Date of publication December 15, 2022.

Discrete Bat Optimizer for Disassembly Line Balancing Problem

QI ZHANG[1] FUGUANG HUANG[2], JIACUN WANG [3], XIWANG GUO[4], SHUJIN QIN[4] LIANG QI[4]

[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 recycling of end-of-life (EOL) products is the primary link in the remanufacturing process. EOL products rely on disassembly lines to retain valuable parts for remanufacturing. In this work, a disassembly line balancing model is established based on an AND/OR graph. It takes precedence relation, cycle time restriction, failure risk, and time uncertainty into consideration and aims to maximize the dismantling profit and minimize the energy consumption. Then, a multi-objective discrete bat optimizer based on Pareto rules is designed according to the problem model, and a precedence preserving crossover operator, a single point mutation operator and a 2-optimization operator are used to simulate the flight strategy of bats to satisfy the search of feasible solutions. To speed up the convergence, we propose an elite strategy to maintain the non-dominate solutions in the external files. By decomposing products of different sizes and analyzing the experimental results, the proposed algorithm is evaluated with the existing multi-objective discrete gray wolf optimizer, artificial bee colony optimizer, non-dominated sorting genetic algorithm II, and multi-objective evolutionary algorithm based on decomposition. The effectiveness of the proposed algorithm in solving this problem is verified.

INDEX TERMS End-of-life products, remanufacturing, disassembly line balancing, AND/OR graph, precedence relation, multi-objective, discrete bat algorithm.

I. INTRODUCTION

The rapid development of economy and automatic manufacturing industry increases people's requirements for diversified and personalized products [1–6]. While enterprises produce large quantity of products every day, a great number of end-of life (EOL) products are weeded out as well, which leads to a huge waste of natural resource and even seriously threatens the healthy and ecological environment. To solve these problems, it is important to recycle and reuse EOL products, which can be done through disassembly processes. Proper and efficient disassembly also helps reduce carbon emissions and combat climate change by reducing the need for new manufacturing and reducing the amount of waste sent to landfills. Furthermore, recycled materials from EOL products can be used in the production of new products, reducing the need for virgin materials and ultimately conserving natural resources. Facing the waste of mass production, a disassembly line can realize standardized, automatic, and efficient disassembly of mechanical and electrical products, which has huge economic benefits.

The DLB problem refers to a set of disassembly operation, and there is precedence relation between each job operation. On the premise of not exceeding a given cycle time and meeting operation constraints. Disassembly operations are reasonably allocated to workstations, so that the number of workstations, idle time of each workstation, damage of parts, and demand index is as little as possible, and the disassembly efficiency is as high as possible [7]. Each workstation on the disassembly line has reserved a certain amount of idle time to ensure that the operation time does not exceed the cycle time, and the disassembly line balance plan needs to have a certain ability to cope with fluctuations in operation time.

Similar to assembly lines, the disassembly production lines also have the need of line balancing [8]. Gungor et al. formally propose the disassembly line balancing (DLB) problem for the first time to minimize the total idle time. A multi-objective model of complete disassembly is established with the optimization objectives for maximizing profit and minimizing disassembly hazardous, high demand parts, and direction changes. Subsequently, many scholars have conducted further studies on DLB problem [9, 10]. The main purpose is to get more valuable components with less consumption, which needs to assign tasks to disassembly workstations and uses fewer workstations while satisfying all constraints. Considering the interaction between disassembly tasks [11]. Yin et al. use the improved discrete hummingbird algorithm to solve the disassembly line balance problem and simultaneously optimize the number of workstations, total disassembly time, idle balance index and the number of disassembly tools [12]. Bentaha et al. [13] consider the randomness of disassembly operation time in the DLB problem, a stochastic programming model is established to maximize disassembly revenue, and an accurate solution method is proposed to solve it. Ren et al. [14] use a 2-optimal algorithm to solve the DLB problem based on multi-criteria decision making with interdependent weights.

In the actual disassembly process, every step of disassembly may occur unexpected situations, leading to the change of various factors in the disassembly link. In the disassembly process, it is difficult to predict the scrap of the disassembled parts, which is an important factor leading to the disassembly failure. It is also affected by the skill level of workers. If such a situation occurs in this process, a series of changes will occur in the disassembly process. DLB problem is proposed as a solution with the main idea that in the event of disassembly failure, the components or subsystems that need to be retained can be combined into other feasible tasks or products to reduce loss and waste. In terms of disassembly failure, Gungor and Gupta [15] first proposed DLBP in 2001, the paper considered a case of disassembly failure. The paper pointed out that the task failed due to the priority constraints of the task and the defects of the EOL product, which then led to a series of complex cases such as prohibiting disassembly of part or all remaining parts. This paper considers a series of changes in the disassembly process, gives a certain probability of failure for each disassembly process, and calculates the failure cost in the disassembly process, according to the failure cost judge the pros and cons of the solution.

McGovern Gupta [16] prove that the DLB problem is an NP-hard problem. Koc et al. [17] use precise formulas to solve it. However, most accurate methods are not suitable for dealing with large scale DLB problems. At present, the DLB problem solving methods mainly include the heuristic methods, mathematical programming methods, and intelligent optimization algorithms. In recent years, many studies focus on intelligent optimization algorithms, such as genetic algorithm (GA) [16] , AC algorithm [11], migratory bird optimization algorithm (MB) [18], grey wolf algorithm (GW) [19], and artificial bee colony algorithm (ABC) [20].

Bat algorithm (BA) is a new heuristic search algorithm proposed by Yang in 2010 [21], which also belongs to swarm intelligence algorithms. Its mechanism is to simulate the echolocation principle of bats. Compared with other algorithms. It is far superior to other algorithms in terms of accuracy and effectiveness. It has fewer parameters to be adjusted. Based on these advantages, many scholars have applied it to solve the optimization problems such as shop scheduling [22], wind power dispatch [23, 24], Wireless sensor network optimization, and image classification [25]. This work tries to use BA to solve a discrete and stochastic multi-objective disassembly-line-balancing problem (SMDP).

Considering the optimization objectives of high disassembly profit and low energy consumption, this paper constructs a multi-objective model for the DLB problem. A bat optimization algorithm with Pareto solution set is proposed. By establishing the correspondence between disassembly operation and iterative flying search, a crowd-distance mechanism is proposed that can ensure the diversity of solutions. The contributions of the paper are summarized as follows:

1) It formulates a SMDP model based on an AND/OR graph with disassembly failure risk under consideration. The objectives are to maximize disassembly profit and minimize energy consumption.
2) It designs a multi-objective discrete bat optimization (MDBO) algorithm combining with a stochastic simulation approach to handle the proposed problem. In this algorithm, a quadratic-vector list structure is designed to represent a solution. Based on the flight mechanism of the basic BA, a new individual generation operator and election operator are proposed to search for candidate solutions from the solution space.
3) The proposed algorithm is tested on several cases to verify its performance. Four popular multi-objective optimization algorithms, i.e., multi-objective discrete grey wolf optimizer (MDGWO), multi-objective discrete artificial bee colony algorithm (MDABC), non-dominated sorting genetic algorithm II (NSGA-II) [26] , and multi-objective evolutionary algorithm based on decomposition (MOEA/D) [27], are used for comparison.

The rest of this paper is organized as follows. The considered problem is formulated in Section II. Section III describes the MDBO algorithm. Section IV carries out simulation experiments and discusses the obtained results. Finally, Section V concludes this paper and discusses future work.

II. PROBLEM DESCRIPTION

A. PROBLEM STATEMENT

The disassembly of EOL products can be regarded as a recursive idea, that is, the disassembled components can be understood as a complete EOL product. The AND/OR

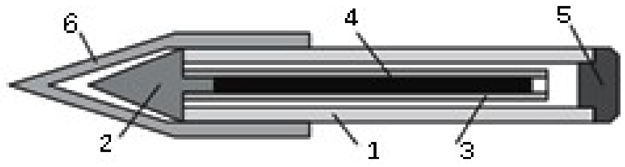


FIGURE 1. Ballpoint pen.

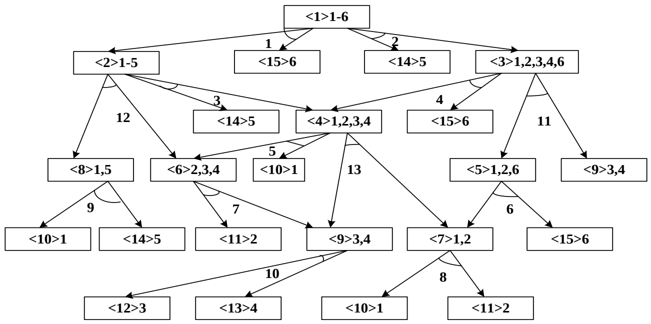


FIGURE 2. Ballpoint pen's AND/OR graph.

graph [17] adopts the idea of divide-and-conquer, so it is very suitable to use the AND/OR graph to represent the relationship between disassembly tasks and components [28]. Fig. 1 shows a schematic diagram of a ballpoint pen. Fig. 2 shows the AND/OR graph. The starting number in the rectangle represents the component number, and the inferior arc emitted by the rectangle represents the operation. The relationship between components and operations is represented by "AND". As can be seen from Fig. 2 , there are two operations in the figure, operation "1" and operation "2". The relationship between the two operations is "OR", because the two operations conflicts with each other and cannot be executed at the same time.

This paper combines with the demolition of waste ballpoint pen dismantling reality. While considering the above optimization objectives, to find an optimal disassembly operation assignment scheme. Under conditions of multiple constraints, realize the collaborative optimization of multiple goals.

Fig. 3 shows the disassembly line layout, which describes the process of disassembly of a ballpoint pen on the workstation. Rectangles represent subassembly and triangles represent disassembly operations. We can see that subassembly $\langle 1,3\rangle$ can get part $\langle 14\rangle$ after operation 2 and 11, subassembly $\langle 5\rangle$can get part $\langle 15\rangle$ after operation 6, subassembly $\langle 7\rangle$ can get part $\langle 10,11\rangle$ after operation 8. The DLB problem is very different from the assembly problem, and it involves a variety of complex factors. Therefore, when modeling the DLB problem in this work, the above complex factors are idealized from the perspective of applicability. In combination with the characteristics of the disassembly line. To facilitate the establishment of the model, this paper makes the following assumptions:

B. QUESTION ASSUMPTIONS

In view of the characteristics of the DLBP, the following assumptions are made.

1) The AND/OR graphs of EOL products to be disassembled are known.
2) The precedence matrix P and incidence matrix D among disassembly operations are known.
3) There are enough EOL products for disassembly.
4) Not all subassemblies can be completely disassembly.
5) The disassembly cost and energy consumption per unit of time of each disassembly operation, and the setup cost and setup energy consumption per unit of time among disassembly operations are known.
6) The recycling value of subassemblies is given.
7) The operating time of the workstation is not greater than its cycle time.

C. NOTION DEFINITION

- i Subassembly index, $i \in \{1, 2, ..., N\}$.,where N denotes the number of subassemblies in a product.
- j, k operation indices $j, k \in \{0, 1, 2, ..., J\}$., where J means the number of operations in a product, and 0 is a virtual operation.
- l, m workstation indices $j, k \in \{1, 2, ..., M\}$., where M is the number of workstations.
- t_j disassembly time of the disassembly operation j.
- t_{jk} setup time of operation k if immediately follows operation j.
- c_j cost per unit of time of performing the disassembly operation j.
- c_{jk} disassembly tool set-up cost per unit of time of operation k if it immediately follows operation j.
- e_j energy consumption per unit of time of operation j.
- e_{jk} disassembly tool set-up energy consumption per unit of time of operation k if it immediately follows operation j
- e_l energy consumption of the l-th workstation operation.
- c_l cost of the l-th workstation operation.
- v_i recycling/reuse value of subassembly i in a product.
- q_{jk} failure probability of operation k if it immediately follows operation j.
- T_l the fixed cycle time of the workstation.
- θ the probability that the failure cost of a disassembly process satisfies the required minimum probability that the failure cost of a disassembly process is less than or equal to its maximum value.
- $\widehat{F}$ maximum failure cost of a disassembly process.
- P disassembly-precedence matrix of a given AND/OR graph.
- D disassembly-incidence matrix of a given AND/OR graph.
- p_{jk} an element in the j-th row and k-th column of P.
- d_{ij} an element in the i-th row and j-th column of D.

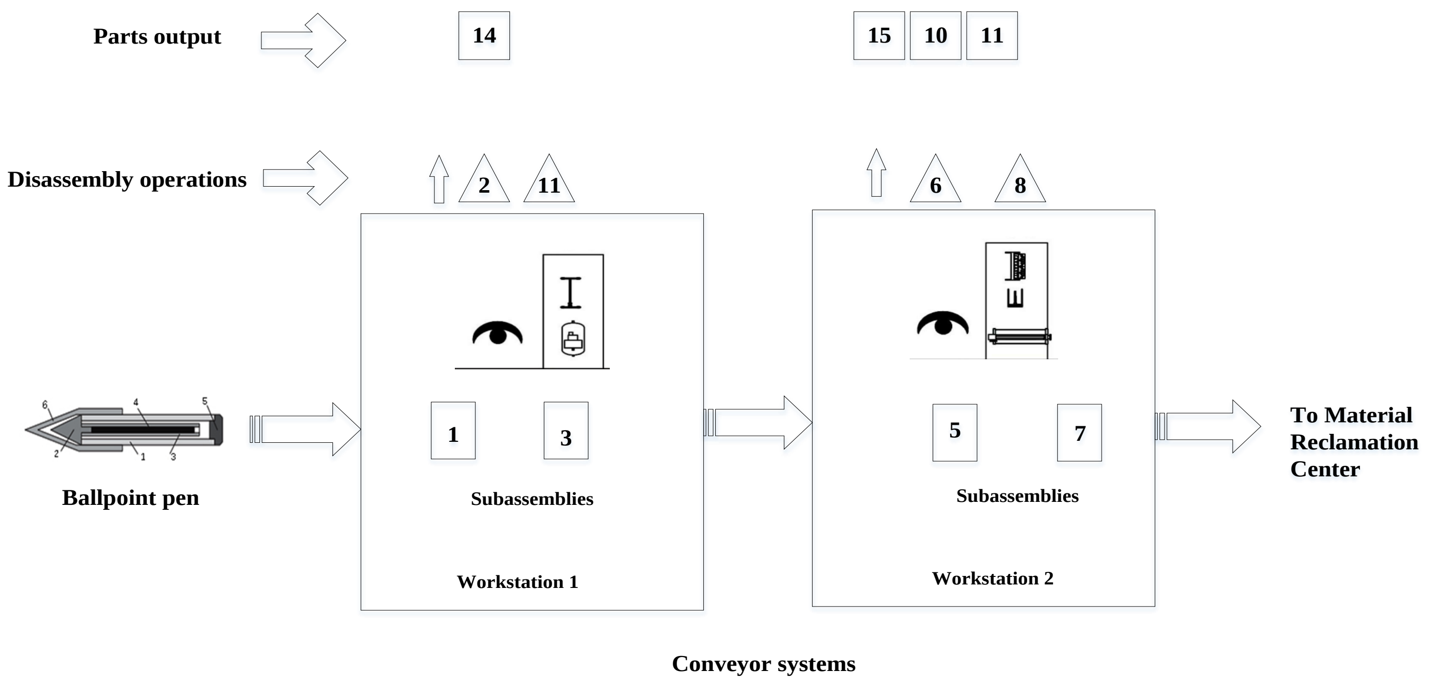


FIGURE 3. Disassembly line layout.

$$x_j = \begin{cases} 1, \text{ if disassembly operation } j \text{ is performed.} \\ 0, \text{otherwise.} \end{cases}$$

$$u_l = \begin{cases} 1, \text{ if workstation is used.} \\ 0, \text{ otherwise.} \end{cases}$$

$$y_{jk} = \begin{cases} 1, \text{ if disassembly operation } k \text{ is performed} \\ \text{after operation } j. \\ 0, \text{ otherwise.} \end{cases}$$

$$z_{jl} = \begin{cases} 1, \text{ if disassembly operation } j \text{ is assigned} \\ \text{to the } l-th \text{ workstation.} \\ 0, \text{ otherwise.} \end{cases}$$

D. MATHEMATICAL MODEL

$$max\ f_1 = \sum_{j=1}^{J}\sum_{i=1}^{N} d_{ij}v_i x_j - \sum_{j=1}^{J} c_j t_j x_j - \sum_{j=0}^{J}\sum_{k=1}^{J} c_{jk}t_{jk}y_{jk} - \sum_{l=1}^{M} c_l u_l \tag{1}$$

$$min\ f_2 = \sum_{j=1}^{J} e_j t_j x_j + \sum_{j=0}^{J}\sum_{k=1}^{J} e_{jk}t_{jk}y_{jk} - \sum_{l=1}^{M} e_l u_l \tag{2}$$

s.t.

$$\sum_{j=1}^{J} x_j \geq 1. \tag{3}$$

$$x_k = \sum_{j=1}^{J} y_{jk}, \forall k \in 1, 2, \cdots, J. \tag{4}$$

$$\sum_{l=1}^{M} z_{jl} \leq 1, \forall j \in 1, 2, \cdots, J. \tag{5}$$

$$\sum_{j=1}^{J} z_{jl} \geq 1, \forall l \in 1, 2, \cdots, M. \tag{6}$$

$$\sum_{l=1}^{M} l z_{jl} \leq \sum_{m=1}^{M} m z_{km}, \forall p_{jk} = 1, \forall j, k \in 1, 2, \cdots, J. \tag{7}$$

$$\sum_{l=1}^{M} l z_{jl} + \sum_{m=1}^{M} m z_{km} \leq 1, \forall p_{jk} = -1, \forall j, k \in 1, 2, \cdots, J. \tag{8}$$

$$\sum_{j=0}^{J}\sum_{k=1}^{J} (z_{kl}t_k x_k + z_{jl}t_{jk}y_{jk} \leq T_l, \forall l \in 1, 2, \cdots, M. \tag{9}$$

$$P_r\{\sum_{j=0}^{J}\sum_{k=1}^{J} (z_{kl}t_k x_k + z_{jl}t_{jk}y_{jk} \leq T_l\} \geq \theta, \forall l \in 1, 2, \cdots, M. \tag{10}$$

Objective function (1) is to maximize expected disassembly profit, which equals the total disassembly revenue minus total disassembly cost. The latter includes the total disassembly cost of each disassembly operation, the total disassembly setup cost of adjacent disassembly operations, the cost of switch-on workstation, and the failure cost. (2) is to minimize expected energy consumption, which includes the total energy consumption of disassembly operations, the total setup energy consumption of adjacent operations in the product, and the energy consumption of switched-on workstations. (3) indicates that at least one disassembly operation is being performed during the disassembly. (4) ensures that the same disassembly operation is not repeated (5) ensures that a disassembly operation can be assigned to one workstation at most once. (6) ensures that the workstation that is open is assigned at least one disassembly operation. (7) means that the feasible disassembly sequence meets the precedence relation constraints. (8) means that the feasible disassembly sequence meets conflicted relation constraints. (9) ensures that the workstation execution time is within the set execution cycle. (10) requires that the probability for the failure cost of a disassembly process being less than or equal to its maximum failure cost is greater than a preset value $\theta > 0$.

III. IMPROVED BAT ALGORITHM

A. BAT ALGORITHM

BA is a kind of swarm intelligence algorithm, which is an optimization technology based on iteration. It relies on the free flight of bats and the flight near the optimal solution to find the optimal solution. Compared with other algorithms, it has the advantages of fast convergence speed, high accuracy, and fewer algorithm parameters.

B. ENCODING

The DLB problem is a kind of discrete combinatorial optimization problem. Combining the characteristics of combinatorial optimization problem and model, a binary-vector list is designed to represent the solution structure. The specific encoding method is shown in Fig. 4. According to the DLB problem's characteristics, we encode the solution as $\pi = (\pi_1, \pi_2)$. $\pi_1 = (o_1, o2, \ldots, oj)$ is a decimal integer string whose length is the sum of all disassembly operations of a product. Each position of π_1 represents a disassembly operation, denoted by o_j . $\pi_2 = (w_1, w_2, \ldots, w_J)$ is a vector of binary elements indicating whether the task at the corresponding position has been disassembled. If $w_j = 1$, the operation in the $j-$th position in π_1 is performed, otherwise $w_j = 0$. Here, each disassembly sequence represents the solution of the problem, and the union of disassembly sequences constitutes a population. For example, Fig. 4 indicates that the disassembly sequence is 4-8-5-3-10.

C. DECODING

The process of decoding is the process of restoring the solution to a specific allocation scheme. Assign disassembly operations to workstations and determine which operations are assigned to which workstations. Each workstation must meet periodic constraints when assigning operations. The detailed steps of decoding are as follows:

1) Set $k = 1$.
2) Determine whether operation k is performed. If $x_k = 1$ and $k = k_f$, where, k_f represents the first operation in the sequence, then switch-on a new workstation l and let $l = 1$. Set the usage time $T_{ld} = 0$ of workstation l, set the S_l of operations assigned to the $l-$th workstation to be empty. If $x_k = 1$ and $k \neq kf$, operation k performs and continuue; otherwise, jump to Step 6
3) The operation transition time is added to T_{ld}. Based on the cycle time of the $l-$th workstation, if $T_{ld} > T_l$, continue to the next step; otherwise, go to Step 6.
4) Switch-on a new workstation and let $l = l+ 1$, $T_{ld} = 0$.
5) The operation transition time is added to T_{ld}.
6) The corresponding operation time is added to T_{ld}. If $T_{ld} > T_l$, continue to the next step; otherwise, jump to Step 9.
7) The operation transition time is added to T_{ld}.
8) Put operation k into the sequence of the $l-$th workstation S_l.
9) If $k \neq J$, set $k = k + 1$, repeat Steps 2 to 9; Otherwise, end and get a feasible solution.

D. MONTE CARLO SIMULATION

In this paper, we use Monte Carlo simulation to solve the objective function evaluation problem. It is a calculation method based on probability and statistical theory. It solves various mathematical problems by constructing random numbers that conform to certain rules. Random variables generated by various probability distributions are the basic means of Monte Carlo simulation. The approximate solution of the problem is obtained by means of statistical simulation or sampling, which is connected with a probability model. In this work, we simulate it randomly for 10 times to get the expected value of the objective function, so that it is closer to the real solution of the problem. It is generally divided into three steps as shown in the pseudocode of Algorithm 1.

E. CALCULATION OF FITNESS

In the multi-objective optimization problem, the mutual restriction between each target may affect the other targets while improving one target. Thus, for the multi-objective optimization problem, its solution is usually a Pareto solution set. The fitness here is expressed as a dominant relationship. For a multi-objective minimization problem whose number of optimized objectives is U, the solution x_1 is denoted as dominated by the solution x_0 if the following relationship is satisfied:

$$\begin{cases} f_i(x_0) \leq f_i(x_1), & \forall i \in \{1, 2, ..., U\} \\ f_j(x_0) < f_j(x_1), & \exists j \in \{1, 2, ..., U\} \end{cases} \tag{11}$$

F. INITIALIZATION OF POPULATION

Pareto MDBO adopts N central positions in the iteration process, each central position is regarded as an individual bat. To solve this problem, multiple bat individuals constitute

---

Algorithm 1 Monte Carlo simulation

Input: Disassembly time given by the individual and target value of the model
Output: New individual target value
  for $i = 1; i < M; i++$ **do**
    According to their stochastic distribution, generate a sample, i.e., disassembly time t_j of disassembly operations. Combine the mathematical model with the current sample to calculate the solution's target value and failure cost
  end for
  $M' \leftarrow \lfloor \theta * M \rfloor$
  find the M'-th largest sample
  if (failure cost $> \hat{F}$) **then**
    return false
  else
    return $f1, f2, fm$ samples of new individual t
  end if

Algorithm 2 Pareto rule

Input: Individuals A, B, number of objectives n
Output: The dominant relationship between A and B
  equal count, less count, greater count $\leftarrow 0$
  for $i = 1; i < n; i++$ **do**
    if A. value i equals B. value i **then**
      equal count $\leftarrow$ equal count + 1
    else if A. value i less B. value i **then**
      less count $\leftarrow$ less count + 1
    else
      great count $\leftarrow$ great count + 1
    end if
  end for
  if equal count $==$ n **then**
    A, B are the same.
  else if equal count + less count $==$ n and less count > 0 **then**
    A $\prec$ B
  else if equal count + less count $==$ n and great count > 0 **then**
    B $\prec$ A
  else
    There is no dominant relationship between A and B
  end if

---

a population. Because of the complexity of the SMDP, the sequence must satisfy the precedence constraint. The essence of the initialization operation is the distribution operation of the problem-solution space. The initialization of the population is the initial solution of the population according to the coding rules. In this work, the initial population is randomly generated and the detailed steps are as follows.

1) Randomly generate a solution , i.e., π_1, π_2.
2) Adjust the disassembly sequence obtained in Step 1 according to P to make the sequence meet the precedence constraint. If the disassembly operation o_j is after ok and p_{jk} =1 in matrix P, then o_j and ok are swapped, continue to traverse the next operation.
3) Adjust the binary values in π_2 based on matrix P to meet the precedence relation between disassembly operations of the product.
4) Adjust the binary values in π_2 based on matrix P to eliminate the conflict relation between disassembly operations of the product.

π_1	4	7	6	8	5	2	...	*J*
π_2	0	1	1	0	1	0	...	0

FIGURE 4. Encoding scheme.

G. INDIVIDUAL EVOLUTION

Bat optimizer is a kind of heuristic swarm intelligence algorithm. It has been proven that BA has more obvious advantages than other algorithms such as particle swarm algorithm [29], simulated annealing algorithm [30], and firefly algorithm [31], and there are not many parameters to be adjusted. However, BA is mainly used to solve the functional optimization problem in the continuous domain. In order to solve the optimization problem in the discrete domain, the BA must be redefined. In combination with the characteristics of the BA, we use the Precedence Preserving Crossover operator (PPX), Position-Based Mutation (PBM) operator, and 2-optimization operator (2-OPT) to solve the disassembly problem. In solving the DLB problem, population and individual updates are as follows:

1) An individual is randomly selected in the current Pareto solution set, and the individual and each individual of the population use PPX operator and PBM operator to generate child individuals, then, go to Step 3;
2) If the random number is greater than the pulse emissivity R_i of bats, then the 2-OPT operator or PBM operator is used to update random individuals for the currently obtained optimal solution. then, go to Step 3;
3) Judge whether the newly generated child new dominates the parent, and judge whether the loudness A_i is less than the set parameter. If true, add the child new to the population. Then, go to Step 4.
4) Perform a non-dominant order on the population.
5) The population is selected by using the ranking and crowding distance. Then go to Step 6.
6) The solution is saved using the dominant relationship and the external archive set V.

IV. EXPERIMENTS

TABLE 1. THE RESULT OF INSTANCE ONE DISASSEMBLY SEQUENCE IS OBTAINED BY MDBO ALGORITHM PROGRAM.

Instance	Disassembly sequence	f1	f2	fc	fm
	1,2,4,8,14,15	275.1	861.0	35.2	2
1	1,2,3,6,10,17,11,20,18	427.8	1374.6	59.3	2
	1,2,4,8,15,25,14,24,33,34,39,11,18,20,28	684.7	1965.2	57.7	3
	2,10,4,15,23,7	572.0	781.9	22.8	1
2	2,10,4,15,22	496.1	612.4	16.8	1
	2,10	231.1	274.3	9.2	1
	2,4,13,8,10	186.8	555.4	26.0	1
3	2,4,5,7,10	277.2	700.2	31.9	1
	2,4,5,7	216.9	637.5	21.9	1

TABLE 2. COMPARISON OF EXPERIMENTAL RESULTS OF INSTANCE ON FIVE ALGORITHMS IN C-METRIC.

Instance	pop	C1	t-test	C2	t-test	C3	t-test	C4	t-test
1	100	0.7317 0.1236	+	0.5792 0.2544	+	0.8006 0.0629	+	0.7259 0.1090	+
	120	0.7348 0.0926	+	0.6393 0.1866	+	0.8573	+	0.7357 0.1200	+
	150	0.7655 0.1137	+	0.5746 0.2166	+	0.7801 0.0907	+	0.7677 0.1039	+
2	100	0.6333 0.0666	+	0.4283 0.2249	+	0.7650 0.0499	+	0.5511 0.1666	+
	120	0.5891 0.1083	+	0.4333 0.1833	+	0.7866 0.0333	+	0.4367 0.2083	+
	150	0.6583 0.1166	+	0.4700 0.2416	+	0.7300 0.0583	+	0.5211 0.2249	+
3	100	0.4500 0.2575	+	0.4000 0.3441	∼	0.5708 0.1941	+	0.4938 0.3266	+
	120	0.5716 0.2416	+	0.4158 0.2933	+	0.5233 0.2358	+	0.5200 0.2500	+
	150	0.4808 0.2733	+	0.4366 0.2783	+	0.5750 0.1400	+	0.4136 0.3066	∼

A. ALGORITHM AND INDEX

We compared the performance of the proposed algorithm that of NSGA-II, MDGWO, MDABC, and MOEA/D. The NSGA-II reduces the complexity of the algorithm by using the non-dominated sorting method, a fast, the convergence of solution set good advantage, become other performance benchmark multi-objective optimization algorithm. MOEA/D decomposes a multi-objective optimization problem into a single-objective optimization problem, which has the advantages of low computational complexity and high solving efficiency. M DGWO a nd M DABC a lgorithms, as meta-heuristic algorithms, have performed well in solving discrete optimization problems in recent years, so the above algorithms are selected to compare with MDBO.

Three products of different level of complexity are used for experiments: hammer drill (HD), radio set (RS) [32] and ballpoint pen (BP) [32]. An HD contains 63 subassemblies and 46 disassembly operations. An RS includes 29 subassemblies and 30 disassembly operations. There are 15 subassemblies in a BP and 13 disassembly operations. Multi-objective algorithm has a variety of indicators to choose from in evaluating the quality of solution set. General evaluation indicators mainly combine the following three dimensions in evaluating the quality of solution set.

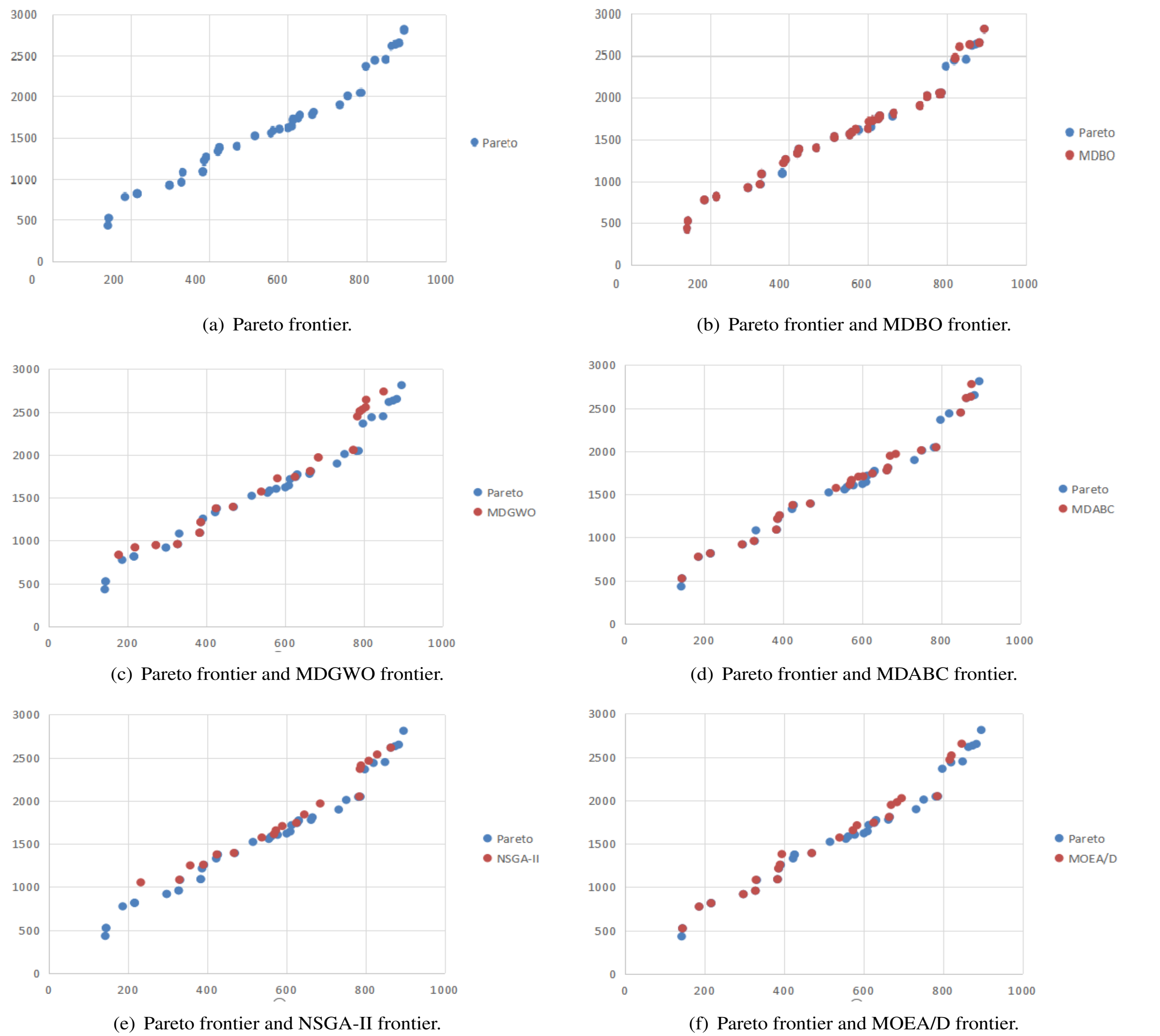
1) Convergence evaluation:
   When the solution set is closer to the real Pareto frontier, the convergence of the solution set is better.
2) Uniformity evaluation:
   The more uniform the distribution of individuals in the solution set is, the better the uniformity of the solution set is.
3) Extensive evaluation:

TABLE 3. COMPARISON OF EXPERIMENTAL RESULTS OF INSTANCE ON FIVE ALGORITHMS IN IGD-METRIC.

Instance	pop	MDBO		MDGWO		MDABC		NSGA-II		MOEA/D	
		mean&var	t-test	mean&var	t-test	mean&var	t-test	mean&var	t-test	mean&var	t-test
1	100	0.0246 (0.0000)	null	0.0667 (0.0002)	+	0.0299 (0.0001)	+	0.0911 (0.0005)	+	0.0736 (0.0012)	+
	120	0.0280 (0.0001)	null	0.0693 (0.0001)	+	0.0441 (0.0003)	+	0.0795 (0.0002)	+	0.0747 (0.0006)	+
	150	0.0298 (0.0000)	null	0.0807 (0.0003)	+	0.0467 (0.0002)	+	0.0971 (0.0002)	+	0.0819 (0.0010)	+
2	100	0.0022 (0.0000)	null	0.0284 (0.0000)	+	0.0237 (0.0001)	+	0.0205 (0.0001)	+	0.0283 (0.0014)	+
	120	0.0012 (0.0000)	null	0.0549 (0.0020)	+	0.0429 (0.0003)	+	0.0367 0.0004)	+	0.0323 (0.0007)	+
	150	0.0012 (0.0000)	null	0.0461 (0.0007)	+	0.0377 (0.0003)	+	0.0370 (0.0003)	+	0.0438 (0.0012)	+
3	100	0.0156 (0.0001)	null	0.0267 (0.0002)	+	0.0175 (0.0001)	∼	0.0163 (0.0001)	∼	0.0416 (0.0004)	+
	120	0.0046 (0.0000)	null	0.0167 (0.0004)	+	0.0142 (0.0000)	+	0.0057 (0.0000)	∼	0.0373 (0.0015)	+
	150	0.0028 (0.0000)	null	0.0222 (0.0003)	+	0.0072 (0.0001)	+	0.0056 (0.0000)	∼	0.0357 (0.0006)	+

TABLE 4. COMPARISON OF EXPERIMENTAL RESULTS OF INSTANCE ON FIVE ALGORITHMS IN HV-METRIC.

Instance	pop	MDBO		MDGWO		MDABC		NSGA-II		MOEA/D	
		mean&var	t-test	mean&var	t-test	mean&var	t-test	mean&var	t-test	mean&var	t-test
1	100	0.5125 (0.0001)	0.0015	0.4562 (0.0004)	+	0.5040 (0.0003)	+	0.4289 (0.0004)	+	0.4468 (0.0015)	+
	120	0.5534 (0.0001)	0.0017	0.4847 (0.0004)	+	0.5255 (0.0011)	+	0.4687 (0.0006)	+	0.4904 (0.0017)	+
	150	0.5553 (0.0001)	0.0020	0.4805 (0.0006)	+	0.5278 (0.0005)	+	0.4549 (0.0005)	+	0.4905 (0.0020)	+
2	100	0.6664 (0.0000)	0.0024	0.6481 (0.0001)	+	0.6533 (0.0000)	+	0.6499 (0.0003)	+	0.6417 (0.0024)	+
	120	0.6350 (0.0000)	0.0001	0.6124 (0.0005)	+	0.6198 (0.0000)	+	0.6148 (0.0004)	+	0.6251 (0.0001)	+
	150	0.6346 (0.0000)	0.0006	0.6115 (0.0011)	+	0.6201 (0.0001)	+	0.6170 (0.0001)	+	0.6168 (0.0006)	+
3	100	0.4697 (0.0000)	0.0001	0.4645 (0.0001)	+	0.4680 (0.0000)	∼	0.4681 (0.0000)	∼	0.4628 (0.0001)	+
	120	0.4697 (0.0000)	0.0001	0.4649 (0.0001)	+	0.4662 (0.0000)	+	0.4687 (0.0000)	∼	0.4637 (0.0001)	+
	150	0.4711 (0.0000)	0.0001	0.4627 (0.0001)	+	0.4690 (0.0000)	+	0.4694 (0.0000)	+	0.4610 (0.0001)	+



(a) Pareto frontier.

(b) Pareto frontier and MDBO frontier.

(c) Pareto frontier and MDGWO frontier.

(d) Pareto frontier and MDABC frontier.

(e) Pareto frontier and NSGA-II frontier.

(f) Pareto frontier and MOEA/D frontier.

FIGURE 5. The Pareto solution set obtained by each algorithm is compared with the Pareto frontier

The more widely the whole solution set is distributed in the target space, the more extensive the solution set is.

In the optimization of multi-objective problems, evaluation indexes such as C-metric, IGD-metric, and Hypervolume-metric are often used. The evaluation indexes are as follows:

1) C-metric: C-metric [33] is called the solution set coverage; the formula is as follows:

$$C(A,B)=\frac{|\{u\in B|\exists v\in A:v\succ u\}|}{|B|} \tag{12}$$

The numerator is the number of solutions in B that are dominated by at least one solution in A. The denominator is the total number of solutions in B.

2) IGD-metric: IGD-metric [34] is called Inverted Generational Distance. This index can be understood as the approximation degree of Pareto. The advantage of this index is that it can evaluate convergence and diversity at the same time, and the calculation cost is small, while the disadvantage is that reference sets are needed formula is as follows:

$$IGD(\rho,\rho^{*})=\frac{\sum_{\pi\in\rho^{*}}d(\pi,\rho)}{|\rho^{*}|} \tag{13}$$

3) Hypervolume-metric[35]: The volume of the region in the target space bounded by the nondominant solution set and the reference point obtained by the algorithm. The higher the HV value is, the better the comprehensive performance of the algorithm is. The formula is as follows:

$$Hypervolume=\bigcup_{i=1}^{|P|}v_i \tag{14}$$

$|\overline{P}|$ is the number of nondominant solution sets. v_i represents the Hypervolume formed by the reference point and the i-th solution in the solution set. The advantage of this index is that it can evaluate convergence and diversity at the same time. The disadvantage is that it has high computational complexity, especially for high-dimensional multi-objective optimization. The selection of reference points determines the accuracy of the Hypervolume metric to a certain extent.

In the index evaluation stage, in order to ensure the accuracy of the data, we obtain a large amount of data through experiments. We repeat 20 tests on the experimental cases to obtain 20 experimental results, and use three evaluation indexes C-metric, IGD-metric, and Hypervolume-metric to evaluate the advantages and disadvantages of the algorithm. Besides, we use the t-test with a degree of freedom of 38 and a significance level of 0.05 to analyze the experimental results. The experimental results of the t-test can be divided into three types, which are significantly better than, significantly worse than, and statistical equivalence than, they can be represented by the symbols “+”, “-”, and “$\sim$”, respectively.

B. PEER ALGORITHMS AND PARAMETER SETTING

In order to test the stability of the algorithm, the population size $|P|$ is adjusted, and the population size is 100, 120, and 150, respectively. The experiment is carried out with fixed parameters without updating the pulse emissivity R_i and pulse loudness A_i.The pulse emissivity R_i of the MDBO proposed in this paper is 0.5, and the corresponding pulse loudness A_i is also 0.5. Parameters of all algorithms are set as follows: mutation probability is 0.3 and the total fitness value f_{tv} is 3*number of operations*number of subassemblies. The parameters involved in the objective functions are set as: T_l is 50, = 0.95, $\widehat{F}$ is 200. In this work, all algorithms run 20 times independently.

All the algorithms are implemented in IntelliJ IDEA2020.1×64, running on the AMD Ryzen 7 4700U CPU (2.00GHz/16.00GB RAM) PC with windows 10 operating system.

C. ANALYSIS OF EXPERIMENTAL RESULTS

In this section, we use MDGWO, MDABC, NSGA-II, MOEA/D, and MDBO to process the three test cases, and C-metric, IGD-metric, and Hypervolume-metric are employed to analyze their experimental results.

Instance 1 represents partial HD disassembly sequences and their target values, Instance 2 represents partial RS disassembly sequences and their target values, Instance 3 represents partial BP disassembly sequences and their target values, where $f1$, $f2$, fc, and fm denote the total profit, energy consumption, failure cost, the number of workstations, respectively in Table 1.

From Table 1, the first four are the data of HD disassembly, In the first solution 1,2,4,8,14,24 represents a complete disassembly sequence, the first four operations 1,2,4,8, are assigned to the first workstation, and the next two operations 14,24 are assigned to the second workstation. Two workstations are used for the entire disassembly process. It can be seen that the profit generated during this operation is 216.5, the energy consumption value is 815.2, and the failure cost is 35.2. By analyzing the disassembly sequence of each group, it can be seen that with the increase of the disassembly sequence, all the target values are increasing.

Table 2 gives the experimental results in terms of C-metric of three cases. C1, C2, C3, C4 respectively represent the comparison between MDBO and MDGWO, MDABC, NSGA-II, and MOEA/D. The top data in each row represent the dominance degree of MDBO over each comparison algorithm, while the bottom data are the dominance degree of each algorithm over MDBO. By comparing the values, it can be seen that MDBO is far better than NSGA-II, MDGWO and MOEA/D have little difference in solving this problem, and MDBO and MDABC have the least difference. Since the disassembly scale of Instance 3 is relatively small, it can be seen that “ ” appears in Instance 3, so it can be concluded that MDBO has more advantages in solving large-scale problems. The data obtained according to different population sizes have no obvious fluctuation, which indicates that the performance of each algorithm is relatively stable.

Table 3 reveals the experimental results of five algorithms in three cases via IGD-metric. Pop stands for population number, and there are three types of population numbers: 100,120,150. From these results, we can conclude that MDBO performs better than MDGWO, MDABC, NSGAII, and MOEA/D for solving the concerned problem because the IGD values of MDBO are smaller than those of MDGWO, MDABC, NSGAII, and MOEA/D. From the observation of Instance 1, it can be seen that the MDBO is slightly better than the MDABC, and the MDBO has achieved a very good effect in Instance 2.

To further reveal the performance of MDBO, the Hypervolume-metric is adopted to analyze the experimental results. Table 4 shows their Hypervolume values. It can be found that the Hypervolume value obtained by MDBO is larger than that of all algorithms, which indicates that the optimization effect of MDBO is better than that of other algorithms. By observing the data, it can also be seen that in Instance 2 and Instance 3, the floating difference of data obtained by the three parameters of MDBO is very small, which is due to the small number of Pareto solutions in the case itself.

In order to show the advantages and disadvantages of each algorithm more intuitively, we draw the Pareto frontier of each algorithm in Fig. 5, where the horizontal axis represents the $f1$ and the vertical axis represents the $f2$. As can be seen from Fig. 5, MDBO is the closest to the real Pareto frontier, followed by MDABC, and the other three algorithms are not much different. They achieve the same effect with the above C-metric, IGD-metric, and Hypervolume-metric, indicating that MDBO has achieved a good effect in solving this problem.

V. CONCLUSION AND FUTURE WORK

This addresses the discrete and stochastic multi-objective DLB problem. A mathematical programming model is established to describe the problem. A multi-objective discrete bat optimization (MDBO) method combined with stochastic simulation is designed to find the best solution. The experimental results show that MDBO is better than MDGWO, MDABC, NSGA-II, and MOEA/D, the four most outstanding algorithms in the domain.

Future research can focus on the following aspects: Considering more restrictive conditions. The current DLB model mainly focuses on reducing idle time and dismantling costs, but in practical applications, other limitations need to be considered, such as safety, environmental protection, quality requirements, etc. Therefore, future research can consider adding more constraints to the model to improve the practicality and reliability of DLB. Optimize DLB using intelligent technology. Currently, intelligent technologies such as artificial intelligence, machine learning [36–38], and the Internet of Things are rapidly developing, which can provide more optimization methods for DLB. For example, machine learning algorithms can be used to optimize task allocation and workstation scheduling, improving the system's adaptability and flexibility. Investigating the modeling and optimization of disassembly lines of multiple products U-shaped layout[39, 40] and parallel layout [41].

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, pp. 1–13, 2021.

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

[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, 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.

[7] E. B. Edis, R. S. Edis, and M. A. Ilgin, "Mixed integer programming approaches to partial disassembly line balancing and sequencing problem," Computers & Operations Research, vol. 138, p. 105559, 2022.

[8] A. Güngör and S. M. Gupta, "Disassembly line in product recovery," International Journal of Production Research, vol. 40, no. 11, pp. 2569–2589, 2002.

[9] X. Guo, M. Zhou, S. Liu, and L. Qi, "Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints," IEEE Transactions on Cybernetics, vol. 50, no. 7, pp. 3307–3317, 2020.

[10] X. Song, W. Zhou, X. Pan, and K. Feng, "Disassembly sequence planning for electro-mechanical products under a partial destructive mode," Assembly Automation, vol. 34, 01 2014.

[11] C. B. Kalayci and S. M. Gupta, "Ant colony optimization for sequence-dependent disassembly line balancing problem," Journal of Manufacturing Technology Management, 2013.

[12] T. Yin, Z. Zhang, and J. Jiang, "A pareto-discrete hummingbird algorithm for partial sequence-dependent disassembly line balancing problem considering tool requirements," Journal of Manufacturing Systems, vol. 60, pp. 406–428, 07 2021.

[13] M. L. Bentaha, O. Battaïa, and A. Dolgui, "An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times," Int. J. Prod. Res., vol. 53, no. 6, pp. 1807–1818, 2001.

[14] Y. Ren, C. Zhang, F. Zhao, G. Tian, W. Lin, L. Meng, and H. Li, "Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-optimal algorithm," Journal of Cleaner Production, vol. 174, pp. 1475–1486, 2018.

[15] A. Gungor and S. M. Gupta, "A solution approach to the disassembly line balancing problem in the presence of task failures," International Journal of Production Research, vol. 39, pp. 1427 – 1467, 2001.

[16] S. M. McGovern and S. M. Gupta, "A balancing method and genetic algorithm for disassembly line balancing," European journal of operational research, vol. 179, no. 3, pp. 692–708, 2007.

[17] 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 Trans, vol. 41, pp. 866–881, 2009.

[18] G. Qin, X. Guo, S. Liu, L. Qi, J. Zhao, Z. Zhao, and Y. Tang, "Multi-objective discrete migrating birds optimizer solving multiple-product partial u-shaped disassembly line balancing problem," in 2021 29th Mediterranean Conference on Control and Automation (MED), 2021, Conference Proceedings, pp. 59–64.

[19] C. Lu, S. Xiao, X. Li, and L. Gao, "An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production," Advances in Engineering Software, vol. 99, pp. 161–176, 09 2016.

[20] C. B. Kalayci and S. M. Gupta, "Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem," Expert Systems with Applications, vol. 40, no. 18, pp. 7231–7241, 2013.

[21] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, 2010, pp. 65–74.

[22] Y. Lu and T. Jiang, "Bi-population based discrete bat algorithm for the low-carbon job shop scheduling problem," IEEE Access, vol. 7, pp. 14 513–14 522, 2019.

[23] H. Liang, Y. Liu, Y. Shen, F. Li, and Y. Man, "A hybrid bat algorithm for economic dispatch with random wind power," IEEE Transactions on Power Systems, vol. 33, no. 5, pp. 5052–5061, 2018.

[24] Y. T. K. Priyanto, M. F. Maulana, and A. Giyantara, "Dynamic economic dispatch using chaotic bat algorithm on 150kv mahakam power system," in 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA). IEEE, 2017, pp. 116–121.

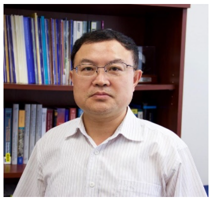
[25] J. Senthilnath, S. Kulkarni, J. A. Benediktsson, and X. S. Yang, "A novel approach for multispectral satellite image

classification based on the bat algorithm," IEEE Geoscience and Remote Sensing Letters, vol. 13, no. 4, pp. 559–603, 2016.
[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.
[27] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.
[28] X. Guo, M. Zhou, S. Liu, and L. Qi, "Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption," IEEE Transactions on Automation Science and Engineering, vol. 18, no. 2, pp. 804–816, 2020.
[29] D. Tian and Z. Shi, "Mpso: Modified particle swarm optimization and its applications," Swarm and Evolutionary Computation, vol. 41, pp. 49–68, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650217307137
[30] J. Liang, S. Guo, B. Du, Y. Li, J. Guo, Z. Yang, and S. Pang, "Minimizing energy consumption in multi-objective two-sided disassembly line balancing problem with complex execution constraints using dual-individual simulated annealing algorithm," Journal of Cleaner Production, vol. 284, p. 125418, 2021.
[31] L. Zhu, Z. Zhang, and Y. Wang, "A pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation," International Journal of Production Research, vol. 56, no. 24, pp. 7354–7374, 2018.
[32] Q. Lu, Y. Ren, H. Jin, L. Meng, L. Li, C. Zhang, 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, 2020.
[33] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: methods and applications," 1999.
[34] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," Trans. Evol. Comp, vol. 7, no. 2, p. 174–188, apr 2003. [Online]. Available: https://doi.org/10.1109/TEVC.2003.810761
[35] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," Evolutionary Computation, IEEE Transactions on, vol. 7, pp. 117 – 132, 05 2003.
[36] E. Tuncel, A. Zeid, and S. Kamarthi, "Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning," J. Intell. Manuf., vol. 25, no. 4, p. 647–659, aug 2014. [Online]. Available: https://doi.org/10.1007/s10845-012-0711-0
[37] J. Gu, J. Wang, X. Guo, G. Liu, S. Qin, and Z. Bi, "A metaverse-based teaching building evacuation training system with deep reinforcement learning," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 4, pp. 2209 – 2219, 2023.
[38] H. Shi, G. Liu, K. Zhang, Z. Zhou, and J. Wang, "Marl sim2real transfer: Merging physical reality with digital virtuality in metaverse," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 4, pp. 2107 – 2117.
[39] Z. Li, I. Kucukkoc, and Z. Zhang, "Iterated local search method and mathematical model for sequence-dependent u-shaped disassembly line balancing problem," Computers & Industrial Engineering, vol. 137, p. 106056, 2019.
[40] Z. Zhang, K. Wang, L. Li, and L. MAO, "Multi-objective optimization for u-shaped disassembly line balancing problem with stochastic operation times," Computer Integrated Manufacturing Systems, vol. 24, no. 1, pp. 89–100, 2018.
[41] S. Smith and P.-Y. Hung, "A novel selective parallel disassembly planning method for green design," Journal of engineering design, vol. 26, no. 10-12, pp. 283–301, 2015.

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.

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 Petrochemical 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.



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.

LIANG QI 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.

FUGUANG HUANG received his B.S. degree in Network engineering from Zhoukou Normal University, Henan, China in 2020. He is currently a Graduate student of the College of Computer and Communication Engineering at Liaoning Petrochemical University, Fushun, China. His research interests are in intelligent optimization algorithm.