

SERVICE-ORIENTED DISASSEMBLY SEQUENCE PLANNING
FOR ELECTRICAL AND ELECTRONIC EQUIPMENT WASTE

Kai Xia^{a,b}, Liang Gao^{a,*}, Lihui Wang^c, Weidong Li^d, Xinyu Li^a, Winifred Ijomah^e

^a*The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*

^b*Wuhan Second Ship Design and Research Institute, Wuhan 430064, China*

^c*Department of Production Engineering, Royal Institute of Technology, Stockholm 100 44, Sweden*

^d*Faculty of Engineering and Computing, Coventry University, Coventry CV1 5FB, UK*

^e*Faculty of Engineering, University of Strathclyde, Glasgow G1 1XJ, UK*

Abstract

Disassembly sequence planning plays an important role in the end-of-life treatment of *electrical and electronic equipment waste* (e-waste). Effective planning methods can improve recovery rates and reduce environmental impacts of e-waste. In previous work, neither mathematical models nor optimization algorithms offered a satisfactory solution for this multi-objective disassembly problem. We present a multi-objective model for the problem and a *modified teaching-learning-based optimization* (MTLBO) algorithm to find the Pareto-optimal frontier. We use numerical simulations to demonstrate and verify the effectiveness and robustness of the algorithm. To do effective disassembly planning, all the participants in the lifecycle of e-waste should work together. Disassembly and recovery of e-waste involve complex processes across the lifecycle. Information support services, disassembly modeling and optimization services must be integrated using computer networks. We also propose a service-oriented framework to support business integration for the participants in the e-waste lifecycle. Effective and optimized disassembly planning can be achieved by invoking the related distributed services. The proposed framework is a novel e-business application for the end-of-life treatment of e-waste.

Keywords: Disassembly sequence planning, e-business, environmental issues, math programming, multi-objective optimization, services, sustainability

1. INTRODUCTION

E-waste has become one of the major and challenging waste streams in terms of quantity and toxicity. Informal treatments result in environmental pollution and secondary resource waste. The Waste Electrical and Electronic Equipment Directive (2002/96/EC) was enacted in 2003 in Europe¹. It aims to prevent the generation of e-waste and promote its recovery to reduce disposal. Many other equivalent directives have been developed and implemented in the world. With legislative pressure for environmental protection, electronics companies have come to recognize that they must take on more responsibility. It is crucial for them to balance economic and environmental objectives, for example, to maximize recovery value and minimize environmental impact.

Disassembly sequence planning plays an important role in end-of-life treatment of e-waste. Effective disassembly planning methods can improve recovery rates and reduce environmental impacts of e-waste. The disassembly sequence planning problem has been proven to be NP-hard, and it has been widely studied in previous research. Different meta-heuristic optimization methods have been proposed and implemented to solve the problem (Kongar and Gupta, 2006; Adenso-Diaz et al., 2007; Xia et al., 2014; Yeh, 2012; Li et al., 2013). Different stakeholders related to e-waste (e.g., remanufacturers, recyclers, and regulators) have different legislative and economic considerations when making disassembly planning arrangements. They have to balance multiple objectives. In order to formulate the multi-objective disassembly sequence planning problem, this paper introduces three disassembly indices to evaluate a disassembly sequence, namely index of diminished toxicity, index of potential recovery value and index of potential recovery weight. Accordingly, we formulated a model with the objective of maximizing these disassembly indices.

The traditional method to solve a multi-objective optimization problem is to weight the

*Corresponding author. Tel.: +86 2787559419; Fax: +86 2787543074.

Email address: gaoliang@mail.hust.edu.cn (Liang Gao)

¹For additional background on this European Commission policy on waste electrical and electronic equipment (WEEE), the interest reader should see ec.europa.eu/environment/waste/weee/index_en.htm, to learn about the prior European policy-related legislation in the 2000s. For the current policy, refer to Directive 2012/19/EU at eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32012L0019.

relative degree of importance of each objective and then transform them into a single-objective optimization problem. However, this method has a drawback for the multi-objective disassembly sequence planning problem. Since disassembly involves different stakeholders with different legislative and economic considerations, one single-objective solution cannot meet the needs of different stakeholders. Furthermore, it makes sense to determine as many non-dominated solutions as possible for a stakeholder to select in order to support better decisions.

Multi-objective evolutionary algorithms have been widely studied and applied to solve multi-objective optimization problems, such as the non-dominated sorting genetic algorithm II (NSGA II) (Deb et al., 2002), the multi-objective evolutionary algorithm based on decomposition (Zhang and Li, 2007), and the multi-objective differential evolution algorithm (Ali et al., 2012). This article proposes a *modified teaching-learning-based optimization* (MTLBO) algorithm to solve the multi-objective disassembly sequence planning problem. Five major components have been designed and incorporated into the algorithm in order to make it applicable for specifying complex disassembly precedence constraints.

Disassembly sequence planning for e-waste always involves globally-distributed participants in the lifecycle of this kind of equipment, such as manufacturers, distributors, retailers, disassemblers, recyclers, remanufacturers, management authorities and so on. To make an effective disassembly plan, all the participants should work together. However, information flows about e-waste in the lifecycle have not yet been effectively established. It is usually not effective to get information and decision support for disassembly planning from the participants (Xia et al., 2015). In this research, we propose a service-oriented framework to support business integration for disassembly planning. Effective and optimized disassembly planning can be done via the Internet by integrating the related distributed services provided by the participants.

E-business is the term used to describe the administration of conducting business via the Internet. Service-oriented technologies are the main supporting technologies for e-business engineering, and there is high interest in their further development (Chao, 2016). Huang and Chung (2003) proposed a service-based framework that adopted a web services-based

approach to developing business integration solutions for e-business application. Rehman et al. (2015) presented a decision-making approach that assisted a cloud service user in selecting a cloud service provider based on the quality of its services. Johnson (2008) proposed a framework for pricing government e-services to help governments to expand online service delivery quickly and broadly, and increase the net benefits to stakeholders.

Enterprises are moving toward service-oriented architecture with web services to modernize their legacy applications by using wrapping techniques (Baghdadi and Al-Bulushi, 2015). Baresi et al. (2016) proposed a solution to integrate existing registries, along with a match-making approach to ease the publication and retrieval of services. The service-oriented disassembly sequence planning framework that we describe is a novel e-business application for the end-of-life treatment of e-waste. The e-service concept that is introduced for the disassembly context will transform the business model of the disassembly industry.

2. PROBLEM FORMULATION

Different considerations have led stakeholders to pursue different objectives for disassembly planning. For instance, according to the Waste Electrical and Electronic Equipment Directive Directive, e-waste regulators need to check whether an end-of-life treatment operator is able to recover (reuse and recycle) at least 75% of the weight, and remove all of the hazardous materials. The components containing hazardous materials need to be removed from the equipment for further processing. Apart from fulfilling these fundamental environmental objectives, remanufacturers want to improve the economic efficiency by prioritizing the valued components during disassembly. We develop a multi-objective disassembly sequence planning model according to these considerations by introducing three disassembly indices as shown in Figure 1. The multi-objective disassembly sequence planning model involves different e-waste stakeholders distributed across different areas. It is a complex business process that needs to be optimized, and the enabling technologies to support e-business engineering need to be implemented to handle it through the Internet.

- Decision variable

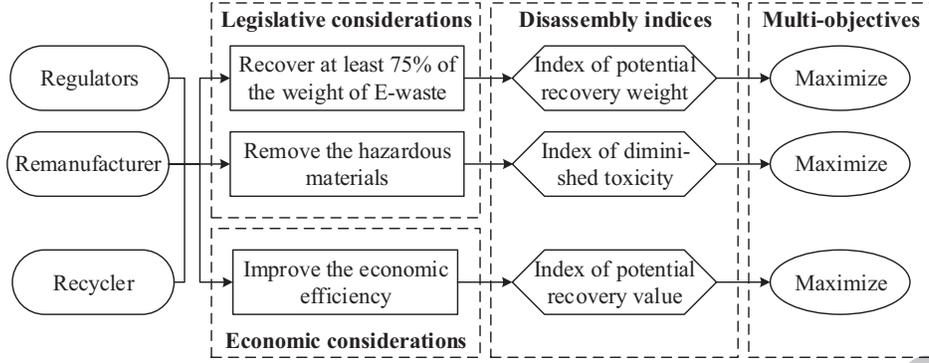


Figure 1: Multi-objective disassembly sequence planning model.

The decision variable $X = (x_1, x_2, \dots, x_m)$ is a disassembly sequence, representing the order in which each disassembly operation is undertaken. Each disassembly operation x_i in X has a corresponding sequence number i . And m is the total number of disassembly operations.

- Objective functions

Each component c has three important properties: toxicity level $h(c)$, potential recovery value $v(c)$ and potential recovery weight $w(c)$. The indices for an X are computed according to these properties of each component and the sequence number of each disassembly operation.

Index of diminished toxicity $f_h(X)$. Some of the components in e-waste are hazardous. The level of their toxicity can be represented by using a qualitative method. For example, we will use the number 5, 3, 1, and 0 to represent high, medium, low and no toxicity, respectively. This index can be computed as follows:

$$f_h(X) = \sum_{i=1}^m \left(\sum_{c \in C_i} h(c) \right) / i, \quad (1)$$

where C_i denotes the set of components disassembled by operation x_i . Higher $f_h(X)$ is achieved when the components with higher toxicity level are disassembled earlier.

Index of potential recovery value $f_v(X)$. The potential recovery value means the value of reused components or the materials recycled from components. The reusable com-

ponent can be all recovered while the recyclable component can be recovered at a rate depending on the material types and recycling conditions. This index can be computed as follows:

$$f_v(X) = \sum_{i=1}^m \left(\sum_{c \in C_i} v(c) \right) / i, \quad (2)$$

Higher $f_v(X)$ is achieved when the components with greater potential recovery value are disassembled earlier.

Index of potential recovery weight $f_w(X)$. The potential recovery value means the value of reused components or the materials recycled form components. The reusable component can be all recovered while the recyclable component can be recovered at a rate depending on the material types and recycling conditions. This index can be computed as follows:

$$f_w(X) = \sum_{i=1}^m \left(\sum_{c \in C_i} w(c) \right) / i, \quad (3)$$

Higher $f_w(X)$ is achieved when the components with heavier potential recovery weight are disassembled earlier.

The objective functions of the multi-objective disassembly sequence planning problem can be set as follows:

$$\text{maximize: } F(X) = (f_h(X), f_v(X), f_w(X)). \quad (4)$$

- Disassembly precedence constraints

A feasible solution for the problem is a disassembly sequence that is subject to disassembly precedence constraints, which come from the geometric and connecting constraints among the components. The disassembly precedence constraints are represented by a disassembly precedence graph. It can be translated to disassembly precedence rules, which are described by pairs of precedent sets and following sets in the form of $Rule_j : Pre_j \rightarrow Fol_j$, where the symbol \rightarrow means the components in set Pre_j have disassembly precedence over the components in set Fol_j . Figure 2 gives an example to illustrate the disassembly precedence

graph and disassembly precedence rules representing the disassembly precedence constraints.

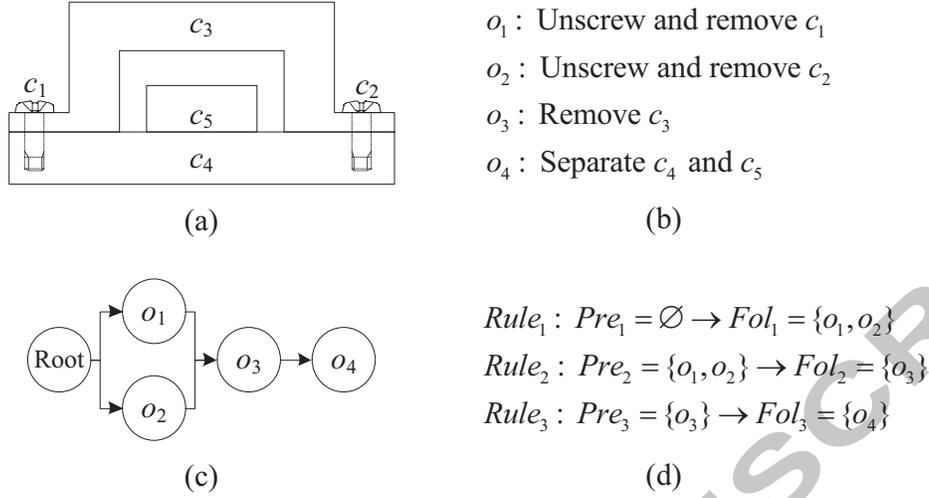


Figure 2: An example of disassembly precedence constraints. (a) 2D product model, (b) list of operations, (c) disassembly preference graph, and (d) disassembly preference rules translated from the disassembly preference graph.

3. MTLBO ALGORITHM: ADDRESSING DISASSEMBLY SEQUENCE PLANNING

This section starts with a brief review on the *teaching-learning-based optimization* (TLBO) algorithm and multi-objective optimization theory, and then presents the proposed MTLBO algorithm in detail and compares it with the TLBO algorithm.

3.1. The TLBO algorithm

The TLBO algorithm was first proposed by Rao et al. (2011). It is a population-based evolutionary algorithm that inspired the teaching-learning process. It includes two fundamental modes of learning: learning from a teacher (teaching phase) and learning from interacting with other learners (learning phase).

Suppose there are m learning subjects offered to n learners. In each generation G , the grade that learner i gets in subject j is denoted as $x_{i,j}^G$. The vector $X_i^G = (x_{i,1}^G, x_{i,2}^G, \dots, x_{i,m}^G)$ formed by all the grades of learner i is the decision vector of the optimization problem. And the objective function value $f(X_i^G)$ is the result of learner i considering all the subject

grades. The learner with the best result is considered as the teacher and presented as $X_{best}^G = (x_{best,1}^G, x_{best,2}^G, \dots, x_{best,m}^G)$. The mean grade of all the learners in subject j is denoted as a_j^G . The new vector generated from X_i^G is denoted as $V_i^G = (v_{i,1}^G, v_{i,2}^G, \dots, v_{i,m}^G)$.

In the teaching phase, a learners result may be improved by learning from the teacher. In the learning phase, a learners result may be improved by learning from another learner. The mutation and selection strategies implemented in the TLBO algorithm are listed below.

- Mutation in teaching phase:

$$v_{i,j}^G = x_{i,j}^G + r(x_{best,j}^G - F \cdot a_j^G), \quad (5)$$

where r is a random number in the range $[0, 1]$ and $F = \text{round}(1 + r)$, which is computed as either 1 or 2 decided by r .

- Selection in teaching phase:

$$X_i^G = \begin{cases} V_i^G, & \text{if } f(V_i^G) \text{ is better than } f(X_i^G), \\ X_i^G, & \text{otherwise.} \end{cases} \quad (6)$$

- Mutation in learning phase:

$$v_{i,j}^G = \begin{cases} x_{i,j}^G + r(x_{i,j}^G - x_{p,j}^G), & \text{if } f(X_i^G) \text{ is better than } f(X_p^G), \\ x_{i,j}^G + r(x_{p,j}^G - x_{i,j}^G), & \text{otherwise,} \end{cases} \quad (7)$$

where p is randomly chosen from the population and $p \neq i$.

- Selection in learning phase:

$$X_i^{G+1} = \begin{cases} V_i^G, & \text{if } f(V_i^G) \text{ is better than } f(X_i^G), \\ X_i^G, & \text{otherwise.} \end{cases} \quad (8)$$

TLBO algorithm has been successfully applied to solve many optimization problems (Rao and Patel, 2013; Pawar and Rao, 2012; Rao and Kalyankar, 2013). Its merits include that

it does not require any parameters to be tuned, except population size and iteration times. It outperforms some other well-known meta-heuristics also.

Directly applying the evolution mechanism from the TLBO algorithm is not suitable to solve the disassembly sequence planning problem, however. In this problem, the decision space grows exponentially in the number of disassembly operations, while the disassembly precedence constraints are complex. These characteristics will cause there to be few feasible solutions for a population if a random solution generation method is embedded in TLBO algorithm. Furthermore, the disassembly precedence constraints cannot be honored when using an arithmetic operation method during the evolutionary iterations needed to achieve an optimal solution. This will lead to only a few feasible solutions in the offspring. Hence, the evolutionary mechanism of TLBO algorithm should be modified to suit the disassembly sequence planning better.

3.2. Multi-objective optimization

A multi-objective optimization problem with k objective functions can be defined as:

$$\begin{aligned} & \text{maximize: } Y = F(X) = (f_1(X), f_2(X), \dots, f_k(X)), \\ & \text{subject to: } x \in \Omega, \end{aligned} \tag{9}$$

where X is the decision vector, Ω is the decision space, and Y is the objective vector.

The objectives in a multi-objective optimization problem are contradictory, namely, when one of the objective functions is improved, the others will be affected and get worse. Hence, it is impossible to achieve the best results for all of the objective functions. The best trade-offs among the multiple objectives can be defined in terms of Pareto optimality.

Let $U = (u_1, u_2, \dots, u_k)$ and $V = (v_1, v_2, \dots, v_k)$. U is said to dominate V , denoted as $U \succ V$, if and only if:

$$\forall i \in (1, 2, \dots, k), u_i \geq v_i \text{ and } \exists i \in (1, 2, \dots, k), u_i > v_i. \tag{10}$$

Let $P \in \Omega$. P is said to be Pareto optimal if:

$$\neg \exists Q \in \Omega, F(Q) \succ F(P). \quad (11)$$

The set of all the Pareto optimal solutions is called the *Pareto-optimal set* (S^*) and the set of all the Pareto optimal objective vectors is called the *Pareto-optimal frontier* (Fr^*) stated as follows:

$$\begin{aligned} S^* &= \{P \in \Omega | \neg \exists Q \in \Omega, F(Q) \succ F(P)\}, \\ Fr^* &= \{F(P) | P \in S^*\}. \end{aligned} \quad (12)$$

NSGA II is an effective multi-optimization optimization algorithm (Deb et al. 2002). The selection mechanism of NSGA II is based on a fast non-dominated sorting approach and a crowding distance sorting approach, which guard the selection process against a uniformly spread Pareto-optimal frontier. These two sorting approaches are implemented in this research.

In a population, the set of all the solutions, whose objective vectors are mutual non-dominated, is called a *non-dominated frontier* in NSGA II. A *population* (Pop) can be separated into different non-dominated frontiers (Fr_1, Fr_2, \dots):

$$\begin{aligned} Pop &= (Fr_1, Fr_2, \dots), \\ \text{where } \forall P \in Fr_{i-1} \text{ and } \forall Q \in Fr_i, F(P) \succ F(Q), (i = 2, 3, \dots). \end{aligned} \quad (13)$$

In a population, the *non-dominated rank* (P_{rank}) of a solution P is defined as:

$$P_{rank} = i, \text{ if } P \in Fr_i, i \in \{1, 2, \dots\}. \quad (14)$$

The *crowding distance* ($P_{distance}$) of P is calculated as the sum of its individual distance values ($P_{distance,i}$) corresponding to each objective i :

$$P_{distance} = \sum_{i=1}^k P_{distance,i}. \quad (15)$$

The computation of $P_{distance,i}$ requires sorting the population according to objective i (f_i) in the descending order. Thereafter, if $f_i(P)$ is the smallest or largest, $P_{distance,i}$ is assigned an infinite value. Otherwise, it is assigned a value equals to the absolute normalized difference in values of f_i of the two adjacent solutions.

3.3. Proposed MTLBO algorithm

An MTLBO algorithm is proposed to solve the multi-objective disassembly sequence planning problem. To make it applicable for the problem with complex disassembly precedence constraints, a *feasible solution generator* was designed to generate feasible solutions, and *teaching phase and learning phase operators* were designed to improve the solutions by applying the *precedence preservation cross-over operation method*. To sort the population for selection, a *fast non-dominated sorter* and a *crowding distance sorter* were designed according to NSGA II.

In the authors previous work, a *simplified teaching-learning-based optimization* (STLBO) algorithm was proposed to solve the single-objective disassembly sequence planning problem (Xia et al., 2014). The MTLBO algorithm was developed based on the STLBO algorithm. It is based on Pareto-optimality and has two parts beyond the STLBO algorithm. They are the fast non-dominated sorter and the crowding distance sorter. Both were modified from those in the STLBO algorithm.

The MTLBO algorithm is illustrated in Figure 3. In the selection phase, parent population P_G and offspring population O_G in G th generation are combined together to form a combined population R_G . R_G is sorted into non-dominated frontiers (Fr_1, Fr_2, \dots, Fr_i) using feasible solution generator. If $|Fr_1| + |Fr_2| + \dots + |Fr_i| = N$ (population size), all the solutions in the top i frontiers are selected for the next generation population P_{G+1} . If $|Fr_1| + |Fr_2| + \dots + |Fr_i| < N$ and $|Fr_1| + |Fr_2| + \dots + |Fr_{i+1}| > N$, all the solutions in the top i frontiers are selected for P_{G+1} , and Fr_{i+1} should be sorted using crowding distance sorter, after which, the top best solutions in Fr_{i+1} are selected to fill the vacant positions of P_{G+1} . In the evolution phase, transitional population Q_{G+1} is generated from P_{G+1} using teaching phase operator, and then O_{G+1} is generated from Q_{G+1} using learning phase

operator.

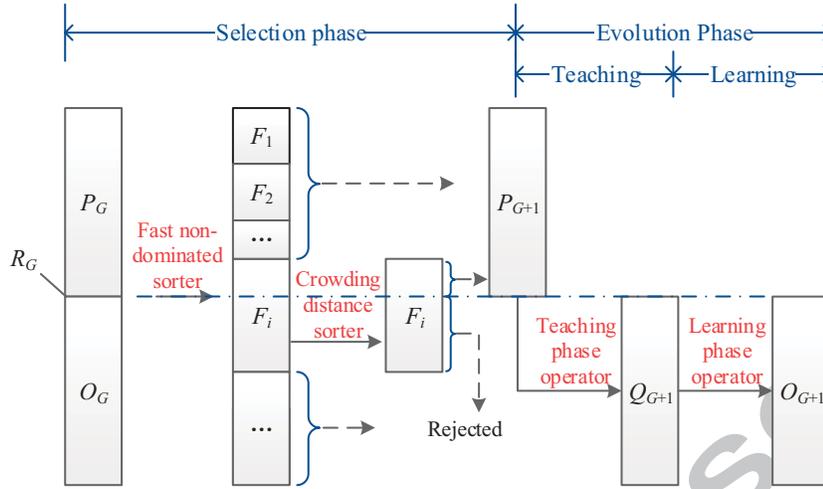


Figure 3: The MTLBO algorithm.

The multi-objective disassembly sequence planning problem is combinatorial. The decision variable is represented and coded as a permutation. For instance, the permutation [2 1 4 3] states that the disassembly operation o_2 is considered first, o_1 second, o_4 third, and o_3 last.

- Feasible solution generator

Feasible solution generator is used to generate feasible solutions randomly according to the disassembly precedence constraints. The algorithm first checks the current disassembly state and precedence constraints to find the available disassembly operations. Second, it randomly chooses an operation from them, pushes it back to the solution and deletes it from the available disassembly operations. The algorithm uses these two steps continue iteratively until the disassembly operations are all pushed back to the solution. The pseudo-code of the feasible solution generator is outlined as follows, where X denotes a feasible solution, A is the set of available disassembly operations and n is the number of disassembly operation.

Pseudo-code 1: feasible solution generator

$X = \emptyset; A = \emptyset;$

For each $Rule_r$ do $flag_r = 0;$ End for

For $i = 1$ to n do

For each $Rule_r$ do
 if ($flag_r = 0 \& \& Pre_r \subseteq X$) then $A = A \cup Fol_r$; $flag_r = 1$;
 End for
 Randomly choose an element e from A ; push e back to X ; delete e from A ;
 End for

- Fast non-dominated sorter

Fast non-dominated sorter was developed according to the fast non-dominated sorting approach, which is used to sort a population (P) into different non-dominated frontiers (Fr_1, Fr_2, \dots). The pseudo-code of fast non-dominated sorter is outlined as follows, where S_X denotes the set of solutions whose objective vectors dominated by that of solution X , n_X denotes the number of solutions whose objective vectors dominate those of solution X , and Fr_{next} denotes the next non-dominated frontier.

Pseudo-code 2: fast non-dominated sorter

For each X in the population do
 $S_X = \emptyset$; $n_X = 0$;
 For each P in the population do
 If ($f(X) \succ f(P)$) then $S_X = S_X \cup P$; else if ($f(P) \succ f(X)$) then $n_X = n_X + 1$;
 End for
 If ($n_X = 0$) then $X_{rank} = 1$; $Fr_1 = Fr_1 \cup X$;
 End for
 $i = 1$;
 While $Fr_i \neq \emptyset$ do
 Let $Fr_{next} = \emptyset$;
 For each X in Fr_i do
 For each P in S_X do
 $n_P = n_P - 1$; If ($n_P = 0$) then $P_{rank} = i + 1$; $Fr_{next} = Fr_{next} \cup P$;
 End for
 End for
 $i = i + 1$; $Fr_i = Fr_{next}$;
 End while

- Crowding distance sorter

Crowding distance sorter was developed according to the crowding distance sorting approach, which is used to sort the solutions with the same non-dominated rank according to the crowding distance in descending order. The pseudo-code for the crowding distance sorter is outlined below, where l denotes the number of solutions in Fr_i , $Fr_i[j]$ denotes j th solution in Fr_i , and $Fr_i[j]_{distance}$ denotes the crowding distance of $Fr_i[j]$.

Pseudo-code 3: crowding distance sorter

$l = |Fr_i|;$

For $j = 1$ to l do $Fr_i[j]_{distance} = 0$; End for

For each objective f_k do

Sort Fr_i according to f_k in descending order;

$Fr_i[1]_{distance} = Fr_i[l]_{distance} = \infty$;

$d = f_k(Fr_i[1]) - f_k(Fr_i[l])$;

For $j = 2$ to $(l - 1)$ do

$Fr_i[j]_{distance} = Fr_i[j]_{distance} + (f_k(Fr_i[j - 1]) - f_k(Fr_i[j + 1]))/d$;

End for

End for

Sort Fr_i according to $Fr_i[j]_{distance}$ in descending order.

- Teaching phase operator

Teaching phase operator is used to generate the transitional population modified from the parent population. The solution which is randomly chosen from the first non-dominated frontier (Fr_1) of P_G is considered as the teacher. The teacher tries to improve the results of learners by giving his or her experiences and knowledge (better disassembly sequence) to the learners with expectation that their results will reach his or her level. A self-adaptive parameter, the *teaching factor* p_t is computed for modifying the existing learners. It is calculated by the *non-dominated rank* r of the modified learner according to the following equation:

$$p_t = \frac{r + 1}{r + 3} \quad (16)$$

Learners with a larger non-dominated rank will have a higher probability of learning from the teacher.

Each solution X_Q in Q_G is generated according to the corresponding learner X and a teacher X_T . A precedence preservative operator is applied to maintain the precedence relationships in the feasible solutions when they are being modified. The pseudo-code of teaching phase operator is outlined as follows, with m defined as the number of disassembly operations (length of a solution).

Pseudo-code 4: teaching phase operator

```

 $Q_G = \emptyset;$ 
For each  $X$  in  $P_G$  do
   $X_Q = \emptyset;$ 
  Randomly choose a teacher  $X_T$  from  $Fr_1$ ; Calculate  $p_t$ ;
  For  $i = 1$  to  $m$  do
    Set  $p \in [0, 1]$  randomly;
    If ( $p < p_t$ ) then  $X_Q[i] = X_T[1]$ ; else  $X_Q[i] = X[1]$ ;
    Erase  $X_Q[i]$  in  $X_T$  and  $X$ ;
  End for
   $Q_G = Q_G \cup X_Q;$ 
End for

```

The teaching-phase operator procedure generates a new solution by choosing and setting the elements one by one from the left to the right. First, the probability is set randomly. Second, if the probability is less than p_t , the leftmost element of the teacher will be chosen and pushed back to the new solution; otherwise the left-most element of the learner will be used. Third, the chosen element will be erased for both the teacher and the learner. Then, these three steps are repeated until the new solution contains all the disassembly operations. In this way, the disassembly precedence in the solutions is preserved. For instance, Table 1 gives an example with 5 elements: a new solution ([2 3 4 5 1]) is generated from a teacher ([2 3 1 4 5]) and a learner ([4 3 5 1 2]). Suppose that Operation 3 has precedence over Operation 5. Then, the precedence will be preserved in the new solution when our approach is used.

- Learning phase operator

Learning phase operator is used to generate the offspring population modified from the transitional population. All the solutions in Q_G are considered as the learners, who learn

Table 1: An example of using teaching phase operator to generate a new solution from a teacher and a learner with 5 elements.

Step	Probability	Teacher	Learner	Solution
1	0.41	[<u>2</u> 3 1 4 5]	[4 3 5 1 2]	[2]
2	0.23	[<u>3</u> 1 4 5]	[4 3 5 1]	[2 3]
3	0.86	[1 4 5]	[<u>4</u> 5 1]	[2 3 4]
4	0.72	[1 5]	[<u>5</u> 1]	[2 3 4 5]
5	0.33	[<u>1</u>]	[1]	[2 3 4 5 1]

Notes: the underlined figure indicates the chosen element in each step.

from each other and try to improve their results. Each solution X_O in O_G is generated according to the corresponding learner (X), another randomly chosen learner (X_A) from Q_G , and a new randomly generated learner X_R . The pseudo-code of learning phase operator is shown next.

Pseudo-code 5: learning phase operator

$O_G = \emptyset$;

For each X in Q_G do

$X_O = \emptyset$;

 Randomly choose $X_A \neq X$ from Q_G ;

 Randomly generate X_R using feasible solution generator;

 For $i = 1$ to m do

 Set $p \in [0, 1]$ randomly;

 If ($p < 0.33$) then $X_O[i] = X[1]$;

 Else if ($p < 0.67$) then $X_O[i] = X_A[1]$;

 Else $X_O[i] = X_R[1]$;

 Erase $X_O[i]$ in X , X_A and X_R ;

 End for

$O_G = O_G \cup X_O$;

End for

3.4. Comparison of MTLBO algorithm with TLBO algorithm

The proposed MTLBO algorithm is different from the TLBO algorithm. First, the generation methods for the initial population are different. In the MTLBO algorithm, the initial population is generated using the feasible solution generator, hence each individual

is a feasible solution, while in the TLBO algorithm, the initial population is generated randomly without considering the individuals feasibility. Second, the realization methods for the evolutionary mechanism are different. In the MTLBO algorithm, learners are modified using both the teaching and learning phase operators, applying the precedence-preserving cross-over operation. So the disassembly precedence constraints will be retained in the modified solutions. In contrast, with the TLBO algorithm, the learners are modified using multidimensional matrix computation. Third, the fast non-dominated and crowding distance sorting approach are implemented in MTLBO, making it suitable for the multi-objective disassembly problem, while TLBO is used to solve single-objective optimization problems.

4. NUMERICAL SIMULATIONS

We conducted simulations to test the performance of the MTLBO algorithm to solve the multi-objective disassembly sequence planning problem. All tested algorithms were coded with C++, and carried out on a PC with a 2 GHz Intel Core2 Duo CPU T5750 and 2 gigabytes of memory.

4.1. Case 1

In Case 1, 10 disassembly operations (o_1, o_2, \dots, o_{10}) are needed to disassemble an e-waste to remove the hazardous or valuable components. The properties of these components (c_1, c_2, \dots, c_8) are listed in Table 2. The disassembly precedence constraints are represented by disassembly preference rules:

$$Rule_1 : Pre_1 = \emptyset \rightarrow Fol_1 = \{o_1, o_2\},$$

$$Rule_2 : Pre_2 = \{o_1, o_2\} \rightarrow Fol_2 = \{o_3, o_6, o_8, o_{10}\},$$

$$Rule_3 : Pre_3 = \{o_3\} \rightarrow Fol_3 = \{o_4, o_5, o_7\},$$

$$Rule_4 : Pre_4 = \{o_5\} \rightarrow Fol_4 = \{o_9\}.$$

To demonstrate the effectiveness and robustness of the MTLBO algorithm, four tests were implemented:

Table 2: Properties of the components in Case 1.

Component	Operation	h	v	w
c_1	o_1	0	5	2
c_2	o_2	1	0	5
c_3	o_4	3	2	4
c_4	o_6	0	4	1
c_5	o_7	0	5	3
c_6	o_8	5	0	1
c_7	o_9	0	5	1
c_8	o_{10}	1	0	5

Notes: h indicates the toxicity level, v denotes the potential recovery value and w is a parameter for the potential recovery weight.

Test 1: maximize $F(X) = (f_h(X), f_v(X), f_w(X))$ using MTLBO, $N = 100$, $G_{max} = 500$;

Test 2: maximize $F(X) = (f_h(X), f_v(X))$ using MTLBO, $N = 100$, $G_{max} = 500$;

Test 3: maximize $F(X) = (f_h(X), f_w(X))$ using MTLBO, $N = 100$, $G_{max} = 500$;

Test 4: maximize $F(X) = (f_v(X), f_w(X))$ using MTLBO, $N = 100$, $G_{max} = 500$.

In Test 1, all the three mentioned objectives are chosen. In Tests 2-4, two of them were chosen for each. The aim of solving a multi-objective disassembly sequence planning problem is to find a set of non-dominated solutions for the decision-makers to select from. Now, see Figure 4.

For Test 1, the acquired non-dominated solutions can be plotted in three dimensions with the axes denoting the three objective function values, as illustrated in Figure 4(a). For Tests 2-4, the acquired non-dominated solutions can be similarly plotted with two axes, as illustrated in Figure 4(b)-4(d).

In Figure 4, each point marked by a circle represents a non-dominated solution. The relationships between the different objectives are shown in Figure 4(a). However this three-dimensional plot is not very intuitive. Figure 4(b), 4(c) and 4(d) give a clearer picture of the relationships between each two of the objectives. The three objectives contradict one another. Choosing a solution from among the acquired non-dominated solution set depends

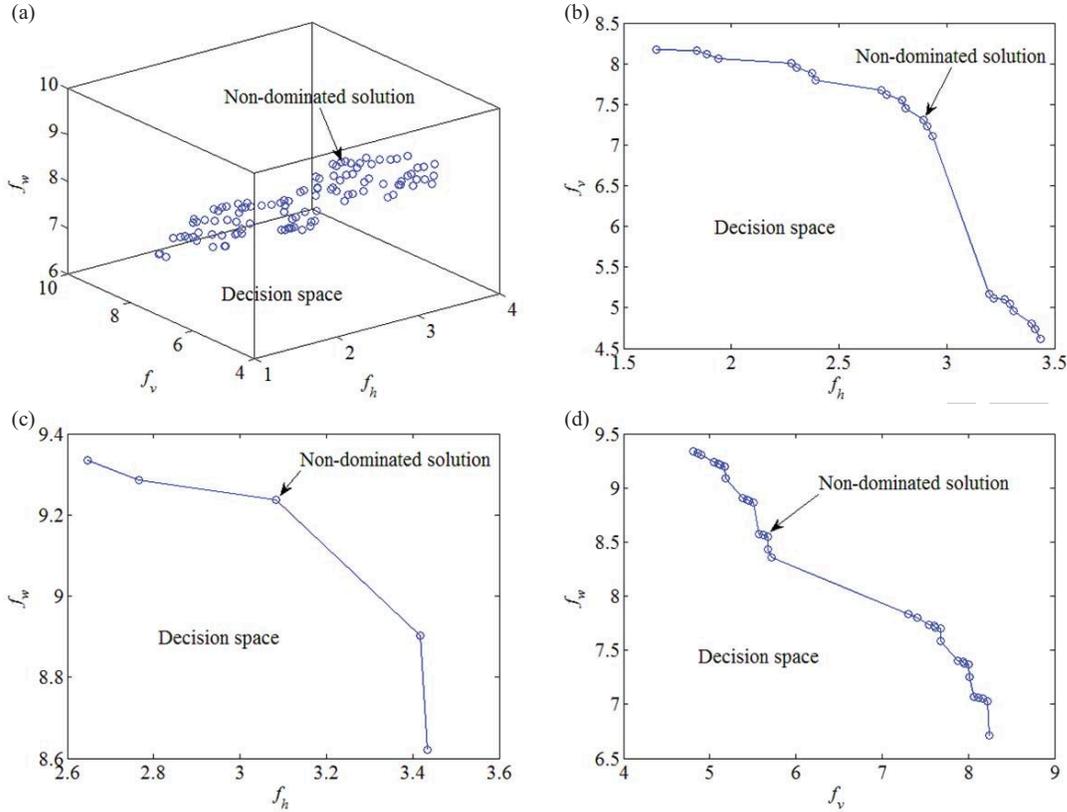


Figure 4: Non-dominated solutions of (a) Test 1, (b) Test 2, (c) Test 3 and (d) Test 4.

on the decision-makers preferences for the objectives. Therefore, the optimization results can help in supporting decisions.

The above results indicate that the proposed MTLBO algorithm is effective and maintains a good spread of non-dominated solutions.

To compare the performance of MTLBO algorithm with that of NSGA II, we conducted an additional test to assess it relative to Test 2, as follows:

Test 5: maximize $F(X) = (f_h(X), f_v(X))$ using NSGA II, $N = 100$, $G_{max} = 500$.

Tests 2 and 5 were run 30 times independently. The MTLBO algorithm found 31.17 non-dominated solutions on average with a standard deviation of 0.75, while NSGA II found only 27.10 on average with a standard deviation of 2.14. One of the test results is illustrated in Figure 5. It shows that the MTLBO algorithm found a better spread of non-dominated solutions and had better convergence to the Pareto-optimal solutions than NSGA-II did.

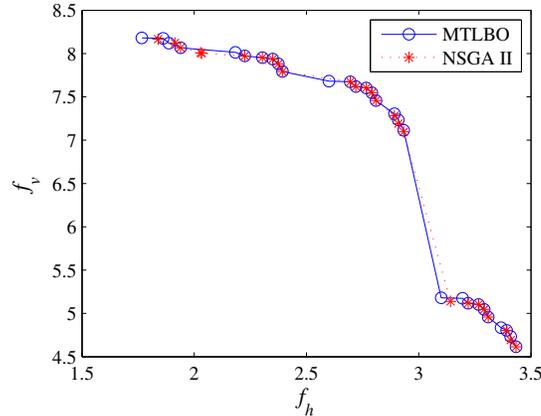


Figure 5: Non-dominated solutions of Test 2 and Test 5.

Two additional tests were also conducted to assess the comparative effects of different settings for G_{max} on the performance of the MTLBO algorithm:

Test 6: maximize $F(X) = (f_h(X), f_v(X))$ using MTLBO, $N = 100$, $G_{max} = 100$;

Test 7: maximize $F(X) = (f_h(X), f_v(X))$ using MTLBO, $N = 100$, $G_{max} = 1000$.

The results of Test 2, 6 and 7 are illustrated in Figure 6, showing that the performance is improved when G_{max} is increased. We found 27 non-dominated solutions in Test 6, 31 in Test 2, and 33 in Test 7. The convergence to Pareto-optimal solutions improved as G_{max} increased from 100 to 500 to 1000. Test 7 had a better spread of solutions than Tests 2 and 6 did. Thus, the reader should recognize that it seems like a larger G_{max} values support better convergence and solution spread.

4.2. Case 2

Case 2 studies an LCD TV. Figure 7 gives its exploded view and the top-level structure of its bill of materials. In an LCD TV, there are some valuable materials, such as plastics, iron (Fe), copper (Cu) and aluminum (Al). There are also some hazardous materials, such as: tin (Sn), lead (Pb) and cadmium (Cd) (in the printed circuit boards); mercury (Hg) and indium (In) (in the LCD screen); polychlorinated biphenyls (PCB) (in the capacitors), and poly-brominated diphenyl ethers (PBDE) contained in the flame retardants used in plastics.

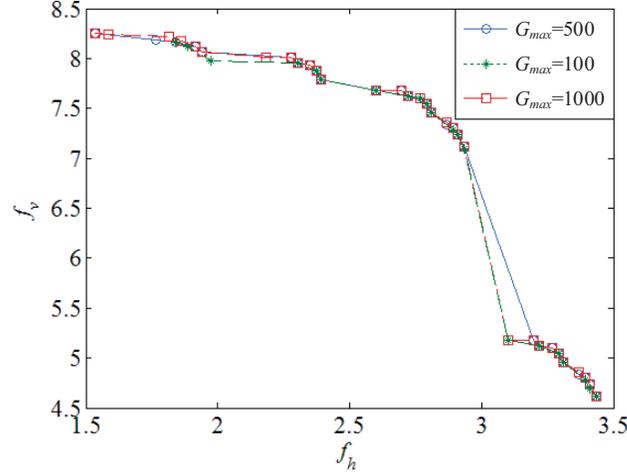


Figure 6: Non-dominated solutions of Test 2, Test 6 and Test 7.

Descriptions of all the disassembly operations and the properties of the disassembled components are given in Table 3. The disassembly precedence constraints are represented by disassembly preference rules:

$$Rule_1 : Pre_1 = \emptyset \rightarrow Fol_1 = \{o_1, o_2\},$$

$$Rule_2 : Pre_2 = \{o_1\} \rightarrow Fol_2 = \{o_3, o_{15}\},$$

$$Rule_3 : Pre_3 = \{o_3\} \rightarrow Fol_3 = \{o_4, o_5, o_6, o_7, o_8, o_{10}\},$$

$$Rule_4 : Pre_4 = \{o_8\} \rightarrow Fol_4 = \{o_9\},$$

$$Rule_5 : Pre_5 = \{o_5, o_7, o_9, o_{10}\} \rightarrow Fol_5 = \{o_{11}\},$$

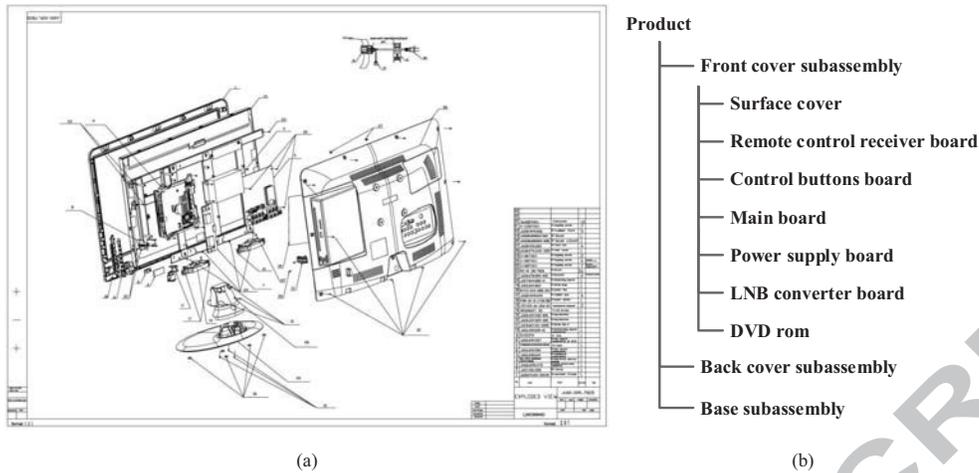
$$Rule_6 : Pre_6 = \{o_{11}\} \rightarrow Fol_6 = \{o_{12}, o_{13}\},$$

$$Rule_7 : Pre_7 = \{o_{13}\} \rightarrow Fol_7 = \{o_{14}\},$$

$$Rule_8 : Pre_8 = \{o_{15}\} \rightarrow Fol_8 = \{o_{16}, o_{17}\}.$$

The MTLBO algorithm was implemented for this case. The objective functions were set up to: maximize $F(X) = (f_h(X), f_v(X), f_w(X))$. Also, the parameters were set as: $N = 100$ and $G_{max} = 500$. The non-dominated solutions that were obtained are shown in Table 4.

A decision can be made based on the Pareto-optimal solutions that we found, according to the decision makers preferences related to the multiple objective functions. For example, if a decision maker gives weight of 0.5, 0.3 and 0.2 to f_h , f_v and f_w respectively, the 3rd



Notes: LNB converter board is a low-noise block converter printed circuit board. It is often used in screen cable TV and broadband Internet technology hardware to support improved signal capture and delivery to enhance the quality of a user's experience with whatever content is being delivery.

Figure 7: Production info, LCD TV: (a) exploded view; (b) top-level bill of materials structure.

solution will be viewed as the best. This is because the value of its objective function, $(0.5f_h + 0.3f_v + 0.2f_w = 120.16)$, is the biggest.

In summary, our proposed MTLBO algorithm seems to have worked quite well, and achieved better convergence and non-dominated solution spread than NSGA II did in solving the multi-objective disassembly sequence planning problem.

5. SERVICE-ORIENTED DISASSEMBLY SEQUENCE PLANNING FRAMEWORK

Effective disassembly sequence planning methods can improve disassembly efficiency, reduce disassembly cost and environmental impact, and improve the recovery value of e-waste. To make disassembly planning more effective, it is necessary to obtain detailed information on e-waste at the outset of the analysis work. However, it is likely to not be effective to get this kind of information support through current e-waste-related information-sharing mechanisms. Also, the disassembly model should be built in a way that ensures the representation of the disassembly process constraints. And finally, there needs to be an optimization algorithm that performs effectively when it is used to solve the disassembly sequence planning problem.

Table 3: Disassembly operations and the properties of the disassembled components.

#	Disassembly operation	h	$v/\text{Chinese Yuan}$	w/g
O_1	Unscrew and remove base subassembly			
O_2	Unscrew and remove cover plate	1	0.09	23.0
O_3	Remove back cover subassembly			
O_4	Disassemble back cover subassembly	1	2.87	723.8
O_5	Remove wire with pin			
O_6	Remove control button part	3	0.02	5.5
O_7	Unscrew and remove main board	5	0.79	196.0
O_8	Unscrew and remove loudspeaker part			
O_9	Unscrew and remove power supply board	5	0.65	118.0
O_{10}	Unscrew and remove metal support	0	0.59	183.0
O_{11}	Unscrew			
O_{12}	Remove Remote control receiver board	3	0.01	3.0
O_{13}	Remove metal mounting plate	0	2.09	639.0
O_{14}	Separate surface frame and LCD screen	6	11.51	3170.8
O_{15}	Disassemble base subassembly	0	0.11	35.0
O_{16}	Disassemble brace part	2	0.28	75.0
O_{17}	Disassemble seat part	0	0.16	50.0

Notes: h indicates the toxicity level, v denotes the potential recovery value and w is a parameter for the potential recovery weight.

Service-oriented technologies promise a way to create the basis for agility so that companies can deliver flexible business processes (Demirkan et al., 2008). They are mission-critical supporting technologies for e-business applications. The disassembly and recovery of e-waste involve complex business processes across the lifecycle of their production and use. So information support services, disassembly modeling, and optimization services need to be integrated together when they are used in computer networks, such as the Internet. We proposed a service-oriented disassembly sequence planning framework to provide a comprehensive and standardized service-based environment for distributed information sharing between the producer and the disassembler. Our framework also provides a service-oriented environment for disassembly modeling and optimization.

Service-oriented disassembly sequence planning offers a useful extended e-business model for the end of the service life involving the kinds of electronic and electrical products that we have studied in this research. We introduced the use of cloud services for the disassembly

Table 4: Non-dominated solutions for Case 2 supporting multi-objective decision-making.

Solution #	f_h	f_v	f_w
1	3.68	2.06	547.21
2	3.81	2.04	541.00
3	2.99	2.23	589.96
4	3.82	2.04	541.60
5	4.43	1.69	450.53
6	3.29	2.20	582.60
7	4.21	1.89	500.53
8	3.39	2.17	572.96
9	4.04	2.00	528.24
10	3.85	2.02	535.57
11	4.35	1.72	454.28
12	3.95	2.02	533.22
13	3.22	2.21	583.99
14	4.10	1.90	503.46
15	3.46	2.17	572.58
16	4.32	1.81	479.96
17	3.34	2.17	572.22
18	3.65	2.10	557.90
19	3.95	2.00	528.50
20	4.05	1.91	507.88

Notes: f_h is the index of diminished toxicity, f_v is the index of potential recovery value and f_w is the index of potential recovery weight.

context, and why it will fundamentally transform the business model that is in use today in the disassembly industry. The hallmarks of our proposed approach involve how the characteristics of customization and distribution can be leveraged for better performance. Our framework also links the relevant stakeholders so they can do more effective disassembly sequence planning together by collaborating via the Internet.

Figure 8 illustrates our service-oriented model. Different stakeholders can access services through the service-oriented platform via the Internet. They can be those who request or provide related services. We considered three different kinds of services: information support, disassembly modelling, and optimization. When a disassembler receives a request to disassemble e-waste, the relevant services can be invoked to create a disassembly sequence

plan.

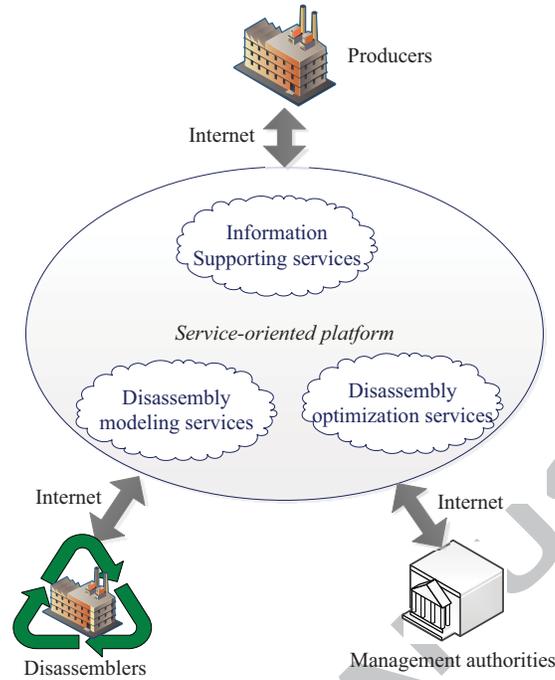


Figure 8: Service-oriented disassembly sequence planning model.

Our three-layer service-oriented disassembly sequence planning framework is shown in Figure 9. The resources layer considers distributed information that is provided by different stakeholders, and the software and algorithms used for disassembly modeling and optimization. The resources are provided as cloud services in the core service layer. The service management layer is used for the description and publication of cloud services, as well as their registration and match-making for their use in the marketplace.

In the three-layer framework, adaptive, secure and on-demand cloud services for disassembly sequence planning are offered over the Internet. The framework provides a service-oriented *request-find-provide business model* for disassembly planning. For instance, for e-waste such as we have discussed, the disassembler will need to request help from information support services. When they are found, the appropriate information providers will provide the services. After that, the disassembler will need to request additional disassembly modelling and optimization services to try to optimize disassembly sequence planning. Again, the relevant services will be invoked when they are identified.

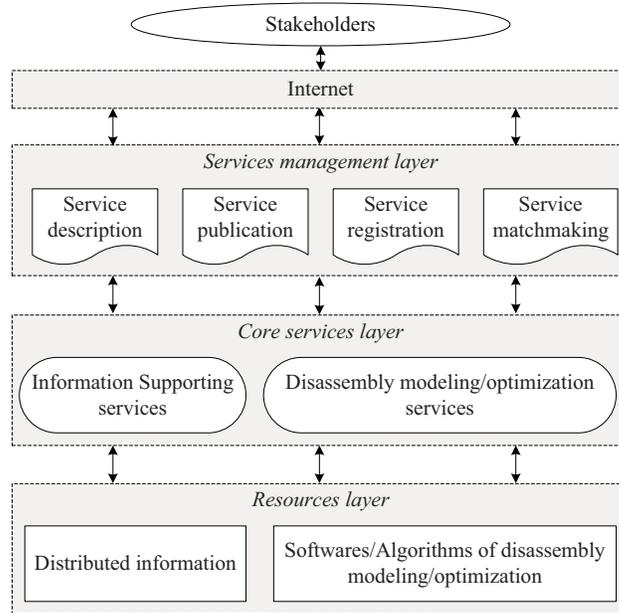


Figure 9: Service-oriented disassembly sequence planning framework.

So the reader should see that disassembly sequence planning services can be invoked by taking advantage of the cloud, with the services delivered by different providers. An interoperable, service-oriented system for this can be realized by implementing our proposed framework. Many valuable and effective optimization algorithms have been proposed to solve different kinds of disassembly planning problems in prior research. They can be developed for our proposed system and be invoked by disassemblers. More importantly, our MTLBO algorithm seems like a good candidate for this.

6. Conclusion

In this article, we proposed an effective MTLBO algorithm to solve multi-objective disassembly sequence planning problem. As many non-dominated solutions as possible should be reviewed by a stakeholder to select in order make better decisions. To this end, we investigated a set of experimental simulations. The simulations showed that our MTLBO algorithm achieved good convergence and spread for the non-dominated solutions it found. In addition, the algorithm performed better than NSGA II to solve the multi-objective disassembly problem. We also proposed a service-oriented disassembly sequence planning framework

to help the stakeholders involved with handling e-waste to more effectively collaborate on disassembly sequence planning via the Internet. Our approach extends the e-business model to include treatment of e-waste at the end of its useful service life. It addresses key questions on how to take advantage of Web services and e-business technologies to support more sustainable business practices.

Some future works include: (1) developing a more practical mathematical model to solve disassembly problems by considering more disassembly issues to optimize, such as disassembly cost, environmental impact, energy consumption, etc.; (2) improving the performance of the optimization algorithms to solve the problems we focus on; and (3) developing a more detailed service-oriented disassembly sequence planning system to transform the business model of the disassembly industry. The latter, in our view, needs to be based on e-business engineering approaches and technologies that will support process performance improvement.

Acknowledgments

We would like to thank the editors, SI guest editors and reviewers, especially Prof. Robert J. Kauffman and Prof. Kuo-Ming Chao, for their valuable comments that help us to improve the quality of the paper. This research work is supported by the National Natural Science Foundation of China under Grant No. 51421062, the National Basic Research Program of China (973 Program) under Grant No. 2014CB046705, the Special Funds for the Scientific and Technological Cooperation with EU from the Ministry of Science and Technology of China under Grant No. 1208, the Independent Innovation Research Fund of Huazhong University of Science and Technology under Grant No. 2014XJGH010.

Appendix. Terminology

TLBO	Teaching-learning-based optimization
STLBO	Simplified teaching-learning-based optimization
MTLBO	Modified teaching-learning-based optimization

NSGA II Non-dominated sorting genetic algorithm II

References

- Adenso-Diaz, B., Garcia-Carbajal, S., Lozano, S., 2007. An efficient grasp algorithm for disassembly sequence planning. *OR Spectrum* 29 (3), 535–549.
- Ali, M., Siarry, P., Pant, M., 2012. An efficient differential evolution based algorithm for solving multi-objective optimization problems. *European Journal of Operational Research* 217 (2), 404–416.
- Baghdadi, Y., Al-Bulushi, W., 2015. A guidance process to modernize legacy applications for soa. *Service Oriented Computing and Applications* 9 (1), 41–58.
- Baresi, L., Miraz, M., Plebani, P., 2016. A distributed architecture for efficient web service discovery. *Service Oriented Computing and Applications* 10 (1), 1–17.
- Chao, K.-M., 2016. E-services in e-business engineering. *Electronic Commerce Research and Applications* 16, 77–81.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6 (2), 182–197.
- Demirkan, H., Kauffman, R. J., Vayghan, J. A., Fill, H.-G., Karagiannis, D., Maglio, P. P., 2008. Service-oriented technology and management: Perspectives on research and practice for the coming decade. *Electronic Commerce Research and Applications* 7 (4), 356–376.
- Huang, Y., Chung, J.-Y., 2003. A web services-based framework for business integration solutions. *Electronic Commerce Research and Applications* 2 (1), 15–26.
- Johnson, C. L., 2008. A framework for pricing government e-services. *Electronic Commerce Research and Applications* 6 (4), 484–489.
- Kongar, E., Gupta, S. M., 2006. Disassembly sequencing using genetic algorithm. *International Journal of Advanced Manufacturing Technology* 30 (5-6), 497–506.
- Li, W. D., Xia, K., Gao, L., Chao, K.-M., 2013. Selective disassembly planning for waste electrical and electronic equipment with case studies on liquid crystal displays. *Robotics and Computer-Integrated Manufacturing* 29 (4), 248–260.
- Pawar, P. J., Rao, R. V., 2012. Parameter optimization of machining processes using teaching-learning-based optimization algorithm. *The International Journal of Advanced Manufacturing Technology* 67 (5-8), 995–1006.
- Rao, R. V., Kalyankar, V. D., 2013. Parameter optimization of modern machining processes using teaching-learning-based optimization algorithm. *Engineering Applications of Artificial Intelligence* 26 (1), 524–531.

- Rao, R. V., Patel, V., 2013. Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *International Journal of Industrial Engineering Computations* 4 (1), 29–50.
- Rao, R. V., Savsani, V. J., Vakharia, D. P., 2011. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design* 43 (3), 303–315.
- Rehman, Z., Hussain, O. K., Chang, E., Dillon, T., 2015. Decision-making framework for user-based inter-cloud service migration. *Electronic Commerce Research and Applications* 14 (6), 523–531.
- Xia, K., Gao, L., Li, W., Chao, K.-M., 2014. Disassembly sequence planning using a simplified teaching-learning-based optimization algorithm. *Advanced Engineering Informatics* 28 (4), 518–527.
- Xia, K., Gao, L., Wang, L., Li, W., Chao, K.-M., 2015. A semantic information services framework for sustainable weee management toward cloud-based remanufacturing. *Journal of Manufacturing Science and Engineering* 137 (6), 061011.
- Yeh, W. C., 2012. Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization. *IEEE Transactions on Systems Man and Cybernetics - Part A: Systems and Humans* 42 (1), 250–261.
- Zhang, Q., Li, H., 2007. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11 (6), 712–731.