Multi-objective discrete strength pareto evolutionary algorithm II for multiple-product partial U-shaped disassembly line balancing problem

FaYang Lu¹, Peisheng Liu¹, Liang Qi², Shujin Qin³, Gongdan Xu⁴, Zhiying Xu⁵

¹ College of Computer and Communication Engineering, Liaoning Petrochemical University, Fushun, 113001, China
² Department of 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, China
⁵ The Department of Biochemistry Engineering at Chaoyang Teachers College, Chaoyang, 122000, P R. China.

Corresponding author’s e-mail : 1206161829@qq.com, lps_1981@sina.com, qiliangsdkd@163.com, sjchin@vip.126.com, xgd222@163.com, lfy6965@163.com.

Abstract. In disassembly industry, the disassembly efficiency closely depends on a reasonable disassembly mode and an efficient disassembly line. In the current manufacturing, the profit of multi-product partial U-shaped disassembly-line-balancing (MPUD) is relatively low and the energy consumption is high. According to the characteristics of the MPUD problem, a mathematical model for maximizing profit and minimizing energy consumption of human-robot disassembly is established. To solve the MPUD problem, a multi-objective strength Pareto evolutionary algorithm II is developed and crossover and mutation operators are improved to promote the convergence and feasibility of the algorithm. We perform the experiments on a ballpoint pen and a washing machine to simulate the disassembly process and compare the proposed algorithm with the nondominated sorting genetic algorithm II and the multi-objective evolutionary algorithm based on decomposition. Experimental results show that the proposed algorithm has obvious advantages in the most important metrics.

1. Introduction

With the rapid development of advanced technology and the acceleration of the upgrading of mechanical, electrical, and electronic equipment, the recycling, and remanufacturing of end-of-life (EOL) products are becoming more and more important [1]. Disassembly is a process of disassembling and separating EOL products to obtain valuable parts by a systematic method. It is an essential link in recycling and remanufacturing engineering of EOL products.

Disassemble line balancing problem (DLBP) is first proposed by Gungor et al. [2]. Because it plays an important role in optimizing the allocation of disassembly resources, it attracts many practitioners and researchers in industry and academia. Zhou et al.[3] propose a dynamic programming method to solve a
multi-objective DLBP. Wang et al. [4] adopt a flower pollination algorithm to solve a random two-sided part DLBP. Li et al. [5] propose an incomplete DLBP for disassembly demand parts and hazardous parts and constructs a multi-objective incomplete DLBP model to optimize disassembly sequence length, the number of workstations, idle time balancing index, and disassembly cost.

With the aggravation of the linear disassembly task, the deficiency of the linear single product disassembly line is becoming more and more obvious. Tiwari and Agrawal [6] study the multi-product disassembly line based on time distribution and show that multi-product disassembly has higher efficiency. And the low efficiency of the linear disassembly line is also exposed, the U-shape layout has the advantages of less area and high production efficiency [7]. Abidin [8] proposes a robot DLBP, which improves the efficiency of the production line by using robots to disassemble. It is verified that human-robot disassembly can be used in the disassembly line. In this paper, an improved SPEA-II is proposed to solve MPUD, and this algorithm is compared on different indicators.

In this work, we use an AND/OR diagram [9] to describe disassembly priority relationships and conflict relationships among products. To increase profits and reduce energy consumption, a multi-objective multi-product human-robot-collaborate U-shaped disassembly-line balancing-problem (MMHUD) is proposed. To solve this problem, this paper proposes an improved algorithm based on SPEA-II. Compared with the existing research, we have the following three contributions:

1) A new MMHUD problem is proposed. Based on this problem, we establish a mathematical model of the MMHUD problem to maximize disassembly profit and minimize disassembly energy consumption by considering modes of human-robot disassembly.
2) The SPEA-II is used to solve MMHUD. The innovation of this time lies in the improvement of the cross-mutation part. The crossover variation of the SPEA-II itself remains, and the code of random crossover is added on this basis, and experiments are carried out under the framework of Jemetal.
3) Ballpoint pen and washing machine are used to simulate the MMHUD, comparing with the non-dominant genetic algorithm (NSGAII) [10] and the multi-objective evolutionary algorithm (MOEA/D) [11], compared with IGD [10] Hypervolume [11] and Epsilon, the results show that the SPEA-II is superior in terms of convergence and stability.

2. Problem Description

2.1. Problem statement

Compared with the linear disassembly line, the U-shaped disassembly line costs less manpower and the number of workstations and is more efficient than the linear disassembly line. To ensure the continuity of disassembly, no task handover is allowed until the disassembly task is completed. MMHUD problem involves allocating a group of disassembly tasks to several workstations according to one or more optimization objectives, including maximizing disassembly profit and minimizing energy consumption. It is necessary to satisfy the task precedence constraints for each product and cycle time constraint. We have the following assumptions:

1) The disassembly time of each subassembly is known, and the disassembly cost per time unit and the disassembly energy consumption per time unit of each subassembly are known. Product disassembly AND/NO diagrams of different products are known.
2) When each disassembly task is assigned to the workstation according to the disassembly sequence, the priority relation matrix and conflict relation matrix should be satisfied (the two matrices are known).
3) Each workstation cannot run for more than a fixed cycle time.

2.2. Notation definition

The notations are summarized as follows:

\( g, \varphi \) EOL product indices, \( g, \varphi \in \{1, 2, \cdots, G\} \), where \( G \) represents the number of disassembled products.

\( i \) Subassembly index, \( i \in \{1, 2, \cdots, N^g\} \), where \( N^g \) denotes the number of subassemblies in product \( g \).
\(j, k\) Task indices, \(j, k \in \{1, 2, \cdots, J^g\}\), where \(J^g\) means the number of tasks in product \(g\).

\(l, m\) the index of a workstation, \(l, m = 1, 2, \cdots, M\), where \(M\) is the number of workstations;

\(w, \tilde{w}\) Location assignment indices, \(w, \tilde{w} \in \{0, 1\}\), if \(w = 0\), assign to the entrance side of the workstation; otherwise, \(w = 1\), assign to the exit side of the workstation.

\(t_{jwl}^h\) Human disassembly time for task \(j\) in product \(g\) assigned to \(w\) side of the \(l\)-th workstation.

\(t_{jwl}^r\) Robot disassembly time for task \(j\) in product \(g\) assigned to \(w\) side of the \(l\)-th workstation.

\(t_{jk}^g\) Setting time of task \(k\) in product \(g\) is performed immediately after task \(j\) in product \(g\).

\(t_{jk}^{\varphi}\) Setting time of task \(k\) in product \(\varphi\) is performed immediately after task \(j\) in product \(g\).

\(e_{jwl}^h\) Human disassembly energy consumption per time unit of task \(j\) in product \(g\) is assigned to \(w\) side of the \(l\)-th workstation.

\(e_{jwl}^r\) Robot disassembly energy consumption per time unit of task \(j\) in product \(g\) is assigned to \(w\) side of the \(l\)-th workstation.

\(t_{jk}^h\) Setting energy consumption per time unit of task \(k\) in product \(g\) is performed immediately after task \(j\) in product \(g\).

\(t_{jk}^{\varphi}\) Setting energy consumption per time unit of task \(k\) in product \(\varphi\) is performed immediately after task \(j\) in product \(g\).

\(c_{jwl}^h\) Human disassembly cost per time unit of task \(j\) in product \(g\) is assigned to \(w\) side of the \(l\)-th workstation.

\(c_{jwl}^r\) Robot disassembly cost per time unit of task \(j\) in product \(g\) is assigned to \(w\) side of the \(l\)-th workstation.

\(c_{jk}^g\) Setting cost per time unit of task \(k\) in product \(g\) is performed immediately after task \(j\) in product \(g\).

\(c_{jk}^{\varphi}\) Setting cost per time unit of task \(k\) in product \(\varphi\) is performed immediately after task \(j\) in product \(g\).

\(v_i^g\) Reuse value of subassembly \(i\) in product \(g\).

\(r_{jg}\) The start time of task \(j\) for product \(g\).

\(T\) Cycle time of the workstation.

\(S^g\) Precedence relationship set, if \((j, k) \in S^g\), then task \(j\) in product \(g\) is the immediate predecessor of task \(k\) in product \(g\).

\(p_{jk}^g\) An element in the \(j\)-th row and \(k\)-th column of \(P\) for product \(g\), \(P\) is precedence matrix of a given AND/OR graph of multiple products.

\(d_{ij}^g\) An element in the \(i\)-th row and \(j\)-th column of \(D\) for product \(g\), \(D\) is the disassembly-incidence matrix of a given AND/OR graph of multiple products.
2.3. Decision variables

\[ x^g_{jwl} = \begin{cases} 
1, & \text{if task } j \text{ in product } g \text{ is performed and is assigned to the } w \text{ side of the } l\text{-th workstation} \\
0, & \text{otherwise.}
\end{cases} \]

\[ y^g_{jklw} = \begin{cases} 
1, & \text{if task } k \text{ in product } g \text{ is performed immediately after task } j \text{ in product } g \text{ and is assigned to the } w \text{ side of the } l\text{-th workstation} \\
0, & \text{otherwise.}
\end{cases} \]

\[ z^\varphi_{jklw} = \begin{cases} 
1, & \text{if task } k \text{ in product } \varphi \text{ is performed immediately after task } j \text{ in product } g \text{ and is assigned to the } w \text{ side of the } l\text{-th workstation} \\
0, & \text{otherwise.}
\end{cases} \]

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

\[ O_j = \begin{cases} 
1, & \text{if the disassembly task } j \text{ is performed by human.} \\
0, & \text{otherwise.}
\end{cases} \]

2.4. Mathematical model

\[
\begin{align*}
\max f_1 &= \sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} d^g_{jul} x^g_{jul} - \sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} t^g_{jul} c^w_{jul} x^g_{jul} O_j - \\
&\sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} t^g_{jul} c^w_{jul} x^g_{jul} (1 - O_j) - \sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} t^g_{jul} c^w_{jul} y^g_{jul} - \\
&\sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} \sum_{k=1}^{N^g} t^g_{jul} c^w_{jul} z^g_{jklw} \quad (1)
\end{align*}
\]

\[
\begin{align*}
\min f_2 &= \sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} t^g_{jul} e^w_{jul} x^g_{jul} O_j + \sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} t^g_{jul} e^w_{jul} x^g_{jul} (1 - O_j) + \\
&\sum_{g=1}^{G} \sum_{j=0}^{J^g} \sum_{k=0}^{J^g} t^g_{jul} e^w_{jul} y^g_{jklw} + \sum_{g=1}^{G} \sum_{j=1}^{J^g} \sum_{l=1}^{M} \sum_{u=0}^{N^g} t^g_{jul} e^w_{jul} z^g_{jklw} \\
&\sum_{j=1}^{J^g} \sum_{l=1}^{M} x^g_{jul} \geq 1, g = 1, 2, \cdots, G. \quad (3)
\end{align*}
\]

\[
\begin{align*}
\sum_{j=1}^{J^g} x^g_{jul} \leq 1, g = 1, 2, \cdots, G, j = 1, 2, \cdots, J^g. \quad (4)
\end{align*}
\]

\[
\begin{align*}
\sum_{j=1}^{J^g} x^g_{jul} \geq 1, l = 1, 2, \cdots, M. \quad (5)
\end{align*}
\]

\[
\begin{align*}
\sum_{j=1}^{J^g} \left( l(x^g_{jul} - x^g_{jul}) + (2M - l)(x^g_{jul} - x^g_{jul}) \right) \leq 0, g = 1, 2, \cdots, G, j, k \in S^g, w = 0, \tilde{w} = 1. \quad (6)
\end{align*}
\]

\[
\begin{align*}
\sum_{j=1}^{J^g} x^g_{jul} + \sum_{l=1}^{M} \sum_{u=0}^{N^g} x^g_{jul} \leq 1, g = 1, 2, \cdots, G, \forall p^g_{jk} = -1. \quad (7)
\end{align*}
\]
Objective function (1) is to maximize profit from the disassembly of multiple products. Objective function (2) is to minimize the energy consumption of the disassembly process. Constraint (3) guarantees that at least one task in each product, excluding task 0, is performed in a disassembly process. Constraint (4) represents each task of multiple products can only be assigned to one side of one workstation. Constraint (5) means that each switched-on workstation must be assigned at least one task. Constraint (6) ensures the tasks on the U-shaped disassembly line must satisfy the precedence relationship. (7) ensures the tasks on the U-shaped disassembly line must satisfy the conflicted relationship. (8) denotes the processing time of each workstation does not exceed the cycle time. (9) The output of each task in the dismantle sequence is 1 (10) The time constraint that adjacent tasks need to meet. \( \delta \) is a sufficiently large positive integer. (11) means the range of decision variables.

3. Strength pareto evolutionary algorithm
SPEA-II is an efficient multi-objective optimization genetic algorithm proposed by Zitzler et al [12]. We have modified the crossover variation based on SPEA-II. This improvement makes SPEA-II performance and computational accuracy better than before. We run the algorithm under jMetal framework[13], to facilitate direct comparison with other algorithms. In encoding and decoding, we use a three-step formula to represent a disassembly sequence (a non-dominated solution) \( \pi = \{\pi_1, \pi_2, \pi_3\} \), and \(\pi_1=\{O_1, O_2, O_3\}\) [9]. In crossover-mutation operator part We reserved the cross-mutation and variation of the SPEA-II and add the code with random crossover and improved the mutation process, which increases the diversity of the population. We improve the ability of global search of the algorithm and avoid falling into the local optimal solution. After the crossover of the SPEA-II, a new population P1 is generated to perform the improved crossover.:  
Step 1: Random selection is performed in Archive P1 and r1 and r2 are selected as parents. Place selected individuals r1 and T2 in the mating pool  
Step 2: Read CrossoverProbability for crossover, Cross two random positions of the sequence.  
Step 3: Save the operands on the inlet side into the new solution one by one. Save the operands on the exit side into the new solution one by one by taking negative numbers. For example, sequences 1-35, 14, 16, 17-28 become sequences 1, 14, 16, 17, 35, 28.  
Step 4: Check and adjust the newly generated sequence according to the conflict matrix and priority relation matrix, and finally store it in P1.  
Step 5: Adjust the U-shaped disassembly sequence.

4. Experimental results and analysis
4.1. Case study
We used a ballpoint pen and a washing machine to simulate the multi-product disassembly task to verify the superiority of human-robot disassembly. All of the algorithms and programs testing were based on the Intel IDEA 2018.2.5 “x64” implemented, and used the operating system Windows 10 AMD Ryzen 5 4600h CPU (3.0 GHz / 16 G RAM) run on a PC, we use 50,100,200 and 300 as population size.
respectively. The largest number of iterations is fixed 100 times, each algorithm ran 20 times. The trial was conducted to compare SPEA, MOEAD, and NSGAI with the data specified above. To make a specific comparison, we chose three indicators to show the advantages of SPEA-II in different aspects: Inverted Generational Distance (IGD), Super volume index (Hypervolume), Epsilon: The Epsilon index gives a factor ε. This factor reflects the convergence of the algorithm.

4.2. Analysis of experimental results

We pursued the maximum profit and minimum energy consumption in the human-robot disassembly line, ballpoint pens, and washing machines were used to verify the SPEA-II. We also combined the radio and hammer drill to simulate the disassembly line, and the results were similar. These experimental results were reflected in the following table (t-test means SPEA-II compared to other algorithms ~ means equal, + means better, - means worse).

From the result, it could be seen that SPEA-II generates the most solutions in terms of the number of solutions, while NSGAI tended to fall into the local optimum and produces the least number of solutions. From the perspective of distribution, SPEA-II had a wide distribution range compared with the other two algorithms, and its convergence was good.

| Population | IGD | Hypervolume (N) | Epsilon (N) |
|------------|-----|----------------|-------------|
|            | mean | variance | t-test | mean | variance | t-test | mean | variance | t-test |
| 50         | SPEA-II | 0.00062 | 1.18E-8 | / | 0.037969 | 0.000107 | / | 0.056431 | 0.000195 | / |
|           | NSGA-II  | 0.001743 | 9.52E-8 | + | 0.139662 | 0.001009 | + | 0.131367 | 0.000479 | + |
|           | MOEA/D  | 0.001421 | 3.11E-8 | + | 0.125296 | 0.000881 | + | 0.122154 | 0.000938 | + |
| 100        | SPEA-II | 0.000323 | 5.33E-9 | / | 0.038448 | 7.44E-5 | / | 0.058724 | 0.000115 | / |
|           | NSGA-II | 0.002301 | 8.12E-7 | + | 0.171888 | 0.000895 | + | 0.140859 | 0.000726 | + |
|           | MOEA/D  | 0.000952 | 1.46E-8 | + | 0.14186 | 0.000578 | + | 0.125955 | 0.000434 | + |
| 120        | SPEA-II | 0.000289 | 4.39E-9 | / | 0.039588 | 5.88E-5 | / | 0.063931 | 6.94E-5 | / |
|           | NSGA-II | 0.00218 | 1.01E-6 | + | 0.160417 | 0.001048 | + | 0.129551 | 0.000568 | + |
|           | MOEA/D  | 0.000985 | 1.48E-8 | + | 0.148907 | 0.000442 | + | 0.131559 | 0.000339 | + |
| 150        | SPEA-II | 0.000173 | 1.79E-9 | / | 0.034439 | 3.42E-5 | / | 0.053146 | 4.39E-5 | / |
|           | NSGA-II | 0.003531 | 1.84E-6 | + | 0.210178 | 0.000836 | + | 0.197347 | 0.002647 | + |
|           | MOEA/D  | 0.000848 | 2.32E-8 | + | 0.141294 | 0.000275 | + | 0.131722 | 0.000254 | + |

5. Conclusion

In this paper, the SPEA-II was applied to the disassembly line, at the same time, we used the mathematical model of human-robot disassembly to improve the U-shape disassembly line. The SPEA-II was chosen because of its excellent algorithm performance similar to NSGAI. It has been proved that SPEA-II has better performance and practicability compared with classical algorithms such as NSGAI and MOEAD. In the future, we will plan to add SPEA-II to solve the parallel disassembly line and bilateral disassembly line and use the algorithm thought in these articles for reference [14-16].

Acknowledgments

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] Wang Y.F. (2021), A MCDM-Based Meta-Heuristic Approach for U-shaped Disassembly Line Balancing Problem. Journal of Physics: Conference Series, 1-3.

[2] Gungor, A., Gupta, S.M. (2001) A solution approach to the disassembly line balancing problem in the presence of task failures, In: International Journal of Production Research.

[3] Zhou Y.S., Guo X.P., and Li D. (2020) Correction to: A dynamic programming approach to a multi-objective disassembly line balancing problem. Annals of Operations Research.

[4] Wang K.P., Li X.Y. and Gao L. (2019) A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. Computers & Industrial Engineering.

[5] Li L. K. (2018) Incomplete Disassembly Line Balancing Optimization and Simulation of Multi-objective Discrete Cuckoo Algorithm. Southwest Jiaotong University.

[6] Agrawal S., and Tiwari M. K. (2008) A collaborative ant colony algorithm to stochastic mixed-model u-shaped disassembly line balancing and sequencing problem. International Journal of Production Research, 46.6:1405-1429.

[7] Zhang Z. Q. (2018) Pareto ant colony genetic algorithm for multi-objective u-shaped disassembly line balancing problem,” Journal of Southwest Jiaotong University. 53.003: 628-637.

[8] Abidin Çil Z., Mete S., Serin F. (2020) Robotic disassembly line balancing problem: A mathematical model and ant colony optimization approach,” Applied Mathematical Modelling.

[9] Wu K., Guo X. W., Zhou M. C., Liu S. X., Qi L., (2020) Multi-objective discrete brainstorming optimizer for stochastic disassembly line balancing problem subject to disassembly failure. 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC).

[10] Deb K., Pratap A., Agarwal S., and Meyarivan T. (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. 182-197, 2002.

[11] Zhang Q. F., and Li H., (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evolut. Comput.6: 712-731.

[12] E Zitzler, M Laumanns, and L Thiele. (2001) SPEA-II: Improving the strength Pareto evolutionary algorithm: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Athens, 95-100.

[13] Juan J. Durillo,Antonio J. Nebro. jMetal: “A Java framework for multi-objective optimization. Advances in Engineering Software. Advances in Engineering Software.42

[14] Guo, X. W., Zhou, M. C., Alsokhiry, F. and Sedraoui, K.(2020) Disassembly sequence planning: a survey In: IEEE/CAA J. Autom. Sinica.

[15] Guo, X. W., Zhou, M. C., Liu, S. X. and Qi, L.(2020) Lexicographic Multiobjective Scatter Search for the Optimization of Sequence-Dependent Selective Disassembly Subject to Multiresource Constraints In: IEEE Transactions on Cybernetics. pp. 3307-3317.

[16] Dual-objective program and scatter search for the optimization of disassembly sequences subjGuo, X. W., Liu, S. X., Zhou, M. C. and Tian, G. D.(2018) Dual-objective program and scatter search for the optimization of disassembly sequences subject to multi-resource constraints IEEE Trans. Autom. Sci. Eng. pp.1091-1103.