Multi-objective Optimization of Mixed-model Two-sided Assembly Lines – A Case Study

Ibrahim Kucukkoc
Department of Industrial Engineering, Faculty of Engineering, Balikesir University, Balikesir, Turkey
ikucukkoc@balikesir.edu.tr

Abstract — **Assembly lines are one of the frequently used mass production techniques for producing homogeneous parts in large quantities. An assembly line is called a mixed-model line when there is more than one product model being assembled on the same line. This paper addresses to mixed-model two-sided lines, on which two or more large-sized product models (like buses or trucks) are assembled on the same line in an intermixed sequence. An ant colony optimization algorithm is proposed for minimizing the cycle time of the line as well as the number of workstations. A real-world problem is solved using the proposed approach and the efficiency of the line is improved.**

Keywords* — *assembly line balancing; mixed-model lines; two-sided lines; multi-objective optimization; ant colony optimization.

I. INTRODUCTION

An assembly line is composed of a sequence of workstations linked to each other through a conveyor or a moving belt. In each workstation (a physical space in which at least one operator works), a set of tasks are performed regarding the semi-product assembled on the line and the semi-product is forwarded to the subsequent workstation. The final product is departed from the last workstation and the time between the departures of two consecutive products is called cycle time. Thus, each workstation is allowed to perform all tasks assigned to it within the cycle time [1].

The assembly line balancing problem is to determine the assignment configuration of tasks to workstations. The wide-spread objective is to maximize the efficiency of the line either by minimizing the number of workstations given the cycle time (referred to as the problem of type-I) or minimizing the cycle time given the number of workstations (referred to as the problem of type-II). There are certain constraints to be satisfied during the task assignment process: capacity constraint, precedence relationship constraint and task occurrence constraint. The capacity constraint ensures that the finishing times of all tasks assigned to a particular workstation do not exceed the cycle time. The precedence relationship constraint specifies the technological or organizational priorities between tasks; i.e. some tasks must have been completed before the initialization of another task. Finally, the occurrence constraint corresponds to the assignment of every task to exactly one workstation to obtain a feasible line balance [2].

The mixed-model production line was introduced by Thomopoulos [3] and has been studied extensively since then. A mixed-model line is a consecutive sequence of workstations in which more than one product model is produced simultaneously. On the contrary, a single model line is able to assemble a single product model at the same time. Therefore, the main advantage of mixed-model lines over single-model lines is that there is no need to construct a new line to produce each product model. Thus, customized customer demands can be fulfilled by the companies in a more economical way. Within this context, the mixed-model lines have been widely utilized in industry from electronics to automotive and home appliances.

Among various studies on assembly line balancing problems, the ones on mixed-model lines usually consider the lines with only one side. However, the two-sided lines [4], across which the product models are assembled through the workstations located in both left and right sides of the line, are usually applied to produce large-sized items (such as trucks, buses or even smaller products) with model variations. Although having a mixed-model two-sided line is getting popular in today's manufacturing industry, the number of studies on mixed-model two-sided lines is limited. An ant colony optimization (ACO) algorithm was developed by Simaria and Vilarinho [5] for solving the mixed-model two-sided assembly line balancing problem considering parallel workstations. The aim was to minimize the number of workstations. A mathematical model and a simulated annealing algorithm were proposed by Ozcan and Toklu [6] with the aim of minimizing the number of mated-stations and the number of workstations. In both studies (Simaria and Vilarinho [5]

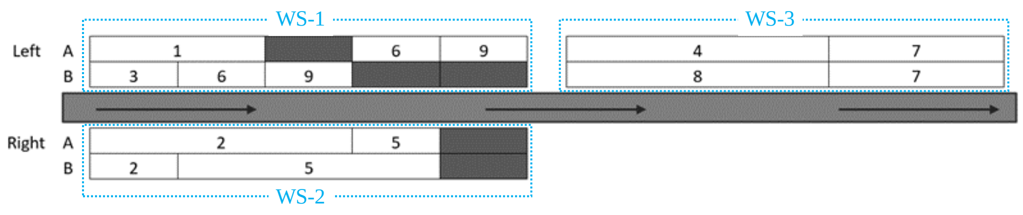

Figure 1. A schematic configuration of tasks in mixed-model two-sided lines [7]

and Ozcan and Toklu [6]) some additional constraints were also adopted, such as synchronous tasks and zoning constraints. Chutima and Chimklai [8] proposed a particle swarm optimization algorithm and aimed to maximize the work-relatedness and workload smoothness. Rabbani et al. [9] proposed a GA-based heuristic and a mixed-integer program to solve the mixed-model two-sided assembly line balancing problem considering the multiple U-shaped layout with the aim of minimizing the cycle time and the number of stations. Kucukkoc and Zhang [10-12] developed various agent-based ant colony algorithms and a mathematical model for balancing and balancing/sequencing the mixed-model parallel two-sided assembly lines.

As seen from this survey, there is no research which aims to simultaneously minimize the cycle time and the number of workstations for mixed-model two-sided lines (referred to as the problem of type-E). Therefore, to the best of author’s knowledge, this research contributes to knowledge by addressing the type-E mixed-model two-sided assembly line balancing problem for the first time. An ant colony algorithm is proposed for solving the problem, due to its NP-hard nature. A real-world problem was taken from Zhang et al. [7] and solved using the proposed approach. The result of the case study indicates that the proposed approach helps improve the efficiency of the line.

II. Problem Definition

A mixed-model two-sided assembly line is comprised of serially linked workstations located in both left and right sides of the line. On the line, two or more product models are produced in an inter-mixed sequence. There is no need for set-up between the product model changes as the models are similar to each other. So, different models of a product can be assembled on the same line in any sequence that satisfies the customized product demands. Figure 1 represents a typical configuration of the mixed-model two-sided lines. The processing times of tasks belonging to two models (A and B) are represented by bars. The length of a particular bar corresponds to the processing time of the task given inside that bar. The gray shaded areas denote the unavoidable idle times usually caused by the precedence relationships and capacity constraints. As seen in Figure 1, there is a total of nine tasks performed in three workstations located in left and right sides of the line.

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

Table 1. Data for the case study

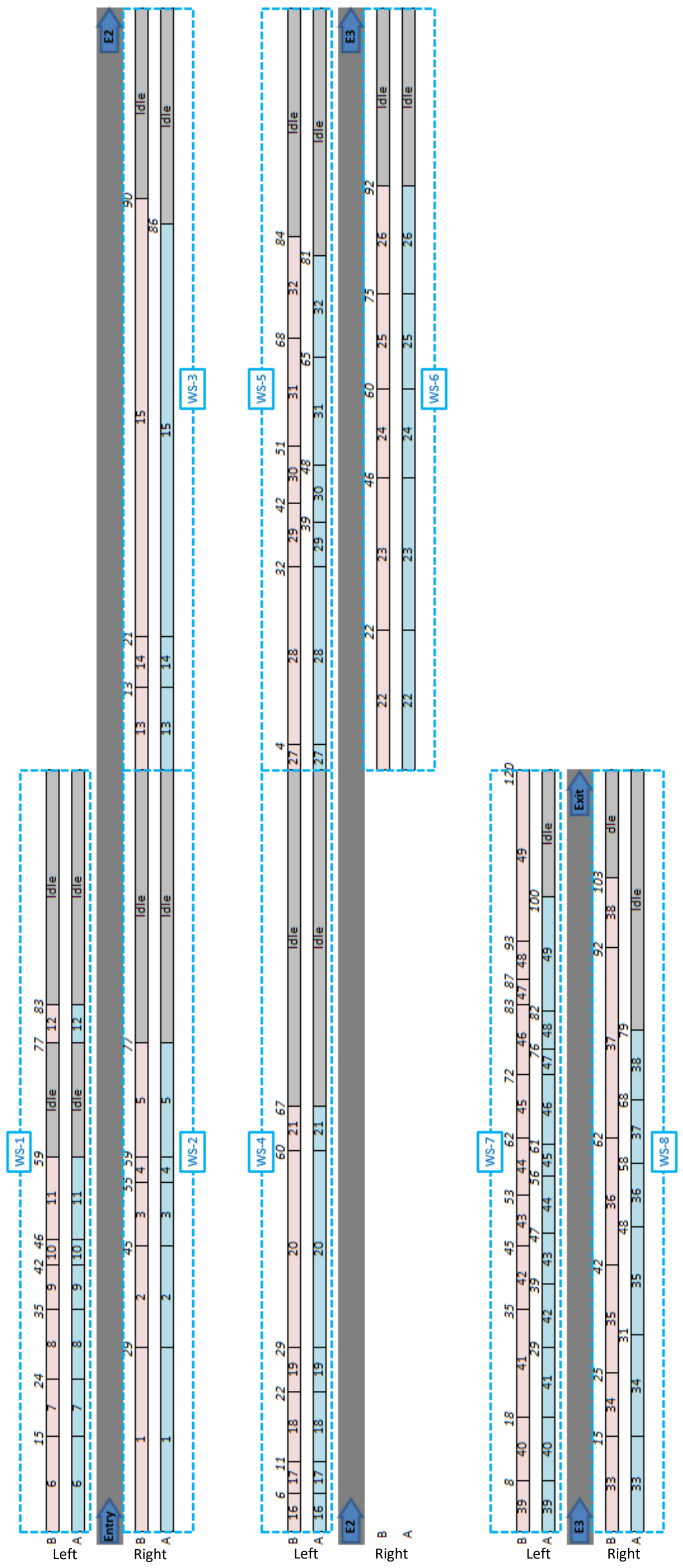
Figure 2. The assignment configuration of tasks in current situation

WS-1 and WS-2 constitutes a 'mated-station' and each of these workstations call the other one as its 'companion'. In mixed-model production, some tasks may not be needed for some models. For example, tasks 1 and 4 are not necessary for model B while tasks 3 and 8 are not necessary for producing model A (see WS-1 and WS-3 in the left side of the line).

The data related to the case study is gathered from Zhang et al. [7] and presented in Table 1. The processing times, precedence relationships and operation sides of 49 tasks are presented in the table. The letters reported in the 'Operation Side' column denote the side of the line in which the corresponding task must be performed; i.e. 'L' means left side, 'R' means right side and 'E' means either side. The tasks marked with asterisk (*) are in the same "*incompatible task group [7]"*, which means they cannot be performed on the same product, simultaneously. No detail information on incompatible task groups will be repeated here due to the page limit, please refer to Zhang et al. [7] for more information. The demands for models are assumed to be the same ($D_A = D_B$). The assignment configuration of tasks in its current form is presented in Figure 2. As seen from the figure, eight workstations are utilized while the companions of WS-3 and WS-4 are not employed. Also, it is seen that the workstation which determines the cycle time is WS-7 as it has the largest processing time of 120 time-units for model B. However, the idle times in the workstations occupy quite a lot of time.

III. SOLUTION METHOD

An ACO approach is proposed for solving the mixed-model two-sided assembly line balancing problem considered in this study. ACO algorithms are one of the most successful examples of swarm intelligent systems and have been successfully applied to a wide range of problem types. The behavior of each ant is inspired by the food searching mechanism of real ants [13] and their communication form with each other, in particular. Ants randomly walk on the path, at the beginning of the search process. After some time, the ant which finds a source of food walks back to the colony, leaving pheromone on the ground. This will attract other ants and they will follow the same path at a certain probability. By time, there will accumulate more pheromone on the shortest path from the nest to the food source [5, 14]. Please refer to Dorigo and Stützle [15, 16] for a comprehensive overview on advances and various applications of ACO algorithms.

The outline of the solution method proposed in this research is presented in Figure 3. As seen in the figure, after the initialization of the parameters, the theoretical minimum of cycle time (C_{min}) is calculated using (1) and accepted as the cycle time ($C \leftarrow C_{min}$) [7].

$$C_{min} = \max_{j \in J} \left\{ \left\lceil \frac{\sum_{i=1}^{N} t_{ij}}{K} \right\rceil^{+} \right\} \quad (1)$$

where N is the total number of tasks, t_{ij} is the processing time of task i for model j ($j \in J$) and K is the total number of workstations. $[X]^+$ denotes the least integer equals to or larger than X.

A new colony is released and a certain amount of pheromone (*initial_pheromone*) is deposited to all edges of the solution space. Please note that the pheromone is deposited between the task and the corresponding workstation in which it is assigned. A balancing solution is built by each ant in the colony considering C (using the procedure given in Figure 4). The selection probability of task i for the position k is calculated using (2) [16]. The algorithm is enhanced with 10 heuristics commonly used in the line balancing domain, and by Kucukkoc and Zhang [12].

$$p_{ik} = \frac{[\tau_{ik}]^{\alpha}[\eta_i]^{\beta}}{\sum_{y \epsilon Z_i}[\tau_{iy}]^{\alpha}[\eta_i]^{\beta}} \tag{2}$$

where α and β are weighting parameters which determine the influence of pheromone and heuristic information in the task selection process, respectively. Z_i is the list of candidate tasks when selecting task i. τ_{ik} is the pheromone amount existing between task i and workstation k, and η_i is the heuristic information of task i that comes from the heuristic selected randomly [10].

The line efficiency values of the solutions obtained are calculated using (3) and pheromones are updated using (4) [16].

$$LE\% = \frac{\sum_{j \varepsilon J}\sum_{i=1}^{N} d_j \, t_{ij}}{K \times C} \times 100 \tag{3}$$

where d_j is the proportional demand of model j ($d_j = D_j / \sum_{j \in J} D_j$), t_{ij} is the processing time of task i for model j and K is the total number of workstations.

$$\tau_{ik} \leftarrow (1-\rho)\tau_{ik} + \Delta\tau_{ik} \tag{4}$$

where $\Delta\tau_{ik} = Q/K$; ρ and Q denote the evaporation rate and a user-determined parameter.

C is increased by C_{inc} and new colonies are released to get new solutions and this cycle continues until C exceeds C_{max}. The best solution which gives the global best $LE\%$ value is determined as the solution of the problem and the algorithm is terminated.

The algorithm was coded in Java and run on an Intel Celeron® CPU N2840 2.16 GHz 4GB platform to solve the problem with the parameter setting of $\alpha = \beta = 0.1$, $\rho = 0.1$, $Q = 50$, $initial\_pheromone = 30$, $MaxNbColonies = 100$ and $ColonySize = 20$.

Figure 5 presents the best solution retrieved when the algorithm was terminated. The solution presented here fulfills all constraints explained in Section II as well as the incompatible task group constraints considered by Zhang et al. [7]. The length of each bar symbolizes the processing time of the task given in it. The finishing/starting times of the tasks are also given over bars to enable a more comprehensible reading. The gray shaded areas denote the unavoidable idle times occurred either towards the end of the workstations (due to capacity constraint) or between tasks (due to precedence or incompatibility constraints). A total of seven workstations is needed to perform all tasks within the cycle time of 107 time-units/item. This gives a line efficiency value of almost 90% when calculated using (3) ($C = 107$, $K = 7$). The efficiency of the line balance

```
**Initialize** the algorithm and all parameters
Calculate the minimum value for cycle time ($C_{min}$)
Initialize the cycle time ($C \leftarrow C_{min}$)
**While** $C < C_{max}$
  **While** *colony_number* < *MaxNbColonies*
    Release a new colony (*colony_number++*)
    Deposit *initial_pheromone* to all edges
    **While** *ant_number* < *ColonySize*
      Release a new ant (*ant_number++*)
      Select a heuristic rule at random
      Build a balancing solution
      Calculate the line efficiency of the solution found
      Update pheromone values
    **End while**
    Determine the best solution (which has the maximum line
    efficiency) in the colony
  **End while**
  Determine the global best solution (which has the maximum
  line efficiency) among the colonies
  $C \leftarrow C + C_{inc}$
**End while**
**Print** the best solution
```

Figure 3. The outline of the proposed algorithm

```
**Initialize** parameters
**While** unassigned tasks list is not empty ($UT \neq \emptyset$) **{**
  Select an operation side at random (left or right)
  Determine the available tasks for the current position
  **If** (there is at least one available task) **{**
    Calculate the selection probability of every available task
    Select a task ($i$) based on the selection probabilities
    Assign task $i$ to the current position and remove from $UT$
    Increase the station time for each model: $st(k)j \leftarrow$
    $st(k)j + (t_{ji})$
  **} else if** (there is no available task due to interference) **{**
    Increase the station time of the current workstation:
    $st(k) \leftarrow st(\underline{k})$, where $\underline{k}$ is the companion of
    workstation $k$ [5]
    Select an operation side at random
  **} else if** (there is no available task due to insufficient
  capacity) **{**
      **If** (both sides of the current line reached full capacity) **{**
          Increase the station number ($k++$)
      **} else if {**
          Alternate the operation side
      **} end if**
  **} end if**
**} end while**
```

Figure 4. The outline of the procedure followed by each ant to build a balancing solution

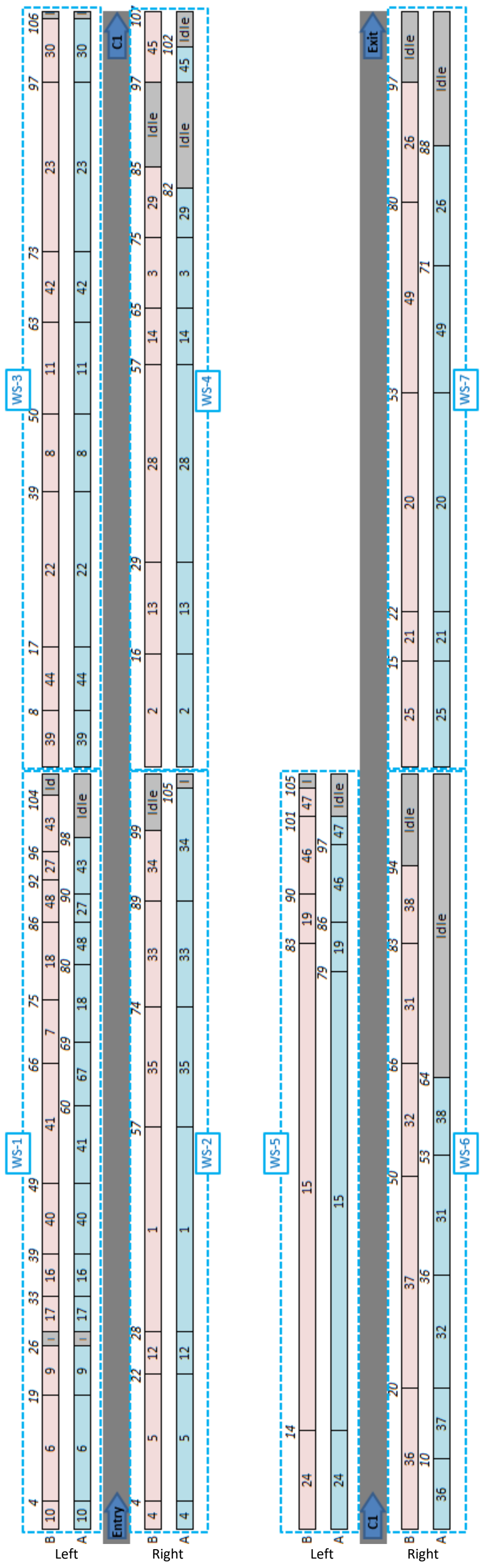

Figure 5. The assignment configuration of tasks according to the best solution obtained

before balancing it using the proposed ACO approach was slightly higher than 70 % (where $C = 120$, $K = 8$). This corresponds to a significant improvement of over 28% in the line efficiency in comparison to the line balance before balancing it. Thus, both the cycle time and the number of workstations have been reduced from 120 time-units/item and eight workstations to 107 time-units/item and seven workstations, respectively. This improvement also resulted in a smoother workload distribution among the workstations across the line. As seen from the figure, WS-3 has the largest workload with 106 time-units for both model A and model B. On the contrary, WS-6 has the smallest workload, i.e. 64 time-units for model A and 94 time-units for model B. The difference in the total workload time between the two models is due to the differences in the processing times of tasks 36 and 37 for models A and B.

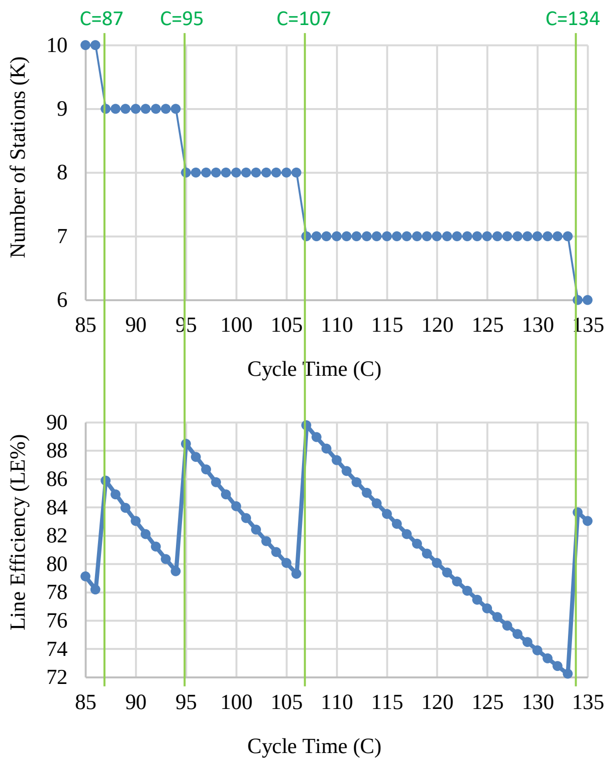

Figure 6. The change in the number of workstations and the line efficiency while the cycle time increases

Figure 6 shows the change in the number of workstations (K) and the line efficiency ($LE\%$) while the cycle time increases up to $C = 135$. As seen in the figure, the ACO algorithm found solutions with 10 workstations when $C = 85$ and $C = 86$ time-units/item. However, a solution with nine workstations ($LE\% = 85.9$) was found when C was increased to 87 time-units/item. Similar breakings have been observed when $C = 107$ and $C = 134$ time-units/item and with $LE\% = 89.8$ and $LE\% = 83.6$, respectively. This confirms that the maximum line efficiency of 89.8% was observed with seven workstations when $C = 107$ time-units/item.

IV. Conclusion

An ACO algorithm, enhanced with commonly used heuristic rules, was proposed for solving the mixed-model two-sided assembly line balancing problem multi-objectively. The aim was to minimize the cycle time of the line as well as the number of workstations (the problem of type-E). To the best of author's knowledge, this is the first attempt in the literature to minimize those conflicting objectives for the mixed-model two-sided lines. The data belonging to an industrial case study was gathered from Zhang et al. [7] and solved using the proposed algorithm coded in JAVA. The cycle time of the line was decreased from 120 time-units/item to 107 time-units/item and the number of workstations was reduced from eight to seven. Eventually, a significant improvement of over 28% was gained in the line efficiency and a smoother workload distribution was obtained in comparison with the situation before balancing the line. The ITG constraint originally introduced by Zhang et al. [7] has also been considered along with other essential constraints (e.g. capacity and precedence relationship). In terms of the practical applications of the study, the line managers can easily apply the method proposed in this paper to other similar problems in industry. Also, the ITG concept can be applied to other problems (e.g. parallel lines and U-shaped lines) or it can be extended (such that there are more than one ITG for the same system) for making it adaptable to more sophisticated implementations in real life. The development of a mathematical model for the type-E mixed-model two-sided assembly line balancing problem has been left to future studies.

References

[1] Boysen N, Fliedner M, Scholl A. A classification of assembly line balancing problems. European Journal of Operational Research. 2007;183(2):674-93.

[2] Battaïa O, Dolgui A. A taxonomy of line balancing problems and their solution approaches. International Journal of Production Economics. 2013;142(2):259-77.

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

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

[5] 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.

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

[7] Zhang DZ, Kucukkoc I, Karaoglan AD. Rebalancing of mixed-model two-sided assembly lines with incompatible task groups: An industrial case study. 46th International Conference on Computers & Industrial Engineering (CIE46), 29-31 October 2016, Tianjin, China2016.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Dorigo M, Di Caro G. The Ant Colony Optimization meta-heuristic. In: Corne D, (Ed.). New Ideas in Optimization. London, UK: McGraw Hill; 1999. p. 11-32.

[14] Dorigo M, Di Caro G, Gambardella LM. Ant Algorithms for Discrete Optimization. Artificial Life. 1999;5:137–72.

[15] Dorigo M, Stutzle T. Ant Colony Optimization: Bradford Books, MIT Press, Cambridge, MA; 2004.

[16] Dorigo M, Stutzle T. Ant Colony Optimization: Overview and Recent Advances. Handbook of Metaheuristics, Second Edition. 2010;146:227-63.