Available online at www.alphanumericjournal.com

alphanumeric journal

The Journal of Operations Research, Statistics, Econometrics and Management Information Systems

Volume 9, Issue 1, 2021

Received: March 04, 2021
Accepted: May 18, 2021
Published Online: June 30, 2021

AJ ID: 2021.09.01.OR.02
DOI: 10.17093/alphanumeric.891406
Research Article

Disassembly Line Balancing by Using Simulation Optimization *

Muhammet Enes Akpınar, Ph.D. *

Res. Assist., Department of Industrial Engineering, Faculty of Engineering, Manisa Celal Bayar University, Manisa, Turkey, enes.akpinar@cbu.edu.tr

Mehmet Ali Ilgın, Ph.D.

Assoc. Prof., Department of Industrial Engineering, Faculty of Engineering, Manisa Celal Bayar University, Manisa, Turkey, mehmetali.ilgin@cbu.edu.tr

Hüseyin Aktaş, Ph.D.

Prof., Faculty of Economics and Administrative Sciences, Manisa Celal Bayar University, Manisa, Turkey, huseyin.aktas@cbu.edu.tr

* Manisa Celal Bayar Üniversitesi, Mühendislik Fakültesi, Şehit Prof. Dr. İlhan Varank Yerleşkesi, Muradiye, Manisa, Türkiye.

ABSTRACT

Increasing environmental awareness in today's society and stricter environmental regulations have forced manufacturing firms to take necessary actions for the recovery of end-of-life (EOL) products through different options (e.g., recycling, remanufacturing,). Disassembly is regarded as a critical operation in EOL treatment of used products since all product recovery options require the disassembly of EOL products at certain levels. This critical operation is generally carried out by forming disassembly lines in product recovery facilities. Miscellaneous methodologies based on heuristics, metaheuristics and mathematical programming have been proposed for the balancing of disassembly lines. Majority of those methodologies assume that disassembly line parameters are deterministic by ignoring the fact that a disassembly line involves great deal of uncertainty mainly due to uncertain conditions of arriving EOL products. Considering this high level of uncertainty, simulation modeling can be an effective tool for the modeling of disassembly lines. In this study, a simulation-based disassembly line balancing methodology is proposed for the explicit consideration of stochastic parameters. First, simulation model of a disassembly line is constructed. Since the disassembly line balancing problem has a combinatorial nature, two commonly used metaheuristics (i.e., genetic algorithms (GAs) and simulated annealing (SA)) are integrated with the simulation model in order to balance the disassembly line. The disassembly sequence and task assignments proposed by GA are compared with the sequence and task assignments proposed by SA. This comparison indicates that GA outperforms SA in four of eight performance measures while both algorithms have the same value for line efficiency measure.

Keywords: Line Balancing, Disassembly, Genetic Algorithms, Simulated Annealing, Simulation

*This paper is a part of the first author's approved Ph.D. thesis.

1. Introduction

Manufacture of high-quality products with the cheapest and fastest way is the main aim of traditional manufacturing systems. However, decrease in natural resources and increased environmental awareness of consumers forced manufacturing companies to consider the environmental impacts of their manufacturing activities. Hence, sustainable manufacturing which tries to minimize the negative impacts of manufacturing systems on environment has emerged as a vital manufacturing concept.

Among various sustainable manufacturing issues, product recovery has gained popularity among researchers in recent years. It includes the recovery of materials and/or components from used products through different options (e.g., recycling, remanufacturing). All of these options call for disassembly which is the systematic separation of a product into its subassemblies, components or other groupings. Disassembly procedures can be fulfilled in a disassembly cell, in a single workstation or in a disassembly line. Disassembly line is the most commonly used layout due to its high efficiency.

The maximum possible yield from a disassembly line can only be obtained if the line is balanced. In another words, work should be evenly distributed among the stations of disassembly lines like assembly lines. However, disassembly lines have many differences compared with assembly lines. For example, in disassembly lines, disassembly of expensive components as early as possible is an important performance measure since the possibility of damage to expensive components is minimized by disassembling them earlier. A similar performance measure is also valid for hazardous components since the early disassembly of hazardous components decreases the risk of dangerous events such as explosions. Considering those differences, assembly line balancing methodologies cannot be employed for disassembly line balancing.

Miscellaneous solution methodologies were proposed for disassembly line balancing problems (Deniz & Ozcelik, 2019; Gungor & Gupta, 1999; Mehmet Ali Ilgin & Gupta, 2010; Özceylan, Kalayci, Güngör, & Gupta, 2019). Among these methodologies, heuristics received increasing attention of researchers. Starting with the study of (Askiner Güngör & Gupta, 2001) and (Aşkiner Güngör & Gupta, 2002), various disassembly line balancing heuristics were developed. Majority of those methodologies are based on a multi-criteria decision making technique including TOPSIS (Avikal, Jain, & Mishra, 2014), PROMETHEE (Avikal, Mishra, & Jain, 2013, 2014; Avikal, Mishra, Jain, & Yadav, 2013) and DEMATEL (Mehmet Ali Ilgin, 2019).

Metaheuristics-based methodologies are also popular in the literature since the disassembly line balancing problem has a combinatorial nature. GAs (Kalayci, Polat, & Gupta, 2016; S. M. McGovern & Gupta, 2007; Seamus M. McGovern & Gupta, 2007; Seidi & Saghari, 2016), evolutionary optimization (Fang, Liu, Li, Laili, & Pham, 2019), reinforcement learning (Tuncel, Zeid, & Kamarthi, 2014), SA(Kalayci & Gupta, 2013d; K. Wang, Li, & Gao, 2019), tabu search (Kalayci & Gupta, 2014), ant colony optimization(Agrawal & Tiwari, 2008; L. P. Ding, Feng, Tan, & Gao, 2010; Kalayci & Gupta, 2013b; Seamus M. McGovern & Gupta, 2006), artificial bee colony algorithm (Kalayci & Gupta, 2013c; Kalayci, Hancilar, Gungor, & Gupta, 2015; Liu et al., 2018; S.

Wang, Guo, & Liu, 2019; Wang, Li, Gao, Li, & Sutherland, 2021), particle swarm optimization (Kalayci & Gupta, 2013a; Xiao, Wang, Yu, & Nie, 2017), firefly algorithm (Zhu, et al., 2018)and artificial fish swarm algorithm (Zhang, Wang, Zhu, & Wang, 2017) are the most commonly used metaheuristic techniques.

There are many disassembly line balancing methodologies based on mathematical programming techniques. Most of those methodologies assume that all disassembly line parameters are deterministic. Mixed integer programming (Altekin, 2017; Altekin & Akkan, 2012; Altekin, Bayındır, & Gümüşkaya, 2016; Altekin, Kandiller, & Ozdemirel, 2008; Özceylan & Paksoy, 2013; Paksoy, Güngör, Özceylan, & Hancilar, 2013), branch and bound (Li, Cil, Mete & Kucukkoc, 2020), mixed integer linear programming (Edis, Ilgin, & Edis, 2019), dynamic programming (Koc, Sabuncuoglu, & Erel, 2009; Zhou, Guo, & Li, 2020), linear physical programming (Ilgin, Akçay, & Araz, 2017) are some of the techniques used in those methodologies. Some researchers integrate mathematical programming techniques and fuzzy logic (Özceylan & Paksoy, 2014b, 2014a).

The number of mathematical programming based methodologies considering stochastic issues in disassembly line balancing is very limited (Bentaha, Battaïa, & Dolgui, 2014; Bentaha, Battaiä, & Dolgui, 2015; Bentaha, Battaïa, Dolgui, & Hu, 2015). In these studies, only disassembly times are modeled as stochastic. However, there are other stochastic processes. For instance, inter-arrival times of used products arriving at a disassembly line are stochastic. Considering those stochastic issues, simulation optimization can be an effective tool for balancing disassembly lines. That is why we propose a simulation optimization based disassembly line balancing methodology in this paper. First, a simulation model of the disassembly line is constructed by considering stochastic disassembly times and stochastic inter-arrival times. Then, two commonly used meta-heuristics GAs and SA are integrated with this simulation model so as to assignment of disassembly tasks to stations and determine the disassembly sequence.

The remainder of the paper is organized as follows. Sections 2 and 3 provide brief information on GAs and SA respectively. The proposed disassembly line balancing methodology and application results are presented in Section 4. Section 5 presents the conclusions and future research directions.

2. Genetic Algorithms

The genetic algorithms involve the application of selection, crossover and mutation processes to a population of individuals. Following the application of these procedures, a new population is created. The old population and the new population are exchanged for each other and each individual has its own regulated value. The newly formed population is selected according to this regulated value and more compatible populations are tried to be formed in each newly created population.

GAs are particularly used in the areas of optimization, automated, mechanical learning, finance, marketing, scheduling, assembly/disassembly line balancing, plant layout and system reliability. The basic characteristics of GAs can be listed as follows;

- Poor solutions tend to disappear while good solutions tend to be used to create better solutions as the population evolves from generation to generation.

- They scan not the whole solution space only part of it.
- They reach a possible solution in shorter time by doing an active search.
- They do not stick to local best solutions by simultaneously examining a population of solutions.

The pseudo code of GAs is as follows (H. Ding, Benyoucef, & Xie, 2003):

Procedure: Genetic algorithms
begin
t←0;
initialize P(t);
evaluate P(t);
while (not termination condition) do
    select P(t+1) from P(t);
    crossover (recombine P(t+1));
    mutation (recombine P(t+1));
    evaluate P(t+1) ;
t←t+1;
end
end

GAs first creates an initial population of individuals coded in accordance with the notation specified in the solution steps. Each chromosome in the initial population represents a possible solution to the problem. Each chromosome has a conformity value indicating the quality of the solution it encodes. The basic working logic of GAs is based on the proliferation of chromosomes with better conformity values, just like in the evolutionary process.

Selection is the process of selecting individuals of a new generation from the existing population according to the selection method chosen. The crossover operator is one of the substantial parameters that affect the performance of the GAs. Crossover creates new offspring by manipulating selected genes in the parent. Following the crossover operation, some of the chromosomes are mutated to increase the diversity of the chromosomes in the generation. The purpose of this process is to identify changes within the population. During the mutation process, the number of genes on the chromosome remains constant.

There are many studies where GAs and simulation optimization are used simultaneously. Spare part inventory policy determination (M. Ali Ilgin & Tunali, 2007), supplier selection (L. P. Ding et al., 2010), facility layout planning (Azadivar & Wang, 2000), production process planning (Amiri & Mohtashami, 2012), scheduling (Lin & Lin, 2015; Zeng, Diabat, & Zhang, 2015), disassembly sequencing (Mehmet Ali Ilgin & Taşoğlu, 2016) and risk assessment(Yin, Wu, & Hsu, 2017) are some of the areas at which GA-based simulation optimization have been successfully implemented.

3. Simulated Annealing

Simulated annealing is a general search algorithm introduced to solve combinatorial optimization problems. The main aim of the annealing process is to increase the temperature of system and then slowly cool the system to achieve the intended structure. Therefore, the annealing process consists of two steps: heating and cooling. First, heating is performed to increase the energy of the physical system, so that the atoms are freely dispersed in the system to obtain an unbalanced structure. The system is then cooled slowly to obtain the desired structure of the system(Dowsland & Thompson, 2012).

SA can be used to solve bound-constrained and unconstrained optimization problems. It uses the same basic steps of the local search methods with one exception. The cooling process of SA is an exponential statement that allows new neighboring solutions to be produced for better results.

The pseudo code of SA is as follows (Eglese, 1990):

```
s ← s0; e ← E(s)
sbest ← s; ebest ← e
k ← 0
while k <kmax and e >emax
        T ← temperature(k/kmax)
        snew ← neighbor(s)
        enew ← E(snew)
        if P(e, enew, T) > random() then
        s ← snew; e ← enew
        if e <ebest then
        sbest ← snew; ebest ← enew
        k ← k + 1
returnsbest
```

As seen in the pseudo code, the presence of an iteration number requires the completion of the current step so as to pass to the next step. The serial algorithm has a single solution at each step and is compared with the existing one. It is possible to produce more than one solution and choose the most suitable one by distributing each step to the appropriate works.

The correct determination of the parameters used for the SA algorithm plays a substantial role in the solution of the problem. The SA algorithm has 5 main parameters: initial temperature, target temperature, number of iterations, cooling rate and stopping criteria. It is very important to determine the starting temperature in SA. The cooling rate is represented by α which takes values between 0 and 1. As the cooling rate approaches 0, the system cools faster and the cooling of the system slows down as it approaches 1.

The target temperature value is used as the stopping rule. The algorithm starts the solution steps with the initial temperature and decreases the temperature with the cooling coefficient determined in each step. The SA algorithm continues its operations until a specified target temperature is reached. The number of iterations refers to multiple operations of the SA algorithm. At each of iteration, SA starts its steps again with the specified parameters. In this way, it is aimed to reach the best solution. The stopping criterion means that the loop is stopped when it meets a certain condition. In physical annealing, the process is automatically ended when it reaches a certain temperature.

SA and simulation models can be integrated for the optimization purposes. Some of the areas in which SA-based simulation optimization have been applied include scheduling (Mattila & Virtanen, 2015; Tasoglu & Yildiz, 2019), production planning and control (Güçdemir & Selim, 2017), decision support systems (Ozcan, Tànfani, & Testi, 2017) and design alternative selection (Ameli, Mansour, & Ahmadi-Javid, 2019).

4. Disassembly Line Balancing by Using Simulation Optimization

This study proposes a four-phase simulation optimization approximation for the balancing of disassembly lines. At first phase, a discrete event system simulation algorithm of the disassembly line under analysis is constructed. Second, a metaheuristic is constructed so as to determination of disassembly sequence. Third, the metaheuristic and the simulation model are integrated for the fitness evaluation of alternative disassembly sequences. Fourth, disassembly tasks are assigned to stations considering the disassembly sequence and the average cycle time. The following sections represent the details about the disassembly system, the simulation model, the design and parameter estimation of GA and SA metaheuristics and the results of the study.

4.1. Disassembly Line

The disassembly line considered in this study is used for the disassembly of wall hung boilers. Table 1 presents the characteristics of 17 components included in a boiler. Among all components, demand exists for component 2 (Heat exchanger) and component 9 (mother board). The only hazardous component is component 11(Plate heat exchanger).

Component Number	Component Name	Price ($)	Demand (per year)	Hazardous Component
1	Cover	-	-	No
2	Heat exchanger	145	250	No
3	Fan	-	-	No
4	Venturi	-	-	No
5	Ignition and ionization spark plugs	-	-	No
6	Flue gas temperature sensor	-	-	No
7	APS	-	-	No
8	Expansion tank	-	-	No
9	Motherboard	50	400	No
10	Water Group	-	-	No
11	Plate heat exchanger	-	-	Yes
12	Cable group	-	-	No
13	Condensate drain siphon	-	-	No
14	Copper Tube Group	-	-	No
15	Gas valve	-	-	No
16	Input / Output temperature water sensors	-	-	No
17	Sheet iron component	-	-	No

Table 1. Component characteristics

Table 2 presents the disassembly-related characteristics of the components (viz., disassembly direction, disassembly time, disassembly precedence relationships). Disassembly times of the components follow normal distribution with the mean and standard deviation values given in Table 2. A pictorial view of the precedence relations among the components is represented in Figure 1.

Component Number	Disassembly Time (min.)	Disassembly Direction	Precedence Relationships
1	Norm(0.5, 0.05)	+x	-
2	Norm(4.5, 0.5)	+x	5,6
3	Norm(2.25, 0.25)	-y	1
4	Norm(0.6, 0.05)	+z	3
5	Norm(1.25, 0.25)	-x	1
6	Norm(0.3, 0.03)	+z	1
7	Norm(1.25, 0.25)	-y	1
8	Norm(1.25, 0.25)	+y	1
9	Norm(0.5, 0.05)	-z	12
10	Norm(0.6, 0.05)	-z	14
11	Norm(0.3, 0.03)	-x	10
12	Norm(1.25, 0.25)	-y	1
13	Norm(1.75, 0.25)	+z	9
14	Norm(2.75, 0.25)	+x	16
15	Norm(0.5, 0.05)	-z	9
16	Norm(0.4, 0.05)	-y	13,15
17	Norm(7.5, 0.5)	+x	7

Table 2. Disassembly characteristics

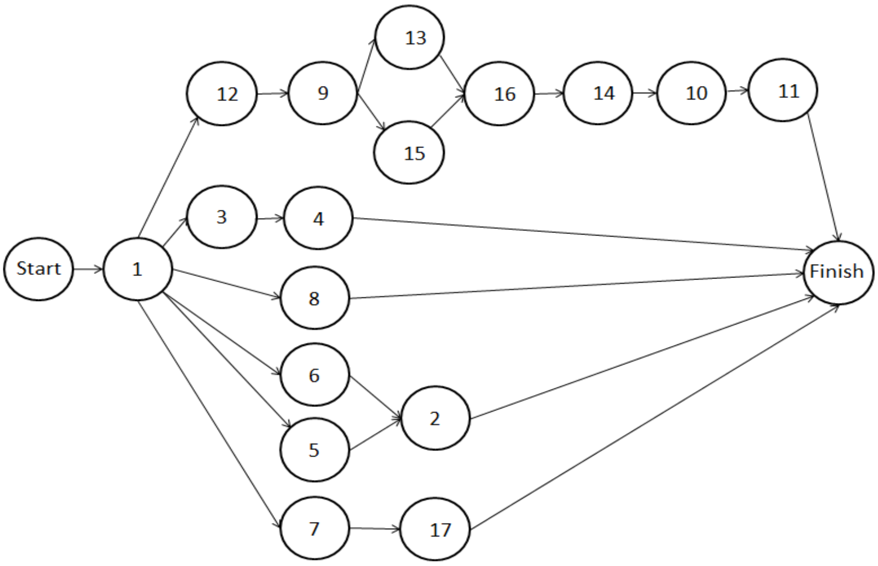

Figure 1. Precedence relations diagram

4.2. Design of Simulation Model

The simulation model of the disassembly line was constructed in ARENA 14.0 simulation software. It was run for one year with one 8-hour shift per day (115,200 minutes). Five replications were carried out for each SA or GA solution. The flow chart of the simulation model is provided in Figure 2.

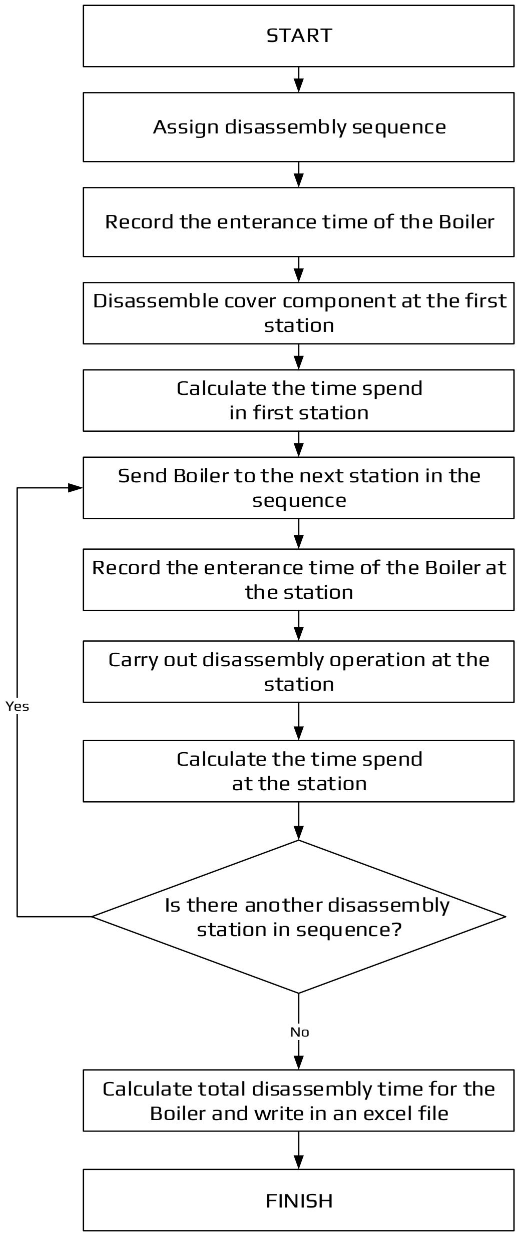

Figure 2. Flow chart of the simulation model

4.3. Design of GA

The flow chart of the GA is provided in Figure 3. Various GA design details including the encoding scheme, fitness evaluation and genetic operators are presented in the following paragraphs.

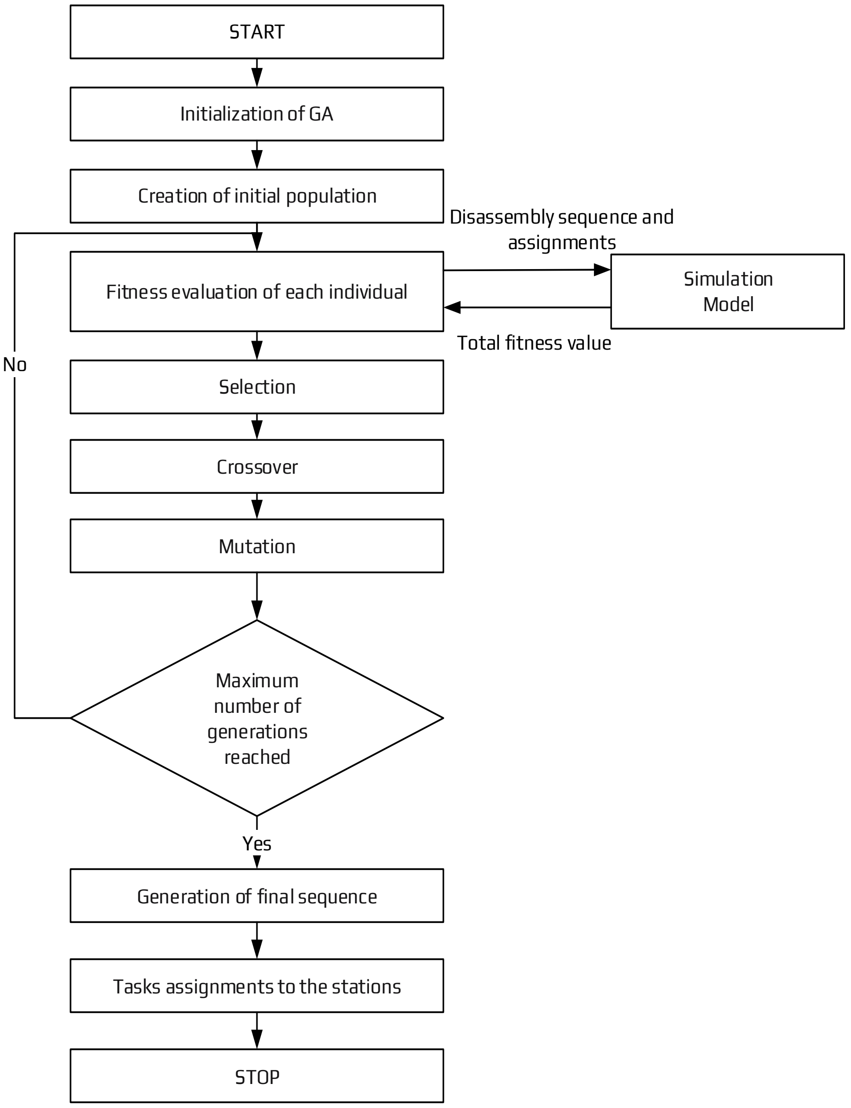

Figure 3. Flowchart of GA-based simulation optimization for disassembly line balancing

Encoding Scheme: The solutions in GAs are encoded as chromosomes based on the features of the problem. Chromosomes are usually constructed by using alphabets, integers, binary digits or other characters. The structure of a chromosome for this study is given in Figure 4. This chromosome involves the permutation of task numbers and represents a possible disassembly sequence.

1	8	2	9	13	15	16	14	10	11	6	5	2	3	4	7	17

Figure 4. Structure of a chromosome

Initial Population: The initial population is constructed by randomly forming chromosomes. The number of chromosomes is equal to population size and the precedence relations among the tasks are considered while constructing the initial population.

Fitness Evaluation: The following performance measures are evaluated in a lexicographic manner for each GA chromosome using the simulation model:

Demand: The components which have high demand disassembled earlier as possible so as to reduce the damage risk during the disassembly operation. This measure is represented as

$$D = \sum_{z=1}^{k} (z.d_{SSz}) \tag{1}$$

where k is the number of components and d_{SSz} illustrate the demand value of the z^{th} component in a sequence (SS_z). This performance measure should be minimized.

Direction: Every reorientation of product increases disassembly time as well as risk of damage. Therefore the number of disassembly direction changes should be minimized. This measure is as follows:

$$DM = \sum_{z=1}^{k} DM_z \quad DM_z = \begin{cases} 1, dm_{SS_z} \neq dm_{SS_{z+1}} \\ 0, otherwise \end{cases} \tag{2}$$

where k is the number of components and dm_{SSz} illustrates the disassembly direction of the z^{th} component in a sequence (SS_z). Disassembly directions are as follows: r_{SSz} = {+1, -1, +2, -2, +3, -3} = {+x, -x, +y, -y, +z, -z}. This performance measure should be minimized.

Revenue: The expensive components should be disassembled at the earliest possible station for minimum component damage (Ilgın 2019). Revenue measure can be represented as follows:

$$R = \sum_{z=1}^{k} (z.r_{SSz}) \tag{3}$$

where k is the number of components and r_{SSz} illustrates the revenue value of the z^{th} component in a sequence (SS_z). This performance measure should be minimized.

Hazardousness: Hazardous components should be disassembled at the earliest instance since the spill of hazardous substances may adversely affect one or more workstations (Ilgın 2019). Hazardousness measure can be represented as follows:

$$H = \sum_{z=1}^{k} (z.h_{SSz}) \quad H_z = \begin{cases} 1, hazardous \\ 0, otherwise \end{cases} \tag{4}$$

where k represents the number of components and h_{SSz} represents whether the z^{th} component in a sequence (SS_z) is hazardous. This performance measure should be minimized.

Idle Time: This measure ensures that the workload among different stations is leveled (Kalaycı and Gupta 2013c). It is presented as follows:

$$C = \sum_{z=1}^{m} (c - t_z)^2 \tag{5}$$

where m is the number of workstations, c is cycle time (maximum time available at each workstation) and t_z is component removal time of component z. Smaller values of C are preferred.

Genetic Operators: The mating population is formed by using roulette wheel selection. One-cut-point crossover is applied after selection. Later, mutation and elitism operators are used and finally chromosomes are repaired with respect to precedence relations in order to attain child chromosomes.

4.4. Design of SA

The flow chart of the SA is provided in Figure 5. Determination of SA parameters has critical importance on the solution quality. Hence, the values of two SA parameters (i.e., initial temperature and cooling rate) are determined by carrying out a design of experiments study (see section 4.5). The performance measures (see equations 1-5) used for the fitness evaluation of GA solutions are also employed for the fitness evaluation of SA solutions.

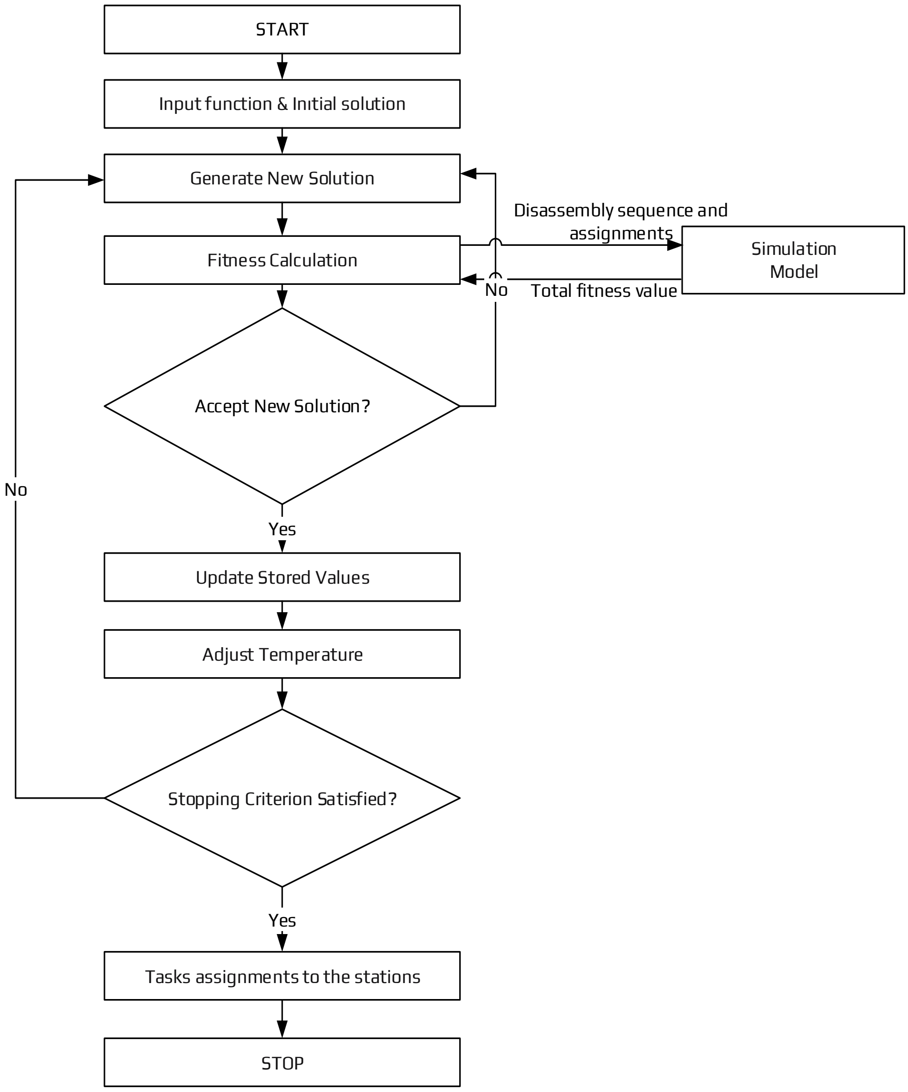

Figure 5. Flowchart of SA-based simulation optimization for disassembly line balancing

4.5. Design of SA and GA Parameter Values Using Full Factor Analysis

The appropriate parameter values for the GAs (e.g., mutation rate, elitism rate) must be decided so as to get better performance of GAs. That is why a full factorial design of experiments study was fulfilled so as to decide the values of five GAs parameters: mutation rate, elitism rate, population size, iteration number and crossover rate. Table 3 represents the parameter levels of GAs. With respect to Table 3, 10 replications for each of experiment require 2,430 experiments (3^5x10). 12,150 simulation replications are needed since 5 replications of the simulation model are carried out for each of experiment. The results of the full factorial design are given in Figure 6.

Parameters	Level 1	Level 2	Level 3
Mutation rate	0.05	0.1	0.2
Elitism rate	0.1	0.2	0.3
Iteration number	250	500	1000
Population size	10	20	30
Crossover rate	0.5	0.7	0.9

Table 3. Factor levels of GA parameters

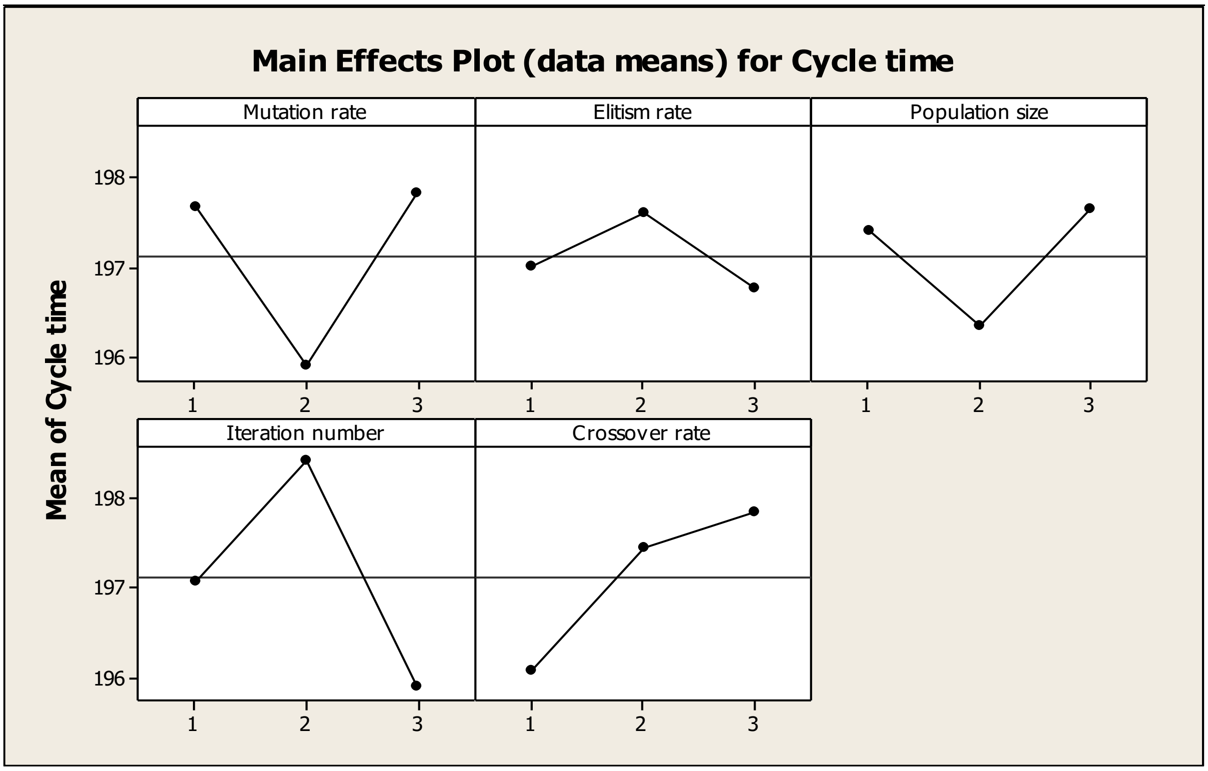

Figure 6. GA Main effects plot for cycle time

The changes in cycle time with respect to changes in various parameters (mutation rate, elitism rate, population size, iteration number and crossover rate) are given in Figure 6. The level at which the cycle time is minimum is preferred for each parameter as given in Table 4.

Parameters	Value
Mutation rate	0.1
Elitism rate	0.3
Iteration number	1000
Population size	20
Crossover rate	0.5

Table 4. GA parameter values

The appropriate values for SA parameters must be decided so as to get better the performance of SA. That is why a full factorial design of experiments study was carried out to decide the values of two SA parameters: initial temperature and cooling rate. Table 5 represents the levels of SA parameters. With respect to Table 5, 10 replications for each of experiment require 250 experiments (5^2x10). 1,250 simulation replications are needed since 5 replications of the simulation model are carried out for each of experiment. The results of the full factorial design are presented in Figure 7.

Parameters	Level 1	Level 2	Level 3	Level 4	Level 5
Initial temperature	100	250	500	750	1000
Cooling rate	0.01	0.025	0.05	0.075	0.1

Table 5. Factor Levels of SA parameters

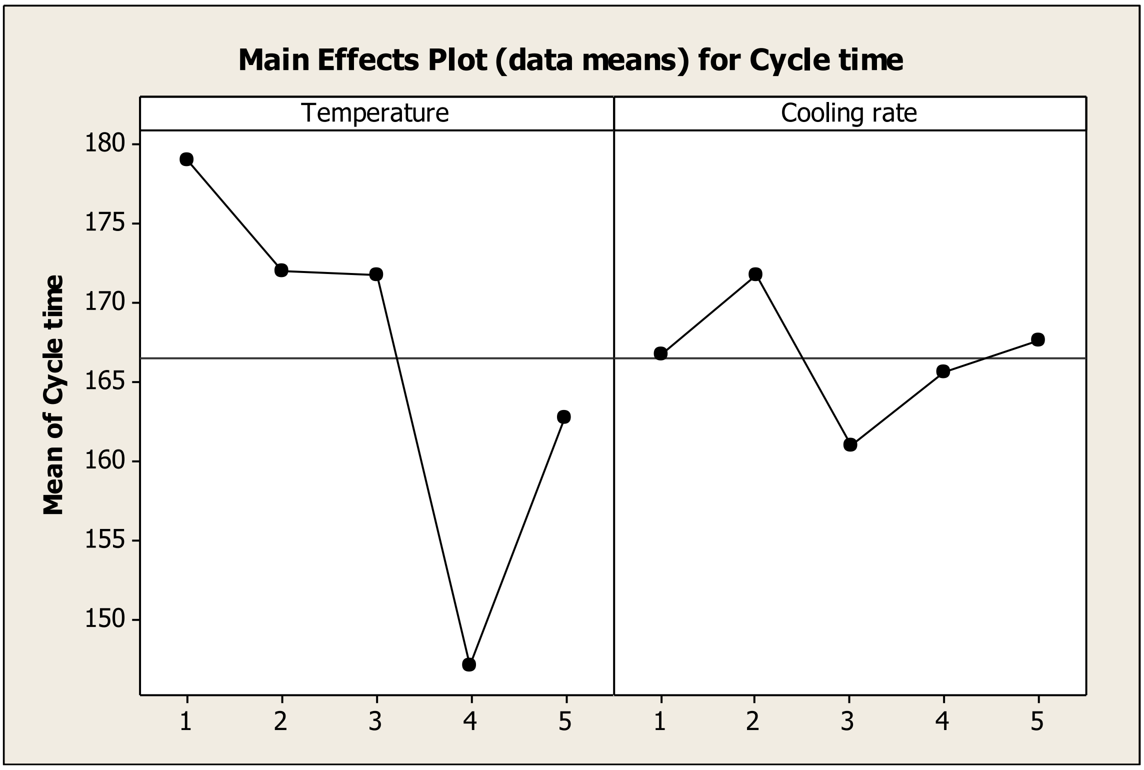

Figure 7. SA Main effects plot for cycle time

The changes in cycle time with respect to changes in various parameters (initial temperature and cooling rate) are given in Figure 7. The level at which the cycle time is minimum is preferred for each parameter as given in Table 6.

Parameters	Value
Initial temperature	750
Cooling rate	0.05

Table 6. SA parameter values

4.6. Results

GAs was operated by considering the parameter values represented in Table 4. The GAs optimization procedure was fulfilled in 604,800s by using a desktop computer with 8 GB RAM and 3.2 GHz Intel Core i5 processor. Regarding the extent of the solution space, converged solution of GAs is gathered in almost a week. This convergence graph is given in Figure 8. With respect to the figure, at first fitness function value is 136 but after 1,000 iterations, the GAs converges to a fitness function value of 122. With iteration number increasing, the GAs converges to a fitness function value of 120. This difference means a %13 positive progress in fitness function compared with the initial solution. Figure 9 represents the chromosome of the converged solution. With respect to the converged solution, the sequence of disassembly tasks is gathered as 1-7-12-9-8-6-17-3-4-13-5-2-15-16-14-10-11.

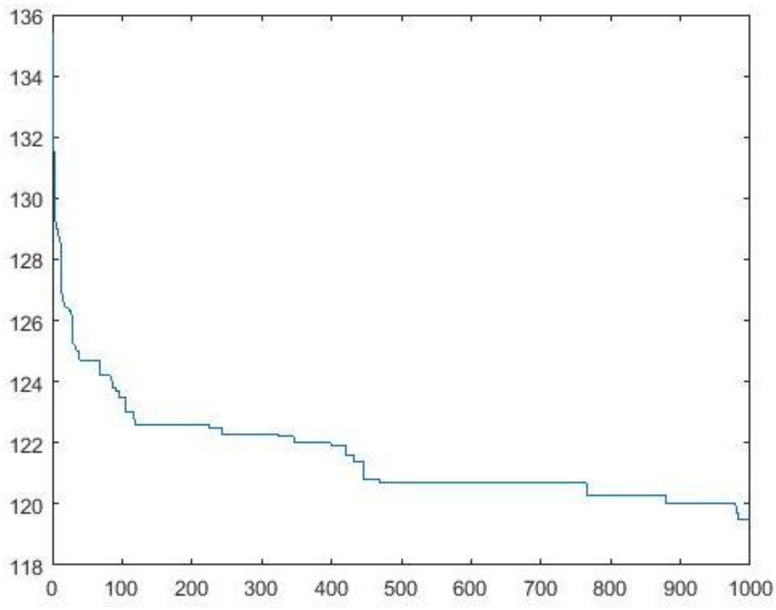

Figure 8. Cycle time convergence graph of GA

1	12	9	13	5	15	16	14	10	11	6	2	3	4	8	7	17

Figure 9. The converged GA solution

Following the determination of the disassembly sequence, the disassembly tasks are assigned to stations. Since the disassembly times are stochastic, an average cycle time is used during the assignment process. Starting with the first task in the sequence, tasks are assigned to stations considering the mean task times and the average cycle time. Whenever it is not possible to assign a task to the current station, a new station is opened. A total of two stations are opened at the end of the assignment process. Table 7 presents the disassembly tasks at each station.

Stations Opened	**Tasks**
Station 1	1, 7, 12, 9, 8, 6, 17, 3, 4, 13, 5, 2
Station 2	16, 14, 10, 15, 11

Table 7. Assignment of tasks using GA-based simulation optimization

SA was also operated by considering the parameter values represented in Table 6. SA optimization procedure was fulfilled in 64,400s with the same computer used as in GA optimization process. Regarding the extent of the solution space, the SA obtained a converged solution in a very small computational time. SA convergence graph is given in Figure 10. With respect to this figure, at first fitness function value is 140 but after 120 iterations, the algorithm converges to a fitness function value of 127. This difference means a %10 positive progress in fitness function gathered compared with the initial solution. Figure 11 represents the converged solution of SA with respect to the converged solution, the sequence of disassembly tasks is gathered as 1-3-12-9-15-6-13-16-14-10-11-5-2-8-7-17-4.

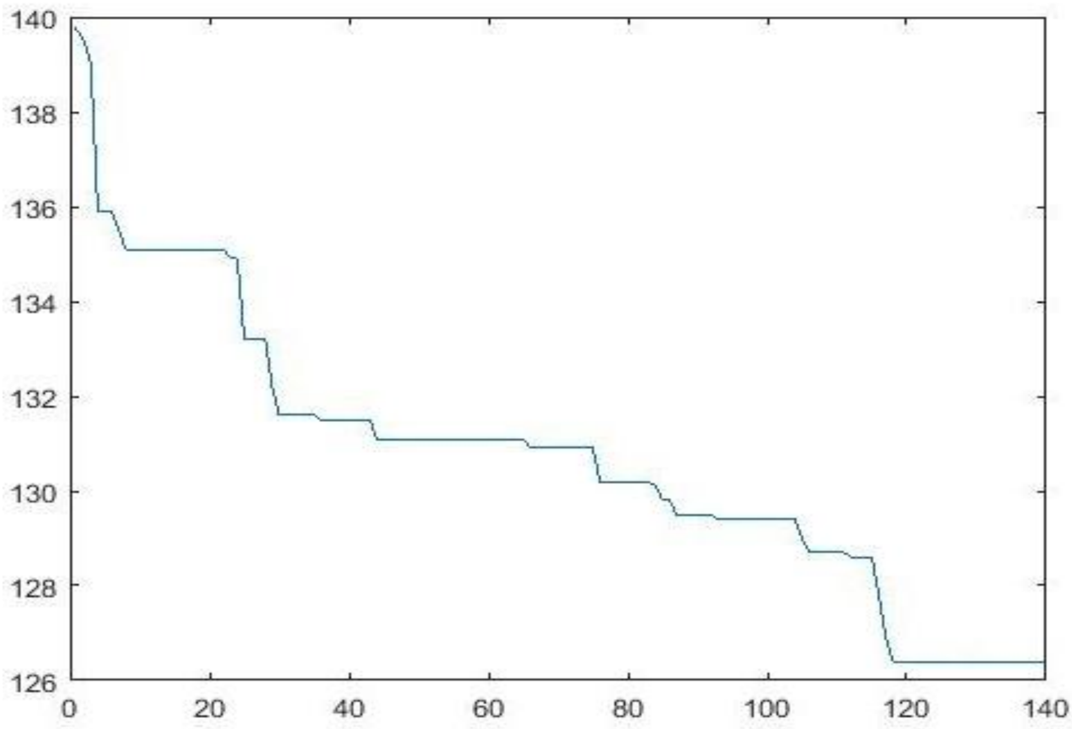

Figure 10. Cycle time convergence graph of SA

1	3	12	9	15	6	13	16	14	10	11	5	2	8	7	17	4

Figure 11. The converged SA solution

Assignment of disassembly tasks to stations is fulfilled using the same procedure with GAs. A total of two stations are opened at the end of the assignment process. Table 8 presents the disassembly tasks at each station.

Stations Opened	Tasks
Station 1	1, 5, 12, 6, 2, 9, 13
Station 2	16, 14, 8, 3, 4, 7, 17, 10, 11

Table 8. Assignment of tasks using SA-based simulation optimization

The most frequently used measures used for the performance evaluation of disassembly line balancing approaches are smoothness index (SI), line efficiency (LE), balance measure (BM), demand measure (D), hazard measure (H), revenue measure (R), disassembly direction measure (DM) and cycle time measure (C) (McGovern and Gupta 2010; Ilgin 2019). Table 9 presents the comparison of GA and SA-based simulation optimization approaches based on those measures. According to Table 9, both approaches have the same performance considering LE. SA outperforms GA in terms of SI, BM and C while GA presents a superior performance on H, D, DM and R.

	LE (%)	SI	BM	H	D	DM	R	C
GA	88.1	15.6	4887	10	4200	9	1890	21.2
SA	88.1	10.2	4746	11	4850	16	2085	17,1

Table 9. Comparison of GA and SA-based simulation optimization

5. Conclusions

Disassembly is an indispensable operation for all product recovery options. The most commonly used layout for disassembly operations is disassembly line mainly due to its efficiency. However, the highest possible efficiency can only be achieved if the disassembly line is balanced. Although there are many disassembly line balancing methodologies, majority of them ignores stochastic issues in disassembly lines. In this study, we developed a metaheuristics-based simulation optimization methodology for the balancing of disassembly lines by considering stochastic disassembly times and product inter-arrival times. The sequence of disassembly tasks together with the task assignments to stations were determined according to the solutions proposed by GA and SA metaheuristics. The disassembly sequence and task assignments proposed by GA were compared with the sequence and task assignments proposed by SA.

The following managerial insights can be obtained from the proposed approach:

In practice, disassembly environment involves many stochastic issues such as stochastic disassembly times and stochastic inter-arrival times of used products. The proposed approach allows decision makers to consider those issues while balancing disassembly lines.

There are some other practical issues in real disassembly lines such as hazardousness, demand and direction change of components. The proposed approach provides decision makers with the opportunity of considering these real life issues.

Limitations of this study can be listed as follows:

If a product with huge number of components were considered, the construction of the simulation model would be very time consuming. In addition, the run-time of the simulation model would be very high.

The metaheuristic algorithms employed in the proposed approach do not guarantee global optimal solutions.

Although GA and SA are powerful metaheuristics to be used in simulation optimization, other metaheuristics such as Tabu search can be used for the development of a simulation optimization-based disassembly line balancing methodology. Various other stochastic issues such line stoppages can also be considered in future studies.

References

Agrawal, S., & Tiwari, M. K. (2008). A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem. International Journal of Production Research. https://doi.org/10.1080/00207540600943985

Altekin, F. T. (2017). A comparison of piecewise linear programming formulations for stochastic disassembly line balancing. International Journal of Production Research. https://doi.org/10.1080/00207543.2017.1351639

Altekin, F. T., & Akkan, C. (2012). Task-failure-driven rebalancing of disassembly lines. International Journal of Production Research. https://doi.org/10.1080/00207543.2011.616915

Altekin, F. T., Bayındır, Z. P., & Gümüşkaya, V. (2016). Remedial actions for disassembly lines with stochastic task times. Computers and Industrial Engineering. https://doi.org/10.1016/j.cie.2016.06.027

Altekin, F. T., Kandiller, L., & Ozdemirel, N. E. (2008). Profit-oriented disassembly-line balancing. International Journal of Production Research. https://doi.org/10.1080/00207540601137207

Ameli, M., Mansour, S., & Ahmadi-Javid, A. (2019). A simulation-optimization model for sustainable product design and efficient end-of-life management based on individual producer responsibility. Resources, Conservation and Recycling. https://doi.org/10.1016/j.resconrec.2018.02.031

Amiri, M., & Mohtashami, A. (2012). Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-011-3802-8

Avikal, S., Jain, R., & Mishra, P. K. (2014). A Kano model, AHP and M-TOPSIS method-based technique for disassembly line balancing under fuzzy environment. Applied Soft Computing Journal. https://doi.org/10.1016/j.asoc.2014.08.002

Avikal, S., Mishra, P. K., & Jain, R. (2013). An AHP and PROMETHEE methods-based environment friendly heuristic for disassembly line balancing problems. Interdisciplinary Environmental Review. https://doi.org/10.1504/ier.2013.054125

Avikal, S., Mishra, P. K., & Jain, R. (2014). A Fuzzy AHP and PROMETHEE method-based heuristic for disassembly line balancing problems. International Journal of Production Research. https://doi.org/10.1080/00207543.2013.831999

Avikal, S., Mishra, P. K., Jain, R., & Yadav, H. C. (2013). A PROMETHEE Method Based Heuristic for Disassembly Line Balancing Problem. Industrial Engineering and Management Systems. https://doi.org/10.7232/iems.2013.12.3.254

Azadivar, F., & Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. International Journal of Production Research. https://doi.org/10.1080/00207540050205154

Bentaha, M. L., Battaiä, O., & Dolgui, A. (2015). An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times. International Journal of Production Research. https://doi.org/10.1080/00207543.2014.961212

Bentaha, M. L., Battaïa, O., & Dolgui, A. (2014). A sample average approximation method for disassembly line balancing problem under uncertainty. Computers and Operations Research. https://doi.org/10.1016/j.cor.2014.05.006

Bentaha, M. L., Battaïa, O., Dolgui, A., & Hu, S. J. (2015). Second order conic approximation for disassembly line design with joint probabilistic constraints. European Journal of Operational Research. https://doi.org/10.1016/j.ejor.2015.06.019

Deniz, N., & Ozcelik, F. (2019). An extended review on disassembly line balancing with bibliometric & social network and future study realization analysis. Journal of Cleaner Production. https://doi.org/10.1016/j.jclepro.2019.03.188

Ding, H., Benyoucef, L., & Xie, X. (2003). Simulation Optimization in Manufacturing Analysis: A Simulation-Optimization Approach Using Genetic Search for Supplier Selection. Proceedings of the 35th Conference on Winter Simulation: Driving Innovation.

Ding, L. P., Feng, Y. X., Tan, J. R., & Gao, Y. C. (2010). A new multi-objective ant colony algorithm for solving the disassembly line balancing problem. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-009-2303-5

Dowsland, K. A., & Thompson, J. M. (2012). Simulated annealing. In Handbook of Natural Computing. https://doi.org/10.1007/978-3-540-92910-9_49

Edis, E. B., Ilgin, M. A., & Edis, R. S. (2019). Disassembly line balancing with sequencing decisions: A mixed integer linear programming model and extensions. Journal of Cleaner Production. https://doi.org/10.1016/j.jclepro.2019.117826

Eglese, R. W. (1990). Simulated annealing: A tool for operational research. European Journal of Operational Research. https://doi.org/10.1016/0377-2217(90)90001-R

Fang, Y., Liu, Q., Li, M., Laili, Y., & Pham, D. T. (2019). Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations. European Journal of Operational Research. https://doi.org/10.1016/j.ejor.2018.12.035

Güçdemir, H., & Selim, H. (2017). Customer centric production planning and control in job shops: A simulation optimization approach. Journal of Manufacturing Systems. https://doi.org/10.1016/j.jmsy.2017.02.004

Gungor, A., & Gupta, S. M. (1999). Issues in environmentally conscious manufacturing and product recovery: A survey. Computers and Industrial Engineering. https://doi.org/10.1016/S0360-8352(99)00167-9

Güngör, Askiner, & Gupta, S. M. (2001). A solution approach to the disassembly line balancing problem in the presence of task failures. International Journal of Production Research. https://doi.org/10.1080/00207540110052157

Güngör, Aşkiner, & Gupta, S. M. (2002). Disassembly line in product recovery. International Journal of Production Research. https://doi.org/10.1080/00207540210135622

Ilgin, M. Ali, & Tunali, S. (2007). Joint optimization of spare parts inventory and maintenance policies using genetic algorithms. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-006-0618-z

Ilgin, Mehmet Ali. (2019). A DEMATEL-Based Disassembly Line Balancing Heuristic. Journal of Manufacturing Science and Engineering, Transactions of the ASME. https://doi.org/10.1115/1.4041925

Ilgin, Mehmet Ali, Akçay, H., & Araz, C. (2017). Disassembly line balancing using linear physical programming. International Journal of Production Research. https://doi.org/10.1080/00207543.2017.1324225

Ilgin, Mehmet Ali, & Gupta, S. M. (2010). Environmentally conscious manufacturing and product recovery (ECMPRO): A review of the state of the art. Journal of Environmental Management. https://doi.org/10.1016/j.jenvman.2009.09.037

Ilgin, Mehmet Ali, & Taşoğlu, G. T. (2016). Simultaneous Determination of Disassembly Sequence and Disassembly-to-Order Decisions Using Simulation Optimization. Journal of Manufacturing Science and Engineering, Transactions of the ASME. https://doi.org/10.1115/1.4033603

Kalayci, C. B., & Gupta, S. M. (2013a). A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-013-4990-1

Kalayci, C. B., & Gupta, S. M. (2013b). Ant colony optimization for sequence-dependent disassembly line balancing problem. Journal of Manufacturing Technology Management. https://doi.org/10.1108/17410381311318909

Kalayci, C. B., & Gupta, S. M. (2013c). Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2013.06.067

Kalayci, C. B., & Gupta, S. M. (2013d). Balancing a sequencedependent disassembly line using simulated annealing algorithm. Applications of Management Science. https://doi.org/10.1108/S0276-8976(2013)0000016008

Kalayci, C. B., & Gupta, S. M. (2014). A tabu search algorithm for balancing a sequence-dependent disassembly line. Production Planning and Control. https://doi.org/10.1080/09537287.2013.782949

Kalayci, C. B., Hancilar, A., Gungor, A., & Gupta, S. M. (2015). Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm. Journal of Manufacturing Systems. https://doi.org/10.1016/j.jmsy.2014.11.015

Kalayci, C. B., Polat, O., & Gupta, S. M. (2016). A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. Annals of Operations Research. https://doi.org/10.1007/s10479-014-1641-3

Koc, A., Sabuncuoglu, I., & Erel, E. (2009). Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph. IIE Transactions (Institute of Industrial Engineers). https://doi.org/10.1080/07408170802510390

Li, Z., Çil, Z. A., Mete, S., & Kucukkoc, I. (2020). A fast branch, bound and remember algorithm for disassembly line balancing problem. International Journal of Production Research, 58(11), 3220-3234.

Lin, Y. K., & Lin, H. C. (2015). Bicriteria scheduling problem for unrelated parallel machines with release dates. Computers and Operations Research. https://doi.org/10.1016/j.cor.2015.04.025

Liu, J., Zhou, Z., Pham, D. T., Xu, W., Yan, J., Liu, A., … Liu, Q. (2018). An improved multi-objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-018-2183-7

Mattila, V., & Virtanen, K. (2015). Ranking and selection for multiple performance measures using incomplete preference information. European Journal of Operational Research. https://doi.org/10.1016/j.ejor.2014.10.028

McGovern, S. M., & Gupta, S. M. (2007). Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem. International Journal of Production Research. https://doi.org/10.1080/00207540701476281

McGovern, Seamus M., & Gupta, S. M. (2006). Ant colony optimization for disassembly sequencing with multiple objectives. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-005-0037-6

McGovern, Seamus M., & Gupta, S. M. (2007). A balancing method and genetic algorithm for disassembly line balancing. European Journal of Operational Research. https://doi.org/10.1016/j.ejor.2005.03.055

Ozcan, Y. A., Tànfani, E., & Testi, A. (2017). Improving the performance of surgery-based clinical pathways: a simulation-optimization approach. Health Care Management Science. https://doi.org/10.1007/s10729-016-9371-5

Özceylan, E., Kalayci, C. B., Güngör, A., & Gupta, S. M. (2019). Disassembly line balancing problem: a review of the state of the art and future directions. International Journal of Production Research. https://doi.org/10.1080/00207543.2018.1428775

Özceylan, E., & Paksoy, T. (2013). Reverse supply chain optimisation with disassembly line balancing. International Journal of Production Research. https://doi.org/10.1080/00207543.2013.784405

Özceylan, E., & Paksoy, T. (2014a). Fuzzy mathematical programming approaches for reverse supply chain optimization with disassembly line balancing problem. Journal of Intelligent and Fuzzy Systems. https://doi.org/10.3233/IFS-130875

Özceylan, E., & Paksoy, T. (2014b). Interactive fuzzy programming approaches to the strategic and tactical planning of a closed-loop supply chain under uncertainty. International Journal of Production Research. https://doi.org/10.1080/00207543.2013.865852

Paksoy, T., Güngör, A., Özceylan, E., & Hancilar, A. (2013). Mixed model disassembly line balancing problem with fuzzy goals. International Journal of Production Research. https://doi.org/10.1080/00207543.2013.795251

Seidi, M., & Saghari, S. (2016). The balancing of disassembly line of automobile engine using Genetic Algorithm (GA) in fuzzy environment. Industrial Engineering and Management Systems. https://doi.org/10.7232/iems.2016.15.4.364

Tasoglu, G., & Yildiz, G. (2019). Simulated annealing based simulation optimization method for solving integrated berth allocation and quay crane scheduling problems. Simulation Modelling Practice and Theory. https://doi.org/10.1016/j.simpat.2019.101948

Tuncel, E., Zeid, A., & Kamarthi, S. (2014). Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning. Journal of Intelligent Manufacturing. https://doi.org/10.1007/s10845-012-0711-0

Wang, K., Li, X., & Gao, L. (2019). Modeling and optimization of multi-objective partial disassembly line balancing problem considering hazard and profit. Journal of Cleaner Production. https://doi.org/10.1016/j.jclepro.2018.11.114

Wang, S., Guo, X., & Liu, J. (2019). An efficient hybrid artificial bee colony algorithm for disassembly line balancing problem with sequence-dependent part removal times. Engineering Optimization. https://doi.org/10.1080/0305215X.2018.1564918

Xiao, S., Wang, Y., Yu, H., & Nie, S. (2017). An entropy-based adaptive hybrid particle swarm optimization for disassembly line balancing problems. Entropy. https://doi.org/10.3390/e19110596

Yin, P. Y., Wu, T. H., & Hsu, P. Y. (2017). Simulation based risk management for multi-objective optimal wind turbine placement using MOEA/D. Energy. https://doi.org/10.1016/j.energy.2017.09.103

Zeng, Q., Diabat, A., & Zhang, Q. (2015). A simulation optimization approach for solving the dual-cycling problem in container terminals. Maritime Policy and Management. https://doi.org/10.1080/03088839.2015.1043362

Zhang, Z., Wang, K., Zhu, L., & Wang, Y. (2017). A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2017.05.053

Zhou, Y., Guo, X., & Li, D. (2020). A dynamic programming approach to a multi-objective disassembly line balancing problem. Annals of Operations Research, 1-24.

Zhu, L., Zhang, Z., & Wang, Y. (2018). A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation. International Journal of Production Research. https://doi.org/10.1080/00207543.2018.1471238

Wang, K., Li, X., Gao, L., Li, P., & Sutherland, J. W. (2021). A Discrete Artificial Bee Colony Algorithm for Multiobjective Disassembly Line Balancing of End-of-Life Products. IEEE Transactions on Cybernetics, (Articke in press).