An s-metric selection evolutionary multi-objective optimization algorithm solving u-shaped disassembly line balancing problem

Wei Shuang Bai¹, Xi Wang Guo¹*, Fangjie Peng²*, Liang Qi³, Shu jin Qin⁴

¹ College of Computer and Communication Engineering, Liaoning Petrochemical University, Fushun, 113001, China
² The Department of Biochemistry Engineering at Chaoyang Teachers College, Chaoyang, 122000, P R. China
³ Department of Computer Intelligent Sci. & Tech., Shandong University of Science and Technology, Qingdao, 266590, China
⁴ College of Economics and Management, Shangqiu Normal University, Shangqiu, 476000, China

*Corresponding author’s e-mail: Xiwang Guo(x.w.guo@163.com), Fangjie Peng(117196455@qq.com)

Abstract. With the continuous consumption of commodities, the number of waste products is also increasing, and its impact on the environment and resources has also attracted great attention. Therefore, the reuse of waste products is one of the ways to solve the problem of increasing waste products at present. In this work, the cycle time constraint of disassembly components is considered in a multi-product partial U-shaped disassembly-line-balancing problem. Moreover, the maximum profit and the minimum idle time are taken as the optimization objectives, and a mathematical model of multi-objective optimization under the cycle time constraints is established. To optimize this problem, an S-metric selection evolutionary multi-objective optimization algorithm (SMS-EMOA) is proposed. The SMS-EMOA is compared with the multi-objective evolutionary algorithm based on decomposition and the indicator-based evolutionary algorithm. The experimental results show the practicability and feasibility of the SMS-EMOA algorithm.

1. Introduction

With the rapid development of science and technology, human consumption demand is increasing, which makes that the life cycle of products is gradually shortened. There are more and more end-of-life (EOL) products[1-3], which do great harm to the environment. In this way, disassembly and recycling are important procedures in the life cycle of products.

Therefore, a Disassembly line balancing problem (DLBP) has been studied by many scholars since it was proposed in 2001[4]. Aiming at the multi-objective problem, Ding et al. [5] introduce a Pareto solution set in the optimization of multi-objective DLBP. Guo et al. [6] use the CPLEX software to solve the multi-objective disassembly optimization problem. It can minimize the number of workstations and minimize the cost.

The above research is carried out under the background of linear disassembly line layout, and the
research on the U-shaped disassembly line balancing (UDLB) problem is relatively less at this stage. A U-shaped disassembly line has the characteristics of compact structure, small footprint, low handling cost, and convenient management. Gu et al. [7] propose a dynamic neighborhood search algorithm to establish an optimization model for the U-shaped disassembly line and minimize the number of direction changes.

The aforementioned studies are all about the UD LB problem of a single product. To study a multi-product problem, a stochastic multi-product multi-objective U-shaped disassembly line balancing (SMMUD) problem is proposed in this paper. A mathematical optimization model is established. To solve the SMMUD problem, our main contributions are as follows:

1) A new SMMUD problem is proposed, which takes into account the uncertainty of disassembly time to maximize the profit and minimum idle time of the workstation.
2) To verify the superiority and feasibility of the SMS-EMOA, several comparison indicators are used to compare SMS-EMOA with the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [8] and the indicator-based evolutionary algorithm (IBEA) [9] algorithm. The experiment proves the superiority of SMS-EMOA in solving the SMMUD problem.

The rest of this work is organized as follows. The problem is described in Section II. An SMS-EMOA is presented in Section III. The experimental results and analysis are given in Section IV. The conclusions and future work are in Section V.

2. Problem Statement

2.1 Problem statement

The U-shaped disassembly method allows the employee to disassemble the products on both sides of the workstations. The proportion of tasks assigned to different workstations is increased, which has better convenience. It is worth noting that the corresponding constraints need to be satisfied when tasks are assigned to workstations. In this paper, an AND/OR diagram is used to describe the priority relationship among various tasks. To better achieve the experimental results, we have the following assumptions:

1) Both the conflict matrix and the association matrix are known.
2) For different products, the setup time and disassembly cost per time unit are the same.
3) The disassembly task cannot be further subdivided.
4) Each task does not interfere with the other and is independent.
5) The disassembly time of each task is known.

2.2 Notation Definition

1) \( P \): all EOL products.
2) \( p, u \): different kinds of disassembly products, \( p, u = 1, 2, \ldots, P \).
3) \( i \): subassembly index, \( i = 1, 2, \ldots, I^p \), where \( I^p \) denote the number of subassemblies for a product \( p \).
4) \( j, k \): the indices of disassembly tasks, \( j, k = 1, 2, \ldots, J^p \) where \( J^p \) is the number of disassembly tasks in \( p \)-th product and 0 is a dummy task.
5) \( l, m \): the indices of workstations, \( l, m = 1, 2, \ldots, M \), where \( M \) is the number of workstations.
6) \( S \): precedence matrix of a given AND/OR graph.
7) \( D \): disassembly incidence matrix of a given AND/OR graph.
8) \( t_{jk}^{pu} \): setting time of \( k \)-th task in \( u \)-th product is performed immediately after \( j \)-th task in \( p \)-th product.
9) \( c_{jk}^{pu} \): preparation cost per time unit when \( k \)-th task in \( u \)-th product is executed immediately after \( j \)-th task in \( p \)-th product.
10) \( c_{j}^{p} \): disassembly cost per time unit when executing \( j \)-th task in \( p \)-th product.
11) \( t_{jk}^{p} \): preparation time of \( k \)-th task in \( p \)-th product when it is performed immediately after \( j \)-th task in \( p \)-th product.
12) $c_{jk}^p$: preparation cost per unit time when $k$-th task in $p$-th product is performed immediately after $j$-th task in $p$-th product.

13) $t_j^p$: disassembly time when executing $j$-th task in the $p$-th product.

14) $\tau$: cycle time of the workstation.

15) $\alpha_j^p$: the start time of $j$-th task of product $p$.

16) $w$: assign location index, when a task is assigned to entrance side, $w = 1$; Otherwise, $w = 0$.

**Decision variables:**

- $X_{jlpw}^p$: if disassembly task $j$ in product $p$ is assigned to the $w$ side of the $l$-th workstation and performed.
- $Z_{jlpwl}^p$: if task $k$ in product $p$ is performed immediately after task $j$ in product $p$ and is assigned to the $w$ side of the $l$-th workstation.
- $B_{jlpwl}^p$: if the $l$-th workstation is used.

**Mathematical model**

\[
\begin{align*}
\max f_1 &= \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} d_{jk}^p X_{jlpw}^p - \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} t_j^p X_{jlpw}^p - \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} t_{jk}^p t_j^p B_{jlpwl}^p - \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} c_{jk}^p c_{jk}^p Z_{jlpwl}^p \\
\min f_2 &= \sum_{l=1}^{L} (u_l \tau - \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} X_{jlpw}^p t_j^p - \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} Z_{jlpwl}^p t_{jk}^p - \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} B_{jlpwl}^p f_j^p) \\
\sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} X_{jlpw}^p \geq 1, w \in \{0, 1\} & \quad (3) \\
X_{jlpw}^p = \sum_{j=1}^{J} Z_{jlpwl}^p, k = 1, 2, 3, ..., J, w \in \{0, 1\} & \quad (4) \\
M \sum_{j=1}^{J} X_{jlpw}^p \leq 1, j = 1, 2, 3, ..., J, p \in P, w \in \{0, 1\} & \quad (5) \\
M \sum_{j=1}^{J} X_{jlpw}^p + \sum_{p=1}^{P} \sum_{j=1}^{J} \sum_{j=1}^{J} \sum_{j=1}^{J} f_{jk}^p Z_{jlpwl}^p \leq Tu_l, l = 1, 2, ..., M & \quad (6) \\
\sum_{u=1}^{U} \sum_{k=1}^{K} Z_{jlpwl}^p = \sum_{u=1}^{U} \sum_{k=1}^{K} Z_{jlpwl}^p = X_{jlpw}^p, \forall p = 1, 2, ..., P, j = 1, 2, ..., J, w = 0, 1, l = 1, 2, ..., M & \quad (7) \\
\alpha_j^p - \alpha_j^w + \lambda(1 - Z_{jlpwl}^p) \geq 0, P, U = 1, 2, ..., P, j = 1, 2, ..., J, w, k = 0, 1, ..., J, w = 0, 1, l = 1, 2, ..., M & \quad (8) \\
\sum_{j=1}^{J} (l(X_{jlpw}^p - X_{jlpw}^p) + (2M - l)(X_{jlpw}^p - X_{jlpw}^p)) \leq 0, p = 1, 2, ..., P, j, k = 1, 2, ..., J & \quad (9)
\end{align*}
\]
\[
\sum_{u=0}^{U} \sum_{j=1}^{P} (X_{j,k}^u + X_{j,k}^u) \leq 1, p = 1, 2, \ldots, P, j, k = 1, 2, \ldots, J^u \tag{10}
\]

\[
\alpha_k^u \geq \alpha_j^p + t_{jk}^u, p, u = 1, 2, \ldots, P, j, k = 1, 2, \ldots, J^p \tag{11}
\]

\[
X_{j,k}^u, Z_{j,k}^u, B_{j,k}^u, u_i \in \{0,1\}, j, k = 1, 2, \ldots, J^p, l = 1, 2, \ldots, M, p = 1, 2, \ldots, P, w \in \{0,1\} \tag{12}
\]

(1) represents the maximum expected profit to disassemble a product. (2) represents the minimum expected workstation idle time. constraint (3) guarantees that at least one task is executed. (4) guarantees that each disassembly task can be executed only once. (5) represents that each task for each product can only be assigned to one workstation. (6) indicates that the disassembly time of each workstation cannot exceed the cycle time. (7) makes sure that the sequence of disassembly can only have one immediately before or after the task. (8) ensures that the number of adjacent tasks that meet the time constraint is large enough. (9) ensures that the feasible disassembly task sequence must meet the precedence relationship. (10) ensures a feasible disassembly task sequence must satisfy conflicting relationships. (11) ensures that the start time of each task should meet the priority relationship. (12) gives the range of decision variables.

### 3. Proposed Algorithm

The main idea of the SMS-EMOA algorithm is to use the S metric [10] to test a generated offspring. If an individual in the population is replaced with a new individual that produces a better solution, then the new individual becomes a member of the population and the replaced individual is eliminated. In this paper, the MPUD problem can be better solved by improving the SMS-EMOA algorithms.

#### 3.1 Encoding and decoding

The encoding method [11] of this algorithm adopted the bidirectional quantity list structure \(\pi = (\pi^1, \pi^2)\). By dividing a feasible disassembly sequence \(\pi^1\) into two subsequences based on the binary string \(\pi^2\), where one sequence was kept unchanged, and the task of another sequence was added with ")" before the task, and then inserted into the previous stationary sequence successively. Thus, a new feasible disassembly sequence \(\pi''\) could be obtained to meet the characteristics of a U-shaped disassembly line layout, where the task represented by ")" was assigned to the U-shaped disassembly line on the exit side. The decoding process of \(\pi\) was shown in Figure 1. It was important to note that the disassembly task needs to meet a time constraint when it was assigned to the workstation.

#### 3.2 Reduce algorithm

In the SMS-EMOA algorithm, after an individual was generated, the individuals with less contribution value (effect to the solution) were eliminated through the reduction algorithm.

#### 3.3 Crossover and Mutation operation

We used the S metric value and a PBX crossover operator[12] to improve the algorithm. The crossover steps were as follows:

- Step1: Picked two individuals at random and called them individual one and individual two.
- Step2: A series of masks were generated randomly, 0 represented a descendant gene from the first individual, and 1 represented a descendant gene from the second individual.
- Step3: Crossed according to the mask to produce a new individual.

The number of genes in the population gradually decreased after continuous iteration of this crossover method. To reduce the occurrence of this situation, we added an indefinite number of genes to the offspring in the mutation operator. After the above operations were completed, the solution might no longer satisfy the priority relationship. To solve this problem, the following methods were given:

- Step 1: Traversal the solutions to determine whether each solution satisfied the priority order.
- Step 2: Deleted one of the solutions if there was a conflict between the two tasks.
- Step 3: If the priority relationship between the two tasks conflicts switched the positions.
Each generation of SMS-EMOA needed to be deleted from the population after the creation of new individuals. Therefore, the selection operator could only compute \( n + 1 \) nondominated solutions at most. Finally, we used the \( S \) metric value to identify our overall population and selected the optimal population, so that the \( S \) metric value was maximized and our results became better.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Case study
The performance of the proposed algorithm was verified by two examples of a Hammer drill and a ballpoint pen, and a radio and Hammer drill. To ensure the true validity of the experimental results, all tests were coded in Java, and all experiments were carried out on a Core (TM) i-3230M 2.60GHz computer with 4G memory. The initial population of the three algorithms was 100, the maximum number of iterations was 2000, and each algorithm was run 20 times independently. In the SMS-EMOA algorithm, the crossover probability and mutation probability were set at 0.9 and 0.05, respectively.

4.2 Analysis of experimental results
To test and analyze the performance of the SMS-EMOA algorithm, we used RHV(N), IGD+(N) [13], Hypervolume (N)[8], and Epsilon(N)[14] to calculate the mean and variance of the maximum profit and the minimum idle time. Tables 1 and 2 showed the data comparison of SMS-EMOA, IBEA, and MOEA/D in the two cases of hammer drill and ball-point pen, hammer drill, and radio, respectively. As could be seen from the data in Table 1, the mean Hypervolume values of the three algorithms in the same hammer drill and ballpoint pen are 0.5065, 0.2389, and 0.5810, respectively. It could be seen that the SMS-EMOA algorithm was better than IBEA and MOEA/D algorithms.

| Table 1. Non-dominated solutions of UDLB problems obtained by three algorithms based on Hammer drill and ballpoint pen |
|-----------------------------------------------|
| Algorithm | MOEA/D | IBEA | SMS-EMOA |
| Hypervolume | mean | variance | t-test | mean | variance | t-test | mean | variance |
| RHV(N) | 0.5065 | 0.033884 | + | 0.2389 | 0.031086 | + | 0.5810 | 0.018031 | null |
| IGD+ | 0.2431 | 0.016555 | ~ | 0.4739 | 0.025478 | + | 0.1829 | 0.005687 | null |
| Epsilon(N) | 0.3619 | 0.030954 | ~ | 0.6455 | 0.025791 | + | 0.2825 | 0.017846 | null |

| Table 2. Non-dominated solutions of UDLB problems obtained by three algorithms based on hammer drill and radio |
|-----------------------------------------------|
| Algorithm | MOEA/D | IBEA | SMS-EMOA |
| Hypervolume | mean | variance | t-test | mean | variance | t-test | mean | variance |
| RHV(N) | 0.0886 | 0.004358 | ~ | 0.0807 | 0.003405 | + | 0.1397 | 0.004689 | null |
| IGD+ | 0.7379 | 0.038065 | + | 0.7613 | 0.029740 | + | 0.5871 | 0.004094 | null |
| Epsilon(N) | 0.3137 | 0.006438 | + | 0.3916 | 0.011291 | + | 0.2635 | 0.004804 | null |

5. CONCLUSION
In this paper, an improved SMS-EMOA algorithm based on Pareto was proposed to solve the U-shaped disassembly line problem. The maximum profit and minimum idle time were mainly taken into consideration. Firstly, the relevant mathematical model was established, and the SMS-EMOA algorithm was processed according to the mathematical model. After several experiments, it was verified that this algorithm had better feasibility and superiority compared with the other two algorithms[15]. In future work, we will build more disassembly models to optimize the algorithm to solve practical problems.[16-17]
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-004a.

REFERENCES

[1] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and L. Qi, “An improved binary cuckoo search algorithm for solving unit commitment problem—Part I: Method Description,” *IEEE Access*, vol. 6, pp. 43535-43545, 2018.

[2] Z. Y. Zhao, S. X. Liu, M. C. Zhou, X.W. Guo, and L. Qi, “Decomposition method for new single-Machine scheduling problems from steel production systems,” *IEEE Trans. Autom. Sci. Eng.*, 2019, doi: 10.1109/TASE.2019.2953669.

[3] Z.Y Zhao, S.X. Liu, M.C. Zhou, D. You, and X. W.Guo, " Heuristic Scheduling of Batch Production Processes Based on Petri Nets and Iterated Greedy Algorithms," in *IEEE Transactions on Automation Science and Engineering*, 2020, Accepted doi:10.1109/TASE.2020.3027532.

[4] X. W. Guo, M. C. Zhou, F. Alsokhiry, and K. Sedraoui, “Disassembly sequence planning: a survey,” *IEEE/CAA J. Autom. Sinica*, 2020, doi: 10.1109/JAS.2020.1003515.

[5] L. P. Ding, J. R. Tan, Y. X. Feng, “Multi-objective optimization for disassembly line balancing based on Pareto ant colony algorithm,” *Comput. Ind. Man.*, vol. 15, no. 7, pp. 1406-1413, 2009.

[6] X. W. Guo, M. C. Zhou, S. X. Liu, and L. Qi, “Multi-resource Constrained Selective Disassembly with Maximal Profit and Minimal Energy Consumption,” *IEEE Trans. Autom. Sci. Eng.*, 2020, doi: 10.1109/TASE.2020.2992220.

[7] Q. X. Xiao , X. P. Guo , and X. J. Gu. “Optimization of stochastic mixed-flow U-shaped disassembly line balancing scheduling problem under multiple constraints”,doi:10.19495/j.cnki.1007-5429.2019.05.012

[8] Q. F. Zhang, and H. Li, MOEA/D: “A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. Evolut. Comput.*, vol. 11, no. 6, pp.712-731, 2007.

[9] W. D. Liao. “Behavior and Improvement of Evolutionary Multi-objective Optimization Algorithm Based on Hypervolume”. *HIT* 2020.

[10] Nicola Beume, Boris Naujoks, and Michael Emmerich. “SMS-EMOA: Multiobjective selection based on dominated hypervolume,” *European Journal of Operational Research*,2006,181(3).

[11] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and L. Qi, “Modified cuckoo search algorithm to solve economic power dispatch optimization problems,” *IEEE. Autom. Sinica*, vol. 5, no. 4, pp. 794-806, 2018.

[12] X. W. Guo, S. X. Liu, M.C. Zhou, and G. D. Tian, “Dual-objective program and scatter search for the optimization of disassembly sequences subject to multi-resource constraints,” *IEEE Trans. Autom. Sci. Eng.*, vol.15, no.3, pp.1091-1103, 2018.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II,” *IEEE Trans. Evolut. Comput.*, vol. 6, no. 2, pp.182-197, 2002.

[14] E. Zitzler, L. Thiele, and M. Laumanns, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Transactions on evolutionary computation*, vol.7, no.2, pp.117-132.2003.

[15] K. Zhang, Y. k. Su, X.W. Guo, L. Qi, and Z. B. Zhao: “MU-GAN: Facial Attribute Editing based on Multi-attention Mechanism”, *IEEE/CAA J. of Autom. Sinica*, CoRR abs/2009.04177 (2020)

[16] Y. Y. Tan, M. C. Zhou, Y. Zhang, X. W. Guo, Liang Qi, and Yanhong Wang, “Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting,” *IEEE Trans. Autom. Sci. Eng.*, Accept, 2020.

[17] X. W. Guo, S. X. Liu, M.C. Zhou, and L. Qi, “Disassembly sequence optimization for large-scale products with multi-resource constraints using scatter search and Petri nets,” *IEEE Trans. Cyber.*, vol. 46, no. 11, pp. 2435-2446, 2016.
[18] X. W. Guo, M. C. Zhou, S. X. Liu, and L. Qi, “Lexicographic Multiobjective Scatter Search for the Optimization of Sequence-Dependent Selective Disassembly Subject to Multi resource Constraints” *IEEE Transactions on Cybernetics.*, vol.50, no.7, pp.3307-3317.2020.