Multi-Objective Ant Lion Optimizer for Stochastic Robotic Disassembly Line Balancing Problem Subject to Resource Constraints

ChengShun Dong¹, Peisheng Liu*, Xi Wang Guo¹, Liang Qi², Shujin Qin³, Gongdan Xu⁴

¹ College of Computer and Communication Engineering, Liaoning Petrochemical University, Fushun, 113001, China
² Department of Computer Intelligent Sci. & Tech., Shandong University of Science and Technology, Qingdao, 266590, China
³ College of Economics and Management, Shangqiu Normal University, Shangqiu, 476000, China
⁴ Institute of System Engineering, Macao University of Science and Technology, Macao, 999078, China

Corresponding author’s e-mail: 406771217@qq.com, lps_1981@sina.com, x.w.guo@163.com, qiliangsdkd@163.com, sjchin@vip.126.com, xgd222@163.com

Abstract. With the development of manufacturing and society, a large amount of end-of-life products are generated but with reuse value. Therefore, it is of great significance to dismantle and reuse them. A robotic linear disassembly line plays an important role and promotes the disassembly efficiency in a disassembly process. In this work, we establish a mathematical model of a linear robotic disassembly line balancing problem, which requires the maximization of profit and minimization of energy consumption. Then, an improved ant lion optimizer is proposed to optimize the objectives of the multi-objective robotic disassembly line balancing problem. The experimental results show that the multi-objective ant lion optimizer has good performance, and the diversity and convergence are better than its peers.

1. Introduction

The recycling of wasted products has always been a concern of the society [1-2]. Our goals are to achieve the optimal reuse value [3]. A Disassembly Line Balancing Problem (DLBP) is an NP-hard problem. Some heuristic algorithms are applied to solve multi-objective DLBPs [4] including gray wolf algorithm [5], artificial bee colony algorithm [6], and cuckoo search [7]. But these algorithms also have some shortcomings. To improve the speed of some algorithms, the diversity of the solutions is sacrificed in the iterative process [8].

Ant lion algorithm is a kind of swarm intelligence technology that comes from ant lion hunting ants [8, 9], which brings us the inspiration like many heuristic algorithms such as Genetic Algorithm [10], Cuckoo Search and Ant Colony Optimizer (ACO).

In this work, we use ant lion algorithm to solve the multi-objective robotic disassembly-line-balancing-problem (MRD) under the premise of resource constraints, failure risk, and minimum energy consumption. This paper has three contributions:
1) The mathematical model of MRD is established to meet an actual disassembly process. It achieves the objectives of minimizing the energy consumption and maximizing the disassembly profit by taking the failure cost, resource constraints, and the priority relationship of the components into consideration.

2) A multi-objective ant lion algorithm (MALA) is designed to solve MRD, and an archive mechanism is used to ensure the diversity and convergence of the proposed algorithm.

3) Multi-objective Evolutionary Algorithm Based on Decomposition (MOEAD) and Non-dominated sorting genetic algorithm II (NSGA-II) are compared with the proposed algorithm. The index analysis of the solution shows that Mala algorithm is better.

2. Problem Statement

2.1 Problem Description

In this paper, we make some assumptions: 1) For each type of robots, the average disassembly time of each disassembly task and the time of conversion task are known. 2) Disassembly cost and energy consumption per time unit of each type of robots are known. 3) The conversion cost per time unit and the conversion energy consumption per time unit when the workstation completes the disassembly tasks are given. 4) The supply of EOL products is unlimited. 5) Only robots are responsible for performing disassembly tasks. A robot can only perform one disassembly task at a time, and each operation must be handled by one robot. 6) AND/OR graphs of multiple EOL products are known. 7) When the disassembly task needs to be operated by a specific robot, the type of robots should be informed to the workstation.

We use an AND/OR graph to describe the priority relationship of disassembly products. We select the mouse to simulate the disassembly line. The disassembly sequence of the linear line disassembly line is shown in Fig. 2. A disassembly sequence 1→4→8→11→13 is executed, task 1 needs robot 1, and robot 3 needs resource 1 and resource 3 to execute the disassembly, and component 2 and component 11 are obtained. As shown in Figure 1.

![Figure 1. Task scheduling of robot linear disassembly line](image1)

![Figure 2. AND/OR graph of a mouse](image2)
2.2 Notation Definition

1) $j$: index of a component, $j = 1, 2, 3, ..., J$, where $j$ is the number of components.

2) $i$, $k$: index of disassembly task, $i, k = 1, 2, 3, ..., I, I$ is the number of disassembly tasks and 0 is a virtual task.

3) $l$, $m$: index of workstations, $l, m = 1, 2, 3, ..., M$, $M$ is the number of workstations.

4) $r$: index of required resources, $r = 1, 2, 3, ..., R$, $R$ represents the number of resource types require.

5) $h$: robot number index, $h = 1, 2, 3, ..., H$, $H$ is the number of robot types.

6) $A$: Priority relation matrix of AND/OR graph.

7) $a_{ik}$: an element in the $i$-th row and $k$-th column of $A$.

8) $B$: disassembly matrix of a AND/OR graph.

9) $b_{ij}$: an element in the $j$-th row and $i$-th column of $B$.

10) $c_{ih}$: disassembly cost per time unit of performing task $i$ by robot $h$.

11) $c_{ik}$: conversion cost per time unit for robot $h$ to execute task $k$ after completing task $i$.

12) $c_{l}$: Disassembly cost of the $l$-th workstation.

13) $D$: a requirement resource matrix for disassembly products.

14) $d_{ir}$: an element in the $i$-th row and $r$-th column of $D$.

15) $e_{ih}$: when robot $h$ performs task $i$, the energy consumption per time unit is required.

16) $e_{ik}$: when the robot completes task $i$ and starts to execute task $k$, the required conversion sets the energy consumption per time unit of task $k$.

17) $e_{l}$: energy consumption generated by using the $l$-th workstation.

18) $F$: the maximum allowable cost of disassembly failure.

19) $E$: the maximum number of resources of a disassembly process.

20) $q_{ik}^h$: the failure probability of robot $h$ executing task $k$ after executing task $i$.

21) $T_l$: cycle time of the $l$-th workstation.

22) $t_i^h$: the time required for task $i$ to complete disassembly on robot $h$.

23) $t_k^h$: the robot $h$ setting time when the task $i$ is executed after executing $k$.

24) $v_j$: reuse value of subassembly $j$.

25) $\theta$: the probability that the failure cost of disassembly should not exceed the minimum probability of the maximum failure cost allowed.

26) $p_{ir}$: disassembly task $i$ needs to use the $r$-th resource.

Decision variables:

\[ x_{ih} = \begin{cases} 
1, & \text{if disassembly task } i \text{ is performed by robot } h \\
0, & \text{otherwise}
\end{cases} \]

\[ y_{ik}^h = \begin{cases} 
1, & \text{if task } k \text{ is performed immediately after} \\
0, & \text{task } i \text{ by robot } h \text{ and assigned} \\
& \text{to the } l \text{ th workstation} \\
& \text{otherwise}
\end{cases} \]

\[ z_l = \begin{cases} 
1, & \text{if the } l \text{ th workstation is used} \\
0, & \text{otherwise}
\end{cases} \]

\[ u_{ii} = \begin{cases} 
1, & \text{if execute task } i \text{ on the } l \text{ th workstation} \\
0, & \text{otherwise}
\end{cases} \]
2.3 Mathematical model

Objective function (1) is for maximizing expect profit of disassembly product. (2) is to minimize expect disassembly energy consumption. (3) guarantees that each task is executed once. (4) ensures that a disassembly task cannot be performed more than once. (5) ensures that disassembly tasks are assigned when the workstation is turned on. (6) ensures that the disassembly sequence satisfies the conflicted relation constraints. (7) ensures that the disassembly sequence satisfies the priority relation matrix (8) represents that a disassembly sequence satisfies the resource constraints in a disassembly process. (9) means that the disassembly time of each workstation cannot exceed the cycle time of workstations. In (10), when disassembly failure occurs, the probability that the failure cost of
disassembly should not exceed the minimum probability of the maximum failure cost allowed. (11) ensures that the workstation performs tasks. (12) indicates the range of some decision variables.

3. Proposed Algorithm

3.1 Population initialization
We can describe the solution of disassembly line as \( \pi = (\pi', \pi'') \), where \( \pi' = \{h_1, h_2, \ldots, h_J\} \) represents the disassembly sequence set of strings, and \( h_j \) represents the disassembly index at position \( J \). \( \pi'' = \{q_1, q_2, \ldots, q_J\} \) is a binary string representing whether the task at this position is executed. If \( q_j = 0 \), the task in the \( j \)-th position in \( \pi' \) cannot be performed; otherwise, \( q_j = 1 \). And initialize the first generation population coding information according to the case information.

3.2 Position-based mutation operation
Before the completion of the iteration, the population needs mutation operation to maintain the diversity of the population. In the mutation operation, a random number is generated to perform a mutation operation, exchange the values of two genes on \( \pi_v' \), reverse the values at a certain position of \( \pi_v'' \), or both, as shown in Algorithm 1.

3.3 Population iteration
Ants are randomly assigned to ant lions, and those with high fitness become ant lions. The elite ant lion is selected by a fitness value, which represents the optimal solution. If the archive is full, we delete some individuals by roulette to store the elite ant lions with better fitness until the maximum number of iterations is met. The elite ant lion represents the optimal solution.

4. Experimental Results and Analysis

4.1 Case study
We verify the effectiveness and feasibility of the algorithm by a copying machine and washing machine. We use the other two algorithms, NSGA-II and MOEAD, to verify the superiority of MALA. The following table shows the performance of the algorithm solution.
4.2 Analysis of experimental results

We use GD(N) index to evaluate the convergence of the algorithm, IGD(N) index evaluates the diversity of algorithms, And Epsilon (N) to evaluate the degree of approximation. Hypervolume to reflect the comprehensive performance of the algorithm. As shown in Table 1 and Table 2.

| Algorithm | MOEA | NSGA-II | MALA |
|-----------|------|---------|------|
| Hypervolume | mean | variance | t-test | mean | variance | t-test | mean | variance | t-test |
| GD(N)     | 0.0150 | 9.85014E-06 | - | 0.0201 | 6.17124E-06 | + | 0.0124 | 3.88242E-06 | null |
| IGD+      | 0.0310 | 7.30633E-06 | - | 0.0191 | 7.30777E-06 | + | 0.0162 | 9.89154E-05 | null |
| Epsilon(N)| 0.1452 | 0.0002 | - | 0.1388 | 0.0016 | + | 0.0884 | 0.0027 | null |

| Algorithm | MOEA | NSGA-II | MALA |
|-----------|------|---------|------|
| Hypervolume | mean | variance | t-test | mean | variance | t-test | mean | variance | t-test |
| GD(N)     | 0.0126 | 6.41135E-05 | + | 0.0107 | 3.90409E-06 | + | 0.0073 | 5.56143E-06 | null |
| IGD+      | 0.01487 | 3.58072E-05 | + | 0.0123 | 9.92322E-06 | + | 0.0047 | 1.80414E-06 | null |
| Epsilon(N)| 0.1608 | 0.0026 | + | 0.1417 | 0.0007 | + | 0.0700 | 0.0011 | null |

5. Conclusion

Aiming at the linear DLBP with multi-objective constraints, the single objective version of ant lion algorithm is improved. The MALA algorithm is designed to maximize the disassembly profit and minimize the disassembly energy consumption. The feasible solution is calculated from the actual disassembly case. The superiority of Mala is verified by comparison with NSGA-II and MOEAD. In the future, we will use more advanced operators to optimize MALA, so as to better solve the problems in the field of disassembly.

Acknowledgment

This work is supported in part by Liaoning Province Education Department Scientific Research Foundation of China under Grant No. L2019027; Liaoning Revitalization Talents Program under Grant No. XLYC1907166; The Natural Science Foundation of Shandong Province under Grant ZR2019BF004; National Natural Science Foundation of China (61573089).Archival Science and Technology Project of Liaoning Province, No.2021-B-004.

References

[1] X. W. Guo, S. X. Liu, M. C. Zhou, and G. D. Tian, “Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets,” IEEE Trans. Cybern., vol. 46, no. 11, pp. 2435-2446, 2016.
[2] Y. P. Fu, M. C. Zhou, X. W Guo, and L. Qi, “Artificial-molecule-based chemical reaction optimization for flow shop scheduling problem with deteriorating and learning effects,” IEEE Access, vol. 7, pp. 53429-53440, 2019.
[3] K. Wegener, W. H. Chen, F. Dietrich, K. Dröder, and S. Kara. “Robot assisted disassembly for the recycling of electric vehicle batteries,” Procedia CIRP, pp. 716-721, 2015, doi:10.1016/j.procir.2015.02.051.
[4] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” ACM Comput. Surv., vol. 35, no. 3, pp. 268–308, Sep. 2003. [Online]. Available: http://doi.acm.org/10.1145/937503.937505
[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Advances in engineering software, vol. 69, pp. 46–61, 2014
[6] C. Kalayci, B., Gupta, S. M., & Nakashima, K. (2011). Bees colony intelligence in solving disassembly line balancing problem. In Proceedings of the 2011 Asian conference of management science and applications (ACMSA2011) (pp. 34–41). Sanya, Hainan, China.
[7] J. Zhao, S.X. Liu, M.C. Zhou, X.W. Guo, L. Qi. Modified Cuckoo Search Algorithm to Solve Economic Power Dispatch Optimization Problems [J], IEEE/CAA Journal of Automatic Sinica, 5(4): 794-806, 2018.
[8] M, Seyedali, P. Jangir, and S. Saremi. "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems." Applied Intelligence, vol. 46, no. 1, pp. 79-95, 2017.
[9] M., Seyedali. "The ant lion optimizer." Advances in engineering software, vol. 83, pp. 80-98, 2015.
[10] S. M. McGovern and S. M. Gupta, “A balancing method and genetic algorithm for disassembly line balancing,” Eur. J. Oper. Res., vol. 179, no. 3, pp. 692-708, 2007.