Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-2 (2020) 10261–10266

Assembly line balancing for personalized production

Francesco Pilati*†, Giovanni Lelli, Maurizio Faccio***, Mauro Gamberi**, Alberto Regattieri****

* University of Trento, Department of Industrial Engineering, Via Sommarive 9, 38123 Trento, Italy
** University of Bologna, Department of Industrial Engineering, Viale del Risorgimento 2, 40136 Bologna, Italy
*** University of Padua, Department of Management and Engineering, Stradella, San Nicola 3, 36100 Vicenza, Italy

† *Corresponding Author*: francesco.pilati@unitn.it

Abstract: Assembly line balancing problems aim to an efficient and effective assignment of all the required tasks to workstations in a flow oriented production system. Nowadays, assembly lines have to face the manufacturing of extremely personalized products (e.g. cars) as requested by an increasingly higher portion of the market demand. Several literature contributions focus on different balancing problems affected by the wide variety of the final product, e.g. mixed and multi model assembly lines. However, no contribution seems to tackle the personalized production of goods. Such products require to assemble a certain number of tasks whatever the final product personalization is, and a variable number of optional of different type determined by the specifications of every single costumer. This paper faces the generalized assembly of personalized goods proposing an innovative two step methodology to optimize the workload balancing between the assembly line stations, considering traditional tasks and the optional required by the product personalization, which could occur with different frequencies and pairings. The first phase of the developed methodology executes a clustering of product options required by the costumers based on a similarity index. This phase leads to the definition of several sets of optional typically requested together by the customer and with similar mounting time. The methodology second phase leverages the defined clusters of optional. Indeed, optional of the same cluster shouldn't be assigned to the same workstation to reduce the overload or underload of the assembly operators. An integer programming model is proposed to assign both traditional tasks and optional to stations, to maximize the assembly line balancing considering the order frequency and assembly time of the clusterized optional. An industrial case study is adopted to test and validate the proposed two steps methodology. The obtained results highlight a consistent time balancing between assembly line workstations and a significant limitation of the operator overloads.

Keywords: assembly line balancing, personalized production, product optional, customization, clustering, customer order.

1. INTRODUCTION AND LITERATURE REVIEW

Great changes are taking place nowadays for production system shifting the company focus from value to customer requests (Martin, 2010; Cohen et al., 2019). Mass personalization is known as the current production paradigm of the industrial context (Wang et al., 2017; Faccio et al., 2019).

In the last years a dramatic increase in number of good variety involved many industrial contexts, e.g. automotive (Pil and Holweg, 2004). Products variety led to new challenges for production systems, forcing to shift from fixed assembly line to more flexible ones. These occurrences involve both manual and automated work content, indeed the share human work in the execution of physical activities is still predominant in manufacturing companies, with a mean values of 70% of the total amount of production activities. Furthermore, fully automatable manual tasks represent to date only the 5% of the total (World Economic Forum, 2018).

Several contributions in the last decades deal with flow-oriented production and assembly system, aiming to minimize a specific objective function such as the number of workstations (Grzechca, 2014), cycle time (Attique et al., 2013), idle time (Tang et al, 2016), total cost (Sarin and Erel, 2007) etc.

Salveson (1955) firstly outlined the mathematical model of assembly lines, defining the assignment of tasks to workstations leveraging binary variables. Such model, as well as the two reference formulations of the assembly line balancing problem (ALBP) proposed by Bowman (1959), tackle single-model assembly lines exploiting linear programming also to define technical precedences.

10.1016/j.ifacol.2020.12.2758

Nomenclature

Indices

i,j = 1 … I tasks or options

k = 1 … K assembly workstations

o = 1 … O customer orders

c = 1 … C clusters

Parameters

t_i assembly time of task/option i [s]
f_i order frequencies of task/option i [ ]
T_c mean assembly time of cluster c [s]
F_c mean order frequencies of cluster c [ ]
CT cycle time [s]
MT mounting time [s]
LBK lower bound of workstations number
g multiplicative parameter of cycle time
$n_{max,c}$ maximum theoretical number of options belong to c, allocable to a workstation
ε_c corrective factor of allocable options belonging to cluster c

$$a_{ij}^{o} = \begin{cases} 1, \text{if order o requires both option i and j} \\ 0, \text{otherwise} \end{cases}$$

$$b_{ij}^{o} = \begin{cases} 1, \text{if o requires only 1 options between i and j} \\ 0, \text{otherwise} \end{cases}$$

$$c_{ij}^{o} = \begin{cases} 1, \text{if order o does not require option i neither j} \\ 0, \text{otherwise} \end{cases}$$

$$d_{i}^{o} = \begin{cases} 1, \text{if order o require optional i} \\ 0, \text{otherwise} \end{cases}$$

Set

P set of couple (s,r): task/option s is an immediately predecessor of task/option r

Variable

$$X_{ik} = \begin{cases} 1, \text{if task or option i is assigned to wokstation k} \\ 0, \text{otherwise} \end{cases}$$

Considering the number of product variants increased by tasks and options combinations require by consumers, the focus of companies shifts to assembly lines which carry out a wide variety of final products, typically designed as mixed or multi model assembly lines (Boysen et al. 2009). In multi model lines the consecutive assembly of two different goods requires set up operations, thereby products carry out in lots. On the other hand, mixed model lines perform a family of similar goods which share assembly tasks with each other and equipped by specific options. In this case set-up times between models can be ignored and the lot size is 1 (Boysen et al., 2007).

Mixed model ALBP is typically tackled creating a virtual average model, which does not physically exist, including the assembly tasks and options of every product model, transforming the environment into a single model ALBP (Bortolini et al., 2018). Thomopoulos (2015) defines the joint precedence graph of a virtual average model, as the overlay of all model precedence graph. The execution time of tasks constituting such joint graph equates to frequency-weighted average time considering all product varieties available, thus allowing to deal with a SALBP. Assuming a set of n independent options, it results 2^n final product variants, thus such method enforceability falls as n increases. Furthermore, it is unrealistic to accurately estimate the market demand of each product variants or even keep an inventory of all the manufactured good variants.

Boysen et al. (2009) designs the joint precedence graph in the same way described above, but overcomes the aforementioned issue considering frequencies of each optional available to customers no matter which models require their assembly, to reduce the ALBP complexity. Determination of assembly time of the tasks required by the average model is a frequency-weighted mean time of each optional. So, the average virtual model is composed only by task and by optional transformed as tasks, therefore this ALBP is not able to consider information about the frequency of each optional.

Another crucial aspect regarding the assembly balancing problem of personalized products is the overload that occurs when two or more optional, assigned to the same workstation, are requested together by certain customers (Zhao et. al., 2004).

Considering the presented literature framework and aforementioned issues, this manuscript proposes a two-step methodology to achieve an optimal workload balance between assembly line stations carrying out personalized goods. The first phase deals exclusively with available product options for customize. Both the customer order list and the mounting time of these options are used to define a similarity index. Such index allows to perform a clustering of options. The second phase leverages the previously identified clusters proposing an innovative linear programming model, that aims to accomplish an optimal workstation balance reducing overloads.

This manuscript is organized as it follows. Section 2 presents the entire innovative methodology to face the assembly of high personalized goods, defining the necessary steps to perform in two phases. Section 3 describes the case study adopted to test and validate the proposed methodology presenting the necessary input data. The achieved results are presented and discussed in Section 4 before drawing the paper conclusions and defining suggestions for further research in Section 5.

2. BALANCING METHODOLOGY

This paper deals with highly personalized goods, tackling the balancing problem, minimizing the cycle time of their assembly lines. The considered assembly lines allow to mount both tasks and options as customers require.

The proposed methodology deals also with XORs options, but they are dissociated from the already defined pure ones. A XOR option set include 2 or more options that are exclusively disjointed to each other, thus one, and only one, is necessary for the assembly process e.g. the wheel type of a car. The methodology assumes each XOR set as a task, with an

assembly time equal to the mean assembly time of options belonging to the set.

The innovative proposed methodology faces the assembly problem in two steps, defining the option clusters in the first phase, while the second one exploits such cluster information to balance workload along the assembly line

2.1 Clustering

The first phase of the proposed methodology exploits customer orders as well as assembly deterministic time to carry out a clusterization of the available options. This phase starts by computing similarity between options considering the order share that requires same options and the comparability of mounting time as the following proposed in Equation (1) similarity index S_{ij} (1) provides.

$$S_{ij} = \sum_o \frac{2 \cdot a_{ij}^o \cdot MT_{ij}}{2 \cdot a_{ij}^o + b_{ij}^o + c_{ij}^o} \qquad \forall i, j : f_i, f_j < 1 \qquad (1)$$

Where:

$$MT_{ij} = \frac{\min(t_i\,;\,t_j)}{\max(t_i\,;\,t_j)} \qquad \forall i, j \qquad (2)$$

$$f_i = \frac{\sum_o d_i}{|O|} \qquad \forall i \qquad (3)$$

Such definition ivolves great values of S_{ij} for the pairs of options typically requested together by customers thanks to the bi-dimensional parameter "a_{ij}^o" which represents the number of orders that require both i and j are. Furthermore, S_{ij} introduces the comparability mounting time parameter "MT_{ij}", explained in Equation (2), to cluster together options characterized by similar assembly time. Such definition of similarity index imposes a value of S_{ij} between 0 and 1 involving just the options, which are characterized by an order frequency, defined in Equation (3), minor then 1. The proposed methodology leverages all couples S_{ij} to fill a square matrix between options and then it exploits the UPGMA algorithm to construct a rooted tree giving priority to pairs of options with high values of S_{ij}. The cutting value of the obtained dendrogram must be chosen in relation to the specific case study, considering number of the workstations and cycle time. Outcome of this methodology phase are sets of options whose splitting between workstations ensure workload balancing.

2.2 Assembly line balancing

Assembly line balancing phase exploits the performed clustering and joint precedences graph as input of an original linear programming model. Aim of such model is the optimization of workstation load, considering technical constraints by respecting the cycle time required to meet the market demand. The proposed methodology leverages the following linear programming model.

Objective function:

$$\text{Min}\left(\underset{k}{\text{Max}} \sum_{i=1}^{I} X_{ik} \cdot t_i \cdot f_i\right) \qquad (4)$$

Subject to:

$$\sum_{k=1}^{K} X_{ik} = 1 \qquad \forall i \qquad (5)$$

$$\sum_{k=1}^{K} k \cdot X_{sk} \le \sum_{k=1}^{K} k \cdot X_{rk} \qquad \forall (s, r) \in P \qquad (6)$$

$$\sum_{i}^{I} X_{ik} \cdot t_i \le CT \cdot g \qquad \forall k \qquad (7)$$

$$\sum_{i=1:(e_{ic} \cdot X_{ik} - n_{max_c}) \ge 0}^{I} (e_{ic} \cdot X_{ik} - n_{max_c}) \cdot T_c \cdot F_c + \sum_{i=1}^{I} X_{ik} \cdot \cdot t_i \cdot f_i \le CT \qquad \forall c, k \qquad (8)$$

$$\sum_{i}^{I} X_{ik} \cdot e_{ic} \le n_{max_c} + \varepsilon_c \qquad \forall c, k \qquad (9)$$

$$X_{ik} \in [0,1] \qquad \forall i, k \qquad (10)$$

Where:

$$T_c = \frac{\sum_{i \in c} t_i}{|c|} \qquad \forall c \qquad (11)$$

$$F_c = \frac{\sum_{i \in c} f_i}{|c|} \qquad \forall c \qquad (12)$$

$$n_{max_c} = \left\lceil \frac{|c|}{K} \right\rceil \qquad \forall c \qquad (13)$$

The objective function (Equation 4) aims to minimize the maximum assembly time between stations, indeed the workstation bottleneck imposes the cycle time for the entire assembly line. Furthermore, (4) ensures workload smoothness between workstations and considers order frequencies of options.

Equation (6) assures the respect of technical precedencies constraint, considering in sequence all the immediate predecessors. Constraint (5) along with (10) guarantee the unique assignment of all tasks and options to just one stations. Equation (7) tackles worst case of assembly balancing, namely when all options are required by a customer introducing a multiplicative factor "g" to cycle time to provide flexibility, avoiding excessive overload to operators. Equation (8) penalizes an over-allocation of options belonging to the same cluster to the same station, thanks to a multiplicative factor equal to $F_c \cdot T_c$. Furthermore, Equation (8) is the only constraint that guarantees compliance with the established cycle time. Finally, Equation (9) faces over-allocation imposing a maximum number of options assignable introducing a parameter "ε_c" to deliver flexibility to the model.

Equations (11) and (12) are respectively mean assembly time and average order frequency of options belonging to cluster c.

To reduce the redundancy of constraints and simplify data input, no variables distinguish options to tasks. The only difference regards the value of the parameter f_i, that is minor then 1 if it references to an option.

3. CASE STUDY

To test and validate the developed balancing methodology this section presents a case study taken from a luxury car manual assembly. Indeed, this context car is characterized by a high possibility of customization.

The presented assembly process includes 100 tasks always mounted by operators (i = 1 … 100) and 24 independent

options (i = 101 … 124) which can be chosen by customers. Case study involves thus more than 16 million of final product combinations available for customers. Fig. 1 represents the precedencies graph of the considered case study, which generates 139 immediate predecessors between tasks and options, where the lasts' circles are yellow coloured.

This case study, as in real applications, consider firstly a target market demand of 2800 $\frac{\text{pieces}}{\text{year}}$. The takt time, considering one 8-hour shift per day results on 2468 $\frac{\text{s}}{\text{pieces}}$.
CT determinates the lower bound (Bortolini et al., 2017) of workstation number by Equation (14), relaxing integrality of variables. Such characteristics of the case study lead to a theoretical number of stations of 5.23, which implies a lower bound of 6 workstations.

$$\text{LBK} = \left\lceil \frac{\sum_i^I t_i \cdot f_i}{\text{CT}} \right\rceil \quad (14)$$

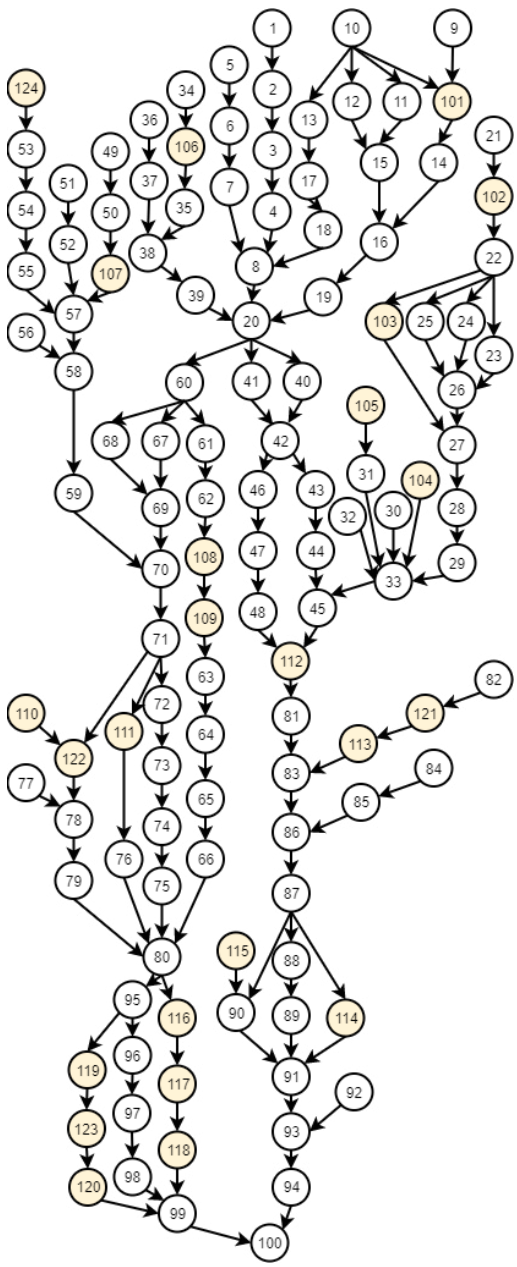


Fig. 1. Precedence graph

The first methodology phase leverages the option order list requested by costumers of a specific car model as well as their assembly time to carry out a clustering of options.

Such clustering information (e_{ic}, T_c, F_c) plus the precedences between tasks and options and their assembly time, allow the linear programming model to minimize the cycle time in the second methodology phase. Considering the share of options on the total operations carry out by the operators, the cycle time is increased by 50% in the case of worst case of assembly, e.g. assuming g = 1.5. Furthermore, to impose high split of options between workstations, ε_c is assumed equal to 1 for all the clusters.

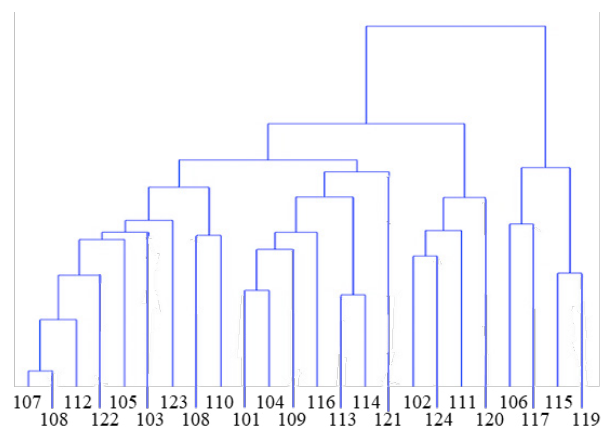
4. RESULTS AND DISCUSSION

This section proposes the results obtained by adopting the entire methodology for ALBP proposed in the former section 2 to the industrial case presented in section 3, distinguishing the two phases.

4.1 Clustering

Order list and deterministic mounting time of available options allow to define the similarity matrix. Such square matrix is the input of UPGMA algorithm, leading to construct the rooted tree represented in Fig. 2.

Fig. 2 Option dendrogram



The used cutting value leads to cluster options in a number of sets minor then K, thus allowing to heavy test Equations (8) and (9) in the second phase. The following Tab. 1 summarises the information of the 5 clusters obtained by cutting the dendrogram involving 23 of 24 available options.

Cluster ID	Number of options belonging to cluster	T_c (s)	F_c
1	9	79.3	0.26
2	6	106.8	0.41
3	4	54.3	0.24
4	2	89	0.39
5	2	91	0.30

Tab. 1. Cluster information

Such data as well as the parameter "e_{ic}" represent the output of the first methodology phase and input of the linear programming model of the second one.

4.2 Assembly line balancing

The last phase assigns tasks and options to workstations based on average option occurrence. Fig.3 shows the average workload distribution along the assembly line. The minimized cycle time results in 2156 seconds, equal to 87% of the takt time. The theoretical production rate thus increases up to 3207 $\frac{\text{pieces}}{\text{year}}$.

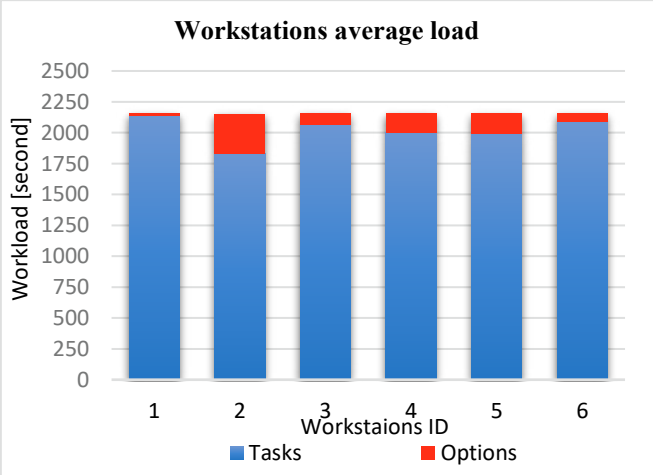


Fig. 3. Average line workload

Despite the assignment of tasks means dramatic difference between assembly operators workload, by introducing options assignment assembly line results well balanced. Indeed, the smoothness index (Equation 15) is minor then 1.

$$SI = \sqrt{\sum_k (\text{Workload}_{\max} - \text{Workload}_k)^2} \quad (15)$$

Furthermore, such assembly line balance ensures a limited overload, even when options belonging to the same cluster are required together by certain customer.

Despite the smoothness index increases till 315, also in the worst case represented in Fig. 4, all workstations load

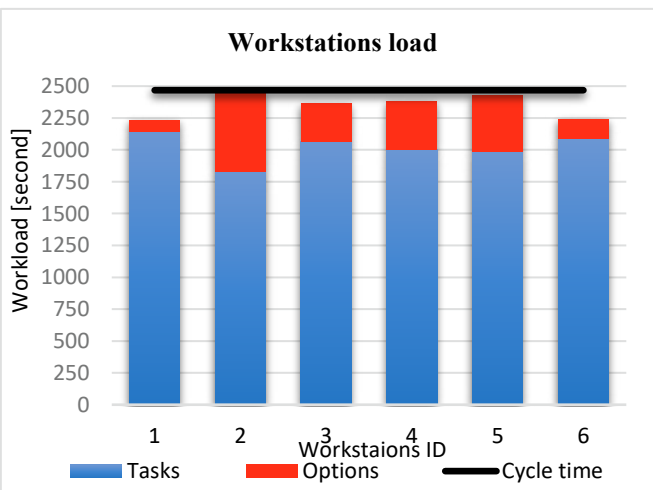


Fig. 4. Line workload in the worst case

guarantees the compliance with the target cycle time.

Constraints (8) and (9) force to split options between different workstations, as presented in Tab. 2, limiting at 2 the maximum number of options belonging to a cluster assigned to the same workstation.

Result concerning cluster 2 emphasizes how the whole methodology allows to divide highly similar options between the stations. Indeed, all the operators have to perform exactly one option belonging to cluster 2.

	Cluster				
Workstation	1	2	3	4	5
1	1	1	1	0	0
2	1	1	1	1	0
3	2	1	0	0	1
4	1	1	0	0	0
5	2	1	1	1	0
6	2	1	1	0	1

Tab. 2. Assigned option to workstation

5. CONCLUSIONS AND FURTHER RESEARCH

This paper proposes an innovative methodology to face highly personalized goods, whose assembly involves tasks and options as the customer require. While this field of research is typically considered as a mixed model ALBP, the proposed methodology exploits order lists to deal with options. Given the unreliability of forecasting the market demand of all final product versions, the proposed methodology is based on order frequency of options.

The proposed methodology suggests to cluster the options that shouldn't be assigned to the same workstation in order to avoid overload. The procedure goes ahead suggesting a linear programming model that exploits the aforementioned clusters to split options between workstations. Constraints of such model considers options both with and without their order frequency.

The overall objective of the methodology is the minimization of assembly line cycle time, which also involves a high balance of workstations load reducing overloads, proved through the value assumed in the proposed test of a real industrial case study by the smoothness index.

Further research should exploit the proposed procedure introducing constraints to linear programming model that considers the options dimension, to store component along the line waiting to be mounted. Furthermore, such aspect should modify even the similarity index.

REFERENCES

Bortolini, M., Faccio, M., Gamberi, M., and Pilati, F. (2017). Multi-objective assembly line balancing considering component picking and ergonomic risk. *Computers & Industrial Engineering*, volume (112:C*)*, 348-367.

Bortolini, M., Faccio, M., Gamberi, M., and Pilati, F. (2018). Motion analysis system (MAS) for production and ergonomics assessment in the manufacturing processes. *Computers & Industrial Engineering*.

Bowman, E. H. (1959). The schedule-sequencing problem. *Operations Research*, volume (7:5), 621-624.

Boysen, N., Fliedner, M., and Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, volume (183:2), 674-693.

Boysen, N., Fliedner, M., and Scholl, A. (2009). Assembly line balancing: Joint precedence graphs under high product variety. *IIE Transactions*, volume (41:3), 183-193.

Cohen, Y., Naseraldin, H., Chaudhuri, A., and Pilati, F. (2019). Assembly systems in Industry 4.0 era: a road map to understand Assembly 4.0. *The International Journal of Advanced Manufacturing Technology*, 1-18.

Faccio, M., Ferrari, E., Gamberi, M., and Pilati, F. (2019). Human Factor Analyser for work measurement of manual manufacturing and assembly processes. *The International Journal of Advanced Manufacturing Technology*, 1-17.

Grzechca, W. (2014). Assembly line balancing problem with reduced number of workstations. *IFAC Proceedings Volumes*, volume (47:3), 6180-6185.

Martin, R. (2010). The age of customer capitalism. *Harvard Business Review*, volume (88:1), 58-65.

Pil, F. K., and Holweg, M. (2004). Linking product variety to order-fulfillment strategies. *Interfaces*, volume (34:5), 394-403.

Salveson, M. E. (1955). The assembly line balancing problem. *The Journal of Industrial Engineering*, volume (6:3), 18-25.

Subhash C. Sarin and Erdal E. (1990) Development of cost model for the single-model stochastic assembly line balancing problem. *International Journal of Production Research*, volume (28:7), 1305-1316.

Tang, Q., Li, Z., and Zhang, L. (2016). An effective discrete artificial bee colony algorithm with idle time reduction techniques for two-sided assembly line balancing problem of type-II. *Computers & Industrial Engineering*, volume (97), 146-156.

Thomopoulos, N. T. (2015). *Demand Forecasting for Inventory Control,* 1-10. Springer.

Wang, Y., Ma, H. S., Yang, J. H., and Wang, K. S. (2017). Industry 4.0: a way from mass customization to mass personalization production. *Advances in Manufacturing*, volume (5:4), 311-320.

World Economic Forum. (2018). The future of jobs report 2018. *World Economic Forum*. Geneva, Switzerland.

Zhao, X., Katsuhisa, O., and Hon-Shiang, L. (2004) "A balancing problem for mixed model assembly lines with a paced moving conveyor." *Naval Research Logistics*, volume (51.3), 446-464.