareto 前沿，见图 3，前沿中的求解结果见表 1。由于 f_1、f_2 和 f_3 的量级差异过大，因此对其归一化生成 f_4、f_5 和 f_6。

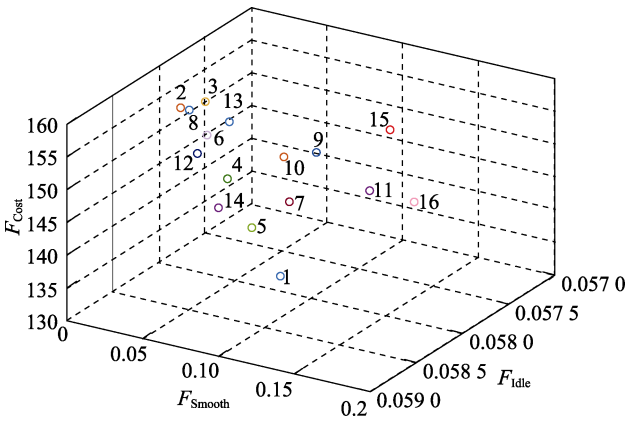

图 3　Pareto 前沿

Fig.3 Pareto frontier

为了验证改进 SEO 算法相比其他算法的有效性，选择蚁群算法[25]和遗传算法[26]与之对比，确定最大迭代次数 $I_{\max}$=200，攻击次数 k=10（对应遗传算法和蚁群算法的种群数为 10），再训练率 α=0.5，攻击参数 β=5π/18，拆卸节拍 T_{b}=600，每个算法运行 30 次。为了验证所提出的 SEO 的有效性，使用非支配排序遗传算法–II（NSGA–II）、蚁群优化算法（Ant Colony

表 1 拆解线平衡 Pareto 最优解集

Tab.1 Optimal solution set of Pareto for disassembly line balancing

序号	线平衡方案	f_1	f_2	f_3	归一化后		
					f_4	f_5	f_6
1	[32, 21, 33, 36, 16, 37, 42] → [29, 47, 46, 44, 14, 4, 39, 43, 31, 52, 3, 2] → [25, 45, 26, 1] → [12, 30, 9] → [34, 11, 23, 38, 15, 27, 49, 18, 17, 35] → [28, 51, 40, 19, 22, 41, 7, 24, 50, 48, 8, 13] → [20, 10, 6, 5]	0.058	0.074	131.0	0.067	1	0
2	[18, 37, 33, 42, 21, 36] → [4, 3, 32, 26, 31, 52, 47] → [1, 9, 12] → [25, 34, 35, 45, 22, 29, 39, 2, 46] → [30, 16, 43, 24, 49, 51, 7, 44, 14, 40, 11] → [23, 38, 41, 17, 15, 50, 27, 19, 28, 48, 8, 13] → [20, 10, 6, 5]	0.058	0.008	151.2	0.167	0	0.924
3	[37, 33, 26, 21, 32] → [29, 36, 14, 42, 4, 39, 11, 18, 31, 52] → [23, 25, 45, 47, 3, 38, 17, 2, 46] → [1, 15, 44, 12, 30] → [35, 24, 9, 49, 16, 27, 28] → [34, 43, 51, 40, 8, 19, 50, 41, 48, 22, 7, 13] → [20, 10, 5, 6]	0.058	0.022	155.0	0	0.248	1
4	[21, 37, 39, 36, 18, 42, 33] → [47, 3, 29, 4, 32, 16, 26, 46] → [44, 30, 14, 1, 12] → [43, 11, 31, 23, 35, 38, 45, 15, 52, 28, 24, 8] → [2, 25, 9, 49, 17] → [51, 27, 19, 34, 40, 41, 50, 22, 7, 13, 48] → [20, 10, 6, 5]	0.058	0.055	144.0	0	0.47	0.541
5	[33, 36, 21, 26] → [29, 32, 4, 37, 16, 42, 3, 18, 47, 14] → [11, 31, 1, 45, 2, 46, 39, 30, 43] → [9, 12, 25] → [44, 23, 52, 35, 38, 15, 49, 27, 34, 22] → [28, 51, 19, 7, 17, 24, 40, 8, 41, 50, 13, 48] → [10, 6, 20, 5]	0.058	0.026	137.4	1	0.714	0.266
6	[21, 29, 26, 36, 14, 4] → [11, 33, 32, 42, 16, 31, 25] → [37, 23, 47, 52, 39, 3, 18, 2, 38, 15, 17, 46] → [30, 44, 49, 43, 45, 28, 1] → [51, 27, 9, 12] → [19, 35, 40, 41, 34, 24, 22, 50, 48, 7, 8, 13] → [20, 10, 6, 5]	0.058	0.081	150.0	0.5	0.265	0.789

Optimization，ACO）、人工蜂群（Artificial Bee Colony，ABC）的模型来解决上面给出的图例。让种群数量 m 分别为 5 和 30，其他参数与上节相同。运行结果通过帕累托解决方案的数量（NPS）、平均理想距离（MID）、增强的反转世代距离（IGD–NS）和超体积（HV）等 4 个指标进行评估。其中，NPS 反映算法的效率，MID 反映算法的部分多样性，IGD–NS 表示真实前沿的每个点距已知前沿的最近欧式距离，反映了算法的收敛性和多样性，HV 表示所有不同的 Pareto 前沿在解空间里的总体积，解决了算法在 3 个以上指标时 Pareto 前沿优势不明显的问题，反映了算法的收敛性和部分多样性。30 次运行后的结果见表 2 和图 4。

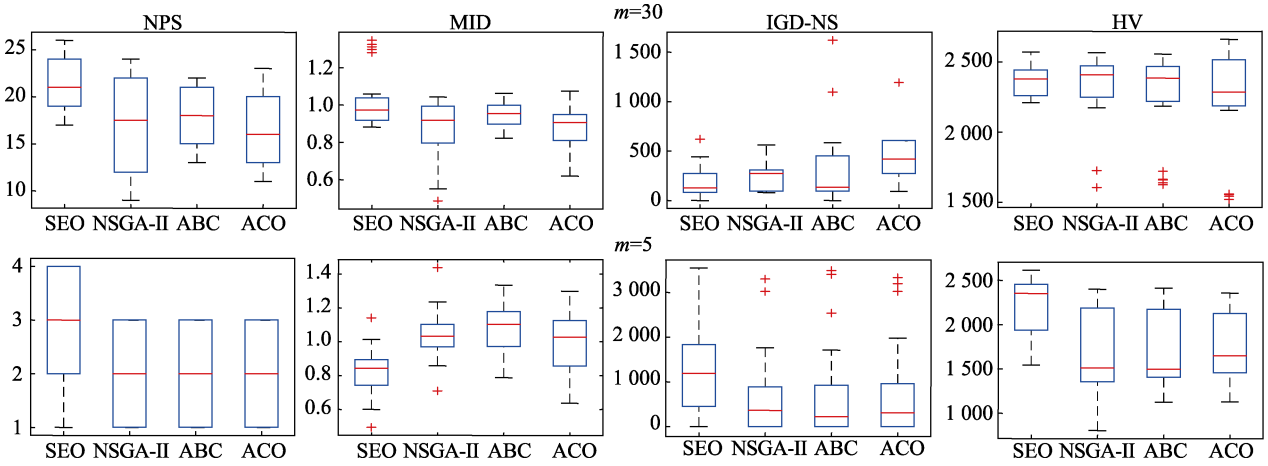

图 4 SEO、NSGA–II、ABC 和 ACO 的不同指标的箱线图（“+”表示数据中的异常值）

Fig.4 Boxplot graph of different metrics for SEO, NSGA-II, ABC and ACO ("+" indicates an outlier in data)

表 2 运行 30 次的各评价指标的平均值

Tab.2 Average value of evaluation metrics for 30 runs

Algorithm	m = 5				m = 30			
	NPS 值	MID 值	IGD-NS 值	HV 值	NPS 值	MID 值	IGD-NS 值	HV 值
SEO	2.87	0.823	1464.9	2 191	21.67	1.107	223.9	2 362
NSGA–II	1.97	1.039	625.5	1 701	16.87	0.981	500.0	2 249
ABC	1.87	1.083	677.1	1 670	17.20	0.977	247.6	2 351
ACO	1.93	0.993	686.7	1 741	18.00	1.035	430.5	2 246

从 4 图中可以看出，当种群数量达到一定程度时，几种方法的性能相似，但 SEO 得到的帕累托前沿分布较广最优解较多。当种群数量较少时，NSGA–II、ACO 和 ABC 求得的 Pareto 前沿较差且分散度较高，而 SEO 能够在攻击次数和迭代次数较低的情况下保留较好的优化精度和分散度。具有良好的收敛性。

4　结语

文中提出了一种基于改进社会工程算法的多目标拆解线平衡优化方法。该算法的主要创新之处在于引入了交换子和交换序列来解决离散拆解线平衡优化问题。在此基础上，定义了基于交换序列的社会工程算法操作算子。使用一个高速电子套结机的案例验证了所提出方法的有效性，使用 4 个指标评价将提出的社会工程算法与 3 种经典算法进行对比，显示了所提出方法的优越性。与经典算法相比，该算法可以有效地解决拆卸序列规划问题。

在实际生产中，拆卸过程中的许多要素是不确定的，如拆卸时间[27]、零件的质量和需求[28-29]、工人的操作水平[30]等。在 DLBP 中加入物联网和智能技术的概念是一种解决方案[31]。在接下来的研究中，将会考虑兼顾模糊和不确定性的 DSP 和 DLBP 组合优化问题。

参考文献：

[1] 朱兴涛，张则强，朱勋梦，等. 求解多目标拆卸线平衡问题的一种蚁群算法[J]. 中国机械工程，2014, 25(8): 1075-1079.
ZHU Xing-tao, ZHANG Ze-qiang, ZHU Xun-meng, et al. An Ant Colony Optimization Algorithm for Multi-Objective Disassembly Line Balancing Problem[J]. China Mechanical Engineering, 2014, 25(8): 1075-1079.

[2] XIA Kai, GAO Liang, LI Wei-dong, et al. Disassembly Sequence Planning Using a Simplified Teaching-Learning-Based Optimization Algorithm[J]. Advanced Engineering Informatics, 2014, 28(4): 518-527.

[3] KAZANCOGLU Y, OZTURKOGLU Y. Integrated Framework of Disassembly Line Balancing with Green and Business Objectives Using a Mixed MCDM[J]. Journal of Cleaner Production, 2018, 191: 179-191.

[4] ALTEKIN F T, AKKAN C. Task-Failure-Driven Rebalancing of Disassembly Lines[J]. International Journal of Production Research, 2012, 50(18): 4955-4976.

[5] RIGGS R J, BATTAÏA O, HU S J. Disassembly Line Balancing under High Variety of End of Life States Using a Joint Precedence Graph Approach[J]. Journal of Manufacturing Systems, 2015, 37: 638-648.

[6] ILGIN M A, AKÇAY H, ARAZ C. Disassembly Line Balancing Using Linear Physical Programming[J]. International Journal of Production Research, 2017, 55(20): 6108-6119.

[7] ALTEKIN F T. A Comparison of Piecewise Linear Programming Formulations for Stochastic Disassembly Line Balancing[J]. International Journal of Production Research, 2017, 55(24): 7412-7434.

[8] TUNCEL E, ZEID A, KAMARTHI S. Solving Large Scale Disassembly Line Balancing Problem with Uncertainty Using Reinforcement Learning[J]. Journal of Intelligent Manufacturing, 2014, 25(4): 647-659.

[9] KALAYCI C B, GUPTA S M. A Particle Swarm Optimization Algorithm with Neighborhood-Based Mutation for Sequence-Dependent Disassembly Line Balancing Problem[J]. The International Journal of Advanced Manufacturing Technology, 2013, 69(1): 197-209.

[10] 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.

[11] 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.

[12] KALAYCI C B, HANCILAR A, GUNGOR A, et al. Multi-Objective Fuzzy Disassembly Line Balancing Using a Hybrid Discrete Artificial Bee Colony Algorithm[J]. Journal of Manufacturing Systems, 2015, 37: 672-682.

[13] 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.

[14] ZHANG Biao, PAN Quan-ke, GAO Liang, et al. An Effective Modified Migrating Birds Optimization for Hybrid Flowshop Scheduling Problem with Lot Streaming[J]. Applied Soft Computing, 2017, 52: 14-27.

[15] ZHENG Fei-feng, HE Jun-kai, CHU Feng, et al. A New Distribution-Free Model for Disassembly Line Balancing Problem with Stochastic Task Processing Times[J]. International Journal of Production Research, 2018, 56(24): 7341-7353.

[16] BENTAHA M L, DOLGUI A, BATTAÏA O, et al. Profit-Oriented Partial Disassembly Line Design: Dealing with Hazardous Parts and Task Processing Times Uncertainty[J]. International Journal of Production Research, 2018, 56(24): 7220-7242.

[17] REN Ya-ping, ZHANG Chao-yong, ZHAO Fu, et al. Disassembly Line Balancing Problem Using Interdependent Weights-Based Multi-Criteria Decision Making and 2-Optimal Algorithm[J]. Journal of Cleaner Production, 2018, 174: 1475-1486.

[18] AVIKAL S, JAIN R, MISHRA P K. A Kano Model, AHP and M-TOPSIS Method-Based Technique for Disassembly Line Balancing under Fuzzy Environment[J]. Applied Soft Computing, 2014, 25: 519-529.

[19] AVIKAL S, MISHRA P K, JAIN R. A Fuzzy AHP and PROMETHEE Method-Based Heuristic for Disassembly Line Balancing Problems[J]. International Journal of

Production Research, 2014, 52(5): 1306-1317.
[20] WANG Wen-yuan, MO D Y, WANG Yue, et al. Assessing the Cost Structure of Component Reuse in a Product Family for Remanufacturing[J]. Journal of Intelligent Manufacturing, 2019, 30(2): 575-587.
[21] WOLPERT D H, MACREADY W G. No Free Lunch Theorems for Optimization[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.
[22] FATHOLLAHI-FARD A M, HAJIAGHAEI-KESHTELI M, TAVAKKOLI-MOGHADDAM R. The Social Engineering Optimizer (SEO)[J]. Engineering Applications of Artificial Intelligence, 2018, 72: 267-293.
[23] KROMBHOLZ K, HOBEL H, HUBER M, et al. Advanced Social Engineering Attacks[J]. Journal of Information Security and Applications, 2015, 22: 113-122.
[24] ZHANG Cheng, FATHOLLAHI-FARD A M, LI Jianyong, et al. Disassembly Sequence Planning for Intelligent Manufacturing Using Social Engineering Optimizer[J]. Symmetry, 2021, 13(4): 663.
[25] 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.
DING Li-ping, TAN Jian-rong, FENG Yi-xiong, et al. Multiobjective Optimization for Disassembly Line Balancing Based on Pareto Ant Colony Algorithm[J]. Computer Integrated Manufacturing Systems, 2009, 15(7): 1406-1413.
[26] 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.
[27] HE Jun-kai, CHU Feng, DOLGUI A, et al. Integrated Stochastic Disassembly Line Balancing and Planning Problem with Machine Specificity[J]. International Journal of Production Research, 2022, 60(5): 1688-1708.
[28] LIU Ming, LIU Xin, CHU Feng, et al. Robust Disassembly Line Balancing with Ambiguous Task Processing Times[J]. International Journal of Production Research, 2020, 58(19): 5806-5835.
[29] LIU Ming, LIU Xin, CHU Feng, et al. An Exact Method for Disassembly Line Balancing Problem with Limited Distributional Information[J]. International Journal of Production Research, 2021, 59(3): 665-682.
[30] LIU Xin, CHU Feng, ZHENG Fei-feng, et al. Distributionally Robust and Risk-Averse Optimisation for the Stochastic Multi-Product Disassembly Line Balancing Problem with Workforce Assignment[J]. International Journal of Production Research, 2022, 60(6): 1973-1991.
[31] MOOSAVI J, NAENI L M, FATHOLLAHI-FARD A M, et al. Blockchain in Supply Chain Management: A Review, Bibliometric, and Network Analysis[J]. Environmental Science and Pollution Research, 2021: 1-15.

责任编辑：陈作