A MCDM-Based Meta-Heuristic Approach for U-shaped Disassembly Line Balancing Problem

Yifeng Wang1*, Yang Xie1, Yaping Ren2 and Chaoyong Zhang1

1 State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, China
2 Department of Industrial Engineering, School of Intelligent Systems Science and Engineering, Jinan University, Guangzhou, China

* Email: yfwang_sz@hust.edu.cn

Abstract. Since the rapid development of technology advancement and accelerated upgrading rate of mechanical, electrical, and electronic equipment, dealing with the recovery, recycling, and remanufacturing of end-of-life (EOL) products are of increasing importance. The disassembly line balancing problem (DLBP) aims at addressing such concern in the most efficient and effective manner. In this paper, a meta-heuristic approach involving both multi-criterion decision making (MCDM) and variable neighborhood search (VNS) is proposed to the U-shaped layout of the disassembly line. Following the MCDM method combining fuzzy set theory, grey relational analysis, and Choquet fuzzy integral, an improved VNS approach with a specially designed decoding method is utilized to balance a U-shaped line with three hierarchical objectives. Finally, computational experiments validate the effectiveness of the proposed approach.

1. Introduction

With the rapid development of science and technology, the rate of obsolescence of mechanical, electrical and electronic equipment has been increased drastically, which will exert a significant pressure on environment and cause considerable economic loss. To address the urgent challenges, recovery, recycling, and remanufacturing are of great importance. The disassembly is a fundamental step towards dealing with end-of-life (EOL) products and disassembly line is the most efficient and effective way to implement the disassembly operation. Therefore, how to appropriately assign disassembly tasks to each workstation in a disassembly line and balance the load in each workstation to promote production efficiency, in other words, the disassembly line balancing problem (DLBP) has become the most principal problem.

Numerous attempts have been made in the literature to solve DLBP which has been proven to be a NP-complete problem to obtain near-optimal or optimal solutions in reasonable time. However, NP-completeness makes DLBP a complicated multi-objective optimization problem which preventing exact methods from solving the large-scale instances, even the median-scale. On the other hand, heuristic and meta-heuristic algorithms have been widely used in optimizing complex problems due to their remarkable global and local searching abilities in the solution space.

Based on the analysis mentioned above, the remainder of this paper is organized as follows. Section 2 offers a review of existing literature about different approaches and different models to DLBP. The detailed description of DLBP is introduced in Section 3, followed by explanation of the
used approach in Section 4. Section 5 presents the results and analysis of computational experiments and Section 6 gives the conclusion.

2. Literature Review
Since the superior performance of heuristic and meta-heuristic algorithms than exact mathematical programming [1] on most products, they attract much attention of researchers who aim at exploring the frontier of DLBP. A number of different heuristic and meta-heuristic algorithms have been employed to solve DLBP, including greedy algorithm and its variants [2], construction and improvement type [3], MCDM [4], [5], genetic algorithm (GA) [6], ant colony optimization (ACO) [7], artificial bee colony (ABC) [8], and etc. Apart from single algorithm, hybrid algorithms like hybrid MCDM [9], [10], [11], GA with VNS [12], and GA with extremal optimization [13] also demonstrated a commendable result in analysing DLBP due to the combination of global and local searching capacity.

In addition to various exploration of optimization algorithms, different forms of layout of disassembly line, namely, U-shaped, two-sided, and parallel, other than the simple straight line have been studied to further improve the efficiency of the whole line. Generally, straight line is the easiest layout to implement with, however, several flaws like too much occupied space and relatively low efficiency [2]. U-shaped line allows workers to operate on both front and back sides of the line simultaneously, which reduces the occupied area and gives more potential to better balance [14]. Two-sided line enables space for improved performance when tackling large-sized products by the cooperation of both the left and right workstations [15]. Workers in parallel line can process two products in one workstation, thus reducing the idle time and enhance teamwork [16].

3. Problem Description
In this section, we present the formal definition of DLBP. It involves $n$ parts assigned to $w$ workstations along the disassembly line properly. Each part, also as known as task, disassembled by one or a group of specific operations with definite processing time and other indices measuring its economic and environmental effects will be allocated to one workstation according to the cycle time constraint and precedence constraint, which can be illustrated in a disassembly precedence diagram in detail.

![Disassembly precedence diagram of a product.](image)

Specifically, the cycle time constraint means that a usually pre-decided and constant Cycle Time (CT) is taken into consideration, which distinguishes a DLBP from a disassembly problem without involving the concept of workstations. Also, the sum of disassembly processing time of all the tasks in each workstation must be guaranteed not exceed CT. In addition to the cycle time constraint, a DLBP includes the precedence constraint in which the disassembly tasks can be classified into AND, OR and complex AND/OR precedence relationships [6]. Figure 1 illustrates an example of disassembly precedence diagram in which the AND precedence relationship is represented as a solid arc while the OR precedence is a dotted arc. Note that the node 0 represents an initial point for convenience.
3.1. The U-shaped Layout
Different from the straight line in which a task can be allocated to a workstation only when all of its prior tasks have been set to the former workstations, U-shaped line requires all of one task’s predecessors or/and successors to be assigned in order to deal with a particular task [14], which generates more both complexity and possibilities of the solution space.

4. MCDM-based VNS Approach
In this section, we present the proposed MCDM-based VNS approach. Along with the logical order, first comes the MCDM method which involves fuzzy set theory, grey relational analysis and Choquet fuzzy integral, followed by the VNS algorithm, specifically the encoding theme, shaking and local searching, and decoding method.

4.1. MCDM Method
In general, five criteria are taken into consideration, namely number of precedent tasks (NPs), profit, demands, hazardous degree (HD) and damage degree (DD), to evaluate and fuzzy set theory, grey relational analysis and Choquet fuzzy integral are employed to generate a rank for each task without ignoring the interdependence between the criteria.

4.1.1. Fuzzy Set Theory. A fuzzy set is defined by a membership function which maps the crisp inputs to degrees of membership within a certain interval. As shown in table 1, the weight of each criterion and the linguistic expression of each qualitative criterion are included, defined by the corresponding triangle fuzzy numbers which are common in MCDM. Additionally, the defuzzification method adopted in [10] is implemented in this paper.

| Fuzzy number | Weight of each criterion | Linguistic expression of each qualitative criterion | Membership function |
|--------------|--------------------------|---------------------------------------------------|---------------------|
| 0            | Negligible               | No                                                | (0, 0, 0.25)        |
| 0.25         | Trivial                  | Slight                                            | (0.25, 0.5, 0.75)   |
| 0.5          | Moderate                 | Moderate                                           | (0.25, 0.5, 0.75)   |
| 0.75         | Important                | Severe                                            | (0.5, 0.75, 1)      |
| 1            | Paramount                | Devastating                                        | (0.75, 1, 1)        |

4.1.2. Grey Relational Analysis. The Grey relational analysis (GRA) was initially proposed to determine the best in a candidate set of alternatives, which is composed of four parts: compatibility processing and normalization, reference sequence selection, computation of grey relational coefficients (GRC), and calculation of grey relational grade and ranking. Technically, different normalizing criteria are employed in this paper, which, lower-bound effectiveness measuring for NPs and upper-bound for others. Note that the parameter $\eta$, namely distinguishing coefficient, in the calculation of GRC is generally set to be 0.5.

4.1.3. Choquet Fuzzy Integral. Because of the low computational complexity and considering the interaction among different criteria, the Choquet fuzzy integral with $\lambda$-fuzzy measure is employed to calculate rank which is generated by sorting the result of integral ascendingly for each task. The basic definition of the Choquet integral with $\lambda$-fuzzy measure is the same in [11].

4.2. Improved VNS Algorithm
Meta-heuristic algorithms are widely used in NP-complete problems to obtain near-optimal or optimal solutions in reasonable time. Three hierarchical objectives are considered in this paper, namely $f_1$, $f_2$ and $f_3$ which are defined as follows

$$\text{MIN} f_1 = w$$
\[
\text{MIN } f_2 = \sum_{j=1}^{w} (CT - ST_j)^2 \\
\text{MIN } f_3 = \left( \sum_{k=1}^{n}[p^0(k) - p(k)]^2 \right)^{1/2}
\]

where \( w, ST_j, n, p^0(k), p(k) \) represent the number of workstations, the work time of the \( j^{th} \) workstation, the number of tasks, the position of \( k^{th} \) task in the initial sequence \( s^0 \) and in the current sequence \( s \), respectively. And the improved VNS algorithm is designed to minimize the three objective functions based on their priority that \( f_1 > f_2 > f_3 \).

### 4.2.1. Encoding

The encoding method adopted in this paper is to generate a sequence of which elements are tasks in the disassembly precedence diagram according to the rank of each task, and simultaneously meet the precedence constraint through a priority matrix (\( P \) of the instance in figure 1 which is shown as figure 3 in the appendix) in which an element of value 1 in the \( i^{th} \) row and \( j^{th} \) column means that task \( i \) is the precedence of task \( j \) while a value 0 means no precedence relationship between corresponding tasks.

### 4.2.2. Shaking and Local Searching

After attaining the initial sequence, a shaking with a variety of neighborhood structures, rather than one with a single operator, is processed to search the solution space and increase the probability to escape from local optima. The included neighborhood structures are, namely, left exchange, right exchange, left insert, and right insert. Apart from the four structures employed in shaking phase, local searching involves 2-opt operator [11].

### 4.2.3. Decoding

The decoding method in this paper is specially designed for the U-shaped layout. Rather than simply assigning tasks into workstations one by one from the top of the sequence, this proposed decoding method considers tasks both from the top and bottom. To be specific, after assigning a task from the top of current subsequence, given that in the middle phase of the conversion from sequence to solution, tasks from bottom are sequentially considered to fill up the gap between CT and currently consumed time until exceeds. Later, the second top task is about to be assigned to the current workstation to check if any possible gap to fill up. This operation both considering the top and bottom over once can effectively broaden the solution space to promote the performance of obtaining a near-optimal/optimal solution.

### 5. Computational Experiments and Analysis

In this section, we show the general process of generating the initial solution via MCDM and the results of computational experiments with parameter settings. To validate the performance of the U-shaped layout, several cases with variety of levels of the number of parts are included in the computational experiments to demonstrate the competitiveness of the U-shaped layout compared to the straight disassembly line.

#### 5.1. Generating the Initial Solution via MCDM

The initial data of the example, as shown in the figure 1, are modified: \( CT = 80, NPs = \{0, 0, 0, 0, 2, 2, 3, 3, 4, 4, 5, 5, 6\} \), disassembly time = \{21, 21, 21, 21, 44, 60, 17, 27, 22, 43, 16, 33, 15\}, profit = \{0, 0, 0, 0, 7, 5, 1.5, 6.5, 14, 4, 9, 10, 11.5\} (which in obtained through subtracting processing cost from recycling revenue), and demands = \{0, 0, 0, 300, 250, 750, 800, 500, 200, 20, 100, 500\}. The weight of each criterion and the qualitative criteria, namely HD and DD, are evaluated by five experts who share the same importance in assessment. Note that task 9 and 11 are OR precedent of task 12 so we randomly remove the OR precedence between task 11 and 12 (between task 9 and 12 is also feasible), which will not change the whole precedence constraint.

Following the step mentioned in Section 4.1, a decision matrix \( D \) of which each entry corresponds to a value of a certain criterion can be constructed and the normalized \( D_{\text{normed}} \) is obtained by the
mixed normalizing criterion, then the GRC are calculated. Additionally, Table 2 provides information of the results of Choquet integral for each task and the corresponding rank after computing out that $\lambda = -0.9998$ by equation (9) in [11]. Considering both rank and precedence relationship of each task, an initial sequence is generated as $s^0 = \{1, 2, 4, 3, 5, 7, 9, 12, 13, 6, 8, 10, 11\}$ and its corresponding solution of the straight line and the U-shaped layout are $S^0_{\text{straight}} = \{W1=\{1, 2, 4\}, W2=\{3, 5\}, W3=\{7, 9, 12\}, W4=\{13, 6\}, W5=\{8, 10\}, W6=\{11\}\}$ and $S^0_{\text{U-shaped}} = \{W1=\{1, 10, 11\}, W2=\{2, 4, 8\}, W3=\{3, 5\}, W4=\{6, 7\}, W5=\{9, 12, 13\}\}$, which are illustrated in figure 2.

| Tasks | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Results | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.6308 | 0.6242 | 1.0001 | 1.0002 | 1.0000 | 0.8901 | 0.5056 | 1.0001 | 0.6659 |
| rank   | 5   | 6   | 7   | 8   | 11  | 12  | 2   | 1   | 4   | 9   | 13  | 3   | 10  |

**Table 2.** Results of Choquet integral and corresponding rank.

![Initial sequence](image1)

![Initial solution of straight line](image2)

![Initial solution of U-shaped layout](image3)

**Figure 2.** (a) Initial sequence. (b) Initial solution of straight line. (c) Initial solution of U-shaped layout.

5.2. Comparing with Straight Line

The proposed approach is conducted in Python and each instance has been tested 20 times to choose the best result. All experiments are run on an Intel Core i7-8750H CPU (2.20 GHz/8.00 GB RAM). Parameter settings are as follows: total number of iterations $\text{Iter}_{\text{max}} = 50$, total number of local searching $\text{LocalSearch}_{\text{max}} = 20$.

Table 3 gives the results of the MCDM-based VNS approach for both U-shaped layout and straight line. Column one and two indicates the instance number and its number of parts. Column three shows different CT settings for a product, referring to different industrial practices. Column four gives the corresponding lower bound of the second objective function with respect to different number of workstations both in two layouts of disassembly line. The following six columns corporate each three evaluation of both layouts.

It can be seen from the table 3 that when considering a small-scale product, the straight line shows a similar competence with the U-shaped layout. When processing median- and large-scale products, the performance of U-shaped layout mainly surpasses that of straight line not only in decrease in $f_2$, but also in reducing the number of workstations which is constituted as a considerable improvement. Especially when CT = 80 of product 4, the U-shaped layout reaches the lower bound of $f_2$ with $f_1 = 3$, which demonstrates the superior performance of U-shaped disassembly line than the traditional straight one.
Table 3. Computational results.

| Product No. | Tasks | CT | Lower bound of $f_2^a$ | Evaluation of U-shaped line | Evaluation of straight line |
|-------------|-------|----|------------------------|-----------------------------|-----------------------------|
|             |       |    | $f_1$ | $f_2$ | $f_3$ | $f_1$ | $f_2$ | $f_3$ |
| 40          | 5     | 145.80 | 5 | 149 | 5.657 | 5 | 149 | 8.944 |
| 50          | 4     | 182.25 | 4 | 193 | 4.000 | 4 | 193 | 7.211 |
| 1           | 11    | 60   | 16.33 | 3 | 17 | 4.000 | 3 | 17 | 2.828 |
| 70          | 7     | 456.33 | 3 | 461 | 6.000 | 3 | 457 | 2.828 |
| 80          | 3     | 1496.33 | 3 | 1761 | 4.000 | 3 | 1597 | 8.000 |
| 60          | 7     | 497.29 | 7 | 747 | 8.367 | 7 | 847 | 6.633 |
| 2           | 13    | 70   | 580.17 | 6 | 619 | 9.274 | 6 | 795 | 5.477 |
| 80          | 5     | 304.2/2360.17 | 5 | 399 | 8.000 | 6 | 2555 | 9.798 |
| 60          | 9     | 484.00 | 9 | 502 | 14.283 | 9 | 588 | 10.488 |
| 3           | 22    | 70   | 36.57/924.50 | 7 | 48 | 8.718 | 8 | 940 | 5.831 |
| 80          | 7     | 1056.57 | 7 | 1076 | 8.944 | 7 | 1160 | 11.136 |
| 60          | 4     | 110.25 | 4 | 111 | 15.232 | 4 | 119 | 8.367 |
| 4           | 27    | 70   | 930.25 | 4 | 1203 | 14.765 | 4 | 959 | 15.124 |
| 80          | 147.00 | 3 | 147 | 13.784 | 3 | 161 | 2.000 |
| 60          | 13    | 555.77/1501.78 | 13 | 583 | 24.083 | 14 | 1635 | 26.344 |
| 5           | 40    | 70   | 511.364 | 11 | 517 | 26.608 | 11 | 631 | 21.633 |
| 80          | 69.44/1102.50 | 9 | 73 | 19.442 | 10 | 1107 | 30.919 |

The lower bound $f_2^a$ is calculated by averagely assigning processing time to each workstation while ignoring the integer nature of ST in one workstation.

Significant improvement as the number of workstations is reduced.

Reaching the lower bound.

6. Conclusions
This paper studies the U-shaped disassembly line layout through a MCDM-based VNS approach with a specially designed decoding method. The MCDM considers multiple criteria and their interdependent correlations utilizing fuzzy set theory, grey relational analysis and Choquet fuzzy integral. The VNS approach with shaking and local searching incorporates a specially designed decoding method to fit the U-shaped layout. The computational results demonstrate that a better performance of U-shaped layout than that of straight line.

Acknowledgments
This project is supported by the Key Research and Development Program of Guangdong Province (No. 2019B090921001) and the Funds for National Natural Science Foundation of China (No. 51861165202).

Appendix

Figure 3. Priority matrix of the product.
References

[1] Ozceylan E and Paksoy T 2013 Reverse supply chain optimisation with disassembly line balancing Int. J. Prod. Res. 51 (20) 5985–6001
[2] McGovern S and Gupta S 2007 A balancing method and genetic algorithm for disassembly line balancing Eur. J. Oper. Res. 179 (3) 692–708
[3] Altekin F, Kandiller L and Ozdemirel N 2008 Profit-oriented disassembly-line balancing Int. J. Prod. Res. 46 (10) 2675–93
[4] Avikal S, Mishra P and Jain R 2014 A Fuzzy AHP and PROMETHEE method-based heuristic for disassembly line balancing problems Int. J. Prod. Res. 52 (5) 1306–17
[5] Avikal S, Jain R and Mishra P 2014 A Kano model, AHP and M-TOPSIS method-based technique for disassembly line balancing under fuzzy environment Appl. Soft Comput. 25 519–29
[6] Gungor A and Gupta S 2002 Disassembly line in product recovery Int. J. Prod. Res. 40 (11) 2569–89
[7] McGovern S and Gupta S 2006 Ant colony optimization for disassembly sequencing with multiple objectives Int. J. Adv. Manuf. Technol. 30 (5–6) 481–96
[8] Kalayci C and Gupta S 2013 Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem Exp. Res. Appl. 40 (18) 7231–41
[9] Igarashi K, Yamada T and Gupta S 2016 Disassembly system modeling and design with parts selection for cost, recycling and CO2 saving rates using multi criteria optimization J. Manuf. Syst. 38 151–64
[10] Ren Y, Zhang C, Zhao F, Tian G, Lin W, Meng L and Li H 2018 Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-Optimal algorithm J. Clean. Prod. 174 1475–86
[11] Ren Y, Zhang C, Zhao F, Triebe M and Meng L 2018 An MCDM-based multiobjective general variable neighborhood search approach for disassembly line balancing problem IEEE Trans. Syst., Man., Cybern. 50 (10) 3770–83
[12] Kalayci C, Polat O and Gupta S 2016 A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem Ann. Oper. Res. 242 (2) 321–54
[13] Pistolesi F, Lazzerini B, Mura M and Dini G 2018 EMOGA: a hybrid genetic algorithm with extremal optimization core for multiobjective disassembly line balancing IEEE Trans. Ind. Inform. 14 (3) 1089–98
[14] Miltenburg G and Wijngaard J 1994 The U-line line balancing problem Manage. Sci. 40 (10) 1378–88
[15] Wang K, Li X and Gao L 2019 A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem Comput. Ind. Eng. 130 634–49
[16] Hezer S and Kara Y 2015 A network-based shortest route model for parallel disassembly line balancing problem Int. J. Prod. Res. 53 (6) 1849–65