Mazandaran University of Science and Technology

Available online at www.iiec2017.com

13th International Conference on Industrial Engineering

(IIEC 2017)

Iran Institute of Industrial Engineering

Balancing and sequencing U-shaped mixed model assembly line problem considering setup times between tasks and demand ratio-base in dynamic environments

Masoud Rabbani [a,*], Reyhaneh Siadatian [a], Neda Manavizadeh [b]

[a] *School of Industrial Engineering, College of Engineering, University of Tehran, P.O.Box: 11155-4563, Tehran, Iran*
Tel: +9821-88021067/ Fax: +9821-88013102

[b] *Department of Industrial Engineering, KHATAM University, Tehran, Iran*

Abstract

In mixed model assembly lines, several products with high similarity are simultaneously produced on an assembly line as per the sequence without any additional setup times between the models. This paper concentrates on mixed model assembly line balancing and sequencing type-II problem. The problem has some particular features such as U-shaped workstations and setup times between two consecutive tasks in dynamic periods wherein each period also affects the flowing period. This research intends to reduce cycle time and inventory costs. To this end, a non-dominated sorting genetic algorithm (NSGA-II) is used to solve the problem. Small scales of the problem are solved using both NSGA-II and GAMS software to evaluate the effectiveness of NSGA-II, and the obtained outcomes are compared. The computational results indicate that the NSGA-II is capable of providing high-quality solutions for small scales of the problem. Finally, conclusion and future research are provided.

Keywords:
Balancing, sequencing, mixed model U-shaped, setup times, dynamic

Introduction

An assembly line is generally used for mass production wherein a flow of units moves continuously through the workstations placed as a sequence. According to the diversity of products, different types of assembly lines are identified. Mixed model assembly line (MMAL) is one of them, which is used in assembly line [1]. In mixed model assembly lines, there is a high rate of similarity among the models so that setup times for stations can be ignored from one model to another one [2]. Hence, several types of products can be simultaneously assembled to the line as per the sequence without any additional setup times and, thereby, various demands of customers can be met. This increases the line flexibility and this can play a crucial role in the promotion of systems since increased flexibility of systems is naturally of high importance in today's highly competitive manufacturing environment in order to fulfill customers' desires [3].

Two important problems occurred in the mixed model assembly line regardless of the line shape. The first one is line balancing that refer to the feasible assignment of tasks to workstations in such a path that some performance measures are optimized according to precedence relationship among the tasks. The second one is model sequencing which is aimed at selecting the order of the products entering the assembly line wherein different models will be produced [4]. This research considered line balancing and sequencing of model simultaneously.

In this research, assembly line problem type-II is used. This means that the number of workstations is known but the cycle time is unknown; and the problem aims to minimize the cycle time. However, problem in type-I, the cycle time is known and the number of workstations is unknown.

U-shaped line is one of the useful line configurations used in assembly line problem. The U-shaped assembly line has become an alternative for assembly production systems because the operator can perform more than one task in different locations of the assembly line. In this line, the entry and exit of these lines lie at the same position; indeed, the products enter the U-shaped assembly line at the front-side and then exit from the backside of the line [5].

Operators in crossover workstations can perform tasks from both front and back sides of the U-line. In this line, if a task is to be performed in front of the line, either the task should possess no predecessors or all its predecessors should be already assigned and completed. However, if a task is to be performed in back of the line, either the task should have no successors or all its successors must have been performed and completed.

Lean manufacturing and Just-In-Time technology are among the advantages of U-shaped over straight line. The most important benefits can be mentioned as increased volume flexibility, increased visibility, operator flexibility, not exceeding the number of workstations required for a U-shaped than that required one on a straight line, teamwork, and the short distance to return the defective product [6].

In this research, mixed model assembly line balancing and sequencing U-shaped line are used.
The remainder of this study is organized as follows: section 2 describes the related literature. The problem definition is explained in section 3. In section 4, the mathematical model is presented. The methodology for tackling the problem is presented in section 5. Section 6 focuses on the numerical results and discussion. Finally, the study ends up with the conclusion in section 7.

2. Literature review

Karimi et al. [7] reviewed the models and algorithms of capacitated and incapacitated single-level lot sizing with exact and heuristic approaches.

Zhang et al. [8] applied a non-dominated sorting genetic algorithm for mixed-model assembly line balancing problem based on demand ratio to minimize cycle time and human resource cost.

Sivasankaran and shahabudeen [9] reviewed assembly line balancing problems and classified them based on the number of models, the nature of task times, and the type of assembly line until 2013. They observed that mixed model deterministic U-line problem type-II was one of the categories that was not solved until that time. It is worth mentioning that this model is studied in this research.

Farkhondeh et al. [10] used goal programming for multi-objective problems where the assembly line operation efficiency was the most important objective for U-line balancing problem. Finally, they solved the problem through data envelopment analysis (DEA) approach.

Hamzadayi and Yildiz [4] solved mixed model assembly U-line balancing and sequencing with parallel workstations and zoning constraints by a priority-based genetic algorithm. Their objectives were to minimize the number of workstations and smooth the workload between and within workstations.

Hamta et al. [11] proposed a hybrid particle swarm optimization (PSO) algorithm for a single model assembly line balancing problem with flexible operation time of tasks, setup times between consecutive tasks and learning effect. They aimed to minimize the cycle time, minimize the total equipment cost, and minimize the smoothness index.

Dong et al. [12] developed a simulated annealing-based algorithm for line balancing and model sequencing problem in stochastic environment. They minimized the expected work overload time for a given cycle time and number of workstations.

Ogan and Azizoglu [13] solved U-shaped assembly line balancing problem with a branch and bound method where each task use a specified set of equipment and each type of equipment has a specified cost. They concluded that this algorithm can solve medium-sized problems in suitable time.

Alavidoost et al. [14] developed a novel bi-objective triangular fuzzy mixed-integer linear programming model to minimize the number of workstations and cycle time. They applied the model in both straight and U-shape assembly lines.

3. Problem Definition

According to the above literature review, a mixed model assembly line balancing and sequencing U-shaped line with a known number of workstations have been used in this paper. There are two objectives to be achieved: the minimization of cycle time and inventory costs. In this problem, the demands of different models are not the same and the cycle time is calculated based on the demands of different models while the demands of all the models during the planning period are known.

When demand rate turns out to be known, but it is not necessarily stable from one period to another, the emergent problem is referred to as dynamic programming. In a dynamic environment, demand volume or type undergoes some changes in different periods. Here, the debatable topic for managers is the determination of production rate in each period from the planning horizon. One of the measures management can take to plan the fluctuating demand is to increase the inventory levels in the periods with fewer demands so that the management can

The idea is that that has been less demand increase cope with a higher number of demands in the future. The production stages of a model must be completed in full and the next period does not start until the production process comes to an end. This study is an attempt to consider the layout of a mixed model assembly line as a U-shaped line in dynamic periods. Dynamic periods are used to determine all variables in different periods to reach efficient solutions where each period influences the next period.

In each period, there are several similar models that should be assigned to their sequential positions; in this way, each of them is produced as per their sequential relevance. Indeed, there are several tasks that should be assigned to each workstation according to the sequence position pertaining to the workstation. In this condition, a setup time is considered between some tasks where these tasks have been performed consecutively without any pause inside the same workstation.
The common assumptions of the problem are listed below:

- Assembly line balancing problem type-II is used.
- The number of workstations are known.
- Precedence diagrams of all model types are known and merged into one combined diagram.
- Task performance's times of each product are known.
- Parameters in the model are considered as deterministic.
- Setup times between models are supposed as insignificant.
- We have a U-line which can also be worked on front side and back side of it.

4. Mathematical Model

Notation and parameters:

i	Index of tasks i= 1, 2, …, I
j	Index of sequence $j = 1, 2, ..., J$
k	Index of workstation k= 1, 2, …, K
s	Sequence position index inside a workstation s= 1, 2, …, S
t	index of period t = 1, 2, ..., T
M	Index of similar models m= 1, 2, …, M
Pr_i	Set of immediate predecessors of task i
Su_i	Set of immediate successors of task i
d_{mt}	expected demand ratio of model m in period t
t_{im}	the time required to perform task i on model m in period t
T_{mt}	time of doing all tasks in model m in period t
C_{mt}	Theoretical minimum cycle time for model m in period t
MT_t	minimum tasks which should assign to each station in period t
q_{mt}	unit production cost of model m produced in period t
r_{mt}	unit holding cost of model m at the end of period t
E_{jmt}	upper bound on the production of model m in sequence j in period t $E_{mt} = \sum_{j=1}^{J} E_{mjt} = \sum_{t'=t}^{T} d_{mt'}$
st_{bi}	Setup time when task i is operated after task b inside the same workstation

Decision variables:

x_{ikst}	1 if the task i is assigned to station k in period t, 0 otherwise
y_{jmt}	1 if the model m is performed in sequence j in period t, 0 otherwise
o_i	A zero-one variable which determines whether or not constraints 6 or constraints 7 is satisfied
w_{bik}	1 if task b is operated immediately before task i at workstation k in the same or in the next cycle
I_{mt}	inventory of model m at the end of period t
NP_{jmt}	Number of production of model m in sequence j in period t

Mathematical Model:

$$\text{Min } z_1 = \sum_{m=1}^{M} \left\{ d_m \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i=1}^{I}\sum_{s=1}^{S}\sum_{t=1}^{T} t_{im} x_{ikst} y_{jmt} \right\} + \sum_{i=1}^{I}\sum_{b=1,b\neq i}^{I}\sum_{k=1}^{K} st_{bi} w_{bik} \quad (1)$$

$$\text{Min } z_2 = \sum_{t=1}^{T}\sum_{m=1}^{M}\left(q_{mt}\sum_{j=1}^{J} NP_{jmt} + r_{mt} I_{mt} \right) \quad (2)$$

Subject to:

$$\sum_{k=1}^{K}\sum_{s=1}^{S} x_{ikst} = 1 \qquad \forall i, \forall t \quad (3)$$

$$\sum_{m=1}^{M} y_{jmt} = 1 \qquad \forall j, \forall t \quad (4)$$

$$\sum_{j=1}^{J} y_{jmt} \geq 1 \qquad \forall m, \forall t \quad (5)$$

$$\sum_{k'=1}^{K} k' \, x_{bk's't'} - \sum_{k=1}^{K} k \, x_{ikst} \leq M \, o_i \qquad b \in Pr_i, \forall i, \forall t' \leq t, \forall s, \quad \forall k' \leq k \quad (6)$$

$$\sum_{k'=1}^{K} k' \, x_{nk's't'} - \sum_{k=1}^{K} k \, x_{ikst} \leq M \, (1 - o_i) \qquad n \in Su_i, \forall i, \forall t' \leq t, \forall s, \quad \forall k' \leq k \quad (7)$$

$$\sum_{i=1}^{I}\sum_{s=1}^{S} x_{ikst} t_{im} + \sum_{i=1}^{I}\sum_{b=1,b\neq i,k=k'}^{I} st_{bi} w_{bik} \leq C_{mt} \qquad \forall m, \forall k, \forall t \quad (8)$$

$$C_{mt} = \frac{T_{mt}}{d_{mt}} \qquad \forall m, \forall t \quad (9)$$

$$\sum_{i=1}^{I}\sum_{s=1}^{S} x_{ikst} \geq MT \qquad \forall k, \forall t \quad (10)$$

$$\sum_{i=1}^{I} x_{ikst} \leq 1 \qquad \forall k, \forall s, \forall t \quad (11)$$

$$\sum_{i=1}^{I} x_{ik,s+1,t} - \sum_{i=1}^{I} x_{ikst} \le 0 \quad (12)$$
$$\forall k, \forall s = 1,\dots,S-1, \forall t$$

$$x_{bkst} + x_{ik,s+1,t'} \le 1 + w_{bik} \quad (13)$$
$$\forall k, \forall s = 1,\dots,S-1, \forall t' \ge t,$$
$$\forall (b,i) | (b \ne i) \wedge (i \notin Pr_b)$$

$$NP_{jmt} \le y_{jmt} E_{jmt} \quad (14)$$
$$\forall j, \forall m, \forall t$$

$$I_{m,t-1} - I_{mt} + \sum_{j=1}^{J} NP_{jmt} = d_{mt} \quad (15)$$
$$\forall m, \forall t > 1$$

$$I_{m,0}, I_{m,T} = 0 \quad (16)$$

$$x_{ikst}, y_{jmt}, w_{bik} \in \{0,1\}, \quad (17)$$
$$I_{mt}, NP_{mt} \ge 0$$

The first objective (1) of the model is to minimize the cycle time of the assembly line based on the demand ratio of each model during the planning period and to minimize the setup time between two successive tasks inside the same workstation [8]. The second objective (2) is to minimize the production and holding costs [7].

Constraint (3) ensures that each task is assigned to a single workstation during one sequence inside a workstation in each period [11]. Constraint (4) guarantees that only one product is assigned to each position in a sequence in each period. According to constraint (5), each model should be produced in at least one sequence in each period. Constraint (6-7) ensures that the precedence conditions are met. This means that before the assignment of task *i* to station *k*, all the predecessors/ successors must have been assigned to either workstation *k* or other workstations.

Constraint (8) shows that every workstation processing time with the desired setup times doesn't exceed the cycle time [15]. Equation (9) represents a theoretical definition of the cycle time in model m [16]. Constraint (10) guarantees that the minimum number of tasks in each period should be greater than MT in each station [16].

Constraint (11) ensures that there will be at most one assigned task in each sequence position inside each workstation during each period [11]. Equation (12) guarantees that the tasks should be assigned to the increasing sequence positions in the schedule of every workstation [15]. Constraint (13) computes variable w_{bik}. It is equal to 1 whenever task *b* is assigned to position *s* and task *i* is assigned to position *s*+1 in the schedule of workstation *k* [11].

Constraint (14) shows that the number of products is contingent on the production rate. This means that if no production is fulfilled ($y_{jmt} = 0$), the number of products is zero while the number of products can go beyond its capacity if production is fulfilled [7].

Constraint (15) represents inventory balancing constraint that indicates the conditions for different periods to be dynamic. In these circumstances, the production rate in each period is such that it can meet the demand of that period because the demand for different periods varies. If there is still some produced products available after the fulfillment of the period demand, the remaining balance of that period is used at the beginning of the next period where the required cost for its maintenance has been considered. Since no shortage is allowed, the initial balance is zero. Finally, one of the objectives of this problem is to direct the production in such a way that the cost of production and inventory maintenance can be minimized [17].

Equation (16) ensures that the inventory at the beginning of period 1 and at the end of the last period is equal to zero without loss of generality [7].

4. Methodology

Multi-objective problems are present in the majority of orders in today's studies. Attempt to solve a multi-objective problem seems to be more complete and sensible in many issues (Deb et al., 2000). A general single objective problem is defined as Max or Min $f(x)$ while a general multi-objective problem is represented as Max or Min $F(x) = [f_1(x), f_2(x), \dots, f_k(x)]$. In multi-objective issues, the main objective is to discover great bargains rather than a single solution; thus, the documentation of the optimum solution is introduced as Pareto Optimum. In Pareto dominance, vector ´u` dominates vector ´v` if and only if ´u` is partially worse than ´v`.

According to the related literature review, genetic algorithm outperforms other algorithms. An algorithm solution designed to discover logical Pareto solutions is based on non-dominated sorting genetic algorithm (NSGA-II). In this section, this metaheuristic approach is presented to solve the problem. The steps involved in this algorithm are schemed as follows:

4.1. Initial population

The initial population is created randomly and each individual is represented through four chromosomes. The first chromosome, which is shown below as the number of periods, includes some rows and columns according to the number of tasks. This indicates which task should be assigned to which workstations. For example, there are two rows that show the number of periods and 4 tasks which should be assigned to 2 workstations in this matrix. In this matrix, the row numbered 2 and column numbered 3 are assigned to the task numbered1 which means that the task number 1 is assigned to workstation number 3 in period 2.

$$X1 = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

The second chromosome shows an ordered sequence of all tasks, which is assigned to the same workstations that have been obtained from the first chromosome. Once more, each row indicates periods and each column indicates tasks. For example, the tasks numbered 1, 3, and 4 are assigned to the workstation numbered 1 in the first chromosome in period 2. In addition, the sequence of these tasks is randomly operative in the workstation numbered 1 according to the precedence diagram shown in the matrix below. It is indicated that the task numbered 1 is performed in the first sequence in workstation 1 in period 2 and the task numbered 4 is performed in the second sequence. Finally, the task numbered 3 is performed in the third sequence.

$$X2 = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 1 & 3 & 2 \end{bmatrix}$$

The third chromosome includes variable y_{jmt}. The number of periods, includes some rows in this matrix and columns is indicated sequence of models. Each component of this matrix shows what model should be performed in each sequence in each period.

$$Y = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \end{bmatrix}$$

The fourth chromosome contains NP_{mt} variable. It is a 3D matrix with sequence, model, and period dimensions whose elements are filled according to the third chromosome and E_{jmt} parameter. This indicated the production number of model *m* in sequence *j* and period *t*.

In this algorithm, the fitness function is the same as the objective functions; however, the constraints unmet in population representation as a penalty have been added to it. According to the aforementioned illustrations, all the variables can be calculated.

4.2. Crossover

The single point crossover is used for the first chromosome and divides each of the parents into two parts (head and tail). This point is generated randomly. The recombination of parent one with parent two generates new offspring. The first offspring keeps the head part of the first parent and the tail part of the first offspring is filled with all missing tasks that are transferred to the second parent in order. Hence, the head of the second offspring is built based on the head part of the second parent; and the tail part is filled with all missing tasks that are transferred to the first parent in order.

In the following, the second chromosome is randomly generated after the conduct of a single point crossover on the first chromosome because the second chromosome is dependent on first chromosome.

Similarly, the single point crossover is used for the third chromosome as well. Since the fourth chromosome is dependent on the single point crossover, it is randomly generated after doing crossover on the third chromosome.

4.3. Mutation

In this algorithm, two types of mutations were performed on the first and third chromosomes. In each mutation, one of the two methods is randomly selected and applied to the matrixes. After the implementation of mutation on the first and third chromosomes, the second and fourth chromosomes will be randomly calculated according to the amount of the first and third chromosomes.

The first one is mutation swap where two columns are randomly selected for the new offspring. The locations of these columns are replaced in the chromosome and the other columns of the parent enter unchanged in the offspring matrix. For example, swap mutation in columns 1 and 2 results in the new X1 as follows:

$$X1_{swap} = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 1 & 1 & 1 \end{bmatrix}$$

This mutation for matrix Y is the same as the example of matrix X1.

The next mutation operation is reversion wherein two columns are selected randomly for the new offspring and, then, these columns plus their between columns are arranged in order from end to first. For example, if columns 1 and 4 are selected for X1, the new X1 will be as follows:

$$X1_{revesion} = \begin{bmatrix} 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

Crossover and mutation are used to define how the next generation is created. The strategy illustrative of which individuals will be stay in the population and which ones will be replaced. The individuals of new generation may be created by the before generation and the offspring produced by crossover and mutation.

6. Numerical Results

The performance of the NSGA-II is investigated and the related results are analyzed. The algorithm is coded in MATLAB R2013b and run on Intel CORE i7 2.30 GHz on personal computer with 6 GB RAM.

A numerical example is used to validate the model by GAMS 23.5 software and the results of NSGA-II algorithm are compared by GAMS software in four small scales of the problem. In this example, the combined precedence diagram can be shown as [19] in Figure 1, which is known to us with 20 tasks. For small-sized problems, only some of these tasks are considered. There are 7 sample test problems of different sizes in small-sized problems and 6 sample test problems of different sizes for large-sized problems, which should be performed by NSGA-II.

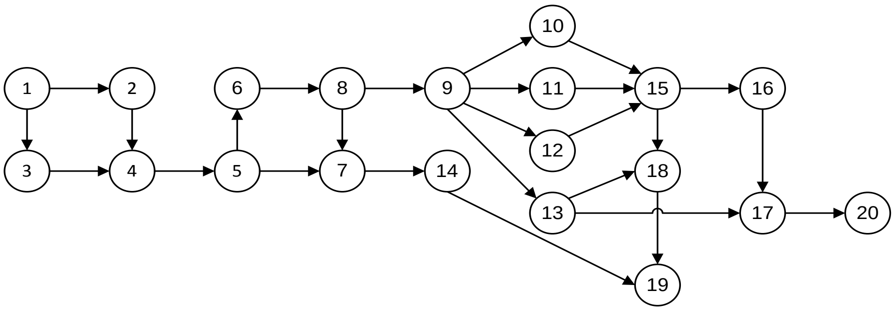

Figure 1- Precedence diagram for the numerical example

In NSGA-II algorithm, test problems were classified in small-sized and large-sized categories with respect to the number of sequences, tasks, workstations, periods, and the demand of each model in each period according to the number of models present in them, respectively. The problem results are shown in table 1 and table 2. In table 1, the exact results of the proposed model for the four small-sized problem obtained by GAMS software have been compared with those obtained by NSGA-II. The optimal obtained sequence and balance for each period is shown in "{}". Number of production of each model in each sequence is shown next to the each model and each sequence is separated from another one with ",". Finally, the tasks assign to each workstation is separated from another workstation with "/".

General parameters of the algorithm are used as follows:

- Number of population size for NSGA-II algorithm is set to 100.
- Experiments for NSGA-II algorithm is repeated 12 times.
- Maximum number of iteration in each run of algorithm is considered 100.
- Crossover rate and mutation rate are in order 0.7 and 0.4.

7. Conclusion

In this paper, we investigated the balancing of tasks in each workstation and simultaneous sequencing of models in dynamic environments where each period also affects the succeeding period. In addition, U-shaped workstations and setup times between two consecutive tasks were considered. The objectives of this problem were to minimize the cycle time and minimize inventory costs when the number of workstations is known. For this purpose, a non-dominated sorting genetic algorithm (NSGA-II) was used to solve the problem. This problem in small scales was solved using both the NSGA-II and GAMS software to evaluate the effectiveness of the NSGA-II. The obtained outcomes were then compared. The results showed that the NSGA-II is capable of providing high-quality solutions with an acceptable speed for the problem in small scales.

For future research, balancing and sequencing U-shaped mixed model assembly line fuzzy problem with setup times between tasks and demand ratio-base in dynamic environments can be used for unknown number of workstations in unknown cycle time.

Table 1 results for the small size problems

Seq. size	No of tasks	No of WSs	No of periods	Exact method		NSGA-II		Optimal sequence and balance	Demand of each model
				Z_1	Z_2	Z_1	Z_2		
3	4	2	2	568	2040	573.2	2292	{A,4A,4B 2/1,3,4} {3A ,4B,A 2/1,3,4}	4,4 4,4
4	5	2	2	914	2040	9886	4056	{A,2A,4B,A 5/1,3,4,2} {A,2A,A,4B 5/1,3,4,2}	4,4 4,4
5	6	2	2	1298	2040	1460	5064	{4B,A,A,A,A 5/6,1,3,4,2} {A,A,4B,A,A 5/6,1,3,4,2}	4,4 4,4
6	7	2	2	1696	2040	1953.8	6072	{3B,3B,A,A,A,A 3/4,5,6,7,1,2} {B,A,A,A,B,A 3/4,5,6,7,1,2}	4,4 4,4
7	8	3	2	-	-	2359.2	3279.6	{2A,A,A,A,4C,A,5B	4,5,4

								2/6,7,8,1,3,4/5} {A,A,C,4B,C,C,2C 1,2,5,6,7,8/3/4}	4,4,5
8	9	3	2	-	-	3215	3686.4	{5A,C,C,C,C,5B,C,C 3/9/1,4,2,5,7,6,8} {B,5A,C,B,B,B,C,C 2/5,1,3,7,8,9,4/6}	5,5,4 5,4,5
9	10	3	2	-	-	4154	3937.2	{6A,C,C,C,C,C,C,5B,C 1,6,3,7,8,9,10,4/5/2} {C,C,B,B,A,5A,B,B,B 9/6/1,3,4,2,10,5,7,8}	6,5,4 6,4,5

Table 2 results for the large size problems

Seq. size	No of tasks	No of WSs	No of periods	NSGA-II		Optimal sequence and balance	Demand of each model
				Z_1	Z_2		
15	13	5	3	17750	9542.4	{7C,6A,4D,B,B,B,B,B,B,B,B,B,B,B,B 9/11/12/7/10,1,13,3,4,2,5,6,8} {A,A,4C,A,A,A,A,A,A,A,A,A,A,4D,D 2/5/7/6,8,9,13,12,11,1,3,4/10} {B,B,B,B,C,C,C,C,6D,B,B,C,D,B,B 4/13/3/6/5,7,8,9,11,12,10,1,2}	6,5,7,4 6,7,4,5 6,4,5,7
17	14	5	3	21767.4	10833.6	{D,D,D,D,6A,D,D,7C,D,D,D,D,D,5B,D,D 3,2/10/1,5,7,6,8,9,11,13,12/14/4} {6A,C,C,C,C,C,C,4D,B,B,B,B,B,B,B,D,C 12/11/14/4,4,5,6,8,9,13,10,2/7} {C,B,C,C,A,A,A,A,A,B,B,B,B,B,B,C, A 13,11,5,6,14,12,8,10,1,3/2/9/4/7}	6,5,7,6 6,7,6,5 6,6,5,7
20	15	5	3	30037.4	12919.2	{A,A,6D,A,A,A,A,C,C,C,C,C,6B,C,C,2B,C,C,C,C 14/4,5,6,8,9,13,12,10,11,1,2/15/7/3} {C,C,C,D,C,6A,C,C,5B,C,B,C,B,D,D,D,D,D,D,D 2/12/1,11,10,15,3,4,5,7,14,6,8/9/13} { A,A,A,A,A,A,B,B,B,B,B,D,D,B,D,D,D,D,D,C 4/1,6,7,2,14,9,12,10,11,13,3/15/5/8}	6,8,7,6 6,7,6,8 6,6,8,7
22	16	5	3	45442.2	13419.6	{B,B,B,8D,5B,C,C,C,C,C,A,A,A,A,A,A,2C,A,A,A, A,A 12/4,14/7,1,3,8,13,2,16/5,9,10,15/6,11} {C,C,C,D,C,3B,C,D,D,D,D,D,D,D,A,B,B,B,2B,C, C,C 9,10,5,7,14,12,15/13,4,6,8,1/3,11/16/2} {C,C,C,C,C,C,C,C,A,A,A,B,A,A,A,A,A,B,B,B,B,7 D 13/4/16,1/11/2,5,7,6,3,14,8,9,12,10,15}	6,8,7,8 6,7,8,8 8,6,8,7
26	18	6	4	66860.4	23152.8	{C,8A,C,C,C,C,C,D,2D,D,D,D,D,D,D,E,E,E,E,E,E, D,7B,D,D,D 2/3/8/9,10,13,12,17,1,4,5,7,14,6,18,11,/16/15} {B,B,B,B,B,B,7C,B,B,B,8E,B,B,B,B,B,B,B,B,B,B, 4A,B,A,A,C 8/4,2/5/18/9,10,1,12,3,6,14,11,15,16,13,17/7} {A,A,A,A,E,A,A,E,E,E,E,E,C,C,C,C,C,C,C,D,D,D, D,2D,D,D 9/14/2/16/1/11,3,4,13,17,10,12,15,18,5,7,6,8} {D,D,E,E,6D,E,E,E,E,E,A,A,A,E,A,A,A,A,C,C,C, C,C,C,C,C 11,15,14,13,17,18,5,7,1,3,2,6,8/4/9/14/12/10}	8,7,7,6,6 6,7,6,7,8 6,8,7,7,6 7,6,6,8,7
30	20	6	4	105019	25080	{B,B,B,B,B,B,B,B,B,7D,6E,C,C,C,C,C,C,C,C,A,A, A,A,A,A,A,A	8,9,8,7,6 6,7,8,9,8

						1/20/6/10,11,2,13,18,16,12,5,7,8,14,19,17,3/4/9,15} {A,A,A,A,A,C,A,A,A,A,A,5D,D,D,8E,A,A,D,D,B, B,B,B,B,7C,B,B,C,C,A 17/3/14,4,5,18,20,1,16,6,8,9,12,/19,10,7/11,15/13,2 } {B,B,B,B,B,5D,B,C,C,C,C,C,C,C,C,D,D,B,D,B, A,A,A,A,A,A,A,6E,D 18,1,3,4,14,16,6,8,9,12,17,20/15/7,10,19/5/2,13/11} {B,B,B,B,D,B,D,B,E,D,D,D,D,D,E,E,E,E,E,E,A,A, C,C,C,C,C,C,C,E 20,16,1,3,4,5,6,8,9,13,11,2,14/18/12/15/10,7,19/17}	7,8,9,8,6 9,6,7,8,8

References

[1] Somon, C. A., van Donk, P. D. and G. Gaalman, “Combined make-to-order and make-to-stock in a food production system,” *International Journal of Production Economics*, Vol.90, 2004, pp.223–235.

[2] Kazemi, S. M., Ghodsi, R., Rabbani, M. and R. Tavakkoli-Moghaddam, “A novel two-stage genetic algorithm for a mixed-model U-line balancing problem with duplicated tasks,” *The International Journal of Advanced Manufacturing Technology*, Vol.*55*, No.9-12 2011, pp.1111-1122.

[3] Manavizadeh, N., Hosseini, N. S., Rabbani, M. and F. Jolai, “A simulated annealing algorithm for a mixed model assembly U-line balancing type-I problem considering human efficiency and Just-In-Time approach,” *Computers & Industrial Engineering*, Vol.64, 2013, pp.669-685.

[4] Hamzadayi, A. and Gokalp Y. “A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints,” *Computers & Industrial Engineering* Vol.62, No.1 2012, pp.206-215.

[5] Avikal, S., Jain, R., Mishra, P.K. and H.C. Yadav, “A heuristic approach for U-shaped assembly line balancing to improve labor productivity. *Computers & Industrial Engineering*, Vol.*64*, No.4 2013, pp.895-901.

[6] Cheng, C. H., Miltenburg, J. and J. Motwani, “The effect of straight and U-shaped line on quality,” *IEEE Transaction on Engineering Management*, Vol.47, No.3 2000 321–334.

[7] Karimi, B., Ghomi, S. F. and J. M. Wilson, “The capacitated lot sizing problem: a review of models and algorithms,” *Omega*, Vol.*31*, No.5 2003, pp.365-378.

[8] Zhang, W. and M. Gen, “An efficient Multi objective genetic algorithm for mixed-model assembly line balancing problem considering demand ratio-based cycle time,” *Journal of Intelligence Manufacture*, Vol.22, 2011, pp.367–378.

[9] Sivasankaran, P. and P. Shahabudeen, “Literature review of assembly line balancing problem,” *The International Journal of Advanced Manufacturing Technology*, vol.73, 2014, pp.1665-1694.

[10] Farkhondeh, H., Hassanzadeh, R., Mahdavi, I. and N. Mahdavi-Amiri, “A DEA approach for comparing solution efficiency in U-line balancing problem using goal programming,” *The International Journal of Advanced Manufacturing Technology*, Vol.61, No.9-12 2012, pp.1161-1172.

[11] Hamta, N., Ghomi, S. F., Jolai, F. and M. A. Shirazi, “A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect,” *International Journal of Production Economics*, Vol.141, No.1 2013, pp.99-111.

[12] Dong, J., Zhang, L., Xiao, T. and H. Mao, “Balancing and sequencing of stochastic mixed-model assembly U-lines to minimise the expectation of work overload time,” *International Journal of Production Research*, Vol.52, No.24 2014, pp.7529-7548.

[13] Ogan, D. and M. Azizoglu, “A branch and bound method for the line balancing problem in U-shaped assembly lines with equipment requirements,” *Journal of Manufacturing Systems*, Vol.36, 2015, pp.46-54.

[14] Alavidoost, M.H., Babazadeh, H. and S. T. Sayyari, “An interactive fuzzy programming approach for bi-objective straight and U-shaped assembly line balancing problem,” *Applied Soft Computing*, Vol.40, 2016, pp.221-235.

[15] Seyed-Alagheband, S.A., Fatemi Ghomi S.M.T. and M. Zandieh, “A simulated annealing algorithm for balancing the assembly line type II problem with sequence dependent setup times between tasks,” *International Journal of Production Research*, Vol.49, No.3 2011, pp.805-825.

[16] Manavizadeh, N., Rabbani, M. and F. Radmehr, “A new multi-objective approach in order to balancing and sequencing U-shaped mixed model assembly line problem: a proposed heuristic algorithm,” *The International Journal of Advanced Manufacturing Technology*, Vol.79, No.1-4 2015, 415-425.

[17] Rabbani, M., Siadatian, R., Farrokhi-Asl, H. and N. Manavizadeh, “Multi-objective optimization algorithms for mixed model assembly line balancing problem with parallel workstations,” *Cogent Engineering*, Vol.3, No.1 2016, pp.1158903.

[18] Deb, K., Agrawal, S., Pratap, A. and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” *Lecture*

notes in computer science, Vol.1917, 2000, pp.849-858.

[19] Hamta, N., Akbarpour Shirazi, M., Fatemi Ghomi, S. M. T. and S. Behdad, “Supply chain network optimization considering assembly line balancing and demand uncertainty,” *International Journal of Production Research*, Vol.53, No.10 2015, pp.2970-2994.