

Simultaneous Determination of Disassembly Sequence and Disassembly-to-Order Decisions Using Simulation Optimization

Mehmet Ali Ilgin¹

Department of Industrial Engineering,
Celal Bayar University,
Muradiye Campus,
Manisa 45140, Turkey
e-mail: mehmetali.ilgin@cbu.edu.tr

Gökçeçek Tuna Taşoğlu

Department of Industrial Engineering,
Celal Bayar University,
Muradiye Campus,
Manisa 45140, Turkey
e-mail: gokcececek.tuna@gmail.com

Strict environmental regulations and increasing public awareness toward environmental issues force many companies to establish dedicated facilities for product recovery. All product recovery options require some level of disassembly. That is why, the cost-effective management of product recovery operations highly depends on the effective planning of disassembly operations. There are two crucial issues common to most disassembly systems. The first issue is disassembly sequencing which involves the determination of an optimal or near optimal disassembly sequence. The second issue is disassembly-to-order (DTO) problem which involves the determination of the number of end of life (EOL) products to process to fulfill the demand for specified numbers of components and materials. Although disassembly sequencing decisions directly affects the various costs associated with a disassembly-to-order problem, these two issues are treated separately in the literature. In this study, a genetic algorithm (GA) based simulation optimization approach was proposed for the simultaneous determination of disassembly sequence and disassembly-to-order decisions. The applicability of the proposed approach was illustrated by providing a numerical example and the best values of GA parameters were identified by carrying out a Taguchi experimental design. [DOI: 10.1115/1.4033603]

Keywords: disassembly sequencing, disassembly-to-order systems, genetic algorithms, simulation optimization, Taguchi experimental design

Introduction

Disassembly can be defined as the process of systematic separation of a product into its components, subassemblies, materials, and other groupings. Although it is a crucial part of maintenance and repair processes, it has gained importance in the past decade mainly due to its role in product recovery which involves the recovery of materials and components from returned or EOL products via recycling [1] and remanufacturing [2]. The profitability of product recovery operations highly depend on the effective planning of disassembly process since almost all product recovery options require some level of disassembly [3].

One of the critical issues in the effective planning of disassembly process is disassembly sequencing which focuses on the determination of the best order of disassembly operations. In other words, it tries to answer the question “How to disassemble?” Disassembly sequence directly affects the time and effort required to disassemble a product. That is why, determination of the most suitable disassembly sequence has an utmost importance in cost effective disassembly.

Development of heuristic procedures is a popular solution approach to disassembly sequencing problem [4,5]. Metaheuristics were also frequently used to solve disassembly sequencing problem due to the combinatorial nature of this problem. Various sequencing methodologies based on GAs [6], genetic programming [7], Ant Colony Optimization [8], Tabu Search [9], Scatter Search [10], and Reinforcement Learning [11] were developed by researchers. Besides metaheuristics and heuristics, several other operations research and artificial intelligence techniques, namely, mathematical programming [12], Petri Nets [13], case based reasoning [14], and neural networks [15], were applied to disassembly sequencing problem.

Another important disassembly-related issue is the disassembly-to-order problem. This problem involves the determination of the optimal number of products to disassemble so as to maximize profit and minimize costs and the negative environmental effects. The solution methodologies developed for DTO problem can be categorized into two main classes based on the modeling of disassembly yield. The studies in the first class assume that disassembly yield is deterministic. In this class, linear physical programming [16], goal programming (GP) [17], and Neural Networks [18] were used to solve the DTO problem associated with various disassembly systems. The studies in the second class consider the uncertainty associated with disassembly yield. One-to-one and one-to-many heuristics proposed by Inderfurth and Langella [19] are the first solution methodologies presented in this class. Imtavanich and Gupta [20] extended Inderfurth and Langella [19] by integrating those heuristic procedures into their GP-based methodology.

It must be noted that majority of DTO methodologies proposed in the literature solve the DTO problem based on a predetermined disassembly sequence. In other words, they ignore the impact of disassembly sequencing on the DTO problem. However, several cost components (viz., disassembly, backorder, and holding costs) considered in DTO problems are affected by the chosen disassembly sequence. This can be attributed to two main reasons. First, disassembly sequencing directly affects the total disassembly time of a product. Second, it determines when a specific component in the product will be disassembled. There are several studies attempting to solve disassembly sequencing and disassembly-to-order problems simultaneously [21,22]. However, those studies ignore the stochastic nature of the disassembly systems by assuming deterministic disassembly times and deterministic EOL product arrivals.

This study fills the gaps in above-mentioned research by integrating simulation and GAs for the simultaneous optimization of disassembly sequencing and DTO problems. The proposed GA has a chromosome structure involving genes related with

¹Corresponding author.

Manuscript received November 26, 2015; final manuscript received May 4, 2016; published online June 22, 2016. Assoc. Editor: Karl R. Haapala.

disassembly sequencing and disassembly-to-order decisions. The fitness evaluation of each GA chromosome is carried out by a simulation model constructed for the disassembly of an example product structure. The simulation model allows us to model precedence relationships between components, stochastic disassembly times, stochastic EOL product arrivals, and component-discriminating demand.

The rest of the paper is organized as follows. The “Genetic Algorithms” section presents brief information on GAs. A discussion on the use of GAs in simulation optimization is provided in the “Simulation Optimization Using Genetic Algorithms” section. The “Joint Optimization of Disassembly Sequencing and Disassembly to Order Decisions Using Simulation Optimization” section provides the details of the proposed approach. Concluding remarks and future research directions are presented in the “Conclusions” section.

GAs

GAs are heuristic search and optimization methods imitating the natural evolution in computer environment [23]. They search for the best solution in a complex multidimensional search space in order to solve NP-hard problems. GAs have been employed to solve optimization problems in various areas including product design, image processing, maintenance planning, production and inventory management, reverse logistics, and disassembly planning.

One of the most distinctive characteristics of GAs is the fact that they work on a population of solutions instead of working on a single solution. This characteristic guarantees that many solution points are evaluated simultaneously in the search space and the probability of reaching the best solution increases. Each potential solution in the population is called as chromosome or individual. The GA process starts with the creation of the initial population consisting of individuals encoded depending on the predetermined representation scheme. The structure of the population changes at each generation due to the application of genetic operators to each chromosome of the population.

Fitness values of chromosomes are calculated at each generation. Selection operator determines the chromosomes to be chosen for the reproduction based on the fitness values. New chromosomes are created by recombining the selected chromosomes. There are two recombination operators: crossover and mutation. Crossover randomly exchanges parts of the two selected chromosomes with a given probability to form better chromosomes and explore the solution space. Mutation is applied following the crossover operation so as not to be trapped in the local optimum.

Simulation Optimization Using GAs

Simulation is a powerful decision-support tool. Its strength can be attributed to several reasons [24,25]. First of all simulation models allow decision makers to model complex systems with a minimum number of simplifying assumptions. Second, all stochastic aspects of a system can be modeled using simulation. Third, what-if type questions can be answered using the simulation model of a complex system. On the other hand, the most commonly cited disadvantage of simulation modeling is the need for the statistical analysis of the results provided by simulation models. In other words, while mathematical models provide optimum results, simulation models provide stochastic output data that needs to be analyzed through statistical techniques in order to answer various questions about the system being studied.

Although simulation is not an optimization technique, it can be integrated with an optimization technique. In that case, simulation takes input parameters from the optimization technique, evaluates the performance measures, and gives the output results back to the optimization technique. This mechanism is called as “simulation optimization.” There are several issues which differentiate simulation optimization problems from generic nonlinear programming

problems [26]. First of all, objective function(s) and constraint(s) are not defined with an explicit mathematical expression in simulation optimization problems. Second, the parameter space is not necessarily continuous. Discrete, continuous, or mixed search spaces are possible. Third, the search space may involve parameter values which are not allowed or impossible in the optimization problem considered. Fourth, the optimization problem may involve many local optimum points for the performance measures.

Heuristics, metaheuristics, gradient based search methods, stochastic approximation methods, sample path optimization are some of the most commonly used optimization techniques in simulation optimization studies [27]. This study employs GAs in simulation optimization process due to their ability to deal with the above cited challenges observed in the simulation-optimization problems. The following GA characteristics are especially beneficial in the simulation optimization process [28]:

- GAs need only objective function value to carry out the optimization process. They are compatible with any type of objective function. They do not require mathematical conditions for the objective function and constraints of the optimization problem.
- Most classical optimization methods utilize the gradient or higher order derivatives of the objective function. In addition, they consider only one point at a time. Improvements on this point are achieved gradually. GAs implement a multidirectional search process by working with a population of solutions. This prevents them from trapping in local optima.
- As previously mentioned, GAs use the encoded variables. Variables become discrete through the encoding mechanism. Hence, GAs can be used in discrete, continuous, or mixed search spaces.

Joint Optimization of Disassembly Sequencing and Disassembly to Order Decisions Using Simulation Optimization

This study proposes a methodology for the joint optimization of disassembly sequencing and disassembly to order decisions. Figure 1 presents the steps of the proposed methodology. First, a simulation model of the disassembly system under consideration is developed. Then a GA is developed by considering disassembly sequencing and disassembly to order related issues. Finally, the simulation model and GA is integrated in order to determine the optimum disassembly sequence and order amounts of product types. The following sections (Disassembly System, Simulation Model and Design of GA) present the details on the disassembly system and describe the steps of the proposed methodology by discussing the development of the simulation model and the GA.

Disassembly System. The product structure considered in this study is a modified version of air conditioner example presented in Ref. [29]. It consists of nine components. Figure 2 presents the precedence relationships among the components. Components B, D, and F have three types. Due to the differences in the types of these three components, there are three product types:

- Product Type 1: A - B (Type 1) - C - D (Type 1) - E - F (Type 1) - G - H - I
- Product Type 2: A - B (Type 2) - C - D (Type 2) - E - F (Type 2) - G - H - I
- Product Type 3: A - B (Type 3) - C - D (Type 3) - E - F (Type 3) - G - H - I

Product types 1, 2, and 3 arrive according to the exponential distributions with mean values of 10, 12, and 15 (products/hour), respectively.

Table 1 presents the component characteristics whose values do not depend on the type of the component. In this table, “direction” and “method” are used to calculate the costs associated with direction and method changes, respectively. There are two disassembly

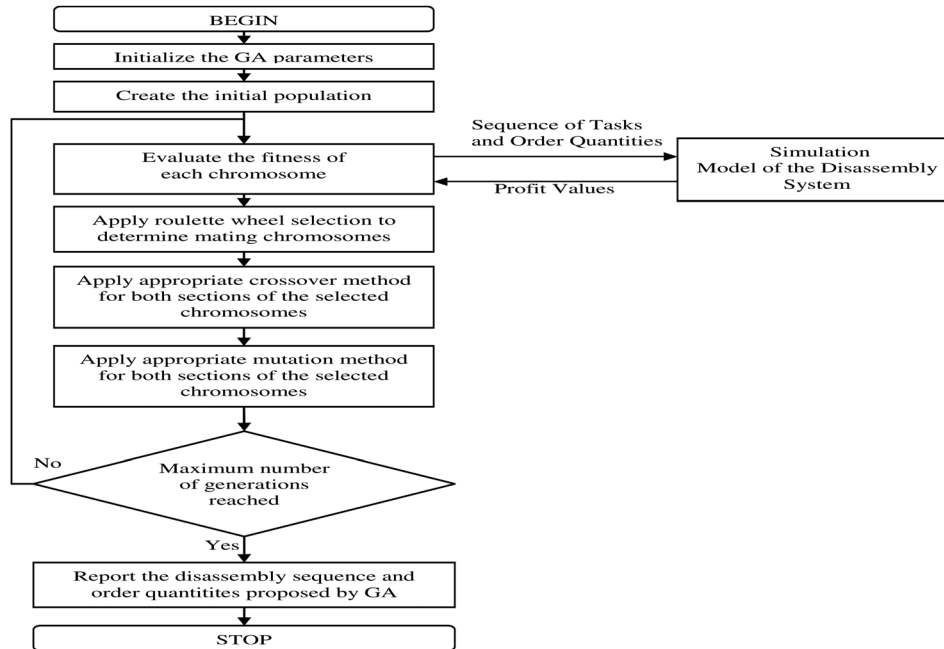


Fig. 1 Steps of the proposed methodology

methods: destructive (D) and nondestructive (ND). Disassembly times are assumed to be distributed according to the exponential distributions with mean disassembly time values presented in Table 1. Weight is used to calculate disposal cost and recycling revenue. Recyclable content (RC) and unit recycling revenue (URR) are required for the calculation of recycling revenue associated with a component. Unit disposal cost (UDC) is employed in disposal cost calculations.

The component characteristics presented in Table 2 depend on the type of the component. "Price," "probability of a missing component," and "probability of a failed component" are used in disposal cost and recycling revenue calculations. It must be noted that the customer demand occurs only for components B, D, E, F, and H. The demand for those components is assumed to be

distributed according to the exponential distributions with mean demand rate values presented in the last column of Table 2.

Simulation Model. Simulation model for the disassembly of the product structure described in the "Disassembly System" section was constructed using ARENA simulation software. The simulation model was run for 6 months with 8-hr shift per day (60,480 min). Ten replications of the model were carried out for each GA chromosome evaluation.

There are two main processes in the simulation model: disassembly-related activities carried out in each disassembly line station and the demand process for the components given in Table 2. Figure 3 presents the flow chart of the activities initiated by the arrival of a semi-disassembled product to the station which disassembles all types of component D. The similar activities are carried out at the other stations. The flow chart for the activities initiated by the arrival of customer demand for type 1 component D is given in Fig. 4. This process is similar for the other component types.

If there is no need for disassembly direction or disassembly method change, mean disassembly time values presented in Table 1 are directly used to generate disassembly times. However, if there is a need for disassembly direction or disassembly method change at a station, the mean disassembly time of this station is increased by 1 min. If both disassembly direction and disassembly method are subject to change, mean disassembly time at the station is increased by 2 min.

Design of GA. The following sections present the details on the design of GA by providing information on encoding scheme, initial population, fitness function, and genetic operators. Use of Taguchi experimental design for the determination of the suitable GA parameters is also discussed.

Encoding Scheme. In GAs, solutions are represented and encoded as chromosomes by considering the problem's features and constraints. Chromosomes are mostly represented as a combination of binary digits, integers, alphabets, or other characters. In this study, the solution is represented by a chromosome consisting of two different sections as depicted in Fig. 5.

The first section represents the sequence of disassembly tasks and it is encoded as a permutation of task numbers. It also

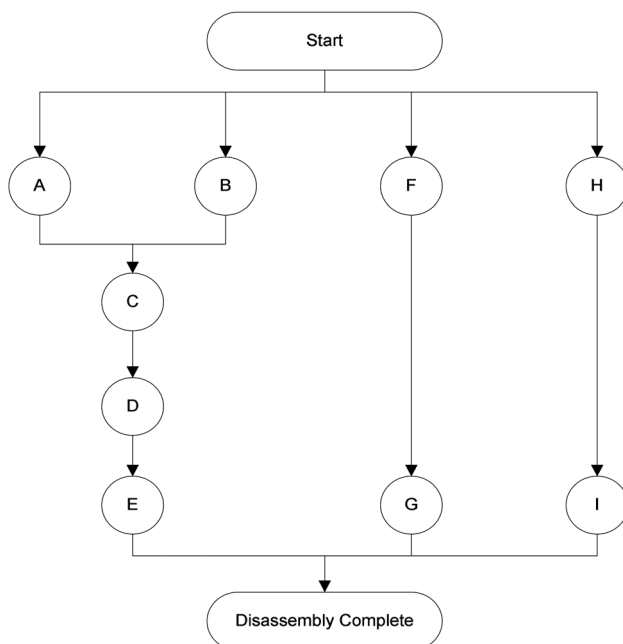


Fig. 2 Precedence relationships for the product

Table 1 Type-independent component characteristics

Comp. Name	Code	Predecessor	Direction	Method	Mean disassembly time (min)	Weight (lbs)	RC	UDC	URR
Evaporator	A	—	+x	D	5	2	30%	0.50	0.80
Control box	B	—	-x	ND	2	0.5	80%	0.35	0.50
Blower	C	A,B	-x	D	10	2.5	35%	0.50	0.90
Air guide	D	A,B,C	+y	ND	5	1	25%	0.25	0.30
Motor	E	A,B,C,D	+z	ND	7	1.5	20%	0.30	0.40
Condenser	F	—	-z	ND	4	3	50%	0.30	0.45
Fan	G	F	-x	D	3	5	75%	0.50	0.50
Protector	H	—	-z	ND	2	1	30%	0.55	0.75
Compressor	I	H	+y	D	6	2	80%	0.30	0.85

Table 2 Type-dependent component characteristics

Component Code	Types	Price (\$)	Probability of a missing component	Probability of a failed component	Mean demand rate (parts/per hour)
B	Type 1	5	0.04	0.10	10
	Type 2	7	0.05	0.15	12
	Type 3	9	0.03	0.10	10
D	Type 1	6	0.04	0.15	8
	Type 2	9	0.02	0.18	12
	Type 3	12	0.05	0.12	10
E	Type 1	10	0.08	0.20	15
F	Type 1	4	0.05	0.11	7
	Type 2	6	0.01	0.08	5
	Type 3	8	0.02	0.09	10
H	Type 1	9	0.10	0.25	15

considers the precedence relations and involves the number of tasks digits. The second section represents the order amount of each product type, respectively, and encoded as the order quantity varying between 50 and 100. In the light of this information, the chromosome given in Fig. 5 can be described as follows: Disassembly tasks must be performed in the order of 6-8-2-1-9-3-4-7-5 and the order quantities of type 1, type 2, and type 3 products should be 100, 99, and 100, respectively.

Initial Population. Initial population is formed by creating chromosomes randomly until the number of chromosomes reaches the population size. While constructing the initial population, precedence relations between the tasks are considered for the first section of the chromosomes. Order quantities are created based on the given range for the second section of the chromosome. Thus, the feasibility of the chromosomes is preserved and there is no need to apply any repair process.

Fitness Function. Fitness function value of each chromosome is evaluated by the simulation model using the following profit function:

$$\text{Profit} = (R_S + R_R) - (C_C + C_{DS} + C_{DY} + C_H + C_B) \quad (1)$$

where R_S is the revenue from the sale of demanded components, R_R is the revenue from the recycling of nonfunctional and undemanded components, C_C is the collection cost associated with the collection of all product types, C_{DS} is the disposal cost associated with the disposal of nonfunctional and undemanded components, C_{DY} is the disassembly cost associated with the disassembly of all product types, C_H is the holding cost of demanded components, and C_B is the backorder cost of demanded components. It must be noted that all cost and revenue components in profit expression are calculated by the simulation model for the simulated time period as follows:

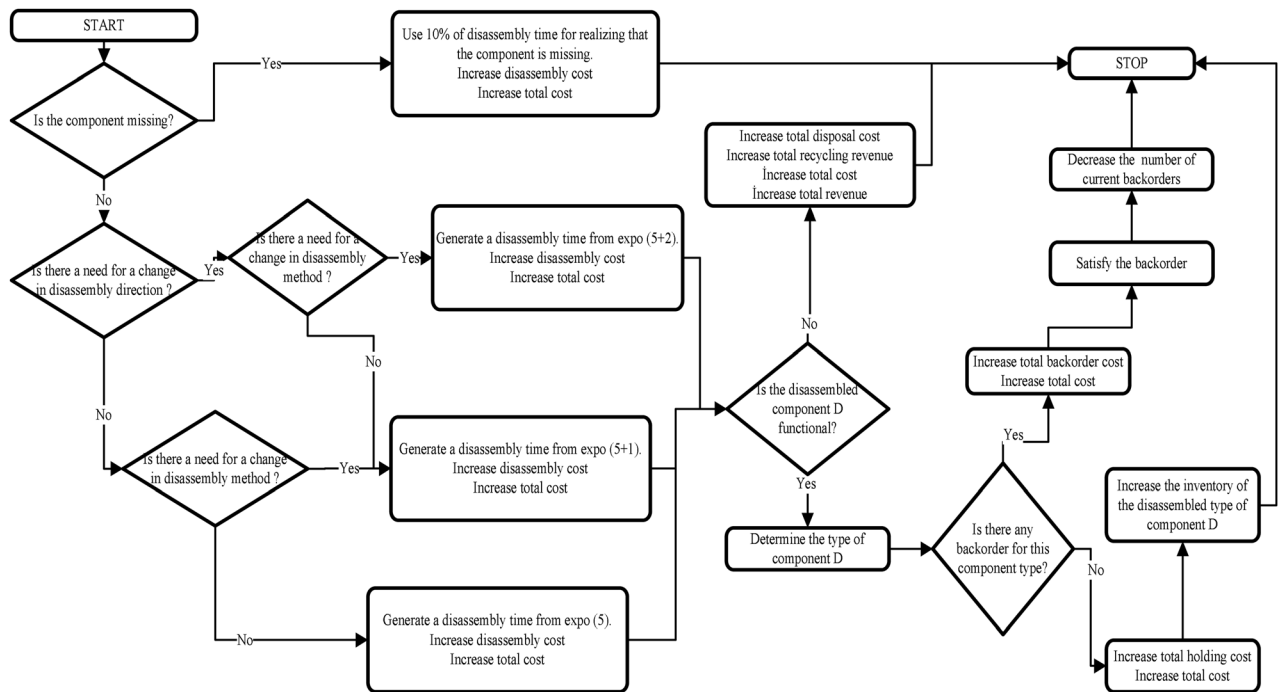


Fig. 3 Flow chart of the activities carried at the station disassembling component D

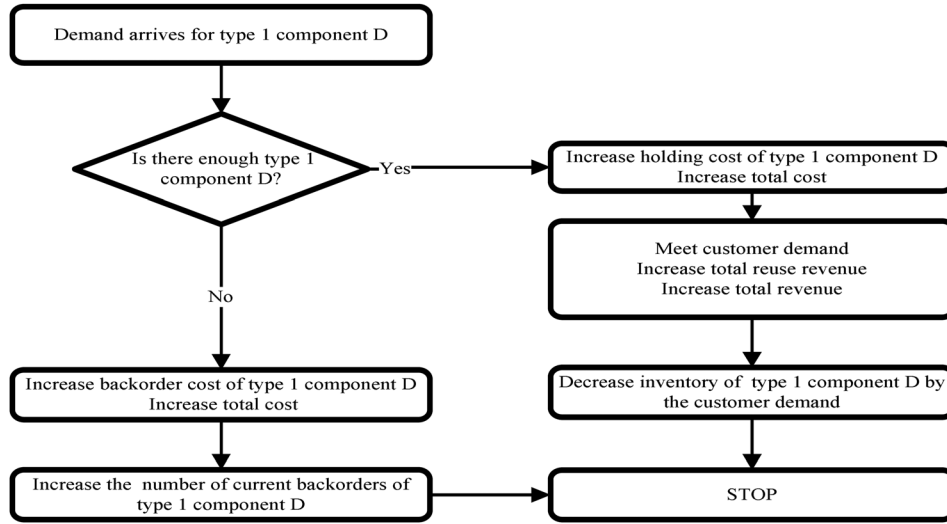


Fig. 4 Flow chart of the demand process of component D (Type 1)

$$R_s = \sum_{i=1}^{N_D} P_i \times NS_i \quad (2)$$

$$C_B = \sum_{i=1}^{N_D} BC_i \quad (8)$$

where P_i is the price of component type i and NS_i is the number of reusable component i disassembled during the simulated time period. N_D is the number of demanded component types. Since there are 11 component types in Table 2, N_D is 11 for this study

where BC_i is the backorder cost of component type i . The changes in the backorder level of the component during the simulated time period, price of the component and backorder cost rate are taken into consideration while calculating the backorder cost. Backorder cost rate is taken as 0.40 for all component types.

$$R_R = \sum_{i=1}^N W_i \cdot RC_i \cdot URR_i \quad (3)$$

where W_i is the weight of component i , RC_i is the RC of component i , and URR_i is the URR of component i . N is the number of components. Since there are nine components in Table 1, N is 9 for this study

Genetic Operators. Roulette wheel selection method is applied in order to determine the mating population. In this method, a mating chromosome is selected with a probability that is proportional to its fitness value. In other words, chromosomes with high fitness values are much more likely to be selected.

$$C_C = \sum_{i=1}^M UC_i \cdot NC_i \quad (4)$$

where UC_i is the unit collection cost of product type i , NC_i is the number of product type i 's collected during the simulated time period, and M is the number of product types. The value of M is 3 for this study since there are three product types. The value of UC_i is taken as \$5 for all product types

Crossover operation is implemented after selection. Two different crossover methods were applied to the each section of the chromosome. Precedence preservative crossover (PPX) methodology was applied to the first section and one-cut-point methodology was applied to the second section.

$$C_{DS} = \sum_{i=1}^N W_i \cdot (1 - RC_i) \cdot UDC_i \quad (5)$$

where W_i is the weight of component i , RC_i is the RC of component i , and UDC_i is the UDC of component i

In PPX methodology [30,31], in addition to the chromosomes of the parents, two more chromosomes, called mask chromosomes, are created for both parents. Mask chromosomes are created with the same size as the parent chromosomes and filled with the real numbers "1" or "2" which represent the order in which the tasks are successively selected from corresponding parents. Thereafter, mask chromosomes are created randomly, and the child chromosomes are obtained as follows:

$$C_{DY} = TDT \cdot UDYC \quad (6)$$

where TDT is the total disassembly time and UDYC is the unit disassembly cost

The leftmost disassembly task in one of the two parents is selected in accordance with the order of parents given in the mask chromosome. After a task is selected, it is deleted in both parents. Finally, the selected task is transferred to the corresponding child chromosome.

$$C_H = \sum_{i=1}^{N_D} HC_i \quad (7)$$

where HC_i is the holding cost of component type i . The changes in the inventory level of the component during the simulated time period, price of the component and holding cost rate are taken into consideration while calculating the holding cost. Holding cost rate is taken as 0.20 for all component types

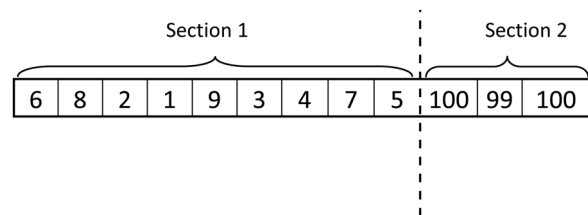


Fig. 5 Structure of a chromosome

First section of parent 1:

8	2	6	1	7	3	4	9	5
---	---	---	---	---	---	---	---	---

First section of parent 2:

2	1	6	3	7	8	4	9	5
---	---	---	---	---	---	---	---	---

Mask for the first section of child 1:

2	2	2	1	2	1	1	2	1
---	---	---	---	---	---	---	---	---

Mask for the first section of child 2:

2	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

First sections of child 1 and child 2 can be obtained as follows:
First section of child 1:

2	1	6	8	3	7	4	9	5
---	---	---	---	---	---	---	---	---

First section of child 2:

2	8	6	1	7	3	4	9	5
---	---	---	---	---	---	---	---	---

The second crossover methodology was applied to the second section. First a single cut point is selected from both parents. Afterward, all digits beyond that point in either parent are swapped between each other. The resulting chromosomes are obtained as follows:

Second section of the parent 1:

60	70	80
----	----	----

Second section of the parent 2

75	55	65
----	----	----

Assuming that the cut-point is determined as 2;

60	70	80
75	55	65

The third digit of each parent is exchanged and the following children are obtained:

Second section of the child 1:

60	70	65
----	----	----

Second section of the child 2:

75	55	80
----	----	----

Following the crossover operation, the mutation process is carried out in an effort to avoid local minima and to ensure that newly generated population is not uniform and incapable of further evolution.

Precedence-preserving shift mutation method which preserves all the precedence relations between disassembly tasks is applied

Table 3 Factor levels

Factors	Levels		
	Level 1	Level 2	Level 3
Population size	30	50	70
Crossover rate	30%	50%	70%
Mutation rate	2%	4%	6%
Generation number	50	75	100

to the first section of the chromosome. Once a disassembly task is selected from a single parent chromosome, it is moved into another position considering the precedence constraints for the task. Since it is known that the disassembly tasks are ordered considering the precedence relations, a new position for the task to be moved should not violate the precedence relations.

In order to preserve the precedence relations, the minimum position of a successor and maximum position of a predecessor of the task to be moved should be determined. If P_{max} is the maximum position of a predecessor and S_{min} is the minimum position of a successor, the new position of the disassembly task should be between S_{min} and P_{max} , so as not to violate the precedence relations and to obtain a feasible chromosome. Therefore, a new position between S_{min} and P_{max} is generated and the task is moved into this new position.

The second section of the chromosome is mutated by changing the value of a random single digit. Once the random product type is selected, the value of the order quantity of the selected product is diversified based on the given feasible range.

Determination of GA Parameter Values Using Taguchi Design.

The suitable values for the variable GA parameters (e.g., crossover rate, mutation rate) must be determined in order to improve the convergence performance of GA. That is why, an experimental design analysis was applied to determine the values of four GA parameters: population size, crossover rate, mutation rate, and number of generations.

Table 3 presents the levels of GA parameters. According to Table 3, a full factorial design with 5 replications for each experiment will require 405 experiments ($3^4 \times 5$). Since the simulation model is replicated 10 times for each experiment, 4050 simulation replications will be required. In order to avoid the high number of experiments required by a full factorial design, Taguchi experimental design [32,33] was employed in this study. Specifically, Taguchi's L9 orthogonal array was used. This orthogonal array can accommodate four factors with three levels. The experimental design scheme presented in Table 4 was obtained by entering the numerical values of factor levels into L9 orthogonal array.

Average S/N ratios for each factor across four levels are provided in Table 5. In this table, delta is the difference between the maximum and minimum average S/N ratios of a factor and

Table 4 L9 orthogonal array based on experimental design employed in this study

Experiment number	Column			
	1	2	3	4
1	30	30%	2%	50
2	30	50%	4%	75
3	30	70%	6%	100
4	50	30%	4%	100
5	50	50%	6%	50
6	50	70%	2%	75
7	70	30%	6%	75
8	70	50%	2%	100
9	70	70%	4%	50

Table 5 Signal to noise (S/N) ratios

Levels	S/N Ratio			
	Population size	Crossover rate	Mutation rate	Generation number
First level	109.6	109.7	109.7	109.7
Second level	109.7	109.7	109.7	109.7
Third level	109.7	109.7	109.7	109.7
Fourth level	109.7	109.7	109.7	109.7
Δ	0.1	0.0	0.0	0.0
Rank	1	4	2	3

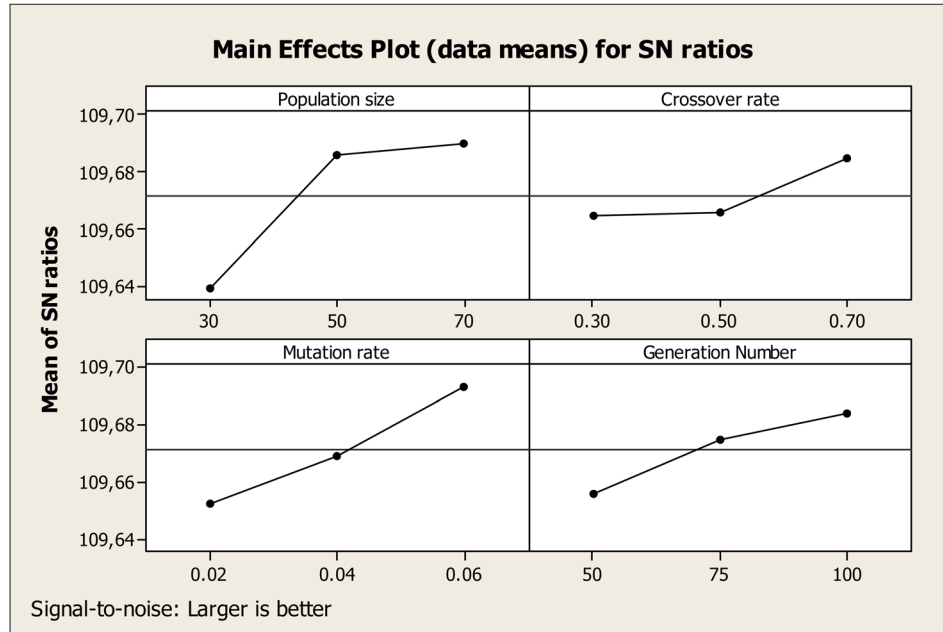


Fig. 6 Main effects plot for S/N ratios

indicates the impact of a factor. The factor with the largest delta value has the largest impact on the response. The ranking of the factors according to the delta values is presented in the last row of Table 5. The most important factor is population size. The other three factors have approximately the same importance level.

The graphical representation of the average S/N ratios is given in Fig. 6. The largest S/N ratio for each factor is preferred. The value of each parameter was determined as given in

Table 6 based on the fact that the largest S/N ratio for a factor is preferable.

Results. GA was run by using the parameter values presented in Table 6. The GA optimization process was completed in 77,903 s using a laptop with a 1.8 GHz Intel Core i5 processor and 4GB RAM. Considering the size of the solution space (4.84×10^{13}), the GA obtained a converged solution in a very

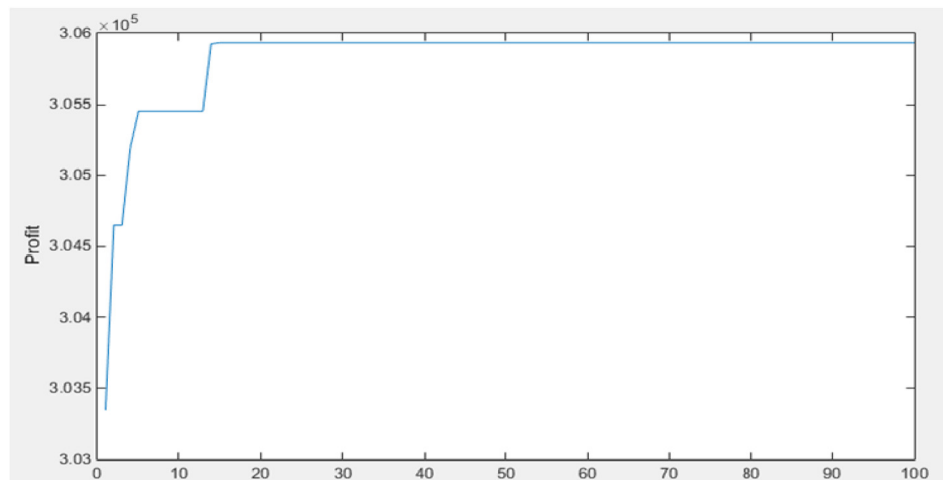


Fig. 7 Convergence graph of GA

Table 6 GA parameter values

Parameter	Value
Population size	70
Crossover rate	0.70
Mutation rate	0.06
Generation number	100

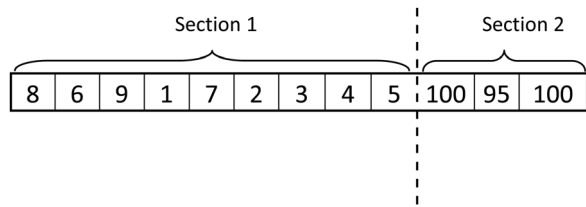


Fig. 8 The converged GA solution

small computational time. Figure 7 presents the convergence graph. According to this figure, after 15 iterations, the algorithm converges to a solution which has a profit value of \$305,930. This means a %1.1 improvement in profit compared to the initial solution which has a profit value of \$302,590. The chromosome associated with the converged solution is given in Fig. 8. This solution determines the sequence of disassembly tasks as 8-6-9-1-7-2-3-4-5 and the order quantities of type 1, type 2, and type 3 products are 100, 95, and 100, respectively.

Conclusions

Disassembly-to-order problem involves the determination of the number of EOL products to process to fulfill the demand for specified numbers of components and materials. The sequence followed to disassemble products has direct impacts on DTO problem. However, disassembly sequencing and DTO problems are analyzed separately in the literature. In order to fill this research gap, we developed a GA-based simulation optimization methodology for the joint optimization of disassembly sequence and order quantities of EOL products. First, a simulation model of a disassembly system was developed. Then, this simulation model was employed by a GA in order to determine the fitness values of solution alternatives. The sequence of disassembly tasks together with the order quantities of EOL product types were determined based on the converged GA solution.

Although GA is a powerful optimization technique and its structure is very suitable for a simulation optimization study, there are many other metaheuristics that can be integrated with simulation. In future research, the performance of the proposed GA based simulation optimization methodology can be compared with the simulation-optimization methodologies employing other metaheuristics such as simulated annealing, tabu search, and ant colony optimization.

References

[1] Haapala, K. R., Zhao, F., Camelio, J., Sutherland, J. W., Skerlos, S. J., Dornfeld, D. A., Jawahir, I. S., Clarens, A. F., and Rickli, J. L., 2013, "A Review of Engineering Research in Sustainable Manufacturing," *ASME J. Manuf. Sci. Eng.*, **135**(4), p. 041013.
 [2] Xia, K., Gao, L., Wang, L., Li, W., and Chao, K.-M., 2015, "A Semantic Information Services Framework for Sustainable WEEE Management Toward Cloud-Based Remanufacturing," *ASME J. Manuf. Sci. Eng.*, **137**(6), p. 061011.
 [3] Ilgin, M. A., and Gupta, S. M., 2010, "Environmentally Conscious Manufacturing and Product Recovery (ECMPRO): A Review of the State of the Art," *J. Environ. Manage.*, **91**(3), pp. 563–591.

[4] Lambert, A. J. D., and Gupta, S. M., 2008, "Methods for Optimum and Near Optimum Disassembly Sequencing," *Int. J. Prod. Res.*, **46**(11), pp. 2845–2865.
 [5] Adenso-Díaz, B., García-Carbajal, S., and Gupta, S. M., 2008, "A Path-Relinking Approach for a Bi-Criteria Disassembly Sequencing Problem," *Comput. Oper. Res.*, **35**(12), pp. 3989–3997.
 [6] Kongar, E., and Gupta, S. M., 2006, "Disassembly Sequencing Using Genetic Algorithm," *Int. J. Adv. Manuf. Technol.*, **30**(5), pp. 497–506.
 [7] Shimizu, Y., Tsuji, K., and Nomura, M., 2007, "Optimal Disassembly Sequence Generation Using a Genetic Programming," *Int. J. Prod. Res.*, **45**(18–19), pp. 4537–4554.
 [8] McGovern, S. M., and Gupta, S. M., 2006, "Ant Colony Optimization for Disassembly Sequencing With Multiple Objectives," *Int. J. Adv. Manuf. Technol.*, **30**(5), pp. 481–496.
 [9] Alshibli, M., Sayed, A., Kongar, E., Sobh, T. M., and Gupta, S. M., 2016, "Disassembly Sequencing Using Tabu Search," *J. Intell. Rob. Syst.*, **82**(1), pp. 69–79.
 [10] Guo, X., and Liu, S., 2014, "A Scatter Search Approach for Multiobjective Selective Disassembly Sequence Problem," *Discrete Dyn. Nat. Soc.*, **2014**, pp. 1–9.
 [11] Reveliotis, S. A., 2007, "Uncertainty Management in Optimal Disassembly Planning Through Learning-Based Strategies," *IIE Trans.*, **39**(6), pp. 645–658.
 [12] Lambert, A. J. D., 2006, "Exact Methods in Optimum Disassembly Sequence Search for Problems Subject to Sequence Dependent Costs," *Omega*, **34**(6), pp. 538–549.
 [13] Tang, Y., Zhou, M., and Gao, M., 2006, "Fuzzy-Petri-Net-Based Disassembly Planning Considering Human Factors," *IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans*, **36**(4), pp. 718–726.
 [14] Veerakamolmal, P., and Gupta, S., 2002, "A Case-Based Reasoning Approach for Automating Disassembly Process Planning," *J. Intell. Manuf.*, **13**(1), pp. 47–60.
 [15] Hsin-Hao, H., Wang, M. H., and Johnson, M. R., 2000, "Disassembly Sequence Generation Using a Neural Network Approach," *J. Manuf. Syst.*, **19**(2), pp. 73–82.
 [16] Odemir, O., and Gupta, S. M., 2014, "A Multi-Criteria Decision Making Model for Advanced Repair-to-Order and Disassembly-to-Order System," *Eur. J. Oper. Res.*, **233**(2), pp. 408–419.
 [17] Kongar, E., and Gupta, S. M., 2002, "A Multi-Criteria Decision Making Approach for Disassembly-to-Order Systems," *J. Electron. Manuf.*, **11**(2), pp. 171–183.
 [18] Gupta, S. M., Imtavanich, P., and Nakashima, K., 2009, "Using Neural Networks to Solve a Disassembly-to-Order Problem," *Int. J. Biomed. Soft Comput. Human Sci.*, **15**(1), pp. 67–71.
 [19] Inderfurth, K., and Langella, I., 2006, "Heuristics for Solving Disassembly-to-Order Problems With Stochastic Yields," *OR Spectr.*, **28**(1), pp. 73–99.
 [20] Imtavanich, P., and Gupta, S. M., 2006, "Calculating Disassembly Yields in a Multi-Criteria Decision-Making Environment for a Disassembly to Order System," in *Applications of Management Science: In Productivity, Finance: Operations*, K. D. Lawrence, and R. K. Klimberg, eds., Elsevier Ltd., Amsterdam, The Netherlands, pp. 109–125.
 [21] Kwak, M., and Kim, H. M., 2010, "Evaluating End-of-Life Recovery Profit by a Simultaneous Consideration of Product Design and Recovery Network Design," *ASME J. Mech. Des.*, **132**(7), p. 071001.
 [22] Kang, C. M., and Hong, Y. S., 2012, "Dynamic Disassembly Planning for Remanufacturing of Multiple Types of Products," *Int. J. Prod. Res.*, **50**(22), pp. 6236–6248.
 [23] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Boston, MA.
 [24] Banks, J., Carson, J. S., Nelson, B. L., and Nicol, D. M., 2001, *Discrete-Event System Simulation*, Prentice Hall, Upper Saddle River, NJ.
 [25] Law, A. M., 2007, *Simulation Modelling and Analysis*, McGraw-Hill, New York.
 [26] Paul, R. J., and Chaney, T. S., 1998, "Simulation Optimisation Using a Genetic Algorithm," *Simul. Pract. Theory*, **6**(6), pp. 601–611.
 [27] Azadivar, F., 1999, "Simulation Optimization Methodologies," 31st Conference on Winter Simulation Conference, pp. 93–100.
 [28] Ilgin, M. A., and Tunali, S., 2007, "Joint Optimization of Spare Parts Inventory and Maintenance Policies Using Genetic Algorithms," *Int. J. Adv. Manuf. Technol.*, **34**(5–6), pp. 594–604.
 [29] Ilgin, M. A., and Gupta, S. M., 2011, "Evaluating the Impact of Sensor-Embedded Products on the Performance of an Air Conditioner Disassembly Line," *Int. J. Adv. Manuf. Technol.*, **53**(9–12), pp. 1199–1216.
 [30] Bierwirth, C., Mattfeld, D., and Kopfer, H., 1996, "On Permutation Representations for Scheduling Problems," *Parallel Problem Solving From Nature IV*, Springer, Berlin, Heidelberg, pp. 310–318.
 [31] Blanton, J. L., and Wainwright, R. L., 1993, "Multiple Vehicle Routing With Time and Capacity Constraints Using Genetic Algorithms," Fifth International Conference on Genetic Algorithm, San Mateo, CA, pp. 452–459.
 [32] Taguchi, G., 1986, *Introduction to Quality Engineering: Designing Quality Into Products and Processes*, Asian Productivity Organization, Tokyo.
 [33] Phadke, M. S., 1989, *Quality Engineering Using Robust Design*, Prentice Hall, Upper Saddle River, NJ.