An enhanced group teaching optimization algorithm for multi-product disassembly line balancing problems

Pei Liang1 · Yaping Fu1 · Kaizhou Gao2 · Hao Sun1

Received: 26 March 2021 / Accepted: 19 July 2021 / Published online: 10 August 2021
© The Author(s) 2021

Abstract
Big data have been widely studied by numerous scholars and enterprises due to its great power in making highly reliable decisions for various complex systems. Remanufacturing systems have recently received much attention, because they play significant roles in end-of-life product recovery, environment protection and resource conservation. Disassembly is treated as a critical step in remanufacturing systems. In practice, it is difficult to know the accurate data of end-of-life products such as disassembly time because of their various usage processes, leading to the great difficulty of making effective and reliable decisions. Thus, it is necessary to model the disassembly process with stochastic programming method where the past collected data are fitted into stochastic distributions of parameters by applying big data technology. Additionally, designing and applying highly efficient intelligent optimization algorithms to handle a variety of complex problems in the disassembly process are urgently needed. To achieve the global optimization of disassembling multiple products simultaneously, this work studies a stochastic multi-product disassembly line balancing problem with maximal disassembly profit while meeting disassembly time requirements. Moreover, a chance-constrained programming model is correspondingly formulated, and then, an enhanced group teaching optimization algorithm incorporating a stochastic simulation method is developed by considering this model’s features. Via performing simulation experiments on real-life cases and comparing it with five popularly known approaches, we verify the excellent performance of the designed method in solving the studied problem.

Keywords Multi-product disassembly · Scheduling · Disassembly line balancing · Enhanced group teaching optimization algorithm · Stochastic simulation

Introduction

Nowadays, big data attain much attention from many scholars and enterprises, because it has huge potential in making high-quality decisions for various complex systems [1]. To be specific, it is intractable to collect and acquire large amounts of data from systems in a short time owing to diverse factors such as limited human and financial resources [2]. To predict the true data as accurately as possible, a large amount of obtained data is usually fitted into approximate distributions of parameters with big data technology [3–5]. As a result, a great deal of data collected from practical systems can be formulated as approximate expressions such as stochastic and fuzzy distributions, which makes modeling and optimization of real-world systems more challenging [6]. Intelligent optimization algorithms have been widely and resoundingly employed to handle diverse complicated optimization problems because of their powerful performance, e.g., strong search ability and high computation efficiency [7–12]. Recently, remanufacturing systems have received great concern from governments, because they play positive roles in environment protection and resource conservation [13]. Remanufacturing end-of-life (EOL) products usually faces much uncertainty because of their diverse usage processes in various kinds of customers [14]. In practice, managers and engineers of remanufacturing enterprises often collect a great deal of data which can be formulated as the approximate distributions as the above-mentioned approach.

* Yaping Fu
  fuyaping0432@163.com

* Hao Sun
  rivaldoking@126.com

1 School of Business, Qingdao University, Qingdao 266071, China
2 Macau Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China
The recent years have witnessed that much attention is given to efficiently promoting operation efficiency of manufacturing and remanufacturing systems via various approaches, e.g., planning and scheduling [15–19]. Disassembly is a very critical step in remanufacturing systems [20]. In a disassembly process, the disassembly line is usually employed to perform disassembly tasks [21, 22]. Currently, disassembly line balancing problems (DLBPs) are regarded as an essential issue in improving disassembly efficiency [23]. They aim at assigning a group of disassembly tasks to workstations while meeting precedence relationships of disassembly tasks to optimize some specified criteria [24]. Recently, DLBPs have attained wide concern from academic and industrial areas [25, 26]. To solve DLBPs, we have to determine a set of performed tasks and their assignment among workstations while obeying complex precedence constraints of disassembly tasks. Thus, it is difficult to effectively solve them, and they have proven to be NP-complete [27]. As a consequence, many meta-heuristic approaches have been used to handle DLBPs owing to their powerful search ability [28]. Zhang et al. [29] present an improved artificial fish swarm algorithm to minimize the number of workstations, total idle time, the sum of the maximum disassembly costs and disassembly direction change frequencies in a DLBP. Agrawal and Tiwari [30] adopt a collaborative ant colony optimization algorithm to deal with a stochastic disassembly line balancing and sequencing problem with minimizing the number of workstations, idle time and the probability of line stoppage. Wang et al. [31] aim at maximizing disassembly profit and minimizing the number of workstations, workload smoothness and energy consumption in a DLBP with uncertainty. They present a multi-objective discrete flower pollination algorithm to solve it. Zhu et al. [32] propose a parallel DLBP with minimizing the cycle time, number of workstations, idle index and quantity of disassembly resource. They use a multi-objective hybrid group neighborhood search algorithm to cope with it. Kalayci and Gupta [33] study a sequence-dependent DLBP with consideration of efficiency, safety and revenue. They design a hybrid approach combining a particle swarm optimization (PSO) and a neighborhood-based mutation operator to deal with it. Wang et al. [34] provide a genetic simulated annealing (SA) algorithm to achieve disassembly profit maximization as well as reach number of workstations and smoothness index minimization in a parallel DLBP. Cīl et al. [35] design an ant colony optimization method to minimize the cycle time in a robotic DLBP. Xiao et al. [36] consider maximizing the total profit in a DLBP under uncertainty. An improved migrating birds optimization algorithm is developed to address it. Wang et al. [37] aim at maximizing the profit and minimizing the number of stations, the difference in workstation load and the energy consumption in a DLBP. They design an improved genetic algorithm to tackle it.

Through analyzing the existing studies, it can be found that most of them concentrate on disassembling only one EOL product on a disassembly line. Nevertheless, in practice, multiple products usually need to be disassembled together on a disassembly line in a planning horizon. Consequently, it is of great significance to deal with a multi-product disassembly line balancing problem to realize a global optimization of disassembling them together. We give an example with disassembling two products, i.e., $A_1$ and $A_2$, to describe the multi-product disassembly. Let $x_i$ and $x_j$ denote their disassembly plans that are made independently, while $x_i$ and $x_j$ represent their disassembly plans that are made together. Then, this work formulates their concise disassembly line balancing (DLB) models in Table 1, where $f(x_i)$ indicates the disassembly profit of $x_i$, $t_f(x_i)$ denotes the disassembly time of $x_i$, $i \in \{1, 2, 3, 4\}$, $j \in \{1, 2\}$, $T_1$ and $T_2$ are two given disassembly time limitations, and $\Omega_1$ and $\Omega_2$ are the constraints regarding product structures of $A_1$ and $A_2$.

By modeling the two DLBs for disassembling two products independently and together, respectively, we observe that the latter is difficult to solve and is more likely to achieve a globally optimal solution, because it has a larger solution space than the former. As a result, we prefer to deal with a DLB model by considering simultaneously disassembling multiple products in a planning horizon to reach a globally optimal solution. To this end, this work studies a multi-product disassembly line balancing problem (named as MDLBP for short). Additionally, it is necessary to take the uncertainty into consideration, because the disassembly process usually suffers from it as the result of diverse unpredictable and uncontrollable factors. Specifically, it is hard for decision-makers to accurately evaluate the disassembly time and set-up time of disassembly tasks in advance, since the various usage processes of EOL products result in the intractability of exactly knowing their information. To overcome this difficulty, the disassembly time and set-up time of disassembly tasks are expressed as random numbers obeying their known stochastic distributions. Moreover, a decision-maker needs to concern profit-desired and time-related criteria in practice. Hence, this work considers a stochastic MDLBP with the aim to reach the expected disassembly profit maximization while satisfying disassembly

| Table 1 Models of DLB for disassembling $A_1$ and $A_2$ |
|----------------------------------------------------------|
| DLB models for disassembling $A_1$ and $A_2$ independently | DLB model for disassembling $A_1$ and $A_2$ together |
| Model 1: $\max f(x_i)$ | Model 3: $\max f(x_j) + f(x_i)$ |
| $s.t. \ t_f(x_i) \leq T_1$ | $s.t. \ t_f(x_i) + t_f(x_j) \leq T_1 + T_2$ |
| $x_i \in \Omega_1$ | $x_i \in \Omega_1$ |
| Model 2: $\max f(x_j)$ | |
| $s.t. \ t_f(x_j) \leq T_2$ | |
| $x_j \in \Omega_2$ | |
| Model 3: $\max f(x_j) + f(x_i)$ | |
| $s.t. \ t_f(x_i) + t_f(x_j) \leq T_1 + T_2$ | |
| $x_i \in \Omega_1$ | $x_j \in \Omega_2$ |
time constraints. By making comparisons between this work and the existing studies, we have mainly made the following three contributions:

1. This work proposes a novel stochastic MDLBP to achieve a global optimization for disassembling multiple products simultaneously. Then, a chance-constrained programming model with the objective of maximal disassembly profit is established to describe this problem mathematically.

2. We particularly design an enhanced group teaching optimization algorithm (EGTOA) incorporating a stochastic simulation approach to deal with the formulated model. In this method, EGTOA with some special strategies such as solution representation, decoding and local-best search approaches is designed to search candidate solutions from a solution domain, while the stochastic simulation approach is developed to assess the performance and feasibility of the obtained solutions.

3. By performing simulation experiments on several real-life cases and comparing the proposed method with five popularly known algorithms, i.e., group teaching optimization algorithm (GTOA) [38], PSO [39], improved gravitational search algorithm (IGSA) [40], variable neighborhood search (VNS) [41] and SA [42], we validate the effectiveness of the proposed method in addressing the studied problem.

The rest of this paper is organized as follows. The investigated problem is described in the section “Problem formulation”. Then, the section “Proposed method” introduces the basic GTOA and illustrates the solution algorithm. Next, simulation experiments are conducted and the results are analyzed in the section “Experimental study”. Finally, we conclude this paper and explore our future research in the section “Conclusion”.

Problem formulation

Problem statement

This work considers simultaneously disassembling multiple EOL products on a disassembly line in a planning period, and we propose a stochastic MDLBP with maximizing the expected disassembly profit subject to a given disassembly time restriction. MDLBP aims at assigning a series of disassembly tasks of each EOL product to workstations. The disassembly process must obey the precedence relationships of disassembly tasks. In the past years, AND/OR graphs have been triumphantly applied to describe the precedence constraints [43–45]. This work also employs AND/OR graphs to formulate the complex relationships of disassembly tasks.

Let \( i \) indicate a component index, and \( j \) and \( k \) represent disassembly tasks indices. To construct an AND/OR graph for a product, this work defines the following three matrices:

1. Precedence matrix \( C = [c_{jk}] \): It demonstrates precedence and conflict relations of two disassembly tasks as
   \[
   c_{jk} = \begin{cases} 
   1, & \text{if task } j \text{ can be executed after task } k \\
   -1, & \text{if task } j \text{ and task } k \text{ conflict with each other} \\
   0, & \text{otherwise} 
   \end{cases}
   \]

2. Succession matrix \( S = [s_{jk}] \): It denotes the relations between two adjacent disassembly tasks as
   \[
   s_{jk} = \begin{cases} 
   1, & \text{if task } k \text{ can be executed next to task } j \\
   0, & \text{otherwise} 
   \end{cases}
   \]

3. Incidence matrix \( D = [d_{iy}] \): It describes the relations between components and disassembly tasks as
   \[
   d_{iy} = \begin{cases} 
   1, & \text{if component } i \text{ is obtained by task } j \\
   -1, & \text{if component } i \text{ is disassembled via task } j \\
   0, & \text{otherwise} 
   \end{cases}
   \]

This work intends to simultaneously disassemble multiple products on a disassembly line. Thus, multiple AND/OR graphs are supposed to be adopted to describe their corresponding products. By employing AND/OR graphs, we can formulate the precedence constraints for all the disassembled products. Besides, the workload of each workstation is not allowed to exceed a given value, i.e., cycle time. To depict this constraint, the total disassembly time of the assigned disassembly tasks to a workstation cannot be larger than the cycle time.

Mathematical model

To formulate the mathematical model for the concerned problem, we define the notations as:

| (1) Parameters | \( p \) | \( p \in \{1,2,\ldots,P\} \) | \( p \in \{1,2,\ldots,P\} \) | \( P \) indicates the number of disassembled products |
|----------------|--------------------|-----------------------------|-------------------------------|
| \( i \) | \( i \in \{1,2,\ldots,N_p\} \) | \( N_p \) represents the number of components of product \( p \) |
| \( j, k \) | \( j, k \in \{0,1,2,\ldots,J_p\} \) | \( J_p \) means the number of tasks of product \( p \), and 0 is a dummy task |
| \( m \) | \( m \in \{1,2,\ldots,M\} \) | \( M \) denotes the number of workstations |
| \( t_{pj} \) | executing time of \( j \) on product \( p \) |
\[ p^{r}\] set-up time of executing task \( k \) if it is executed after task \( j \) on product \( p \).

\[ c^{r}_{pj} \] cost per time unit of executing task \( j \) on product \( p \).

\[ c^{s}_{pk} \] cost per time unit of setting up task \( k \) if it is executed after task \( j \) on product \( p \).

\( F \) fixed cost of opening a workstation

\( r_{pi} \) recycling value of component \( i \) of product \( p \)

\( CT \) cycle time

\( T \) a given disassembly time limitation

\( \alpha \) confidence level of the disassembly time regarding \( CT \)

\( \beta \) confidence level of the disassembly time regarding \( T \)

\( D_{p} \) incidence matrix of an AND/OR graph of product \( p \)

\( S_{p} \) success matrix of an AND/OR graph of product \( p \)

\( d_{pj} \) an element in row \( i \) and column \( j \) of \( D_{p} \)

\( s_{pk} \) an element in row \( j \) and column \( k \) of \( S_{p} \)

It is worth mentioning here that \( t_{pj}^{e} \) and \( t_{pk}^{e} \) are random.

(2) Decision variables

\[ w_{pj} \] task indication. If task \( j \) is executed on product \( p \), \( w_{pj} = 1 \). Otherwise, \( w_{pj} = 0 \)

\[ x_{pjm} \] assignment indication. If task \( j \) is assigned to workstation \( m \) on product \( p \), \( x_{pjm} = 1 \). Otherwise, \( x_{pjm} = 0 \)

\[ y_{pk} \] adjacent indication. If task \( k \) is executed after task \( j \) on product \( p \), \( y_{pk} = 1 \). Otherwise, \( y_{pk} = 0 \)

\[ z_{m} \] open indication. If workstation \( m \) is opened, \( z_{m} = 1 \). Otherwise, \( z_{m} = 0 \).

By adopting the above-mentioned indices, a chance-constrained programming model for the considered MDLBP can be established as

\[
\max E \left\{ \sum_{p=1}^{P} \sum_{j=1}^{J_{p}} \sum_{i=1}^{N_{p}} d_{pj} r_{pi} w_{pj} - \sum_{p=1}^{P} \sum_{j=1}^{J_{p}} \sum_{i=1}^{N_{p}} t_{pj}^{e} c_{pj}^{e} w_{pj} - \sum_{p=1}^{P} \sum_{j=1}^{J_{p}} \sum_{k=1}^{J_{p}} t_{pk}^{e} t_{pk}^{r} y_{pk} - \sum_{m=1}^{M} z_{m} f \right\}
\]

s.t.

\[
\sum_{m=1}^{M} x_{pjm} = 1, \quad j = 1, 2, \ldots, J_{p}, \quad p = 1, 2, \ldots, P. \tag{2}
\]

\[
\sum_{m=1}^{M} x_{pjm} \geq \sum_{m=1}^{M} x_{pkm}, \quad \forall y_{pk} = 1, \quad p = 1, 2, \ldots, P. \tag{3}
\]

\[
s_{pk} - y_{pk} \geq 0, \quad j, k = 1, 2, \ldots, J_{p}, \quad p = 1, 2, \ldots, P. \tag{4}
\]

\[
0 \leq \sum_{j=0}^{J_{p}} d_{pj} w_{pj} \leq 1, \quad i = 1, 2, \ldots, N_{p}, \quad p = 1, 2, \ldots, P. \tag{5}
\]

Here, (1) demonstrates that the objective is to achieve the expected disassembly profit maximization. The disassembly profit is random, since it relates to random disassembly time and set-up time. \( E(\eta) \) indicates an expected value of a random variable \( \eta \). (2) guarantees that a disassembly task can be allocated to a workstation at most. (3) represents that the precedence relationships among tasks must be satisfied. (4) reveals that a feasible disassembly task sequence of an EOL product should meet precedence and conflict relationships. (5) denotes the incidence relationships between tasks and components of an EOL product. (6) indicates a chance constraint for the actual workload per workstation, where \( \alpha \) is the probability that the actual workload of a workstation is not larger than \( CT \). (7) illustrates a chance constraint for the disassembly time and \( \beta \) is the probability that the disassembly time is less than or equal to \( T \). (8) gives the range of decision variables.

### Proposed method

Compared with disassembling only one product on a disassembly line in the literature [40], the studied problem has some unique characteristics. For example, it has a larger decision space, leading to the difficulty of finding the optimal solutions. Moreover, acquiring feasible solutions is very hard because of the disassembly time requirements, workstation availability, and complicated precedence relationships of disassembly tasks. Thus, we have to design an effective optimizer to search promising and feasible solutions as quickly as possible. Furthermore, the disassembly process usually faces a great deal of uncertainty, because it
is intractable to accurately know the information of EOL products beforehand as a result of their various usage processes in diverse customers [46–50]. Therefore, we need to cope with the noise from random disturbance. To overcome the above challenges, we introduce an intelligent optimization algorithm named as EGTOA incorporating a stochastic simulation approach.

**Basic framework of GTOA**

GTOA is a novel population-based meta-heuristic algorithm to solve continuous optimization problems [38]. By simulating the group teaching mechanism in which students are taught according to their aptitude, it can effectively deal with varied optimization problems [38]. In GTOA, a student represents an individual, i.e., a solution, and his knowledge denotes the fitness value of this solution. GTOA mainly contains four phases, i.e., teacher allocation, ability grouping, teacher, and student phases. These four phases are repeated until a given termination criterion is reached. In teacher allocation phase, the teacher is determined and allocated. In ability grouping phase, all students are divided into two groups, i.e., outstanding and average groups, based on their ability of accepting knowledge. In teacher phase, a student can learn knowledge from the teacher and teaching plans are different between outstanding group and average group. In student phase, a student can acquire knowledge by self-learning and interaction with other students. This work sums up the procedure of basic GTOA as [38]:

1. **Step 1:** Set algorithm parameters including the maximum number of fitness evaluations \(T_m\), the current number of fitness evaluations \(T_c\), and population size \(Q\).
2. **Step 2:** Randomly generate and evaluate an initial population.
3. **Step 3:** Allocate the teacher \(T_t\) by selecting the top three individuals and calculate \(T_t\).
4. **Step 4:** Divide the population into an outstanding group \(v^*_g\) and an average group \(v^*_b\) based on their fitness values.
5. **Step 5:** Conduct teaching and learning phases by using the following two steps:
   - **Step 5.1:** Perform the teacher and student phases on \(v^*_g\), and then obtain \(v^{g+1}_g\).
   - **Step 5.2:** Perform the teacher and student phases on \(v^*_b\), and then obtain \(v^{b+1}_b\).
6. **Step 6:** Construct a new population which comprises of \(v^{g+1}_g\) and \(v^{b+1}_b\).
7. **Step 7:** Stop this iteration process if a given termination criterion is met. Otherwise, it goes to Step 3.

GTOA has many merits such as utilizing fewer parameters than most of other meta-heuristic algorithms, easy realization and great computation efficiency. It has shown admirable performance in handling varied complex optimization problems such as shading problems in photovoltaic systems [51] and engineering design problems [52]. However, as far as we know, GTOA has not been adopted to address DLBPs and its potential in industrial applications needs to be further investigated. Thus, to extend it to handle the proposed problem and explore its performance, this work designs an enhanced GTOA with some special strategies such as the solution representation and a local-best search method according to the features of the concerned problem. Its detailed design is demonstrated below.

**The framework of EGTOA**

**Solution representation and population initialization**

In EGTOA, a double-linked list structure with real values is taken as an encoding form based on the features of the concerned problem. An individual, i.e., a solution, can be represented as \(v = (v_1, v_2, \ldots, v_p)\), where \(v_p\) indicates a disassembly decision of product \(p\), and it is expressed as \(\{v'_p, v''_p\}, p = 1, 2, \ldots, P\). \(v'_p = \{o_{p1}, o_{p2}, \ldots, o_{pp}\}\) denotes a disassembly task sequence decision of product \(p\), while \(v''_p = \{x_{p1}, x_{p2}, \ldots, x_{pp}\}\) indicates a disassembly task execution decision of product \(p\). Notice that \(\{v'_p, v''_p\}\) is defined in continuous decision space. It must be transformed into corresponding \(\{\varphi'_p, \varphi''_p\}\) locating in the discrete decision space for evaluations. To do it, we design a decoding method as:

1. For the disassembly task sequence decision of product \(p\), we map the \(j\)-th smallest value in \(v'_p\) to the \(j\)-th task in \(\varphi'_p\).
2. For the disassembly task execution decision of product \(p\), if a value in \(v''_p\) is larger than 0.5, the value on the corresponding position in \(\varphi''_p\) is 1, and otherwise 0. According to this method, we can transform \(\{v'_p, v''_p\}\) into \(\{\varphi'_p, \varphi''_p\}\) effectively.

Note that \(\varphi''_p\) is a binary vector, and each element in \(\varphi''_p\) indicates whether the disassembly task on the corresponding position in \(\varphi'_p\) is executed. If it is equal to 1, this disassembly task is executed, and otherwise, it is not executed. To describe the above-mentioned transformation process more...
clearly, we give an example with 5 disassembly tasks of product \( p \), as shown in Table 2.

In addition, to guarantee the search efficiency of EGTOA in the tremendous decision space, we have to generate feasible solutions that meet precedence and conflict relationships. When infeasible solutions arise, this work uses the following method to adjust them: (1) The disassembly task sequence \( \varphi_p^* \) is repaired to satisfy precedence relationships based on precedence matrix \( C \), \( p = 1, 2, \ldots, P \). (2) The disassembly task execution decision \( \varphi_p^{**} \) is modified to meet the precedence and conflict relationships according to precedence matrix \( C \) and succession matrix \( S \), \( p = 1, 2, \ldots, P \).

Notice that the above solution representation does not give the assignment of disassembly tasks among workstations. This work proposes the decoding rules as: (1) On a disassembly line, each EOL product is disassembled according to its disassembly decision shown in a solution. (2) A workstation is initially opened. Each executed disassembly task is sequentially assigned to the earliest available workstation based on the disassembly order of products that it belongs to. If the workload of a workstation exceeds the given \( CT \), another workstation is opened.

According to the above-mentioned methods, an initial population containing \( Q \) feasible solutions can be randomly produced.

### Solution evaluation

To describe uncertainty mathematically, this work defines the disassembly time and set-up time of disassembly tasks as random numbers which obey their known random distributions. Consequently, the objective function value is expressed as an expected value, and thus, we cannot directly calculate it like addressing deterministic optimization problems. Monte Carlo simulation is regarded as an effective way to assess probability functions [53, 54]. It adopts the average objective function value of a vast number of samples to approximate the expected value. In theory, this approach can achieve the true objective function value if the number of samples reaches infinite. Nevertheless, we have to do such simulation with the limited computation resources in reality. Therefore, to make good use of the given computation resources for evaluating the performance and feasibility of obtained solutions in a search process, this work develops a stochastic simulation approach. Let \( \Phi \) be an evaluated solution and \( L \) be the number of samples. The method is described as follows:

| \( \varphi_p' \) | 0.36 | 0.11 | 0.13 | 0.54 | 0.86 | \( \varphi_p^{**} \) | 0.24 | 0.95 | 0.18 | 0.59 | 0.16 |
|-----------------|------|------|------|------|------|-----------------|------|------|------|------|------|
| \( \varphi_p'' \) | 3    | 1    | 2    | 4    | 5    | \( \varphi_p^{**} \) | 0    | 1    | 0    | 1    | 0    |

**Step 1:** Initialize \( \overline{A}(\Phi) = 0 \), \( T_l(\Phi) = 0 \), \( F_l(\Phi) = 0 \), \( N_l(\Phi) = 1 \), and \( L = 30 \).

**Step 2:** Generate \( L \) samples and each sample \( \gamma_l \) includes the disassembly time and set-up time produced from their random distributions, \( l = 1, 2, \ldots, L \).

**Step 3:** Calculate disassembly time \( T_l(\Phi) \) and the number of opened workstations \( N_l(\Phi) \), \( l = 1, 2, \ldots, L \).

**Step 4:** Calculate the fixed cost of opened workstations \( F_l(\Phi) \) and the disassembly profit \( A_l(\Phi) \), \( l = 1, 2, \ldots, L \).

**Step 5:** Sort \( T_l(\Phi) \), \( l = 1, 2, \ldots, L \), in an ascending order and let \( l^* = \lfloor \beta \cdot L \rfloor \). If \( T_{l^*}(\Phi) \leq T^* \), \( \Phi \) satisfies Eq. (7) and go to Step 6. Otherwise, \( \Phi \) is infeasible, and let \( \overline{A}(\Phi) \) equal to 0.0001.

**Step 6:** Compute the expected disassembly profit \( \overline{A}(\Phi) \) as:

\[
\overline{A}(\Phi) = \frac{\sum_{l=1}^{L} A_l(\Phi)}{L}.
\]

**Step 7:** Return \( \overline{A}(\Phi) \) as the approximate objective function value of \( \Phi \).

### Teacher allocation phase

Basic GTOA adopts the top three individuals obtained in a search process to guide the population to search promising solutions. Let \( v_1', v_2' \) and \( v_3' \) be the first, second and third best students, respectively. \( T^* \) denotes the teacher, and \( f(x) \) represents the objective function value of \( x \). The teacher allocation can be defined as:

\[
T^* = \begin{cases} 
 v_1' , & f\left(\frac{v_1' + v_2' + v_3'}{3}\right) < f(v_1') \\
 v_2' + v_3', & f\left(\frac{v_1' + v_2' + v_3'}{3}\right) \geq f(v_1'). 
\end{cases}
\]  

### Ability grouping phase

Basic GTOA divides all students into two groups, i.e., an outstanding group and an average group, in terms of their ability of accepting knowledge. According to their fitness values, the top \( Q/2 \) individuals in the class constitute the outstanding group \( v_1' \), while the rest of the class composes the average group \( v_3' \). Notice that the ability grouping is dynamic. It is performed again after a learning cycle. In addition, to accelerate the convergence of this algorithm, outstanding and average groups share the same teacher.
Teacher phase

Basic GTOA considers that students are able to gain knowledge from a teacher. The teacher makes different teaching plans for two groups according to their ability of accepting knowledge. For $v_{st}^t$, the teacher attempts to increase the mean knowledge of the whole class. It can be represented as:

$$v_{et}^{t+1} = v_{et}^t + a(T - F(b \cdot M + c \cdot v_{et}^t)),$$

where $v_{et}^{t+1}$ is the knowledge that student $i$ learns from the teacher at time $t+1$, $v_{et}^t$ and $M$ are the knowledge of student $i$ and the mean knowledge of the whole class at time $t$, and $t$ denotes the iteration times. $a$, $b$, and $c$ are random numbers ranging from 0 to 1. The sum of $b$ and $c$ is equal to 1. $F$ is the teaching factor denoting the teaching result of the teacher, and it is a constant equaling to 1 or 2.

Different from $v_{g}^t$, $v_{h}^t$ is poor in accepting knowledge. Generally, a teacher gives more attention to $v_{g}^t$ than $v_{h}^t$. As a result, the teacher tries to particularly enhance the knowledge of the students of $v_{g}^t$. It is presented as Eq. (12):

$$v_{et}^{t+1} = v_{et}^t + 2d(T - v_{et}^t),$$

where $d$ is a random number between 0 and 1.

It is worth noting that students may not be able to acquire knowledge in this phase, which can be formulated by Eq. (13):

$$v_{et}^{t+1} = \begin{cases} v_{et}^{t+1}, & f(v_{et}^t) < f(v_{et}^{t+1}) \\ v_{et}^t, & f(v_{et}^t) \geq f(v_{et}^{t+1}) \end{cases}.$$ 

Student phase

In addition to acquiring knowledge from the teacher, a student can gain knowledge with two methods, i.e., self-learning and interaction with other students. They can be expressed as Eq. (14):

$$v_{si}^{t+1} = \begin{cases} v_{si}^{t+1} + e(v_{si}^{t+1} - v_{ji}^{t+1}) + g(v_{si}^t - v_{ji}^t), & f(v_{si}^{t+1}) < f(v_{si}^t) \\ v_{si}^{t+1} - e(v_{si}^{t+1} - v_{ji}^{t+1}) + g(v_{si}^t - v_{ji}^t), & f(v_{si}^{t+1}) \geq f(v_{si}^t) \end{cases},$$

where $e$ and $g$ are two random numbers between 0 and 1, $j$ denotes a student which is randomly selected from the class, and $i$ represents the current student. $v_{si}^{t+1}$ is the knowledge that student $i$ learns at time $t+1$, while $v_{si}^t$ is the knowledge that student $j$ learns in teacher phase at time $t+1$. In Eq. (14), the self-learning and interaction from other students are defined as the second and third items on the right, respectively.

Notice that a student may not be able to gain knowledge in student phase. Equation (15) is employed to formulate this situation. It is given as:

$$v_{et}^{t+1} = \begin{cases} v_{et}^{t+1}, & f(v_{et}^{t+1}) < f(v_{et}^t) \\ v_{et}^t, & f(v_{et}^{t+1}) \geq f(v_{et}^t) \end{cases},$$

where $v_{et}^{t+1}$ is the knowledge of student $i$ at time $t+1$.

Local-best search method

To strengthen the exploitation ability of EGTOA, a local-best search approach is developed to refine the best solution obtained in the search process. It can be described as: (1) the best solution $v_t$ is chosen. Then, two positions are randomly selected in its first string $v_t$ and values on them are swapped. If the value $\rho$ on the corresponding position in the second string $v^t_2$ is less than 0.5, then $\rho$ equals to 0. Otherwise, $\rho$ is equal to 1. Thus, a neighborhood solution can be produced. (2) The searched neighborhood solution in the continuous decision space is transformed into a corresponding solution in the discrete decision space to evaluate its performance and feasibility. (3) If the transformed neighborhood solution locating in the discrete decision space is infeasible, we adopt the method in the section “Solution representation and population initialization” to adjust it. Otherwise, the new solution and original solution are directly compared. The better one is chosen as the current best solution. (4) If a given termination condition is satisfied, i.e., reaching the maximum number of neighborhood solutions $I$, we output the current best solution. Otherwise, we repeat (1–3). Nevertheless, the computation resources are usually limited. Consequently, to make efficient use of the limited computation resources, the local-best search method is applied with probability $R$ at each iteration.

Procedure of EGTOA

Having discussed all the phases of EGTOA in detail, we summarize its framework as: (1) Algorithm parameters are set, and an initial population containing a set of randomly generated solutions is produced and evaluated. (2) The teacher is allocated. (3) All students are divided into outstanding and average groups. (4) The teacher and student phases are sequentially performed on these two groups. (5) The local-best search is conducted with probability $R$. EGTOA repeats (2–5) until a given termination criterion is
reached. At last, the best solution is output. The procedure of EGTOA is given in Algorithm 1.

Algorithm 1: EGTOA

**Input:** Algorithm parameters.

**Output:** The best solution.

Set algorithm parameters.

Generate an initial population and evaluate it.

While (a termination criterion is not met) Do

Allocate the teacher.

Divide all students into outstanding and average groups based on their fitness values.

Perform the teacher phase on outstanding and average groups.

Perform the student phase on outstanding and average groups.

Conduct local-best search with probability $R$.

End while

Output the best solution.

**Experimental study**

To examine the performance of EGTOA in handing the concerned problem, this work carries out simulation experiments on a set of randomly produced test problems, and makes comparisons between EGTOA and five competitive approaches: PSO [39], IGSA [40], VNS [41], SA [42], and GTOA [38]. PSO and IGSA are two population-based metaheuristics. PSO has been triumphantly extended to deal with varied complex optimization problems [8, 55, 56] including DLBPs [40]. IGSA has been employed to solve DLBPs [40]. This work extends them to deal with the considered problem as the design in the study [40] which also focuses on solving DLBPs. Because EGTOA uses a local-best search method, we also choose two single-solution-based metaheuristics, i.e., VNS and SA, for comparisons. They have also been applied to tackle DLBPs in the existing studies and we extend them to solve the investigated problem as the literature [41, 42]. Furthermore, to test the effectiveness of the local-best search method, GTOA [38] is also taken as a compared method.

**Test instance generation**

This work adopts four products that are widely used for experiments in the existing literature, i.e., a radio set [57], a copy machine [57], a hammer drill [58] and a microwave oven [58], to randomly generate five test problems with $n$ EOL products, $n \in \{2, 4, 6, 8, 10\}$. Each test problem contains three instances. The data for each instance are randomly produced from the intervals in Table 3. Besides, we assume that the disassembly time and set-up time of disassembly tasks follow their corresponding truncated normal distribution. Their mean time is randomly generated from the intervals, as shown in Table 3. Their standard deviation is equal to a product of their mean time and 0.00001. Additionally, the cycle time and fixed cost of opening a workstation are constants, and they are equal to 100 and 10, respectively. Furthermore, to set the given disassembly time limitation of an instance, this work randomly produces a

| Parameters | Intervals |
|------------|-----------|
| $t_{pj}$   | [5, 20]   |
| $p_{jk}$   | [1, 10]   |
| $c_{pj}$   | [1, 20]   |
| $c_{pj}$   | [1, 10]   |
| $r_{pj}$   | [300, 400]|

**Table 3 Test instance generation method**

**Table 4 Parameter settings of PSO, IGSA, VNS and SA**

| Algorithms | Parameters | Value |
|------------|------------|-------|
| PSO        | Cognitive and social constants | 2, 2  |
|            | Inertia weight             | 1     |
|            | Population size            | 50    |
| IGSA       | Alpha                    | 20    |
|            | Initial value             | 100   |
|            | Rnorm, Rpower             | 2, 1  |
|            | Population size            | 50    |
| VNS        | Number of shaking structures | 6     |
|            | Number of local search structures | 6     |
| SA         | Initial temperature       | 1000  |
|            | Cooling factor             | 0.9   |
solution of the instance and calculates its disassembly time, then let this disassembly time multiplied by 3 be the given disassembly time limitation of this instance. The confidence levels $u_1$ and $u_2$ are both equal to 0.90.

**Parameter settings**

The parameter settings of PSO and IGSA are obtained from their original studies [39, 40]. PSO has been successfully used to cope with a DLBP in [40]. It can acquire relatively better results when its population size is set to 50 as the setting in [40]. Other parameter settings of PSO refer to its original study [39]. The parameter settings of IGSA, VNS and SA are derived from existing studies on addressing DLBPs in [40–42]. Table 4 gives the parameter settings of PSO, IGSA, VNS and SA. In addition, to make a fair comparison, the number of fitness evaluations is taken as a stopping criterion equaling to $30\ P \cdot \ U$, where $U$ represents the maximum number of disassembly tasks in the solved instance.

To obtain a relatively better parameter combination for EGTOA, an Orthogonal Experiment [59] is conducted. There are three parameters in EGTOA and each of them has four levels, i.e., $Q \in \{20, 40, 60, 80\}$, $R \in \{0.05, 0.10, 0.15, 0.20\}$ and $I \in \{20, 40, 60, 80\}$. Hence, an orthogonal array $L_{16}(4^3)$ can be constructed. It contains 16 different parameter combinations, as shown in Table 5. EGTOA with each parameter combination performs 20 times for solving an instance with two EOL products. The obtained best disassembly profit value of each time is used as the response variable and the average response variable (ARV) of 20 times is applied to analyze the experimental results.

Table 5 gives the ARVs of EGTOA with all parameter combinations. It can be found that EGTOA can achieve a better result when $Q = 20$, $R = 0.20$ and $I = 80$. Thus, this parameter combination is employed in the following experiments. To make a fair and valid comparison between EGTOA and GTOA, the population size of GTOA is set to 20.

**Comparison analysis and discussion**

In this section, we conduct comparison experiments to test the performance of EGTOA and its peer approaches.

| Ins. | $Q$  | $R$  | ARV  |
|------|------|------|------|
| 1    | 20   | 0.05 | 20   | 5778.80 |
| 2    | 20   | 0.10 | 40   | 5801.87 |
| 3    | 20   | 0.15 | 60   | 5813.37 |
| 4    | 20   | 0.20 | 80   | 5813.88 |
| 5    | 40   | 0.05 | 40   | 5743.02 |
| 6    | 40   | 0.10 | 20   | 5772.95 |
| 7    | 40   | 0.15 | 80   | 5810.55 |
| 8    | 40   | 0.20 | 60   | 5760.77 |
| 9    | 60   | 0.05 | 60   | 5767.33 |
| 10   | 60   | 0.10 | 80   | 5744.63 |
| 11   | 60   | 0.15 | 20   | 5783.78 |
| 12   | 60   | 0.20 | 40   | 5779.58 |
| 13   | 80   | 0.05 | 80   | 5784.20 |
| 14   | 80   | 0.10 | 60   | 5803.68 |
| 15   | 80   | 0.15 | 40   | 5761.70 |
| 16   | 80   | 0.20 | 20   | 5789.07 |

| Table 6 | Comparison results of EGTOA, PSO and IGSA |
|---------|-----------------------------------------|
| Ins.   | EGTOA | PSO | IGSA |
|        | Mean  | Std | Mean  | Std | Mean  | Std |
| 1      | 5854.17 | 91.23 | 5796.30 | 65.47 | 5690.80 | 124.16 |
| 2      | 4829.50 | 58.69 | 4806.40 | 50.81 | 4773.52 | 42.75 |
| 3      | 4972.43 | 77.08 | 5001.40 | 92.47 | 4916.75 | 87.63 |
| 4      | 12,263.20 | 99.21 | 12,140.00 | 118.63 | 11,767.10 | 178.53 |
| 5      | 10,523.60 | 122.69 | 10,457.70 | 95.26 | 10,351.30 | 145.04 |
| 6      | 8874.03 | 165.14 | 8879.28 | 159.66 | 8679.47 | 123.75 |
| 7      | 13,790.40 | 224.07 | 13,795.90 | 193.95 | 13,468.60 | 161.42 |
| 8      | 14,352.70 | 250.71 | 14,244.80 | 172.46 | 13,979.00 | 182.65 |
| 9      | 14,621.20 | 232.10 | 14,691.90 | 214.96 | 14,276.40 | 156.61 |
| 10     | 20,311.70 | 192.71 | 20,260.70 | 203.70 | 19,714.70 | 177.87 |
| 11     | 19,362.00 | 202.45 | 19,312.20 | 192.63 | 18,919.40 | 150.41 |
| 12     | 18,806.40 | 217.91 | 18,810.00 | 206.72 | 18,326.90 | 214.47 |
| 13     | 25,082.50 | 365.82 | 24,842.00 | 175.98 | 24,218.30 | 253.16 |
| 14     | 21,071.40 | 276.26 | 21,134.20 | 205.18 | 20,609.80 | 228.43 |
| 15     | 24,520.40 | 276.26 | 24,217.80 | 243.90 | 23,668.00 | 199.71 |
| Average| 14,615.71 | 186.36 | 14,559.37 | 159.45 | 14,224.00 | 161.77 |
EGTOA, GTOA, PSO, IGSA, VNS and SA are applied to solve each instance 20 times. The mean and standard deviation (std) values over 20 times are used to analyze their performance. Table 6 shows the experimental results of EGTOA and two population-based meta-heuristic methods, i.e., PSO and IGSA. From Table 6, by comparing EGTOA with PSO, it can be observed that EGTOA can obtain better mean disassembly profit values than PSO on 9 out of 15 instances. Via making a comparison between EGTOA with IGSA, we can see that the mean disassembly profit values of EGTOA are larger than those of IGSA on all the instances. Besides, the average values of mean disassembly profit values on all the instances of EGTOA, PSO and IGSA are 14,615.71, 14,559.37 and 14,224.00, respectively. We can find that the average value of mean disassembly profit values on all the instances of EGTOA is better than that of PSO and IGSA.

Table 7 gives the experimental results of EGTOA and two single-solution-based meta-heuristic methods, i.e., VNS and SA. From Table 7, we can observe that EGTOA can obtain better mean disassembly profit values than VNS on all the instances, and EGTOA can gain better mean disassembly profit values than SA on 14 out of 15 instances. The standard deviation values of EGTOA are smaller than those of VNS and SA on all the instances. In addition, the average values of mean disassembly profit values on all the instances of EGTOA, VNS and SA are 14,615.71, 13,154.90 and 13,074.40, respectively. We can observe that EGTOA can achieve a better performance than VNS and SA in terms of the average value of mean disassembly profit values on all the instances. The above-mentioned comparison results demonstrate that EGTOA outperforms PSO, IGSA, VNS and SA in handling the considered problem.

Table 8 Comparison results of EGTOA and GTOA

| Ins. | EGTOA Mean | Std | GTOA Mean | Std |
|------|-------------|-----|-----------|-----|
| 1    | 5854.17     | 91.23 | 5779.63   | 86.01 |
| 2    | 4829.50     | 58.69 | 4784.65   | 70.65 |
| 3    | 4972.43     | 77.08 | 4930.12   | 62.20 |
| 4    | 12,263.20   | 99.21 | 12,211.20 | 140.65 |
| 5    | 10,523.60   | 122.69 | 10,512.70 | 141.91 |
| 6    | 8874.03     | 165.14 | 8763.65   | 130.47 |
| 7    | 13,790.40   | 224.07 | 13,789.90 | 251.50 |
| 8    | 14,352.70   | 250.71 | 14,172.00 | 238.39 |
| 9    | 14,621.20   | 232.10 | 14,227.50 | 236.36 |
| 10   | 20,311.70   | 192.21 | 19,238.10 | 122.33 |
| 11   | 19,362.00   | 202.45 | 18,632.60 | 207.92 |
| 12   | 18,806.40   | 217.91 | 18,963.00 | 353.08 |
| 13   | 25,082.50   | 365.82 | 24,958.40 | 353.08 |
| 14   | 21,071.40   | 219.89 | 21,011.10 | 262.61 |
| 15   | 24,520.40   | 276.26 | 24,347.20 | 149.76 |
| Average | 14,615.71 | 186.36 | 14,528.38 | 174.14 |

Besides, to verify the effectiveness of the local-best search method in EGTOA, comparison experiments between EGTOA and GTOA are conducted. Table 8 gives their experimental results. It is obvious that the mean disassembly profit values of EGTOA are greater than those of GTOA on all the instances. In addition, it can be found that the average value of mean disassembly profit values on all the instances of EGTOA is larger than that of GTOA. The comparison results indicate that EGTOA performs better than GTOA in solving the concerned problem. Thus, it can be verified that...
the local-best search method is beneficial to enhancing the performance of EGTOA.

Additionally, to further validate the effectiveness of EGTOA, the $t$-test with 38 degrees of freedom at a 0.10 level of significance is adopted in this work. The symbol “+” means that EGTOA is significantly better than its peers, “−” indicates that EGTOA is significantly worse than its peers, and “∼” denotes that EGTOA is statistically equivalent to its peers. Table 9 exhibits the comparison results of $t$-test. It is obvious that EGTOA is significantly better than GTOA on 9 instances and statistically equivalent to GTOA on the rest. EGTOA is significantly better than PSO on 7 instances and statistically equivalent to PSO on the rest. EGTOA is significantly better than IGSA and VNS on all the instances. EGTOA is significantly better than SA on 14 out of 15 instances and statistically equivalent to SA on 1 instance. Via analyzing the comparison results of $t$-test, it can be demonstrated that EGTOA is superior to five competitive methods, i.e., GTOA, PSO, IGSA, VNS and SA, in handling the concerned problem.

Moreover, to visually exhibit the experimental results of six algorithms, i.e., EGTOA, GTOA, PSO, IGSA, VNS and SA, this work draws their boxplot and convergence curve graphs on all the instances as given in Figs. 1 and 2, respectively. From Fig. 1, it can be observed that the obtained results of EGTOA are much better and more concentrated than its five peers on most instances. Furthermore, from Fig. 2, we can see that EGTOA can converge to a better disassembly profit value faster than GTOA, PSO, IGSA, VNS and SA on most instances. Thus, the convergence and optimality of EGTOA are much better than those of its five peers. Via analyzing the above-mentioned experimental results, it can be declared that EGTOA performs better than its five compared methods in tackling the concerned problem.

### Conclusion

To efficiently operate and manage the remanufacturing systems and reach the global optimization in the disassembly process, this work addresses a stochastic profit-oriented multi-product disassembly line balancing problem with consideration of disassembly time limitation. Next, according to the characteristics of this problem, a chance-constrained programming model is formulated to describe it mathematically. To solve it, this work develops an intelligent optimization algorithm named as enhanced group teaching optimization algorithm. Finally, the effectiveness of the proposed method is validated by conducting simulation experiments on some real-life cases and comparing it with five competitive approaches in literature, i.e., group teaching optimization algorithm, particle swarm optimization algorithm, improved gravitational search algorithm, variable neighborhood search algorithm and simulated annealing algorithm. The results...
Fig. 1 Boxplot of EGTOA, GTOA, PSO, IGSA, VNS and SA on all the instances
Fig. 2 Convergence curve graphs of EGTOA, GTOA, PSO, IGSA, VNS and SA on all the instances.
can help decision-makers make wise decisions in stochastic environments when there are multiple EOL products that need to be dismantled together on a disassembly line. In the future research, we plan to explore the following topics: (1) to formulate and tackle a new stochastic multi-product disassembly line balancing problem considering energy-aware optimization; and (2) to develop more efficient approaches to handle the concerned problem with high complexity and stochastic features.

Author contributions PL was the main contributor and writer of the paper’s ideas and experimental design. YPF supervised the work and made much improvement about the method and experimental work. KZG and HS did some experimental work, result analysis, and verification and contributed to the revision of this paper. All authors read and approved the final manuscript.

Funding This work is supported in part by National Natural Science Foundation of China under Grant Nos. 61603169 and 61703220; China Postdoctoral Science Foundation Funded Project under Grant No. 2019T120569; Shandong Province Outstanding Youth Innovation Team Project of Colleges and Universities of China under Grant No. 2020RQG011; Shandong Province Colleges and Universities Youth Innovation Talent Introduction and Education Program, the Faculty Research Grants (FRG) from Macau University of Science and Technology, and Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology.

Availability of data and materials All data generated or analyzed during this study are included in this published article.

Code availability The codes of all algorithms are implemented in VC ++ 2017 and run on an Intel Core i5 CPU (3.20 GHz/8.00 GB RAM) PC with windows 10 Operating System.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Tang M, Liao HC (2019) From conventional group decision making to large-scale group decision making: What are the challenges and how to meet them in big data era? A state-of-the-art survey. Omega. https://doi.org/10.1016/j.omega.2019.102141
2. Roh JW, Jang SB, Kim SY, Cho H, Kim J (2021) Steam trap maintenance-prioritizing model based on big data. ACS Omega 6:4408–4416
3. Gupta S, Garg H, Chaudhary S (2020) Parameter estimation and optimization of multi-objective capacitated stochastic transportation problem for gamma distribution. Complex Intell Syst 6:651–667
4. Chu HK, Zhang WS, Bai PF, Chen YH (2021) Data-driven optimization for last-mile delivery. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00293-1
5. Ding ZY, Jiang ZG, Liu Y, Wang Y, Li CB (2018) A big data based cost prediction method for remanufacturing end-of-Life products. Procedia CIRP 72:1362–1367
6. Zhang YF, Ren S, Liu Y, Sakao T, Huisingsh D (2017) A framework for Big Data driven product lifecycle management. J Clean Prod 159:229–240
7. Fu YP, Tian GD, Fathollahi-Fard AM, Ahmadi A, Zhang CY (2019) Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. J Clean Prod 226:515–525
8. Cui SZ, Yin YQ, Wang DJ, Li ZW, Wang YZ (2021) A stacking-based ensemble learning method for earthquake casualty prediction. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.107038
9. Yu HF (2020) Evaluation of cloud computing resource scheduling based on improved optimization algorithm. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00163-2
10. Agrawal P, Ganesh T, Mohamed AW (2021) Solving knapsack problems using a binary gaining sharing knowledge-based optimization algorithm. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00351-8
11. Feng Y, Wang DJ, Yin YQ, Li ZW, Hu ZN (2020) An XGBoost-based casualty prediction method for terrorist attacks. Complex Intell Syst 6:721–740
12. Fu YP, Xiao H, Lee LH, Huang M (2021) Stochastic optimization using grey wolf optimization with optimal computing budget allocation. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107154
13. Tian GD, Chu J, Hu H, Li H (2014) Technology innovation system and its integrated structure for automotive components remanufacturing industry development in China. J Clean Prod 85:419–432
14. Liang P, Fu YP, Ni SY, Zheng B (2021) Modeling and optimization for noise-aversion and energy-awareness disassembly sequence planning problems in reverse supply chain. Environ Sci Pollut Res. https://doi.org/10.1007/s11356-021-14124-w
15. Yin YQ, Wang Y, Cheng TCE, Liu WQ, Li JH (2017) Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. Omega 69:1–28
16. Yin YQ, Cheng SR, Cheng TCE, Wang DJ, Wu C-C (2016) Just-in-time scheduling with two competing agents on unrelated parallel machines. Omega 63:41–47
17. Yin YQ, Cheng TCE, Lan W, Wu C-C, Liu J (2015) Two-agent single-machine scheduling with deteriorating jobs. Comput Ind Eng 81:177–185
18. Yin YQ, Cheng TCE, Wang DJ (2016) Rescheduling on identical parallel machines with machine disruptions to minimize total completion time. Eur J Oper Res 252(3):737–749
19. Wang DJ, Zhu JQ, Wei XW, Cheng TCE, Yin YQ, Wang YZ (2019) Integrated production and multiple trips vehicle routing with time windows and uncertain travel times. Comput Oper Res 103:1–12
20. Guo XW, Zhou MC, Abusorrah A, Alsokhiry F, Sedraoui K (2021) Disassembly sequence planning: a survey. IEEE/CAA J Autom Sinica 8(7):1308–1324

21. McGovern SM, Gupta SM (2007) A balancing method and genetic algorithm for disassembly line balancing. Eur J Oper Res 179(3):692–708

22. Bentaha ML, Battaiia O, Dolgui A, Hu SJ (2015) Second order conic approximation for disassembly line design with joint probabilistic constraints. Eur J Oper Res 247(3):957–967

23. Ren YP, Zhang CY, Zhao F, Tian GD, Lin WW, Meng LL, Li HL (2018) Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-Optimal algorithm. J Clean Prod 174:1475–1486

24. Ren YP, Zhang CY, Zhao F, Triebe MJ, Meng L (2020) An MCDM-based multiobjective general variable neighborhood search approach for disassembly line balancing problem. IEEE Trans Syst Man Cybernet 50(10):3770–3783

25. Özcelayn E, Kalayci CB, Gungör A, Gupta SM (2019) Disassembly line balancing problem: a review of the state of the art and future directions. Int J Prod Res 57:4805–4827

26. Gao KZ, He ZM, Huang Y, Duan PY, Suganthan PN (2020) A survey on meta-heuristics for solving disassembly line balancing, planning and scheduling problems in remanufacturing. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2020.100719

27. McGovern SM, Gupta SM (2007) Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem. Int J Prod Res 45:4485–4511

28. Fu YP, Hou YS, Wang ZF, Wu XW, Gao KZ, Wang L (2021) Distributed scheduling problems in intelligent manufacturing systems. Tsinghua Sci Technol 26(5):625–645

29. Zhang ZQ, Wang KP, Zhu LX, Wang Y (2017) A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem. Expert Syst Appl 86:165–176

30. Agrawal S, Tiwari MK (2008) A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem. Int J Prod Res 46(6):1405–1429

31. Wang KP, Li XY, Gao L, Garg A (2019) Partial disassembly line balancing for energy consumption and profit under uncertainty. Robot Comput-Integr Manuf 59:235–251

32. Zhu LX, Zhang ZQ, Guan C (2020) Multi-objective parallel partial disassembly line balancing problem using hybrid group neighborhood search algorithm. J Manuf Syst 56:252–269

33. Kalayci CB, Gupta SM (2013) A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. Int J Adv Manuf Technol 69:197–209

34. Wang KP, Li XY, Gao L, Li PG, Gupta SM (2021) A genetic simulated annealing algorithm for parallel partial disassembly line balancing problem. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107404

35. ÇiìZA, Mete S, Serin F (2020) Robotic disassembly line balancing problem: a mathematical model and ant colony optimization approach. Appl Math Model 86:335–348

36. Xiao QX, Guo XP, Li D (2020) Partial disassembly line balancing under uncertainty: robust optimisation models and an improved migrating birds optimisation algorithm. Int J Prod Res. https://doi.org/10.1080/00207543.2020.1744765

37. Wang KP, Li XY, Gao L, Li PG (2020) Energy consumption and profit-oriented disassembly line balancing for waste electrical and electronic equipment. J Clean Prod. https://doi.org/10.1016/j.jclepro.2020.121829

38. Zhang YY, Jin ZG (2020) Group teaching optimization algorithm: a novel metaheuristic method for solving global optimization problems. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113246

39. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95 - International Conference on Neural Networks 4:1942–1948 Perth, Australia.

40. Ren YP, Yu DY, Zhang CY, Tian GD, Meng LL, Zhou QX (2017) An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem. Int J Prod Res 55(24):7302–7316

41. Kalayci CB, Polat O, Gupta SM (2015) A variable neighbourhood search algorithm for disassembly lines. J Manuf Technol Manag 26(2):182–194

42. Wang KP, Li XY, Gao L, Li PG, Gupta SM (2021) Energy consumption and profit-oriented disassembly line balancing for waste electrical and electronic equipment. J Clean Prod. https://doi.org/10.1016/j.jclepro.2020.121829

43. Tian GD, Zhou MC, Li PG (2018) Disassembly sequence planning considering fuzzy component quality and varying operational cost. IEEE Trans Autom Sci Eng 15(2):748–760

44. Guo XW, Liu SX, Zhou MC, Tian GD (2018) Dual-objective program and scatter search for the optimization of disassembly sequences subject to multi-resource constraints. IEEE Trans Autom Sci Eng 15(3):1091–1103

45. Tian GD, Ren YP, Feng YX, Zhou MC, Zhang HH, Tan JR (2019) Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony. IEEE Trans Industr Inf 15(4):2456–2468

46. Fu YP, Zhou MC, Guo XW, Qi L (2020) Stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling via fruit fly optimization algorithm. J Clean Prod. https://doi.org/10.1016/j.jclepro.2020.123364

47. Wang KP, Li XY, Gao L (2019) A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. Comput Ind Eng 130:634–649

48. Fu YP, Zhou MC, Guo XW, Qi L, Sedraoui K (2020) Multivariate optimization algorithm for stochastic biobjective disassembly sequence planning subject to operation failures. IEEE Trans Syst Man Cybernet. https://doi.org/10.1109/TSMC.2021.3049323

49. Tang Y, Zhou MC (2006) A systematic approach to design and operation of disassembly lines. IEEE Trans Autom Sci Eng 3(3):324–329

50. Tian GD, Zhou MC, Chu JW, Liu YM (2012) Probability evaluation models of product disassembly cost subject to random removal time and different removal labor cost. IEEE Trans Autom Sci Eng 9(2):288–295

51. Zafar MH, Al-shahraani T, Khan NM, Feroz Mirza A, Mansoor M, Qadir MU, Khan ML, Naqvi RA (2020) Group teaching optimization algorithm based MPPT Control of PV systems under partial shading and complex partial shading. Electronics. https://doi.org/10.3390/electronics9111962

52. Zhu SK, Wu Q, Jiang X, Xing W (2021) A novel multi-objective group teaching optimization algorithm and its application to engineering design. Comput Ind Eng. https://doi.org/10.1016/j.cie.2021.107198

53. Fu YP, Wang HF, Wang JW, Pu XJ (2020) Multi-objective modeling and optimization for scheduling a stochastic hybrid flow with maximizing processing quality and minimizing total tardiness. IEEE Syst J. https://doi.org/10.1109/JSYST.2020.3014093

54. Fu YP, Zhou MC, Guo XW, Qi L (2020) Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm. IEEE Trans Syst Man Cybernet. 50(12):5037–5048

55. Wang DJ, Qiu HX, Wu C-C, Lin W-C, Lai KJ, Cheng S-R (2019) Dominance rule and opposition-based particle swarm optimization for two-stage assembly scheduling with time cumulated learning effect. Soft Comput 23:9617–9628

56. Wei XW, Qiu HX, Wang DJ, Duan JH, Wang YZ, Cheng TCE (2020) An integrated location-routing problem with post-disaster
relief distribution. Comput Ind Eng. https://doi.org/10.1016/j.cie.2020.106632

57. Lambert AJD (2007) Optimization disassembly processes subjected to sequence-dependent cost. Comput Oper Res 46(2):536–551

58. Guo XW, Zhou MC, Liu SX, Qi L (2020) Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption. IEEE Trans Autom Sci Eng. https://doi.org/10.1109/TASE.2020.2992220

59. Montgomery DC (2005) Design and analysis of experiments. Wiley, Hoboken

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.