Rebalancing of mixed-model two-sided assembly lines with incompatible task groups: An industrial case study

David Z. ZHANG[1*], Ibrahim KUCUKKOC[2], Aslan Deniz KARAOGLAN[2]

[1] College of Engineering, Mathematics and Physical Sciences,
University of Exeter, Exeter EX4 4QF, England, United Kingdom
d.z.zhang@exeter.ac.uk
[2] Department of Industrial Engineering, Faculty of Engineering,
Balikesir University, Cagis Campus, Balikesir, Turkey
i.kucukkoc@exeter.ac.uk, deniz@balikesir.edu.tr

ABSTRACT

Assembly lines, which are mostly used flow-oriented production techniques in mass production, are classified in different ways, e.g. considering their configurations and/or product variety produced on the line. Mixed-model two-sided assembly lines are usually constructed for the production of large-sized items, such as automobiles or trucks, in an inter-mixed sequence. While the literature on two-sided lines is extensive, there is no research concerning minimizing the cycle time in mixed-model two-sided lines, to the best of the authors' knowledge. Therefore, this paper introduces the type-II mixed-model two-sided assembly line balancing problem benefiting from the real data gathered through an industrial case study. This is the major contribution of this paper. This paper also contributes to knowledge by incorporating incompatible task groups, different from negative zoning constraints. Any two tasks existing in the same incompatible task group cannot be accomplished at the same time in the same mated-station.

Keywords: Assembly line balancing, type-II, mixed-model lines, two-sided lines, incompatible task groups.

1 INTRODUCTION

An assembly line consists of a sequence of workstations linked to each other via a conveyor or moving belt. Each workstation completes its operations on the product (or product model) produced on the line and the product moves to the downstream workstation [1]. The tasks are performed in workstations considering the precedence relationships between tasks and the capacity constraints. The capacity constraint is determined by the cycle time of the line, which is the maximum time each workstation is allowed to complete its tasks. The assembly line balancing (ALB) problem is to assign tasks to workstations considering the capacity and precedence relationship constraints such that a performance measure is optimized. The performance measure can be optimized by minimizing the number of workstations given the cycle time (referred to as type-I ALB problem), or minimizing the cycle time given the number of workstations (referred to as type-II ALB problem). In some cases, both performance measures (i.e. the number of workstations and the cycle time) are aimed to be minimized, which is called a type-E ALB problem [2].

The assembly lines are classified as one-sided lines and two-sided lines based on the formation of the workstations across the corresponding line. In a one-sided assembly line,

---

[*] Corresponding Author

the workstations are located in only one side of the line whereas the workstations are located in both (left and right) sides of the line in a two-sided assembly line [3]. Assembly lines are also divided into three groups based on the variety of the products assembled on the line: (i) single-model lines, (ii) mixed-model lines and (iii) multi-model lines. Single model lines are utilized to produce only one model of a product in mass quantities while two or more product models are assembled on mixed-model lines with no need for set-up between model changes [4]. Multi-model lines are also utilized to produce two or more product models. However, different from mixed-model lines, multi model-lines need set-up operations between model changes. Due to this drawback, mixed-model lines are favored in industry.

While there are numerous studies on single-model two-sided lines (e.g., Kim *et al.* [5], Lee *et al.* [6], Baykasoglu and Dereli [7], Ozcan and Toklu [8], Ozbakir and Tapkan [9], Tapkan *et al.* [10], Khorasanian *et al.* [11], Purnomo *et al.* [12]), very little attention has been paid to mixed-model two-sided lines, which consider the production of two or more product models on a two-sided assembly line. Simaria and Vilarinho [13] developed an ant colony optimization algorithm for solving the mixed-model two-sided assembly line balancing (MTALB) problem considering parallel workstations. The aim was minimizing the number of workstations considering some additional constraints, such as synchronous tasks and zoning constraints. Ozcan and Toklu [14] modelled the MTALB problem mathematically and developed a simulated annealing algorithm with the aim of minimizing the number of mated-stations and the number of workstations considering synchronous tasks, positional constraints and zoning constraints. Chutima and Chimklai [15] proposed a particle swarm optimization algorithm with negative knowledge when solving the MTALB problem considering work-relatedness and workload smoothness. Rabbani *et al.* [16] addressed the MTALB problem with multiple U-shaped layout and proposed a GA-based heuristic and a mixed integer program for minimizing the cycle time and the number of stations.

Kucukkoc and Zhang [17] developed a flexible agent-based ant colony optimization algorithm for the mixed-model parallel two-sided assembly line balancing problem (without model-sequencing mechanism) to minimize the line length and the number of workstations. Kucukkoc and Zhang [18] combined the model-sequencing and line balancing problems and introduced the mixed-model parallel two-sided assembly line balancing and sequencing problem, and Kucukkoc and Zhang [19] proposed a mathematical model and an agent-based solution approach for solving it. In their latter research [20], a GA-based model sequencing mechanism was integrated into the agent-based ant colony optimization algorithm to solve the problem more efficiently. This paper addresses to the mixed-model two-sided assembly line system and contributes to knowledge as follows:

i) Different from the common tendency to minimize the number of workstations in mixed-model two-sided lines, the cycle time is minimized in this research. Therefore, *type-II mixed-model two-sided assembly line balancing problem* is introduced.
ii) A new constraint between tasks, called *incompatible task groups*, is introduced. This concept will be explained in details in the following section.
iii) Furthermore, this research contributes to knowledge by an industrial case study conducted in a tractor cabin manufacturing plant. As stated by Baykasoglu and Dereli [7], "*real-life case studies can provide very useful research contributions*" for two-sided lines. In their research on mixed-model two-sided lines, Simaria and Vilarinho [13] also highlighted the need for additional efforts in "*matching theoretical procedures and practical applications*".

The remainder of this paper is organized as follows. The type-II mixed-model two-sided assembly line balancing problem with incompatible task groups is defined in Section 2. The solution method is described in Section 3 and an industrial case study is provided in Section 4 together with a discussion. Finally, the paper is concluded with some future research directions and practical implications of the study in Section 5.

2 PROBLEM DEFINITION

The configuration of the classical mixed-model two-sided assembly line is represented in Figure 1. As seen from Figure 1, two product models (model-I and model-II) are produced on the same two-sided line, simultaneously. Note that, in practice, there may be more than two product models on the line. The workstations are located on both left and right sides of the line, and one operator performs in each workstation. Opposite workstations are called the *companion* of each other, such as WS-1 is the companion of WS-2, or vice versa. The two companion workstations are called a *mated-station*; e.g. WS-1 and WS-2 constitute a mated-station. Thus, there are four mated-stations in Figure 1, which equivalents to eight workstations in total.

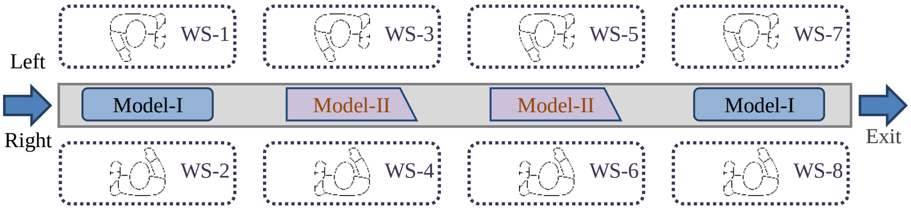

Figure 1. Representation of a mixed-model two-sided assembly line.

In two-sided lines, there are certain constraints; i.e. capacity constraints, precedence relationships constraints and operation side constraints to be satisfied to have a feasible solution. The capacity constraints are determined by the cycle time, which is the maximum time each workstation is allowed to complete tasks assigned to it. The precedence relationship constraints occur due to the technological requirements or organizational restrictions of the manufacturing environment. Considering the precedence relationships between tasks when building a line balancing solution is particularly important, or the line balance to be obtained eventually can be infeasible. Thus, a task can start after all of its predecessors have been completed. In two-sided lines, some tasks may require to be performed at a specific side, i.e. left or right side, while some others can be performed at either side. A task that needs to be performed on the left (right) side of the line is called a left (right) side task.

Table 1 presents the data of an example problem about the classical mixed-model two-sided assembly line balancing problem taken from Ozcan and Toklu [14]. The second column provides the information about which side the corresponding task must be assigned to. The letters L, R and E denote the left side, right side and either side, respectively. The third and fourth columns provide the processing time of the related task for each model, model-A and model-B, respectively. Processing times of tasks may vary from one model to another. For example, task 2 requires 3 time-units for product model A while 1 time-unit for product model B. The precedence relationships data are provided in the last column. For example, tasks 2 and 3 must be completed to initialize task 6.

The balancing solution of the example problem is given in Figure 2. The length of the bars represents the processing time of the task given in the corresponding bar. The gray shaded areas denote the unavoidable idle times. As seen from the figure, three workstations are needed when the cycle time is considered 5 time-units/item. Tasks 1, 3, 6 and 9 have been assigned to the left side of the first mated-station while tasks 2 and 5 have been assigned to the right. Tasks 4, 8 and 7 have been assigned to the left of the second mated-workstation. No tasks have been assigned to its opposite side as all tasks have already been assigned. As seen from the figure, the workload time of any workstation does not exceed the cycle time for any model.

The problem introduced in this paper is the type-II mixed-model two-sided assembly line balancing problem with incompatible task groups. The aim is to minimize the cycle time of a paced (or synchronous) mixed-model two-sided assembly line given the number of workstations.

Table 1. The data of the example problem [14].

Task No	Side	Processing Time (time-units)		Immediate Predecessor(s)
		Model A	Model B	
1	L	2	0	-
2	R	3	1	-
3	E	0	1	-
4	L	3	0	1
5	R	1	3	2
6	E	1	1	2,3
7	E	2	2	4,5
8	L	0	3	5
9	E	1	1	6

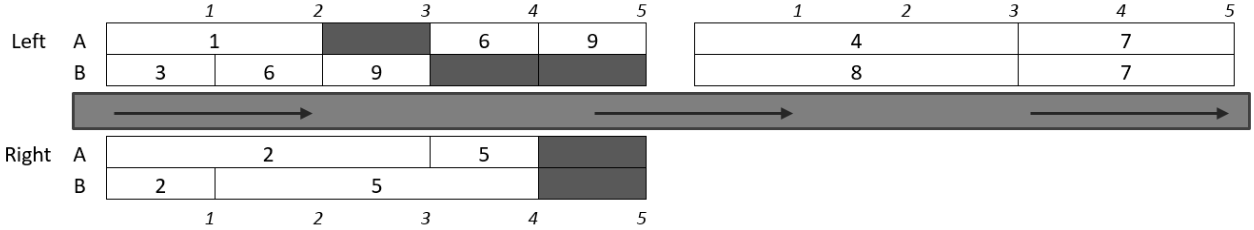

Figure 2. The balancing solution of the example problem [14].

In addition to the constraints explained above, this paper considers a new concept to represent the real-world manufacturing environment more efficiently: incompatible task groups (ITGs). In the literature, tasks have positive or negative zoning constraints. If there is a positive zoning constraint between two tasks, these two tasks must be assigned to the same workstation. On the contrary, the negative zoning constraint determines the two tasks that must be performed in different workstations. However, on certain occasions, it is not possible to perform two tasks at the same time on both sides of the line. For example, if the left-side operator needs to get in the product model being assembled on the line, the right-side operator cannot get in the product model at the same time as there is no enough space for two operators. Therefore, these two incompatible tasks must be performed at different times no matter they are assigned to the same or different workstations. Therefore, tasks do not have binary relationships between each other. Instead, a task existing in an ITG cannot be performed at the same time with another task existing in the same ITG. The ITGs can be categorized if there is more than one ITG for the same system.

3 SOLUTION METHOD

The pseudocode of the solution algorithm used in this research to conduct the case study (to be given in the next section) is presented in Figure 3. As seen from the figure, the cycle time value is initialized ($C \leftarrow C_{min}$) and a number of solutions ($maximumNumberOfSubIterations = 200$ iterations) are produced for each C value using 10 different line balancing rules, most of which are used commonly in the line balancing domain: Computer Method of Sequencing Operations for Assembly Lines - Comsoal [21], Ranked Positional Weight Method - RPWM [22], Reverse Ranked Positional Weight Method - RRPWM (produced from RPWM), Longest Processing Time - LPT [23], Shortest Processing Time - SPT [24], Smallest Task Number - STN [25], Maximum Number of Predecessors - MNP (produced from Baykasoglu [24]), Least Number of Predecessors - LNP, (produced from MNP), Maximum Number of Successors - MNS (produced from Tonge [26]), and Least Number of Successors - LNS (produced from MNS). Each of these 200 balancing solutions is built using a randomly picked rule and the solution which gives the minimum number of stations is retrieved. If the minimum number of stations found by the algorithm is equal to the number of stations aimed, the algorithm is terminated and the solution which gives the minimum number of stations is designated as the best solution. If not, C is increased by $cycle\_time\_increment$

value and 200 solutions are built again using randomly picked heuristics. This cycle continues until $numberOfStationsAimed$ is achieved and the current cycle time is determined as the solution of the problem ($C^* \leftarrow C$). When there is more than one solution with the same number of stations, the solution which has smoother workload distribution is favored. Note that C_{min} is the theoretical minimum cycle time calculated as $C_{min} = \max_{j\in J}\left\{\left\lceil\frac{\sum_{i=1}^{N} t_{ij}}{K}\right\rceil^{+}\right\}$ where N is the total number of tasks, t_{ij} is the processing time of task i for model j ($j \in J$), K is the total number of workstations. $[X]^{+}$ denotes the least integer equals to or larger than X.

```
Start (a heuristic rule is selected randomly)
Initialize all parameters, $C \leftarrow C_{min}$
While ($numberOfStations > numberOfStationsAimed$) {
    While ($sub\_iteration\_number < maximumNumberOfSubIterations$) {
        Pick a heuristic rule randomly and build a balancing solution using the procedure given in Figure 4.
        Evaluate the solution obtained (calculate the $numberOfStations$)
    } End while
    $sub\_iteration\_number += 1$
} End while
$C += cycle\_time\_increment$
Terminate and retrieve $C$ ($C^* \leftarrow C$)
```

Figure 3. The pseudocode of the heuristic algorithm coded in Java.

```
Start (a heuristic rule is selected randomly)
Initialize all sets (i.e., available tasks and unassigned tasks) and parameters (i.e., $st(k)j \leftarrow 0$ , where $st(k)j$ represents the station time of workstation k for model j)
While (there is one or more unassigned task) {
    Select an operation side randomly and determine all available tasks for the current position
    If (there is at least one available task) {
        Select a task (i.e., task i) using the heuristic rule
        Assign task i to the current available station and increase the station time: $st(k)j \leftarrow st(k)j + (t_{ij})$
        Increase the earliest starting times of all successors of task i
        Remove task i from the unassigned tasks list of the relevant line
    } else if (there is no available task due to interference) {
        Increase the station time of the current workstation: $st(k)j \leftarrow st(\underline{k})j$, where $\underline{k}$ is the companion of k
    } else if (there is no available task due to insufficient capacity) {
            If (both sides of the line reached full capacity) {
                Increase the station number (k++)
            } else if (at least one side of the current line has not reached full capacity) {
                Alternate the operation side
            } End if
    } End if
} End while
Terminate
```

Figure 4. The procedure for building a balancing solution.

4 CASE STUDY

The case study has been conducted at an automotive company which produces safety cabins for tractors, excavators, and forklifts. The production operations of the company have been illustrated in Figure 5. As seen from the figure, the formed sheet metal and formed profile are welded in the welding line. The cabin frame is painted by a powder coating operation and moved to the assembly line, in which all functional units and accessories are sequentially mounted on to the cabin. More than five different cabin models are assembled on the same line which is constituted by 13 sequentially located mated-stations (which equivalents to a total of 26 workstations). Due to the page limit, the assembly process of the two mostly produced cabin models in the first eight workstations were selected for this case study. Thus, the aim is to minimize C given $K = 8$. The data on the processing times, operation sides and precedence relationships of the 49 tasks are provided in Table 2. The tasks existing in the same incompatible task group are marked with '1' in the 'ITG' column. The assignment configuration of these tasks in its current status is also presented in Table 3.

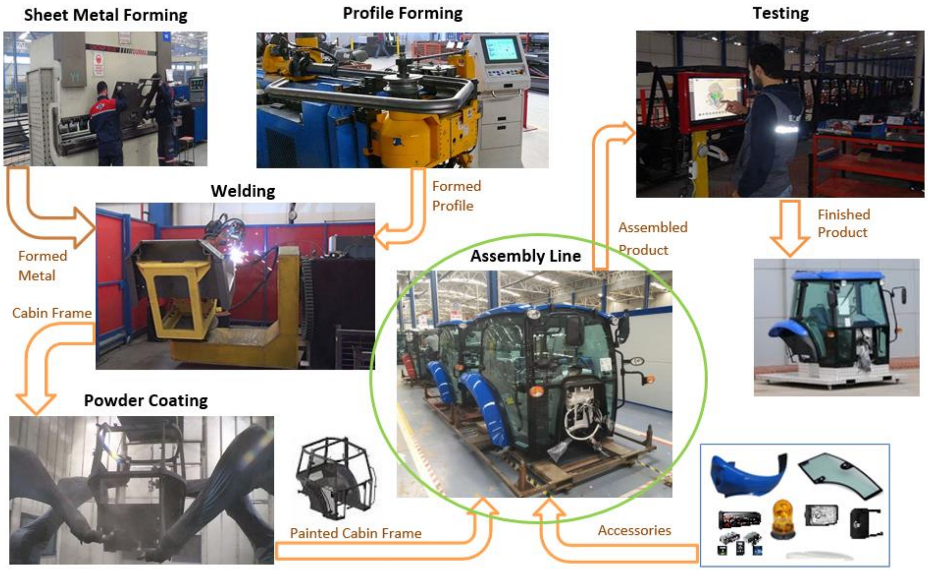

Figure 5. The production system of the company.

Table 2. Data for the case study.

Task	Processing Time (time-unit)		Side	ITG	Immediate Predecessor(s)	Task	Processing Time (time-unit)		Side	ITG	Immediate Predecessor(s)
	A	B					A	B			
1	29	29	E	1	-	**26**	17	17	E	1	24
2	16	16	R	-	7	**27**	4	4	L	-	12
3	10	10	R	-	7	**28**	28	28	E	-	27
4	4	4	E	-	-	**29**	7	10	E	-	28
5	18	18	E	-	4	**30**	9	9	E	-	28
6	15	15	E	-	-	**31**	17	17	E	-	29
7	9	9	E	-	-	**32**	16	16	E	-	30
8	11	11	E	-	7	**33**	15	15	E	-	10
9	7	7	E	1	6	**34**	16	10	R	-	12
10	4	4	E	-	-	**35**	17	17	R	-	12
11	13	13	E	1	2,9,10	**36**	10	20	R	-	35
12	6	6	E	-	5	**37**	10	30	R	-	36
13	13	13	E	-	12	**38**	11	11	R	-	37
14	8	8	E	-	13	**39**	8	8	L	-	10
15	65	69	E	1	14	**40**	10	10	E	-	12
16	6	6	E	-	5	**41**	11	17	E	-	12
17	5	5	E	-	12	**42**	10	10	E	1	40,41
18	11	11	L	-	12,17	**43**	8	8	E	1	40,41
19	7	7	E	1	12,15	**44**	9	9	E	1	43
20	31	31	E	1	11,19	**45**	5	10	E	1	44
21	7	7	E	1	19	**46**	11	11	E	1	2,3,11,15,43
22	22	22	E	-	12	**47**	4	4	E	1	42,45,46
23	24	24	E	1	22,27	**48**	6	6	L	-	12
24	14	14	E	1	23	**49**	18	27	E	-	47
25	15	15	E	1	22,24,27						

Figure 6 exhibits the change in the number of stations as the cycle time increases. As seen, the best solution obtained when the cycle time was 88 time-units requires 10 workstations. As the cycle time increases by one time-unit, the number of workstations decreases and the algorithm is terminated as soon as it reaches to eight workstations eventually.

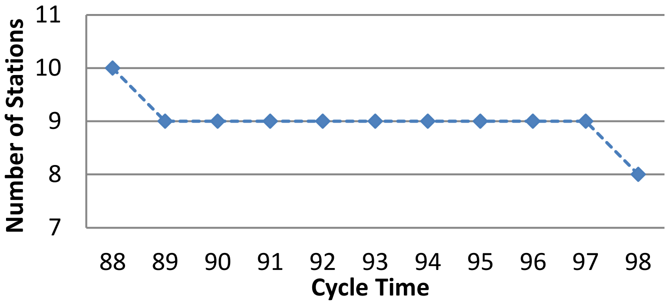

Figure 6. The number of stations obtained for the corresponding cycle time value.

The solution obtained when the algorithm was terminated is presented in Table 3 (see "Proposed Line Balance" column). As seen from the table, the maximum of the station workloads in the new solution is 98 time-units, which is much smaller than the one in current line balance (120 time-units). That means the cycle time of the line was reduced from 120 time-units to 98 time-units, which corresponds to a 22.4% improvement. Therefore, the weighted line efficiency is increased from 70% to 85.8%, with an enhancement of 22.4%. Moreover, in the current situation, the line requires 5 mated-stations as the left side of the second workstation (2-L) and the right side of the third workstation (3-R) are not utilized. However, the proposed solution requires four mated-stations since both of the operation sides have been utilized. Thus, the line length was also reduced in the proposed balance.

Table 3. The assignment configuration of the current and the proposed line balances.

Current Line Balance				Proposed Line Balance			
Station	Assigned Tasks	Station Workload		Station	Assigned Tasks	Station Workload	
		A	B			A	B
1-L	6,7,8,9,10,11,12	65	65	1-L	7,10,6,9,33,8,27,22,17,16	**98***	**98***
1-R	1,2,3,4,5	77	77	1-R	4,5,12,1,13,14,40	88	88
2-L	-	-	-	2-L	15,11,19	85	89
2-R	13,14,15	86	90	2-R	2,41,28,35,43	80	86
3-L	16,17,18,19,20,21	67	67	3-L	24,44,46,42,45,47,29,48	66	74
3-R	-	-	-	3-R	23,3,36,37,30	63	93
4-L	27,28,29,30,31,32	81	84	4-L	20,26,32,18,39	83	83
4-R	22,23,24,25,26	92	92	4-R	49,31,25,34,38,21	84	87
5-L	39,40,41,42,43,44, 45,46,47,48,49	100	**120***				
5-R	33,34,35,36,37,38	79	103				

* The maximum station workload time for the corresponding situation

5 CONCLUSION

The mixed-model two-sided assembly line balancing problem was addressed considering ITGs. Different from the studies on mixed-model two-sided assembly lines in the literature, the main aim was to minimize the cycle time while maximizing the workload smoothness among the workstations. A case study was conducted using data gathered from an automotive company. The line was rebalanced using heuristic method coded in Java and the cycle time was reduced by 22.4%. The newly introduced ITG concept limits performing the tasks existing in the same ITG group on either side of the line. While there may be more than one ITG in some cases, only one ITG was considered in this research. This concept can also be implemented in other line concepts or industries where it is inconvenient to perform some group of tasks on the same item, simultaneously. As a limitation of the work, the number of tasks considered in the case study was kept to a minimum due to page limit.

ACKNOWLEDGMENT

The authors acknowledge engineers Mr. Korunay Ucok and Ms. Sevil Duran for their valuable collaboration in gathering the industrial data.

REFERENCES

[1] Kucukkoc I, Zhang DZ. Balancing of parallel U-shaped assembly lines. Computers and Operations Research. 2015;64:233–44, doi: http://dx.doi.org/10.1016/j.cor.2015.05.014.

[2] Kucukkoc I, Zhang DZ. Type-E parallel two-sided assembly line balancing problem: Mathematical model and ant colony optimisation based approach with optimised parameters. Computers and Industrial Engineering. 2015;84:56–69, doi: http://dx.doi.org/10.1016/j.cie.2014.12.037.

[3] Bartholdi JJ. Balancing 2-Sided Assembly Lines - a Case-Study. International Journal of Production Research. 1993;31(10):2447-61.

[4] Thomopoulos NT. Line Balancing-Sequencing for Mixed-Model Assembly. Management Science. 1967;14(2):B-59-B-75.

[5] Kim YK, Kim YH, Kim YJ. Two-sided assembly line balancing: a genetic algorithm approach. Production Planning & Control. 2000;11(1):44-53.

[6] Lee TO, Kim Y, Kim YK. Two-sided assembly line balancing to maximize work relatedness and slackness. Computers & Industrial Engineering. 2001;40(3):273-92.

[7] Baykasoglu A, Dereli T. Two-sided assembly line balancing using an ant-colony-based heuristic. International Journal of Advanced Manufacturing Technology. 2008;36(5-6):582-8.

[8] Ozcan U, Toklu B. A tabu search algorithm for two-sided assembly line balancing. International Journal of Advanced Manufacturing Technology. 2009;43(7-8):822-9.

[9] Ozbakir L, Tapkan P. Bee colony intelligence in zone constrained two-sided assembly line balancing problem. Expert Systems with Applications. 2011;38(9):11947-57.

[10] Tapkan P, Ozbakir L, Baykasoglu A. Modeling and solving constrained two-sided assembly line balancing problem via bee algorithms. Applied Soft Computing. 2012;12(11):3343-55.

[11] Khorasanian D, Hejazi SR, Moslehi G. Two-sided assembly line balancing considering the relationships between tasks. Computers & Industrial Engineering. 2013;66(4):1096-105.

[12] Purnomo HD, Wee HM, Rau H. Two-sided assembly lines balancing with assignment restrictions. Mathematical and Computer Modelling. 2013;57(1-2):189-99.

[13] Simaria AS, Vilarinho PM. 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines. Computers & Industrial Engineering. 2009;56(2):489-506.

[14] Ozcan U, Toklu B. Balancing of mixed-model two-sided assembly lines. Computers & Industrial Engineering. 2009;57(1):217-27.

[15] Chutima P, Chimklai P. Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. Computers & Industrial Engineering. 2012;62(1):39-55.

[16] Rabbani M, Moghaddam M, Manavizadeh N. Balancing of mixed-model two-sided assembly lines with multiple U-shaped layout. International Journal of Advanced Manufacturing Technology. 2012;59(9-12):1191-210.

[17] Kucukkoc I, Zhang DZ. Mixed-model parallel two-sided assembly line balancing problem: A flexible agent-based ant colony optimization approach. Computers & Industrial Engineering. 2016;97:58-72.

[18] Kucukkoc I, Zhang DZ. Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. International Journal of Production Research. 2014;52(12):3665-87.

[19] Kucukkoc I, Zhang DZ. Mathematical Model and Agent Based Solution Approach for the Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines. International Journal of Production Economics. 2014;158, :314-33.

[20] Kucukkoc I, Zhang DZ. Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines. The International Journal of Advanced Manufacturing Technology. 2015: doi: http://dx.doi.org/10.1007/s00170-015-7320-y.

[21] Arcus A. COMSOAL, A Computer Method of Sequencing Operations for Assembly Lines. The International Journal of Production Research. 1966;4(4):259-77.

[22] Helgeson WB, Birnie DP. Assembly Line Balancing Using the Ranked Positional Weight Technique. The Journal of Industrial Engineering. 1961;12(6):394-8.

[23] Talbot FB, Patterson JH. An Integer Programming Algorithm with Network Cuts for Solving the Assembly Line Balancing Problem. Management Science. 1984;30(1):85-99.

[24] Baykasoglu A. Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. Journal of Intelligent Manufacturing. 2006;17(2):217-32.

[25] Arcus AL. An analysis of a computer method of sequencing assembly line operations: Ph.D. Dissertation, University of California, Berkeley, USA; 1963.

[26] Tonge FM. Summary of a Heuristic Line Balancing Procedure. Management Science. 1960;7(1):21-42.