Multi-objective discrete salp swarm optimizer for solving multiple-product U-shaped disassembly line balancing problem

Jiawei Li¹, Xiwang Guo*, Fangjie Peng**, Liang Qi³, Shujin Qin⁴
¹ College of Computer and Communication Engineering, Liaoning Petrochemical University, Fushun, Liaoning, 113001, China
² The Department of Biochemistry Engineering at Chaoyang Teachers College, Chaoyang, Liaoning, 122000, China
³ Department of Computer Intelligent Sci. & Tech., Shandong University of Science and Technology, Qingdao, Shandong, 266590, China
⁴ College of Economics and Management, Shangqiu Normal University, Shangqiu, Henan, 476000, China
*Corresponding author’s e-mail: Xiwang Guo(x.w.guo@163.com), Fangjie Peng(117196455@qq.com)

Abstract. The disassembly line balancing problem is one of the most significant methods in handling large quantities of waste electrical and electronic equipment efficiently problems. In this paper, we propose a U-shaped layout disassembly line and set up a mathematical model to maximize the disassembly profit and minimize the disassembly time of hazardous parts. Therefore, an improved Multi-objective Salp swarm algorithm based on Pareto which combines with a stochastic simulation method is proposed to solve such a U-shaped disassembly line balancing problem. Some cases are used to verify the effectiveness of the proposed method. The results show the proposed method can solve the U-shaped disassembly line balancing problem and outdo the other two optimization algorithms: Nondominated Sorting Genetic Algorithm-II (NSGA-II) and Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) in the quality of solutions.

1. Introduction

With people's pursuit of quality life, more and more personalized goods are constantly produced and put into use. At the same time, a large number of wasted products will be weeded out. Therefore, recycling end-of-life (EOL) products are crucial in the development of sustainability [1-3]. Due to the different service life, working conditions and fault degree, improper disposal of the EOL products not only produce a large amount of waste but also cause environmental pollution problems. Valuable parts disassembled by EOL products can be turned into reused components, which can achieve the goal of green and sustainable development [4-5]. Therefore, the disassembly line aims to minimize the amount of waste sent to landfills utilizing recycling to bring the product to a desired level of quality in this paper.

Disassembly Line Balancing Problem (DLBP) is a classical problem which is proposed in [6] firstly. Especially in the face of increasingly severe resource and environmental problems, the research on DLBP is of special significance. Some scholars use heuristic algorithms to get good solutions, but they still have...
shortcomings in large-scale problems. Therefore, optimization algorithms have been proposed. Thousands of studies make the optimization of DLBP, but most of them are based on traditional linear type layout. Therefore, a U-shaped layout has been proposed. The solution to the U-shaped Disassembly Line Balancing (UDLB) problem has its traits and innovation point. U-shaped layout has been widely used in many fields due to its advantages of compact structure, convenient movement of workers and materials. According to these advantages, we research the U-shaped disassembly line in this paper.

At present, there are relatively little researches about the UDLB problem. Agrawal et al. [7] propose a stochastic mixed-model UDLB problem with an ant colony algorithm, Wang et al. [8] research on a UDLB problem with a flower pollination algorithm. There are inevitably hazardous parts in the product, and these parts should be removed quickly. Meanwhile, we have to put different types of objectives into consideration. DLBP needs to consider the collaborative optimization of multiple objectives, but predecessors’ study can only convert the problem to a single objective. Therefore, a multi-objective mathematical model should be built to carry out the experiments to solve the problem.

As a sub-heuristic algorithm, the Salp Swarm Algorithm (SSA) has been tested on several optimization problems in a variety of fields [9]. However, to makes the SSA algorithm more preferable, we propose a Multi-objective Salp Swarm Algorithm (MSSA) to verify the feasibility and effectiveness of this algorithm. This paper makes three main contributions over the research fields:

1. A mathematical model that includes two objective functions is proposed with maximizing disassembly profit and minimizing disassembly hazardous index. And it’s worth noticing that the latter one is the first time to be mentioned in similar questions.

2. MSSA algorithm is used in solving MUDLB problem, but the algorithm is improved in changing it from single objective to multi-objective problem. Therefore, this is the first time we solve such a problem with MSSA. We design an appropriate encoding and decoding process to obtain a feasible solution.

3. Compared with two algorithms NSGA-II [10] and MOEA/D [11] in the simulation experiments to show that MSSA has superior Pareto fronts and has better results in solving the proposed problem.

2. Problem statement
Most of the previous disassembly line problems are solved by linear type layout. To improve the work efficiency and balance rate, we propose a U-shaped layout. Through the U-shaped working range, as shown in Figure 1, workers can move around both sides of the transmission device. The entrance and exit sides are on the same side, which can reduce the change time of directions. What’s more, the number of workstations will be rationally reduced by a reasonable assignment of tasks. According to these merits, this paper makes the following assumptions: to satisfy the priority and conflict relationship in the AND/OR graph when allocating each process to workstations and minimize the number of workstations. Each workstation should have the same cycle time. The specified cycle time of each workstation is the same on the premise of the actual task execution time of each workstation not exceeding the cycle time. The disassembly time for each task is known.

Figure 1. a U-shaped layout.
On the premise of satisfying the disassembly priority relation, each process is reasonably allocated to the workstation and keeps the number of workstations minimized according to the cycle time constraint and precedence constraint. Since the disassembly line is to decompose EOL products, the disassembly process should also consider whether there are hazardous parts and the disassembly cost, etc. The U-shaped disassembly line builds the feasible solution according to the feature that the entrance and exit are in the same position in the task priority relation diagram.

For the sake of describing the precedence constraint, we adopt AND/OR graphs to make the whole line clear. In this paper, we select a hammer drill, a reduced copy machine and a radio.

2.1. Notation definition
1) \( g, \varphi \) EOL product indices, \( g, \varphi \in \{1,2,\cdots,G\} \), where \( G \) represents the number of products.
2) \( i \) Subassembly index, \( i \in \{1,2,\cdots,N_{\varphi}\} \), where \( N_{\varphi} \) denotes the subassemblies in product \( g \).
3) \( j, k \) Task indices, \( j, k \in \{1,2,\cdots,J_{g}\} \), where \( J_{g} \) means the number of tasks in product \( g \).
4) \( l, m \) Workstation indices, \( l, m \in \{1,2,\cdots,M\} \), where \( M \) is the number of workstations.
5) \( w, \bar{w} \) Location assignment indices, \( w, \bar{w} \in \{0,1\} \), if \( w = 0 \), assign to the entrance side of the workstation; otherwise, \( w = 1 \), assign to the exit side of the workstation.
6) \( t_{jlw}^{g} \) Disassembly time that task \( j \) in product \( g \) is assigned to \( w \) side of the \( l \)-th workstation.
7) \( t_{jk}^{\varphi} \) Setting time that task \( k \) in product \( \varphi \) is performed immediately after task \( j \) in product \( g \).
8) \( c_{jlw}^{g} \) Cost per time unit that task \( j \) in product \( g \) is assigned to \( w \) side of the \( l \)-th workstation.
9) \( c_{jk}^{\varphi} \) Setting cost per time unit that task \( k \) in product \( \varphi \) is performed after task \( j \) in product \( g \).
10) \( \nu_{jlw}^{g} \) Reuse value of subassembly \( i \) in product \( g \).
11) \( T \) Cycle time of the workstation.
12) \( P \) Precedence matrix of multiple products.
13) \( D \) Disassembly-incidence matrix of multiple products.
14) \( H \) Define whether the subassembly product is hazardous.
15) \( S_{g}^{\varphi} \) Precedence relationship set, if \( (j, k) \in S_{g}^{\varphi} \), then task \( j \) in product \( g \) is the immediate predecessor of task \( k \) in product \( g \).
16) \( p_{jk}^{g} \) An element in the \( j \)-th row and \( k \)-th column of \( P \).
17) \( d_{ij}^{\varphi} \) An element of \( D \).
18) \( h_{k}^{\varphi} \) An element of \( H \).
19) \( \alpha_{lj} \) The start time of task \( j \) of product \( g \).

2.2. Decision variables
\[
x_{jlw}^{g} = \begin{cases} 1, & \text{if task } j \text{ in product } g \text{ is performed and is assigned to the } w \text{ side of the } l \text{-th workstation.} \\ 0, & \text{otherwise.} \end{cases}
\]
\[
u_{jl} = \begin{cases} 1, & \text{if the } l \text{-th workstation is used.} \\ 0, & \text{otherwise.} \end{cases}
\]
\[
z_{jkw}^{\varphi} = \begin{cases} 1, & \text{if task } j \text{ in product } \varphi \text{ is performed immediately after task } j \text{ in product } g \text{ and is assigned to the } w \text{ side of the } l \text{-th workstation.} \\ 0, & \text{otherwise.} \end{cases}
\]

2.3. Mathematical model
\[
\begin{align*}
\max f_1 &= \sum_{g=1}^{G} \sum_{j=1}^{J_{g}} \sum_{l=1}^{L} \sum_{w=0}^{1} \sum_{i=1}^{N_{\varphi}} d_{jlw}^{g} \nu_{jlw}^{g} - \sum_{g=1}^{G} \sum_{j=1}^{J_{g}} \sum_{l=1}^{L} \sum_{w=0}^{1} \sum_{i=1}^{N_{\varphi}} c_{jlw}^{g} x_{jlw}^{g} - \sum_{g=1}^{G} \sum_{j=1}^{J_{g}} \sum_{l=1}^{L} \sum_{w=0}^{1} \sum_{i=1}^{N_{\varphi}} t_{jlw}^{g} \alpha_{jl} \nu_{jl} - \sum_{g=1}^{G} \sum_{j=1}^{J_{g}} \sum_{l=1}^{L} \sum_{w=0}^{1} \sum_{i=1}^{N_{\varphi}} t_{jlw}^{g} c_{jk}^{\varphi} x_{jlw}^{g}
\end{align*}
\]
\[
\min f_2 = \sum_{g=1}^{G} \sum_{j=1}^{J_{g}} \sum_{l=1}^{L} \sum_{w=0}^{1} \sum_{i=1}^{N_{\varphi}} d_{jlw}^{g} x_{jlw}^{g} \alpha_{jl} \nu_{jl} + \sum_{g=1}^{G} \sum_{j=1}^{J_{g}} \sum_{l=1}^{L} \sum_{w=0}^{1} \sum_{i=1}^{N_{\varphi}} t_{jlw}^{g} c_{jk}^{\varphi} x_{jlw}^{g}
\]
\[
\sum_{j=1}^{M} \sum_{l=1}^{I} \sum_{w=0}^{1} x_{jul}^g \geq 1, g = 1, 2, \ldots, G. 
\] (3)

\[
\sum_{j=1}^{M} \sum_{l=1}^{I} x_{jul}^g \leq 1, g = 1, 2, \ldots, G, j, l = 1, 2, \ldots, J^g. 
\] (4)

\[
\sum_{j=1}^{M} \sum_{l=1}^{I} \sum_{w=0}^{1} x_{jul}^g \geq u_l, l = 1, 2, \ldots, M
\] (5)

\[
\sum_{l=1}^{I} \left( (x_{jul}^g - x_{jwl}^g) + (2M - l)(x_{jul}^g - x_{jwl}^g) \right) \leq 0, g = 1, 2, \ldots, G, j, k, \varphi = 0, 1, l = 1, 2, \ldots, M. 
\] (6)

\[
\sum_{g=1}^{G} \sum_{l=1}^{I} x_{jul}^g = \sum_{g=1}^{G} \sum_{l=1}^{I} x_{jwl}^g = x_{jul}^g, \forall g = 1, 2, \ldots, G, j = 1, 2, \ldots, J^g, w = 0, l = 1, 2, \ldots, M. 
\] (7)

\[
\sum_{j=1}^{M} \sum_{l=1}^{I} \sum_{w=0}^{1} x_{jul}^g \leq 1, g = 1, 2, \ldots, G, \forall p_{jk}^g = -1. 
\] (8)

\[
\sum_{g=1}^{G} \sum_{l=1}^{I} \sum_{w=0}^{1} t_{jul}^g + \sum_{g=1}^{G} \sum_{l=1}^{I} \sum_{w=0}^{1} t_{jwl}^{g\varphi} \leq T, l = 1, 2, \ldots, M. 
\] (9)

\[
\alpha_{g\varphi} - \alpha_{gk} + \lambda(1 - z_{jwl}^{g\varphi}) \geq t_{jul}^g + t_{jwl}^{g\varphi}, g, \varphi = 1, 2, \ldots, G, j, k \in S^g. 
\] (10)

\[
\alpha_{gk} \geq \alpha_{g\varphi} + t_{jwl}^{g\varphi}, \forall p_{jk}^g = 1. 
\] (11)

\[
x_{jwl}^{g\varphi}, u_l \in [0, 1], g = 1, 2, \ldots, G, j, l = 1, 2, \ldots, J^g, w = 0, l = 1, 2, \ldots, M. 
\] (12)

Objective function (1) is to maximize profit from the disassembly line. Objective function (2) is to minimize the hazardous index to remove hazardous parts as quickly as possible. Constraint (3) guarantees that at least one task in each product, excluding task 0, is performed in the process. Constraint (4) represents each task 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. Constraint (7) ensures the disassembly sequence can only have one input per task. Constraint (8) ensures the tasks on the U-shaped disassembly line must satisfy the conflicted relationship. Constraint (9) denotes the processing time of each workstation does not exceed the cycle time. Constraint (10) ensures that adjacent tasks meet the time constraint \( \lambda \) is a number large enough. Constraint (11) is to ensure that the start time of each task should satisfy the precedence relationship. Constraint (12) means the range of decision variables.

3. Proposed algorithm
Salp Swarm Algorithm (SSA) is still a new-type and burgeoning algorithm [9], which mimics the foraging process of salp groups and can be applied to various fields to get optimal results for many problems. Aiming to improve the property of the algorithm to let it solve the large-scale problem, we propose an improved MSSA algorithm to solve the MUDLB problem to show the superiorities.

3.1. Encoding and decoding
The proposed algorithm used a double-vector list structure \( \pi = (\pi^1, \pi^2) \) for encoding. \( \pi^1 \) represented the whole disassembled task sequence. \( \pi^2 \) was to judge whether the task was performed.

The whole task sequences in disassembled line were assigned to each workstation according to \( \pi^1 \) during the decoding process, the subsequences of the workstations were determined at the same time. As shown in Figure 1, set three workstations into the whole disassembly line and each workstation was
assigned at least one worker. Operations 4 and 7 were assigned to workstation 1. Operation 4 was assigned to the entrance side of workstation 1 while operation 7 was assigned to the exit side.

3.2. The initialization and search process of MSSA
First, we initialized the population in our workspace, valued it as \(n\). The probability that a new individual was generated from one old individual. We made the population distribution random according to the characteristic of the problem, the resulting initial solution would be evenly distributed in the result set.

In the numerous kinds of problem solutions about MSSA, it was mainly used to solve decimals problems. Therefore, we built new crossover and mutation operators to improve the algorithm.

For example, tasks 1-7 represented radio instances, tasks 8-22 represented the reduced copy machine instances through the whole disassembly sequence. Step 1: marked symbol parent 1 and parent 2; Step 2: a mask code would be generated at random. Masks ranged from 0 to 1. When the mask equals 0, child 1 duplicated the value of task parent 1. About the mutation operator, step 1: chose a catastrophe point randomly; Step 2: the preceded and followed tasks would be found; Step 3: the mutation task could insert itself stochastically. Then, accelerated the convergence and increased the distribution of the algorithm by proper selection way. The new individual set and current population were combined, and the first \(n\) individuals were chosen as the next population by rank and crowding distance approached.

As for building a U-shaped disassembly line, we could put fast nondominated sorting into effect to sort the individual in our group. Therefore, we chose a salp with the best conditions in all aspects as the leader to make the algorithm, then the results became better.

4. Experimental results and analysis

4.1. Case study
To show the superiority of our algorithm, we chose some other algorithms to make a comparison. As two of the well-known optimal algorithms, NSGA-II and MOEA/D, which have a great deal of authority in this area, are selected to be compared. All algorithms were taken on a laptop with 2.50 GHz Intel Core i5-7200U CPU, 8 GB memory using IntelliJ IDEA 2020.2.4. To enrich the experiment results, ran 20 times in all the algorithms. The probabilities of crossover and mutation were set to 0.7 and 0.5, respectively. \(\text{Spread}(N)[12]\), \(\text{Hypervolume}(N)[13]\), \(\text{Epsilon}(N)[14]\), inverted generational distance plus \(\text{IGD}^+ (N)\) [15] were selected to be the comprehensive indicators for better evaluating.

4.2. Performance analysis
We combined three instances, reduced copy machine, hammer drill and radio into 4 multi-product cases: reduced copy machine and radio as case 1; reduced copy machine, hammer drill and radio as case 2; hammer drill and radio as case 3; reduced copy machine and radio as case 4, and chose them into experiments. Table 1 shows the result of nondominated solutions under different indicators.

|                  | MSSA          | NSGA-II       | MOEA/D        | NSGA-II       | MOEA/D        |
|------------------|---------------|---------------|---------------|---------------|---------------|
|                  | mean | variance | t-test | mean | variance | t-test | mean | variance | t-test | mean | variance | t-test |
| **Case 1**       |      |          |       |      |          |       |      |          |       |      |          |       |
| MSSA             | 1.6088 | 0.0039  | /     | 0.5141 | 0.0036  | /     | 0.1696 | 0.0032  | /     | 0.0904 | 0.0016  | /     |
| NSGA-II          | 1.4447 | 0.0028  | +     | 0.3371 | 0.0004  | +     | 0.3339 | 0.0003  | +     | 0.2218 | 0.0004  | +     |
| MOEA/D           | 0.9789 | 0.0039  | +     | 0.3638 | 0.0001  | +     | 0.3585 | 0.0004  | +     | 0.1985 | 0.0001  | +     |
|                  | 1.4217 | 0.0047  | /     | 0.6141 | 0.0018  | /     | 0.1641 | 0.0009  | /     | 0.0790 | 0.0008  | /     |
| **Case 2**       |      |          |       |      |          |       |      |          |       |      |          |       |
| MSSA             | 1.3360 | 0.0011  | +     | 0.4767 | 0.0003  | +     | 0.2451 | 0.0001  | +     | 0.1749 | 0.0001  | +     |
| NSGA-II          | 0.9704 | 0.0039  | +     | 0.5239 | 0.0001  | +     | 0.2866 | 0.0002  | +     | 0.1429 | 0.0000  | +     |
| MOEA/D           | 1.4031 | 0.0119  | /     | 0.7121 | 0.0030  | /     | 0.1150 | 0.0011  | /     | 0.0673 | 0.0011  | /     |
|                  | 1.2992 | 0.0107  | +     | 0.5715 | 0.0002  | +     | 0.2328 | 0.0002  | +     | 0.1665 | 0.0001  | +     |
It was observed that MSSA had quite better results than that of NSGA-II and MOEA/D in approximation and distribution through the whole nondominated sets. From all the experiments above, it is not too hard to find that the MSSA algorithm is a good way to solve the problems we proposed.

5. Conclusion
In this paper, a Multi-objective Salp Swarm Algorithm (MSSA) was proposed to solve the MUDLB problem. The performance of this algorithm showed its feasibility and superiority to solve this kind of problem by a lot of large-scale simulation experiments. This paper proposed a U-shaped layout that fits the present environment to avoid trouble and adjust the balance in the actual disassembly process. A mathematical model was established by the maximum profit and the minimum hazardous index as the optimization objectives. The proposed algorithm was applied to several examples, and the results were compared with well-known algorithms. The proposed algorithm had a strong ability and could be used to solve the proposed problem. Look forward to the future, it is necessary to set more practical disassembly models and incorporate more optimization strategies to solve this problem better [16-17].

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] Guo, X. W., Zhou, M. C., Alsokhiry, F. and Sedraoui, K. (2020) Disassembly sequence planning: a survey In: IEEE/CAA J. Autom. Sinica.
[2] Guo, X. W., Zhou, M. C., Liu, S. X. and Qi, L. (2020) Multi-resource Constrained Selective Disassembly with Maximal Profit and Minimal Energy Consumption IEEE Trans. Autom. Sci. Eng.
[3] 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.
[4] Guo, 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.
[5] Guo, X. W., Liu, S. X., Zhou, M. C. and Qi, L. (2016) Disassembly sequence optimization with multiresource constraints using scatter search and Petri nets IEEE Trans. Cyber., pp. 2435-2446.
[6] Gungor, A. and Gupta, S. M. (2001) A solution approach to the disassembly line balancing problem in the presence of task failures Int. J. Prod. Res. pp. 1427-1467.
[7] Agrawal, S. and Tiwari, M. K. (2008) A collaborative ant colony algorithm to stochastic mixedmodel U-shaped disassembly line balancing and sequencing problem Int. J. Prod. Res. pp. 1405-1429.
[8] Wang, K. P., Gao, L. and Li, X. Y. (2020) A multi-objective algorithm for U-shaped disassembly line balancing with partial destructive mode In: Neural Comput. Appl.
[9] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. and Mirjalili, S. M. (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, In: Engineering Software, pp. 163-191.
[10] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II IEEE Trans. Evolut. Comput., pp. 182-197.
[11] Zhang, Q. F. and Li, H. (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition IEEE Trans. Evolut. Comput., pp. 712-731.
[12] Wu, J. and Azarm, S.(2001) Metrics for quality assessment of a multiobjective design optimization solution set Transactions of the ASME, Journal of Mechanical Design, 123(1): 18-25.
[13] Zitzler, E., Thiele, L., Laumanns, C. M. and Fonseca, V. G. (2003) Performance assessment of multiobjective optimizers: an analysis and review IEEE Trans. Evolut. Comput. pp. 117-132.
[14] Gacic, N., Zecevic, A. I. and Siljak, D. D. (1998) Coherency Recognition Using Epsilon Decomposition IEEE Trans on Power Systems, 13(2): 314-319.
[15] Fu, Y. P., Ding, J. L., Wang, H. F. and Wang, J. W. (2018) Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in Industry 4.0-based manufacturing system In: Appl. Soft Comput., pp. 847-855.
[16] Zhao, Z. Y., Liu, S. X., Zhou, M. C., Guo, X. W. and Qi, L.(2019) Decomposition method for new single-Machine scheduling problems from steel production systems IEEE Trans. Autom. Sci. Eng.
[17] Zhao, Z.Y., Liu, S.X., Zhou, M.C., You, D. and Guo, X. W.(2020) Heuristic Scheduling of Batch Production Processes Based on Petri Nets and Iterated Greedy Algorithms In: IEEE Transactions on Automation Science and Engineering.