Date of publication December 15, 2022.

An Improved Adaptive Genetic Algorithm for U-shaped Disassembly Line Balancing Problem Subject to Area Resource Constraint

QI ZHANG[1] , WEISHUANG BAI[2], JIACUN WANG [3], XIWANG GUO [4], SHUJIN QIN [5], and LIANG QI [6]

[1]College of Information Science and Engineering, Shenyang University of Chemical Technology, Shenyang 110142, P R. China (e-mail: qizhang@syuct.edu.cn).
[2,4]Computer and Communication Engineering College, Liaoning Petrochemical University, Fushun, 113001, P R. China, (e-mail: hfgedu@163.com, x.w.guo@163.com.
[3]Dept of Comp. Sci. and Soft. Eng. at Monmouth University, W. Long Branch, NJ, USA (e-mail: jwang@monmouth.edu).
[5]Department of Logistics Management, Shangqiu Normal University, Shangqiu, 476000, P R. China, (e-mail: sjchin@vip.126.com).
[6]Department of Computer Science and Technology at Shandong University of Science and Technology, Qingdao, P. R. China (qiliangsdkd@163.com).

Corresponding author: Jiacun Wang (e-mail: jwang@monmouth.edu.)

This work is supported in part by NSFC under Grant Nos. 61573089, 62073069 and 51405075. Liaoning Province Education Department Scientific Research Foundation of China under Grant No. L2019027; Liaoning Province Dr. Research Foundation of China under Grant No. 20170520135. LiaoNing Revitalization Talents Program under Grant No. XLYC1907166; The Natural Science Foundation of Shandong Province under Grant ZR2019BF004.

ABSTRACT The disassembly, recovery, and reuse of waste products are attracting more and more attention. It not only saves resources and protects the environment but also promotes economic development. In a disassembly process, the disassembly line balancing problem is one of the most important problems.At present, the consideration of the space area of workstations is relatively small, and the relatively large use of the area of workstations can also better reduce costs. Aiming at the balancing problem of u-shaped disassembly line, a single-objective optimization mathematical model with area constraints is established with the goal of maximizing profits. In order to solve this problem, we refer to Adaptive Genetic Algorithm and improve its crossover and mutation operator. We adopt elite strategy to avoid premature convergence and improve the global search ability. Its effectiveness is proved by comparison with the optimization results of CPLEX. Experimental results also verify the feasibility of the proposed model and the superiority of the improved Adaptive Genetic Algorithm when solving large-scale instances over another algorithm. At the same time, the experimental results also verify the superiority and effectiveness of the improved Adaptive Genetic Algorithm algorithm by comparing with Random Search.

INDEX TERMS Disassembly line balancing; Adaptive genetic algorithm; U-shaped disassembly line.

I. INTRODUCTION

With the rapid development of science and technology, the demand for human consumption becomes more diverse, which makes the life cycle of products shorter and useless [1–5]. In addition, some of these discarded products cause great harm to the environment by littering. However, for partial end-of-life (EOF) products, we can disassemble useful parts and reuse them to reduce environmental pollution. In this way, disassembly and recycling are important links in the life cycle of products [6–11]. Through the disassembling and recycling of the product, the recycled valuable parts can be repaired and reused.However, for the hazardous parts, disassembling may reduce their harm and make them reach the green standard [12–16].

Since the concept of the Disassembly Line Balancing Problem (DLBP) is proposed [17], scholars have conducted a lot of research to disassemble products and arrange workstations. There are also various problems in the actual disassembly. For example, some workstations may be very busy, some workstations may not work, some workstations may place too many parts, and some workstations may place too few parts, etc. Therefore, reasonable planning of disassembly task allocation becomes a key factor. At present, intelligent optimization algorithms such as artificial bee colony algorithm [18], variable neighborhood search algorithm [19], and ant colony algorithm [20] are widely used in DLBP. Kalayci et al. establish DLBP mathematical model by considering the uncertainty of disassembly operation time, the number of opened workstations, and the priority to disassemble parts with high harm and high demand, and adopt an artificial swarm algorithm for optimization [21]. Wang et al. use the variable domain search algorithm to solve the workload problem of the balanced workstation and establish the corresponding DLBP mathematical model. A parallel dynamic neighborhood depth search algorithm is designed to maximize the reasonable assignment of employees [22]. Bentaha et al. [23] consider the issue of disassembly time and optimize the start-up cost of workstations and the disassembly cost of harmful parts by establishing a constrained binary programming model. Altekin et al. [24] successfully realize the profit maximization in the whole disassembly time by establishing a hybrid planning model. Koc et al. [25] establish the corresponding DLBP mathematical model according to the priority relationship of tasks, and solve the model by integer programming and dynamic programming to achieve the goal of minimizing the total number of workstations.

The traditional way of product disassembly is mainly comprised of a single person and a single machine, which takes a long time and is inefficient. The U-shaped disassembly [26] is a large-scale wany of production and presents a U-shaped distribution state. On each workstation, employees stand in the middle and disassemble products on both sidesof the line. Its main feature is that in a process of search, it can search from the front and back directions and then assign the corresponding disassembly task. That is to say, the first and last tasks in the disassembly sequence can be assigned to the same workstation. This line may also reduce the walking distance and time of the employees, therefore the efficiency of the U-shaped disassembly line is significantly increased [27–33]. Gu et al. [34] consider the uncertainty of disassembly time based on a U-shaped disassembly line and explore the balance problem of a random U-shaped disassembly line. Cai et al. [35] consider the material quality of waste products and the metal quality of different parts have a very important relationship with the cost, so they establish a U-shaped disassembly line mathematical model of parts classification. Zhang et al. [36] create a mathematical model of a multi-objective U-shaped disassembly line based on the cost of resource allocation, to solve the disassembly task interruption caused by equipment failure of the disassembly line. Wang et al. [37] consider some destructive disassembly problems and disassembly time, and establish a corresponding mathematical model of U-shaped disassembly line destruction, and adopt a discrete flower pollination algorithm to solve this problem.

The above papers consider disassembly time, disassembly parts, and damage problems, while the space area of workstations has not been considered much by researchers at present. In an actual process of disassembling, we encounter different sizes of workstations, different types of products, and different parts of the same procuts. This results in a very messy layout of workstations that is not easy to manage. When allocating the parts to be disassembled, we should consider the size of the space occupied by the parts, so that the area of the workstation is evenly distributed. In this way, a more standardized disassembly production line is established to improve the disassembly efficiency of the workstation. Therefore, it is necessary to explore the single-objective DLBP of workstations under the constraint of space area. In this paper, we propose a single-product U-shaped disassembly-line-balancing problem (SUDP). Adaptive Genetic Algorithm has been applied in many fields. Adaptive Genetic Algorithm has simple algorithm, strong directionality of genetic operator operation and good convergence. On this basis, the Adaptive Genetic Algorithm is improved to solve the above problems. The main contributions are as follows:

1) The problem of space area constraint based on U-shaped disassembly line is proposed, so the mathematical model of single-product and single-objective is established to maximize the profit.
2) The validity of the proposed model is verified by the results of CPLEX [38] and the improved AGA (IAGA), and the simulation results prove the efficiency and effectiveness of the IAGA algorithm in the SUDP problem.
3) To verify the feasibility of the IAGA algorithm in SUDP, it is compared the decomposition-based Random search (RS) [39] algorithm. Results clearly verify the effectiveness of the IAGA algorithm in solving such problems.

The rest of this paper is as follows. The description of the issues is presented in Section II. The explanation of the IAGA algorithm is presented in Section III. Section IV mainly describes experimental results and analysis. The last Section summarizes this article.

II. PROBLEM DESCRIPTION

A. PROBLEM STATEMENT

In traditional factory disassembly, the disassembled product moves from one workstation to the next at a speed on a conveyor belt, and at the same time the disassembled parts be classified into the material storage area of the workstation, and waiting for secondary transfer. However, this situation may cause problems. Because the length of our workstation is limited to a certain extent, but the size of the parts of the products may be small or large. If we assign a number of tasks containing large parts to the same workstation, the space of the workstation will increase and the cost increase, which is not a small problem for the factory. At the same time, the factory will constantly change the product as time goes by. However, the size of workstation is unchanged, which lead to space area problems and need to rearrange

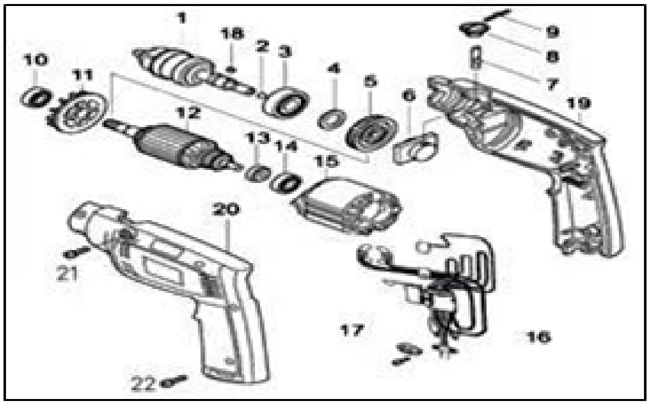


FIGURE 1. The schematics of a hammer drill.

a new reasonable disassembly plan. The characteristic of the U-shaped disassembly line is that one employee can disassemble tasks on both sides of the disassembly line at the same time, and the disassembled parts can be placed in the parts placement area to facilitate the re-transfer of parts. The maximum area of the workstation is a value obtained by considering the floor area of the parts of various products, so that the factory can replace different products. In the disassembly line problem under the area constraint, the area where parts are placed, the size of parts and the footprint of the workstation are considered. When each task is assigned to the workstation, the relationship between the overall footprint and working time is more balanced.

To make the task allocation scheme more reasonable,and to describe the relationship between each component more clearly, this paper adopts AND/OR diagram to represent the priority relationship between each component.This paper chooses the hammer drill case to study. Fig. 1 is the schematic diagram of a hammer drill, Fig.2 is the U-shaped disassembly line flow chart of a hammer drill, and Fig. 3 is the AND/OR diagram of a hammer drill. In Fig. 2, we can see the whole process of a set of disassembly sequences of a hammer drill. The number inside the circle represents the operation in the AND/OR diagram. The disassembly product enters the disassembly line from the entry side, after disassembly tasks reasonable allocate, produce the disassembly sequence, namely "1 - > 2 - > 4 - >8- > (-15) - > (-25) - > (-34)", Task 1 and task 34 are assigned to the first workstation.Task 1 on the entrance side and task 34 on the exit side. Task 2, 4, and 25 are assigned to the second workstation, task 2 and task 4 on the entrance side, and task 25 on the exit side. Task 8 and task 15 are assigned to the third workstation, task 8 on the entrance side and task 15 on the exit side. Finally, the waste can be treated.Tasks in the disassembly sequence are assigned to corresponding workstations for disassembly through time constraints, area constraints, priority constraints, and conflict constraints.

In a process of disassembling parts, it is impossible to disassemble them on the conveyor belt. We need to disassemble them on the workstation. However, the disassembled parts occupy a certain position on the workstation, which we call the floor area. This floor area must have its maximum value, which ensures our maximum profit under this condition. And we have to ensure that the parts we disassembled can be placed on the workstation. Take the disassembly of a product as an example. When a component in the workstation needs further disassembly, we first release its position on the workstation and recalculate the area occupied by its disassembled sub-parts. When a product is disassembled, we transfer the components on all workstations, that is, release all the area resources of the workstation.

Fig. 4 shows the changes in the floor area of each workstation during the disassembly of a product. During $t_1 \sim t_2$, workstation 1 performs task 1, that is, disassemble component 1. At this time, the floor area of workstation 1 is the floor area of the whole product at the beginning. $t_2 \sim t_3$ perform task 2. What is disassembled is component 3 disassembled from workstation 1 that can be further disassembled. At this time, the floor area of the parts in workstation 1 that do not need to be disassembled is shown in the figure. $t_3 \sim t_4$ time to execute task 4, which is carried out in workstation 2 and the component 3 disassembled by task 2 is the component disassembled by task 1 and needs to be disassembled. Perform task 8 within $t_4 \sim t_5$ time. Component 7 disassembled by task 8 is the component disassembled by task 4 and needs further disassembly. From t_5 to t_6, task 15 is executed. Task 15 is assigned to the exit side of workstation 3 for disassembly, but it is still disassembled on workstation 3. Assembly 11 is disassembled into assembly 46 and assembly 15. Task 25 is executed in $t_6 \sim t_7$ time, and task 25 is assigned to the exit side of workstation 2. Therefore, the floor area of workstation 2 at this time is the area of component 4 and component 43 that do not need to be disassembled before plus the floor area of component 15 that needs to be disassembled in task 25. Task 34 is executed in $t_7 \sim t_8$ time. Task 34 is assigned to the exit side of workstation 1. At this time, the floor area of workstation 1 is the area of component 2 that does not need to be disassembled before plus the area of components disassembled by task 34 at this time. At this point, the disassembly process of a product ends. $t_8 \sim t_9$ represents the display of the floor area of each part of workstation 1 and the recovery time.

The specific area changes on the inlet side and outlet side of workstation 1 for disassembling a product are shown in Fig. 5. $0 \sim t_1$ represents the appearance of the entrance side of workstation 1. $t_1 \sim t_2$ indicates the status of the workstation that has just started to allocate the disassembled product. At this time, component 1 is removed from the workstation. $t_2 \sim t_9$ represents the status of component 1 after disassembly. At this time, workstation 1 only performs task 1, and only component 2 remains on workstation 1. $0 \sim t_7$ represents the exit side of workstation 1. When the time goes to t_7, the exit side of the workstation 1 starts to perform task 34, that is, start to disassemble component 19. When the time goes to t_7, the exit side of the workstation 1 starts to perform task 34, that is, start to disassemble component 19. At this time,

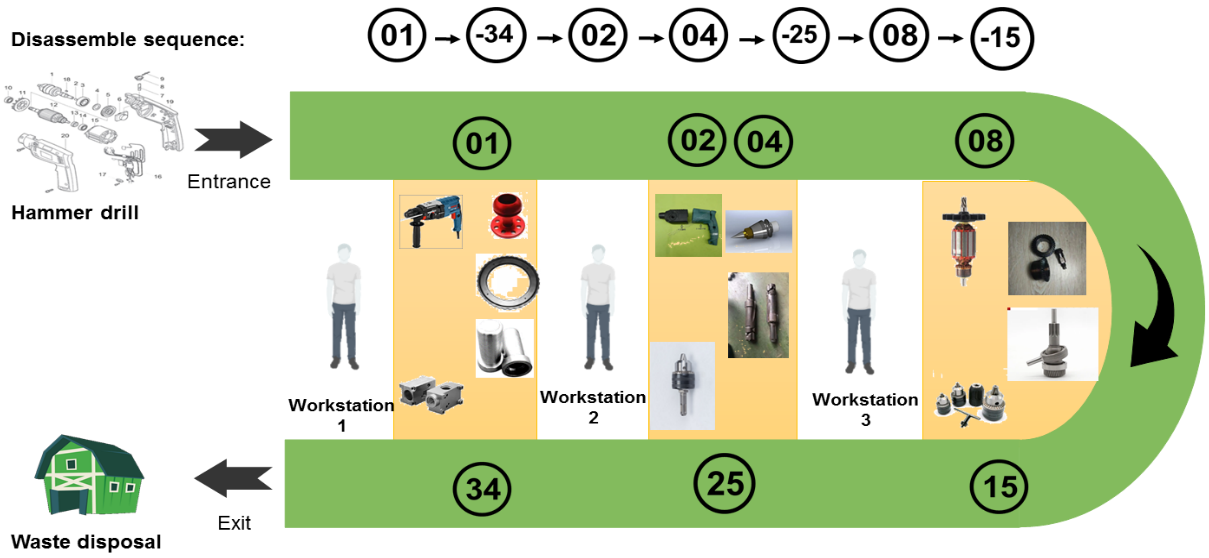


FIGURE 2. U-shaped disassembly line flow chart of hammer drill.

$t_7 \sim t_8$ represents the workstation status of disassembly task 34. $t_8 \sim t_9$ represents the distribution of components at the exit side of workstation 1 after task 34 is removed. At this time, the removal task of workstation 1 is over. Finally, the distribution of the disassembled components of workstation 1 is shown.

According to the above description of the AND/OR diagram, we introduce two matrices S and D to describe this problem.

1) A precedence matrix $S = [S_{jk}]$ is used to describe the sequence of disassembly operations, where j and k represent disassembly tasks.

$$S_{jk} = \begin{cases} 1, \text{if operation } j \text{ can be performed before operation } k \\ -1, \text{if operation j and k conflict with each other} \\ 0, \text{otherwise} \end{cases}$$

The precedence matrix for the washing machine is shown below:

$$S = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & -1 & 0 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 0 & -1 & -1 & 0 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & -1 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & 1 & -1 & -1 \\ 0 & -1 & 0 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & -1 & 1 & -1 \\ 0 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & -1 & 1 & -1 & 1 & -1 \\ 0 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & 1 & -1 & -1 & 0 \end{bmatrix}$$

2) A disassembly incidence matrix $D = [d_{ij}]$ where d_{ij} means the relationship between subassembly i and disassembly operation j:

$$d_{ij} = \begin{cases} 1, \text{if subassembly } i \text{ is obtained by operation } j \\ -1, \text{if subassembly } i \text{ is disassembled by operation } j \\ 0, \text{otherwise} \end{cases}$$

The conflict matrix for the washing machine is shown below:

$$D = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

In the work of this paper, we have the following assumptions

1) The matrices S and D mentioned above are known.
2) The workstations for removing wires are arranged in order.
3) Not all components in EOL products can be completely disassembled.
4) The workstation starts with the first task and ends when all tasks are assigned.

5) The actual disassembly time of each workstation is less than the specified cycle time, and the disassembly time of each disassembly task is known.
6) The floor area of the parts can be measured and never change, and the workstation is designed into a similar rectangle or square according to the area of most products to ensure that the parts can be placed.
7) Although the disassembled product is single, it is enough to make the disassembly line run normally without interruption.
8) There is no damage to the disassembled parts.

B. NOTATION DEFINITION

The symbols involved in the mathematical model are as follows:

- i Component index, $i \in \{1, 2, \ldots, I\}$ where I indicate the number of product components .
- j Disassembly Task index, $j \in \{1, 2, \ldots, J\}$, where J is the number of product disassembly tasks, and 0 is the virtual task.
- R_j The set of preorder tasks of task j .
- l Workstation Index, $l \in \{1, 2, \ldots, L\}$, where L is the number of workstations.
- a_i Footprint of component i.
- S Given the priority matrix of an AND/OR graph.
- D The association matrix of the given AND/OR graph.
- d_{ij} A member of the i-th row and the j-th column of D.
- t_j The disassembly time for the j-th task .
- c_j Disassembly cost per unit time for the j-th task.
- c_l Cost per unit area of workstation l .
- A_l Maximum storage area of workstation l.
- T Workstation cycle time.

Decision variables:

$$x_{jlw} = \begin{cases} 1,\text{If the product task } j \text{ is assigned to side} \\ \quad w \text{ of workstation } l \text{ and executed} \\ 0, \text{otherwise} \end{cases}$$

$$u_l = \begin{cases} 1,\text{if workstation } l \text{ is in use} \\ 0, \text{otherwise} \end{cases}$$

C. MATHEMATICAL MODEL

$$\max f_1 = \sum_{j=1}^{J}\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{w=0}^{1} d_{ij}v_i x_{jlw} - \sum_{l=1}^{L} c_l A_l u_l - \sum_{j=1}^{J}\sum_{l=1}^{L}\sum_{w=0}^{1} t_j c_j x_{jlw} \tag{1}$$

$$\sum_{j=1}^{J}\sum_{i=1}^{I}\sum_{w=0}^{1} x_{jlw} \geq 1. \tag{2}$$

$$\sum_{l=1}^{L}\sum_{w=0}^{1} x_{jlw} \leq 1, \quad j \in \{1, 2, \cdots, J\}. \tag{3}$$

$$\sum_{j=1}^{J}\sum_{w=0}^{1} x_{jlw} \geq u_l, \quad l \in \{1, 2, \cdots, L\}. \tag{4}$$

$$\sum_{j=1}^{J}\sum_{w=0}^{1} x_{jlw} \leq B{*}u_l, l \in \{1, 2, \cdots, L\}. \tag{5}$$

$$u_l \geq u_{l+1}, l \in \{1, 2, \cdots, L-1\}. \tag{6}$$

$$\sum_{j=1}^{J}\sum_{w=0}^{1} t_j x_{jlw} \leq T u_l, l \in \{1, 2, \cdots, L\}. \tag{7}$$

$$x_{jlw} \leq \sum_{j' \varepsilon R_j}\sum_{l'=1}^{l-1}\sum_{w'=0} x_{j'l'w'} * S_{jj'}, \\ j \in \{3, \cdots, J\}, l \in \{1, 2, \cdots, L\}, w = 0. \tag{8}$$

$$x_{jlw} \leq \sum_{j' \varepsilon R_j}\sum_{l'=1}^{L}\sum_{w'=0} x_{j'l'w'} * S_{jj'} + \sum_{j' \varepsilon R_j}\sum_{l'=l+1}^{L}\sum_{w'=1} x_{j'l'w'} * S_{jj'} + \sum_{j' \varepsilon R_j}\sum_{w'=1} x_{j'lw'} * S_{jj'}, j \in \{3, \cdots, J\}, \\ l \in \{1, 2, \cdots, L\}, w = 1 \tag{9}$$

$$\sum_{w=0}^{1}\sum_{l=1}^{L}(x_{jlw}+x_{klw}) \leq 1, j, k \in \{1, 2, \cdots, J\} \tag{10}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{w=0}^{1}(a_i d_{ij} x_{jlw} u_l - a_i \sum_{w'=0}^{1} d_{ij} x_{jw'}) \leq A_l, \\ l \in \{1, 2, \cdots, L\} \tag{11}$$

$$x_{jlw}, u_l \epsilon \{0, 1\}, j \in \{1, 2, \cdots, J\}, l \in \{1, 2, \cdots, L\}, \\ w \epsilon \{0, 1\} \tag{12}$$

The objective function (1) represents the maximum expected profit of disassembled products. Constraint (2) ensure that at least one task is performed. Constraint (3) ensure that each disassembly task can be executed only once. Constraint (4) indicates that each task per product can be assigned to only one workstation. Constraint (5) ensure that the workstation is started and you can start the disassemble task. Constraint (6) turn on workstations in turn,and no idle workstations are allowed. Constraint (7) indicates that the disassembly time of each workstation cannot exceed the cycle time. Constraint (8) and Constraint (9) the order of feasible disassembly tasks must satisfy the priority relationship.

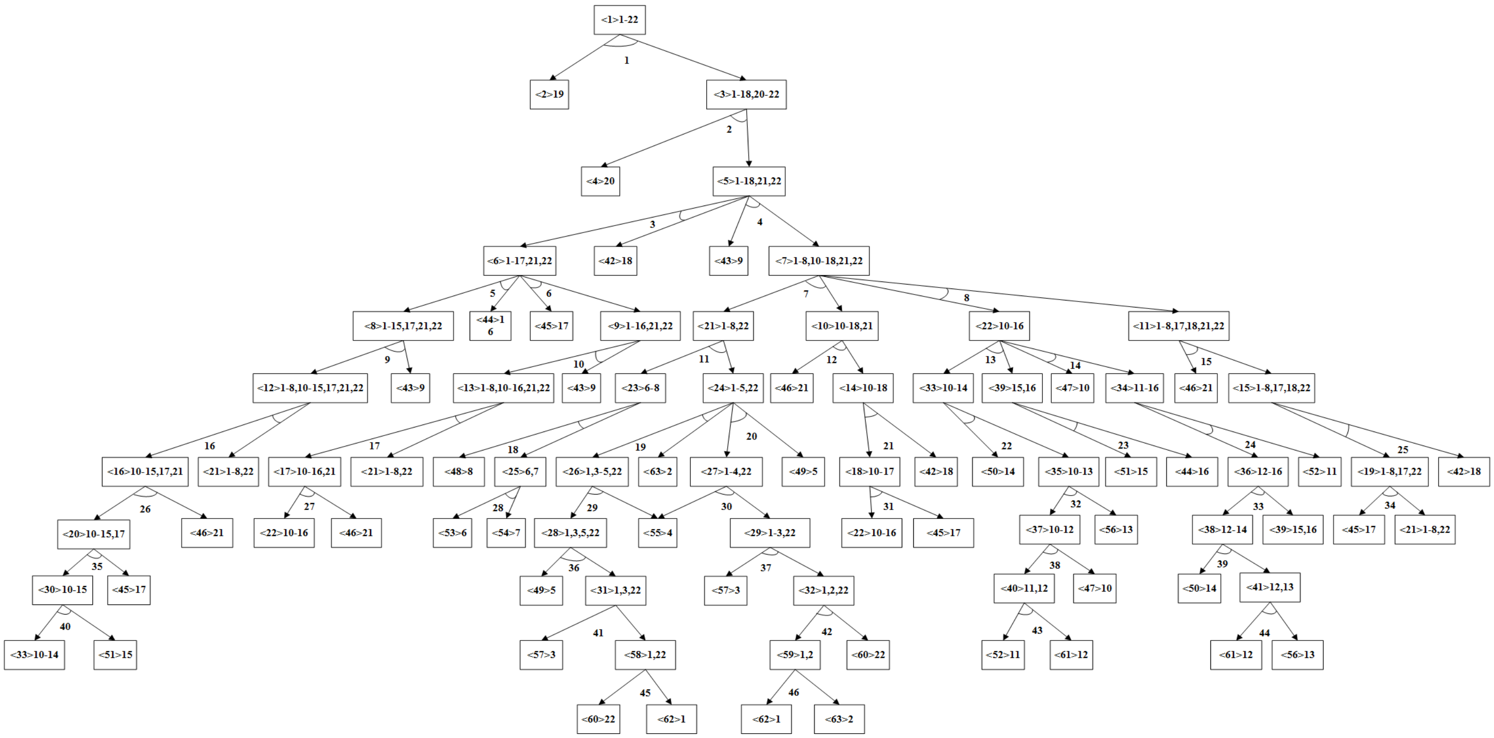


FIGURE 3. the AND/OR graph of a hammer drill.

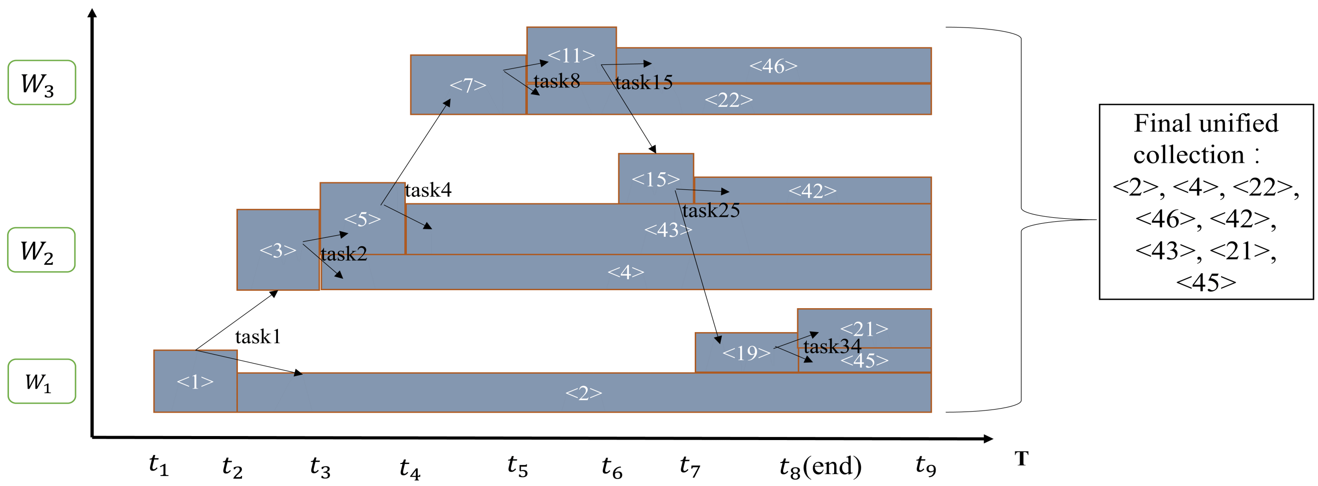


FIGURE 4. workstation area change diagram for disassembling a product.

Constraint (10) ensure a feasible disassembly task sequence, conflict relationships must be satisfied. Constraint (11) means that the actual storage area of each workstation shall not be greater than the specified area. Constraint (12) gives the range of decision variables.

III. PROPOSED ALGORITHM

AGA algorithm [40] is a kind of adaptive genetic algorithm, which has been applied in many fields. It has the advantages of simple algorithm, strong directional operation of genetic operator and good convergence. The IAGA algorithm changes the crossover probability and mutation probability according to the adaptivity value of each individual. When most individuals stay in the local optimal position, it needs to increase the crossover and mutation probability to improve the optimization ability of the algorithm. On the contrary, when individuals in the population are scattered in the solution space, the probability of crossover and mutation needs to be reduced to improve the convergence ability of the algorithm. The IAGA algorithm is more beneficial to solve the SUDP problem.

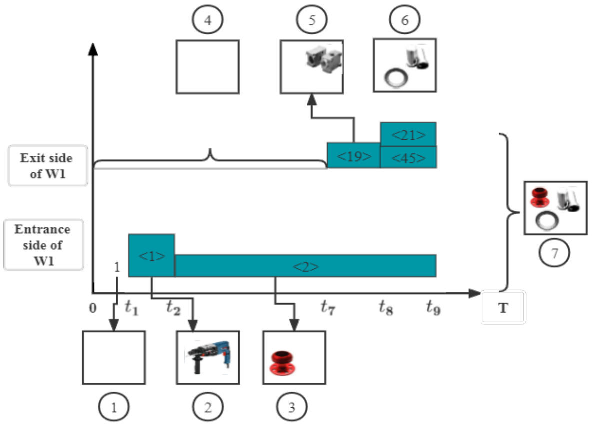


FIGURE 5. Schematic diagram of area change at the inlet and outlet side of workstation1.

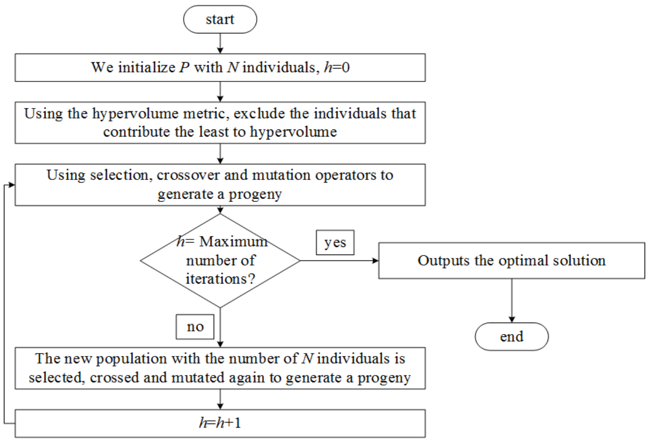


FIGURE 6. Flow chart of IAGA algorithm.

A. INTRODUCTION AND IMPROVED OF BASIC AGA

The main flow chart of the IAGA algorithm is shown in Fig. 6. The IAGA algorithm improves the crossover mutation operator on the basis of the traditional genetic algorithm, and adopts the elite strategy to avoid the problem of premature convergence and improve the global search ability. Meanwhile, it avoids the problem of sawtooth and has relatively strong local search ability.

The main idea of the IAGA algorithm is to start from the initial population, calculate the adaptive value, and use the cross mutation operator to generate a new individual. In order to ensure that the individual population remains the same size, the elite individual joins the next generation of the new population. If there is no individual in the population with a higher fitness value than the elite individual, the elite individual is added to the new generation of the population, And the individual with the smallest fitness value in the new species group is eliminated. The pseudo-code process of the IAGA algorithm is shown in Algorithm 1.

Algorithm 1 IAGA algorithm

Input: population P
Output: best individual
n = Population size
$P[t]$ = initialize $P(n)$
while Termination ConditionMet() == false **do**
  fitnessvalue=fitness(n);
  for t=1 **do**
    gbestfitness0 = max(fitnessvalue);
    parents = select parents($P[t]$);
    $P[t+1]$ = crossover(parents);
    $P[t+1]$ = mutate($P[t+1]$);
    fitnessvalue = fitness($P(n,t)$);
    gbestfitness1 = max(fitnessvalue);
    if gbestfitness1 $<$ gbestfitness0 **then**
      pop(gbestindividual0)
      fitnessvalue(gbestindex1) = gbestfitness0;
    end if
    fitnessvalue = fitness($P(n,t)$);
  end for
end while

To enrich the whole population, some bias towards the inferior state solution must also be established, so we set up the Reduce algorithm. In order to rank the solutions of considerable importance differently, while keeping the runtime complexity relatively small, the individual with the highest $d(s, p(t))$ value in the worst front will be eliminated; otherwise, the individual with the highest $d(s, p(t))$ value equal to 0 will be replaced by the y option. Reduce is shown in algorithm 2.

Algorithm 2 Reduce(P)

Input: a population P
Output: The result of eliminating the detection element P/r
if $v < 1$ **then**
  $r = argmin_{s\epsilon y_v}\left[d\left(s,p\right)\right]$
else if $v > 1$ **then**
  $r = argmin_{s\epsilon y_1}\left[y\left(s,y_1\right)\right]$
end if

B. THE INTERPRETATION OF ENCODING AND DECODING

According to the characteristics of the disassembly line, two search strategies should be obtained. Firstly, there should be a disassembly sequence of disassembly products. Secondly, a scheme of disassembly task allocation to workstations should be given, that is, how to allocate the disassembly sequence is the most reasonable. We represent a solution as a stringβ=(β_1,β_2) ,among this β_1=(θ_1,θ_2,θ_3,$\ldots$,θ_j,$\ldots\theta_J$), β_1=β_2=$x_1, x_2, x_3, \ldots, x_j, \ldots x_J$,$\beta_1$ represents the sequence of disassembly tasks, consisting of the index number of the disassembly tasks. β_2 consists of a set of binary strings, zeros, and ones. If x_j=1, then the task on the j index in the disassembly sequence be executed, otherwise not.

Obviously, randomly generate solutions may be infeasible, so we need a series of adjustment work to adjust infeasible solutions into feasible solutions step by step, and the complete adjustment should not only meet the priority relationship but also meet the cycle time and area constraints of the workstation. The adjustment process is divided into three stages. The first stage is to find out the two adjacent tasks that do not conform to the priority relationship by traversing all the tasks and achieving the goal of conforming to the priority relationship by changing their positions. The second stage is to adjust the array of binary strings to a priority array, which is essentially the same process as the first stage until all tasks have been traversed completely. The third stage is to avoid the solution being affected by the conflict relationship, that is, to traverse all tasks, find the task with conflict relationship between two tasks, and set it to 0. The adjustment process of the three stages is shown in Fig.7. After three stages, we have a new solution β'. The decoding process of β' is shown in Fig. 8.

C. CROSSOVER OPERATOR

The main idea of the crossover operator is to cross-combine the children of two parents. The children of the first parent and the children of the second parent are combined based on randomly generated binary strings. The function of a binary string is to add the child of the first parent generation to the new solution when the binary is 0, and the child of the second parent generation to the new solution when the binary is 1, until all the children are added to the solution, and finally a new solution is obtained. However, this solution may not meet the requirements of priority relation, so the sequence adjustment shown in Fig. 7 above is needed to adjust the newly obtained solution and obtain the solution that meets the requirements. The crossover diagram is shown in Fig. 9.

D. MUTATION OPERATOR

The idea of the mutation operator is to find a random mutation point in the parents, the pre-task and post-task are found according to the mutation point, and then the task of the mutation point is randomly inserted into the parent generation except for the pre-task and post-task. Finally, a new solution with priority relation is obtained. The mutation diagram is shown in Fig. 10.

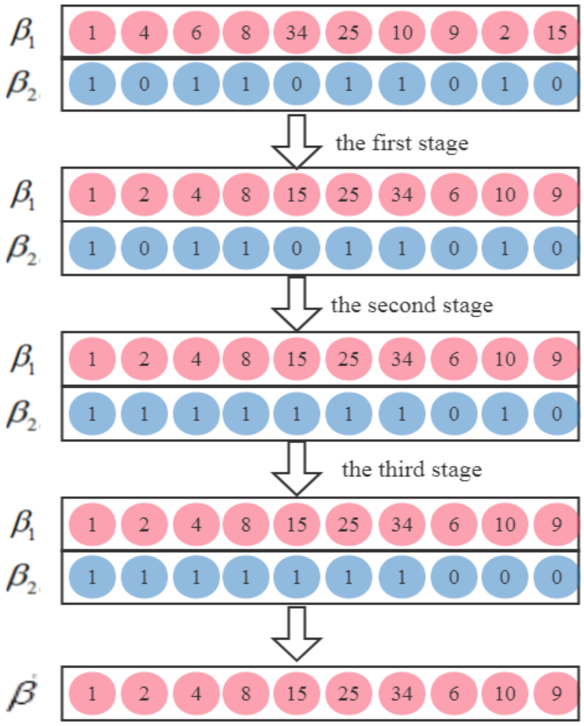


FIGURE 7. Disassembly sequence adjustment process.

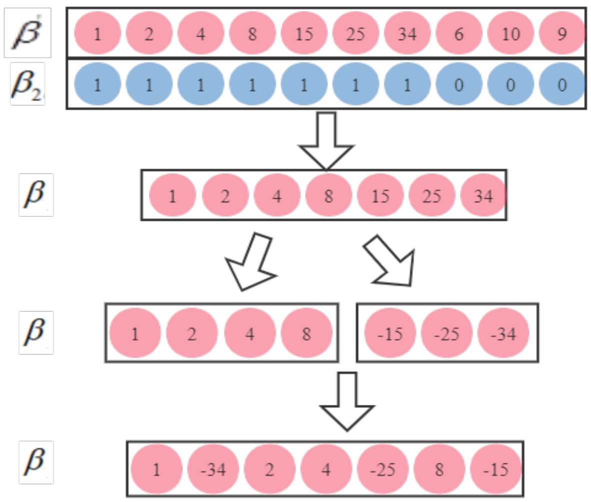


FIGURE 8. The decoding process.

TABLE 1. The solution result of flashlight under IAGA algorithm

Num	Disassemble scheme	profit	Space Area
1	$(1, 3, 7, 6, 9, 10)$	1188	1.48
2	$(1, 3, 9, 10)$	963.3	1.30
3	$(1, 9, 3, 6, 10)$	1070.2	1.45
4	$(1, 9, 10, 3)$	906.3	1.37

IV. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the correctness of the mathematical model in this paper, the IAGA and CPLEX are used to verify the model respectively. The operating environments are the Jmetal framework [41] and IBM ILOG CPLEX Optimization Studio. Meanwhile, to demonstrate the superiority of the IAGA algorithm, we chose the RS algorithm for comparison. To ensure the fairness and effectiveness of the experiment, all experiments are conducted on a Core (TM) I-3230m 2.60ghz 4G computer.

It can be seen from Table 2 and Table 2 that the corresponding profit and space area of different disassembly schemes are also different. When the demolition schemes are similar, the profit and area change within a certain range.

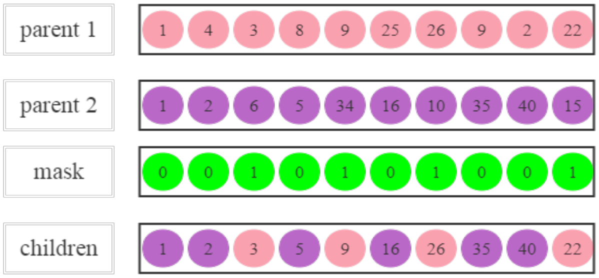


FIGURE 9. The crossover diagram.

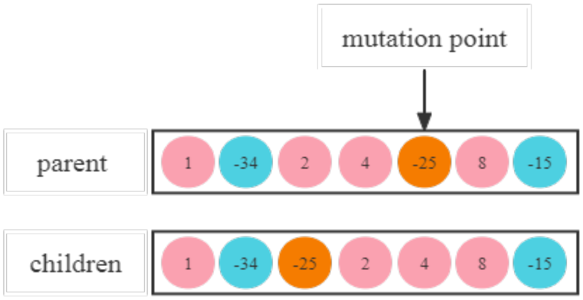


FIGURE 10. The mutation diagram.

In Table 1, the maximum profit is 1188, and the change range of space area is 1.30 1.48. In Table II, the maximum profit is 1775.1, and the change range of space area is 1.41 1.44. Therefore, leaders can choose the disassembly scheme suitable for their factory according to different cases and the actual situation. If it is a small case and wants a high profit, select case 1 in Table 1. At this time, the space area of the workstation is 1.48; If it is a large case and wants a high profit, choose scheme 4 in Table 2. At this time, the space area of the workstation is 1.42. From the data of a small case and a large case above, it can be seen that to enable the same factory to disassemble multiple products, the area of the workstation needs to be set to an appropriate size so that the disassembled products can be replaced more conveniently. Through the analysis of the cases, the area of the workstation can be kept unchanged and the disassembled products can be replaced at any time. At the same time, to maximize the benefits, leaders need to choose the appropriate scheme for disassembly.

TABLE 2. The solution result of hammer drill under IAGA algorithm

Num	Disassemble scheme	profit	Space Area
1	$(1, 2, 4, 7, 12, 21, 31, 13, 23, 11, 18, 31, 30)$	1501.4	1.41
2	$(1, 2, 4, 7, 12, 21, 31, 11, 18, 20, 30, 37, 13, 23)$	1566.8	1.43
3	$(1, 2, 4, 8, 13, 23, 15, 25, 34, 11, 18, 19, 29, 36)$	1772.2	1.44
4	$(1, 2, 3, 6, 10, 17, 27, 11, 18, 28, 20, 30, 13, 23)$	1775.1	1.42

A. TEST INSTANCES

We use four cases, Flashlight [42] Ballpoint Pen [43] Washing Machine [44], and Hammer Drill [45] as test cases.The number of tasks of these cases is 10, 13, 13, and 46, respectively, and the number of components is 15, 15, 15, and 63, respectively.

B. RESULTS COMPARED WITH CPLEX

We use the IAGA algorithm and CPLEX respectively to compare the profit and running time of the four cases. The results of running are shown in Tables 3 and 4, respectively. The population size of the IAGA algorithm is 100, and the maximum number of iterations is 1000, and the probability of crossover and mutation is 0.9 and 0.3 respectively, and the number of runs is 20. In order to show the real validity of the test results, there is a large difference in the number of tasks used. The smallest case is a flashlight with 10 tasks, and the largest case is a hammer drill with 46 tasks, the difference in the number of tasks is 36.

Table 3 shows the solution results of the instance set based on the IAGA, and Table 4 shows the solution results of the instance set based on CPLEX. To test the speed and effectiveness of the experimental scheme, the instance set solution result of the IAGA is compared with that of CPLEX.

As shown in Table 4, from the perspective of maximum profit, CPLEX is always the optimal solution no matter it is a small case or a large case, and the numerical value is similar to that of the IAGA. However, from the perspective of running time, the running time of the IAGA flashlight is 0.17 seconds, while that of the CPLEX flashlight is 0.21 seconds, which is slower than that of the IAGA for small cases; the IAGA hammer-drill run time is 0.45 seconds and the CPLEX hammer-drill run time is 0.74 seconds. In the large case, the IAGA algorithm is faster than the CPLEX and in the remaining two cases the IAGA is also faster than the CPLEX. So in general, CPLEX runs slower.

TABLE 3. Results of solving the instances set based on the IAGA.

Products	Result	Profit	time
Flashlight	$(1, 3, 7, 6, 9, 10)$	1188	390 ms
Ballpoint Pen	$(2, 6, 10, 11, 13)$	924	171 ms
Washing Machine	$(1, 2, 5, 11, 10)$	890	312 ms
Hammer Drill	$(1, 2, 4, 7, 12, 21, 31, 11, 20, 13, 23)$	1511.5	375 ms

TABLE 4. Results of solving the instances set based on CPLEX.

Products	Result	Maximum Profit	Computation time
Flashlight	$(1, 3, 7, 6, 9, 10)$	1188	2990 ms
Ballpoint Pen	$(2, 6, 10, 11, 13)$	924	3240 ms
Washing Machine	$(1, 2, 5, 10, 11)$	890	3630 ms
Hammer Drill	$(1, 2, 4, 7, 12, 21, 31)$	1512	11300 ms

C. COMPARE WITH RANDOM SEARCH

Table 5 shows different iteration data results of the two algorithms based on a Hammer Drill, and Table 6 shows

TABLE 5. Data results of different iterations of TWO ALGORITHMS based on hammer drill.

Iterations	the IAGA	RS
10	982	998
20	1000	1035
30	1064	1052
40	1074	1054
50	1108	1054
60	1117	1054
70	1117	1054
80	1117	1054
90	1117	1054

TABLE 6. Data results of different iterations of TWO ALGORITHMS based on radio.

Iterations	the IAGA	RS
50	134.2	228.6
100	360.1	273.7
200	456.9	328.7
300	599.4	435.1
400	632.7	598.6
500	671.2	598.6
600	671.2	598.6
700	671.2	598.6

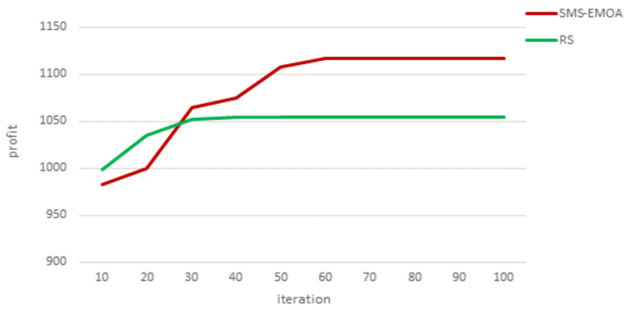


FIGURE 11. Comparison of hammer drill target values.

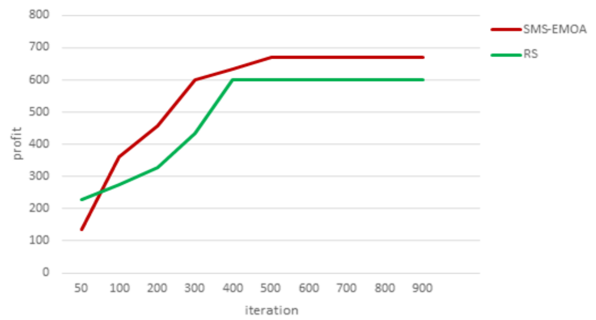


FIGURE 12. Comparison of radio target values.

different iteration data results of the two algorithms based on Radio. It is obvious from the data in the two tables that the comparison of the data in the two tables shows the advantages of the IAGA algorithm.

It can be seen from Table 5 that after 20 iterations, the profit value of the IAGA is significantly higher than that of RS, and the same effect is achieved with more iterations in the later. The case tasks of the ballpoint pen are relatively few, so the number of iterations be relatively large before it changes. When the iteration is 100 times, the profit value of the IAGA began to be higher than that of RS. As the number of iterations increase, the profit value of the IAGA and RS algorithms also increase. When the iteration is 200 times, RS has priority as the optimal solution, but the IAGA could continue to search. RS has already fallen the local optimal solution, which shows the superiority of the IAGA algorithm and the genetic operator it uses can better avoid the local optimal situation.

Fig. 11 is a comparison of target values for hammer drills and Fig. 12 is a comparison of target values for radio. These two pictures can more vividly explain the situation described above. It can be seen from the figure that the profit value change curves of hammer drill and ballpoint pen in different iterations. In the hammer drill case, RS starts to level off after less than 50 iterations, while the IAGA starts to level off after 50 iterations. In the case of the ballpoint pen, the effect is even more obvious, with RS becoming smooth after 200 iterations and the IAGA becoming smooth after 300 iterations. Through the above statements, we can more definitely see the feasibility and superiority of the IAGA algorithm.

V. CONCLUSION

The IAGA algorithm is proposed to solve the single product U-shaped disassembly line balance problem. Firstly, to verify the correctness of the mathematical model designed, CPLEX is used to test the mathematical model and the running disassembly sequence is correct. Compare with CPLEX running time, the larger the case information, the smaller the IAGA running time, and the better the superiority. The mathematical model mainly considers the maximum profit. After several experiments, the accuracy and superiority of the algorithm are verified by comparison with the other two algorithms. There is a very small difference in the number of tasks of the cases sought in this paper. In future work, we will find more cases with different tasks to verify the algorithm, making the experiment more rigorous and more convincing.

REFERENCES

[1] X. Guo, Z. Zhang, L. Qi, S. Liu, Y. Tang, and Z. Zhao, "Stochastic hybrid discrete grey wolf optimizer for multi-objective disassembly sequencing and line balancing planning in disassembling multiple products," IEEE transactions on automation science and engineering: a publication of the IEEE Robotics and Automation Society, no. 19-3, 2022.

[2] X. Cui, X. Guo, M. Zhou, J. Wang, S. Qin, and L. Qi, "A discrete whale optimization algorithm for disassembly line balancing with carbon emission constraint," IEEE Robotics and Automation Letters, vol. 8, no. 5, pp. 3055–3061, 2023.

[3] S. Qin, J. Li, J. Wang, X. Guo, S. Liu, and L. Qi, "A salp swarm algorithm for parallel disassembly line balancing considering workers with government benefits," IEEE Transactions on Computational Social Systems, 2023.

[4] X. Guo, Z. Bi, J. Wang, S. Liu, S. Qin, and L. Qi, "Reinforcement learning for disassembly system optimization problems: A survey," International Journal of Network Dynamics and Intelligence, vol. 2, no. 1, pp. 1–14, 2023.

[5] S. Qin, S. Zhang, J. Wang, S. Liu, X. Guo, and L. Qi, "Multi-objective multi-verse optimizer for multi-robotic u-shaped disassembly line balancing problems," IEEE Transactions on Artificial Intelligence, 2023.

[6] X. Guo, T. Wei, J. Wang, S. Liu, S. Qin, and L. Qi, "Multiobjective u-shaped disassembly line balancing problem considering human fatigue index and an efficient solution," IEEE Transactions on Computational Social Systems, 2022.

[7] L. Ding, J. Tan, and Y. X. Feng, "Multi-objective optimization for disassembly line balancing based on pareto ant colony algorithm," Computer Integrated Manufacturing System, vol. 15, no. 7, p. 9, 2009.

[8] G. Tian, M. Zhou, and J. Chu, "A chance constrained programming approach to determine the optimal disassembly sequence," IEEE T AUTOM SCI ENG, vol. 10, no. 4, pp. 1004–1013, 2013.

[9] X. W. G. Y.P. Fu, M. C. Zhou and L. Qi, "Stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling via fruit fly optimization algorithm," Journal of Cleaner Production, vol. 278, 2020.

[10] H. J. Kim and P. Xirouchakis, "Capacitated disassembly scheduling with random demand," Int. J. Product. Res, vol. 48, no. 23, pp. 7177–7194, 2010.

[11] E. Zceylan, C. B. Kalayci, A. Güngr, and S. M. Gupta, "Disassembly line balancing problem: a review of the state of the art and future directions," International Journal of Production Research, no. 1, pp. 1–23, 2018.

[12] J. Jiang, Z. Q. Zhang, M. K. Xie, and N. Cai, "An improved wolf pack algorithm for multi-objective disassembly line balancing problem with spatial constraints," Computer Integrated Manufacturing Systems, vol. 27, no. 6, pp. 1569–1581, 2021.

[13] K. Zhang, Y. Su, X. Guo, L. Qi, and Z. Zhao, "Mu-gan: Facial attribute editing based on multi-attention mechanism," IEEE/CAA Journal of Automatica Sinica, no. 008-009, 2021.

[14] S. M. Mcgovern and S. M. Gupta, "Ant colony optimization for disassembly sequencing with multiple objectives," international journal of advanced manufacturing technology, vol. 30, no. 5-6, pp. 481–496, 2006.

[15] Y. Wang, Y. Xie, Y. Ren, and C. Zhang, "A mcdm-based meta-heuristic approach for u-shaped disassembly line balancing problem," Journal of Physics Conference Series, vol. 1828, no. 1, p. 012159, 2021.

[16] S. Agrawal and M. K. Tiwari, "A collaborative ant colony algorithm to stochastic mixed-model u-shaped disassembly line balancing and sequencing problem," International Journal of Production Research, vol. 46, no. 6, pp. 1405–1429, 2008.

[17] S. M. Mcgovern and S. M. Gupta, "Combinatorial optimization methods for disassembly line balancing," vol. 179, no. 3, pp. 692–708, 2004.

[18] Z. Zeqiang, H. U. Yang, and C. Chong, "Improved artificial bee colony algorithm for disassembly line balancing problem," Journal of Southwest Jiaotong University, vol. 51, no. 5, pp. 910–917, 2016.

[19] Y. Su, Z. Zhang, and H. Yang, "A variable neighborhood search algorithm for disassembly line balancing problem," Modern Manufacturing Engineering, no. 10, pp. 19–25, 2016.

[20] X. Zhu, Z. Zhang, X. Zhu, J. Hu, S. J. University, and Y. N. University, "An ant colony optimization algorithm for multi-objective disassembly line balancing problem," China Mechanical Engineering, vol. 25, no. 8, pp. 1075–1079, 2014.

[21] Can, B., Kalayci, Arif, Hancilar, Askiner, Gungor, Surendra, M., and Gupta, "Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm - sciencedirect," Journal of Manufacturing Systems, vol. 37, no. Part 3, pp. 672–682, 2015.

[22] J. L. S. W. Wang, X. P. Guo, "Modeling of class ii disassembly line equilibrium problem and optimization," Chinese Management Science, vol. 29, no. 10, pp. 118–125, 2019.

[23] M. L. Bentaha, O. Battaïa, and A. Dolgui, "An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times," International Journal of Production Research, vol. 53, no. 6, pp. 1807–1818, 2014.

[24] F. T. Altekin, L. Kandiller, and N. E. Zdemirel, "Profit-oriented disassembly line balancing," International Journal of Production Research, vol. 46, no. 10, pp. 2675–2693, 2008.

[25] A. Koc, I. Sabuncuoglu, and E. Erel, "Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an and/or graph," IIE Transactions, vol. 41, no. 10, pp. 866–881, 2009.

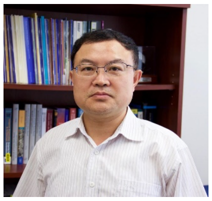
[26] Q. X. Xiao, X. P. Guo, and G. U. Xin-Jun, "The stochastic mixed-model u-shaped disassembly line balancing and sequencing optimization problem with multiple constraints," Industrial Engineering and Management, pp. 1007–5429, 2019.

[27] H. Jiang, J. Yi, S. Chen, and X. Zhu, "A multi-objective algorithm for task scheduling and resource allocation

in cloud-based disassembly," Journal of Manufacturing Systems, vol. 41, pp. 239–255, 2016.
[28] L. Qi, M. Zhou, and W. Luan, "A dynamic road incident information delivery strategy to reduce urban traffic congestion," IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 5, pp. 934–945, 2018.
[29] Y. Tan, M. C. Zhou, Y. Wang, X. Guo, and L. Qi, "A hybrid mip-cp approach to multistage scheduling problem in continuous casting and hot-rolling processes," IEEE Transactions on Automation Science and Engineering, vol. PP, no. 99, pp. 1–10, 2019.
[30] X. T. Zhu, Z. Q. Zhang, and X. M. e. Zhu, "To solve the multi-objective disassembly line balancing problem of a kind of ant colony algorithm," Chinese Journal of Mechanical Engineering, vol. 25, no. 8, p. 1075 1079, 2014.
[31] Y. Tan, M. C. Zhou, Y. Zhang, X. Guo, and Y. Wang, "Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting," IEEE Transactions on Automation Science and Engineering, vol. PP, no. 99, pp. 1–15, 2020.
[32] X. W. G. Y. P. Fu, M. C. Zhou and L. Qi, "Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via the bi-population evolutionary algorithm," IEEE Trans. Syst. Man. Cy-s, 2020.
[33] Y. F. Wu, E. M. Chen, and J. Ni, "U-shaped disassembly line balance optimization for low carbon and high efficiency," Modular machine tool and automatic processing technology, no. 03, pp. 153–158, 2021.
[34] G. U. Xin-Jun and X. P. Guo, "Multi-objective evolutionary algorithm optimization of stochastic mixed-model u-shaped disassembly line balancing and sequencing problem," Operations Research and Management Science, vol. 26, no. 9, p. 52 61, 2017.
[35] H. Cai, T. Yu, and C. Xia, "Quality-oriented classification of aircraft material based on svm," Mathematical Problems in Engineering,2014,(2014-11-6), vol. 2014, no. 3, 2014.
[36] Q. Zhang, Z.K.and Tang and L. Zhang, "Multi-objective u-shaped assembly line balancing under multi-resource restriction," Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS, vol. 23, no. 11, pp. 2431–2441, 2017.
[37] K. Wang, L. Gao, and X. Li, "A multi-objective algorithm for u-shaped disassembly line balancing with partial destructive mode," Neural Computing and Applications, vol. 32, no. 16, pp. 12 715–12 736, 2020.
[38] Xiwang, Guo, Shixin, Liu, MengChu, Zhou, Guangdong, and Tian, "Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and petri nets." IEEE transactions on cybernetics, vol. 46, no. 11, 2016.
[39] C. M. Li, "Random search versus optimal search," Military Operations Research and Systems Engineering, vol. 2001, no. 2, pp. 18–20.
[40] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," IEEE Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 100–116, 2003.
[41] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," Advances in Engineering Software, vol. 42, no. 10, pp. 760–771, 2011.
[42] Y. Tang, M. C. Zhou, E. Zussman, and R. Caudill, "Disassembly modeling, planning, and application," Journal of Manufacturing Systems, vol. 21, no. 3, pp. 200–217, 2002.
[43] Q. Lu, Y. Ren, H. Jin, L. Meng, and J. W. Sutherland, "A hybrid metaheuristic algorithm for a profit-oriented and energy-efficient disassembly sequencing problem," Robotics and Computer-Integrated Manufacturing, vol. 61, 2019.
[44] P. Nowakowski, "A novel, cost efficient identification method for disassembly planning of waste electrical and electronic equipment," Journal of Cleaner Production, vol. 172, no. pt.3, pp. 2695–2707, 2018.
[45] Y. Fu, J. Ding, H. Wang, and J. Wang, "Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in industry 4.0-based manufacturing system," Applied Soft Computing, vol. 68, pp. 847–855, 2017.

QI ZHANG received her B.S. degree and M.S. degree in applied mathematics from Shenyang Normal University, Shenyang, China in 2012 and 2015, respectively, and Ph.D. degree in Systems Engineering from Northeastern University, Shenyang, China in 2022. She is currently a lecturer of the College of Information Engineering at Shenyang University of Chemical Technology. Her research focuses on bin packing problem, steel production scheduling and planning, mathematical programming, and intelligent optimization algorithm.

WEISHUANG BAI She received her bachelor's degree from Liaoning Shihua University in 2020.Currently, she is a graduate student majoring in electronic information in Liaoning Shihua University, Fushun, China.



JIACUN WANG (SM'00) received the Ph.D. degree in computer engineering from Nanjing University of Science and Technology (NUST), China, in 1991. He is currently a Professor of software engineering at Monmouth University, West Long Branch, New Jersey, USA. From 2001 to 2004, he was a member of scientific staff with Nortel Networks in Richardson, Texas. Prior to joining Nortel, he was a research associate of the School of Computer Science, Florida International University at Miami. His research interests include software engineering, discrete event systems, formal methods, machine learning, and real-time distributed systems. He authored Timed Petri Nets: Theory and Application Kluwer, 1998), Real-time Embedded Systems (Wiley, 2018) and Formal Methods in Computer Science (CRC, 2019), edited Handbook of Finite Stat Based Models and Applications (CRC, 2012), and published over 130 research papers in journals and conferences. Dr. Wang was an Associate Editor of IEEE Transactions on Systems, Man and Cybernetics, Part C, and is currently Associate Editor of IEEE/CAA Journal of Automatica Sinica. He has served as general chair, program chair, and special sessions chair or program committee member for many international conferences. He is a senior member of IEEE.

XIWANG GUO received his B.S. degree in Computer Science and Technology from Shenyang Institute of Engineering, Shenyang, China, in 2006, M.S. degree in Aeronautics and Astronautics Manufacturing Engineering. from Shenyang Aerospace University, Shenyang, China, in 2009, Ph. D. degree in System Engineering from Northeastern University, Shenyang, China, in 2015. He is currently an associate professor of the College of Computer and Communication Engineering at Liaoning Shihua University. From 2016 to 2018, he was a visiting scholar of Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He has authored 60+ technical papers in journals and conference proceedings, including IEEE Transactions on Cybernetics, IEEE Transactions on System, Man and Cybernetics: Systems, IEEE Transactions on Intelligent Transportation Systems, and IEEE/CAA Journal of Automatica Sinica. His current research interests include Petri nets, remanufacturing, recycling and reuse of automotive, intelligent optimization algorithm.

SHUJIN QIN received his B.S. degree in Information and Computing Science from Tianjin Polytechnic University, Tianjin, China in 2008, M.S. degree in Operational Research and Cybernetics from University of Science and Technology Liaoning, Anshan, China in 2011, and Ph. D. degree in System Engineering from Northeastern University, Shenyang, China in 2019. He joined Shangqiu Normal University, Shangqiu, China in 2019, and is now a lecturer of logistics management. His current research interests include large-scaled integer programming, cutting stock problem, vehicle routing problem, traveling repairman problem and intelligent optimization algorithm.

LIANG QI (S'16-M'18) received his B.S. degree in Information and Computing Science and M.S. degree in Computer Software and Theory from Shandong University of Science and Technology, Qingdao, China, in 2009 and 2012, respectively, and Ph.D. degree in Computer Software and Theory from Tongji University, Shanghai, China in 2017. He is currently a lecturer of Computer Science and Technology at Shandong University of Science and Technology, Qingdao, China. From 2015 to 2017, he was a visiting student in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He has authored over 50 technical papers in journals and conference proceedings, including IEEE Transactions on System, Man and Cybernetics: Systems, IEEE Transactions on Intelligent Transportation Systems, and IEEE/CAA Journal of Automatica Sinica. He received the Best Student Paper Award-Finalist in the 15th IEEE International Conference on Networking, Sensing and Control (ICNSC'2018). His current research interests include Petri nets, discrete event systems and optimization algorithms.