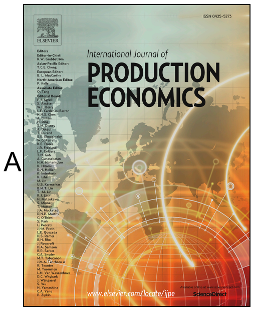


Balancing of mixed-model two-sided assembly lines with underground workstations: A mathematical model and ant colony optimization algorithm

Ibrahim Kucukkoc, Zixiang Li, Aslan D. Karaoglan, David Z. Zhang

PII: S0925-5273(18)30325-6

DOI: 10.1016/j.ijpe.2018.08.009

Reference: PROECO 7127

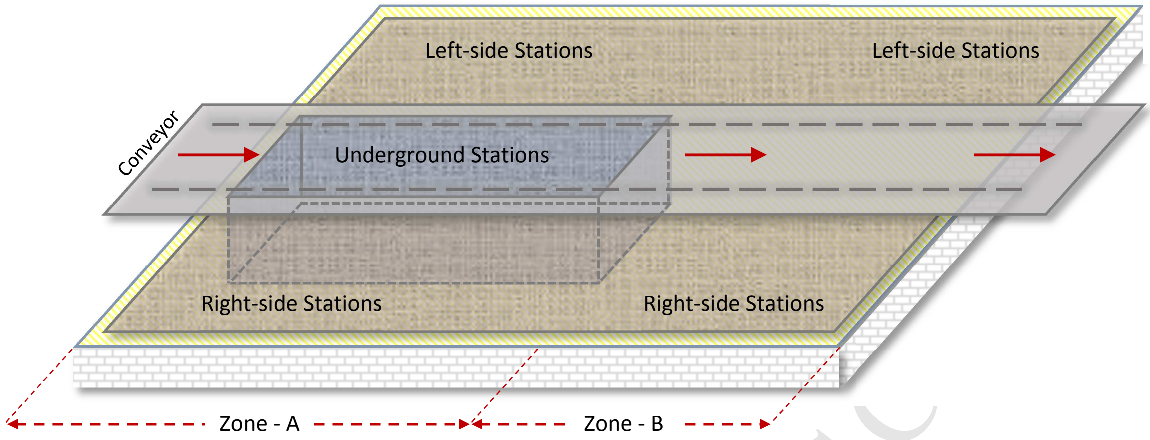
To appear in: *International Journal of Production Economics*

Received Date: 10 May 2017

Revised Date: 15 July 2018

Accepted Date: 7 August 2018

Graphical Abstract



Highlights

- Mixed-model two-sided assembly lines with underground stations is introduced.
- A mixed-integer linear programming (MILP) model is developed and coded in GAMS.
- Ant colony optimization (ACO) algorithm is developed and parameters are optimized.
- A new lower bound formulation is developed and a case study is conducted.
- Computational experiments verified the performance of the developed ACO algorithm.

Balancing of mixed-model two-sided assembly lines with underground workstations: A mathematical model and ant colony optimization algorithm

Ibrahim Kucukkoc[a1], Zixiang Li[b], Aslan D. Karaoglan[a], David Z. Zhang[c]

[a] *Industrial Engineering Department, Balikesir University, Cagis Campus, Balikesir 10145, Turkey*

[b] *Industrial Engineering Department, Wuhan University of Science and Technology, China*

[c] *College of Engineering, Mathematics and Physical Sciences, University of Exeter, North Park Road, Streatham Campus, Exeter EX4 4QF, United Kingdom*

Email: ikucukkoc@balikesir.edu.tr, zixiangliwust@gmail.com, deniz@balikesir.edu.tr, d.z.zhang@exeter.ac.uk

Abstract

Mixed-model assembly lines allow the production of different product variants in mass quantities on the same assembly line. In studies addressing mixed-model assembly with two-sided lines, assembly line stations are classified as left-side or right-side stations depending on the operation side to which they are allocated. However, underground stations are also utilized in industry to perform tasks that need to be done underneath the product being assembled on the line. This paper introduces and mathematically formulates a mixed-model, two-sided assembly line balancing problem considering underground stations. The precedence relationships between tasks being performed in the three types of stations are defined and considered in the model. A numerical example is solved in GAMS (with CPLEX solver) and the detailed balancing solution is provided. A new ant colony optimization algorithm, in which the parameters are optimized using response surface methodology, is also developed to solve real-world problems. A total of 78 test problems are derived from the literature and their lower bounds are calculated to test the performance of the ACO algorithm. ACO finds optimum solutions for the majority of small and medium-sized test problems. In comparing the ACO results to the lower bounds for the large-sized problems, ACO finds near optimum solutions in majority of the test cases.

[1] Corresponding Author. Tel: +902666121194 (6402)

Keywords: two-sided assembly line balancing; mixed-models; underground workstations; ant colony optimization; response surface methodology; MILP

Acknowledgment

The authors appreciate the constructive comments of the area editor Bart MacCarthy and anonymous reviewers which have helped increase the coherence and readability of the manuscript.

Balancing of mixed-model two-sided assembly lines with underground workstations: A mathematical model and ant colony optimization algorithm

Abstract

Mixed-model assembly lines allow the production of different product variants in mass quantities on the same assembly line. In studies addressing mixed-model assembly with two-sided lines, assembly line stations are classified as left-side or right-side stations depending on the operation side to which they are allocated. However, underground stations are also utilized in industry to perform tasks that need to be done underneath the product being assembled on the line. This paper introduces and mathematically formulates a mixed-model, two-sided assembly line balancing problem considering underground stations. The precedence relationships between tasks being performed in the three types of stations are defined and considered in the model. A numerical example is solved in GAMS (with CPLEX solver) and the detailed balancing solution is provided. A new ant colony optimization algorithm, in which the parameters are optimized using response surface methodology, is also developed to solve real-world problems. A total of 78 test problems are derived from the literature and their lower bounds are calculated to test the performance of the ACO algorithm. ACO finds optimum solutions for the majority of small and medium-sized test problems. In comparing the ACO results to the lower bounds for the large-sized problems, ACO finds near optimum solutions in majority of the test cases.

Keywords: two-sided assembly line balancing; mixed-models; underground workstations; ant colony optimization; response surface methodology; MILP

1. Introduction

An assembly line is a sequence of workstations (referred to as station hereafter) linked to each other by a conveyor or moving belt. A set of tasks (or operations) is performed in each station whose capacity is restricted by the cycle time (Li et al., 2017a). The cycle time is determined by the line manager based on the demand for the product(s) and the planning horizon in which they need to be completed (Baybars, 1986). The assembly line balancing (ALB) problem is assigning tasks to stations with the aim of optimizing one or more optimization criteria, i.e. minimizing the number of stations (referred to as type-I ALB problem), minimizing the cycle time (referred to as type-II ALB problem), or minimizing both simultaneously (referred to as type-E ALB problem) (Gurevsky et al., 2012). In doing so, certain constraints need to be satisfied, i.e. precedence relationship constraints between tasks (some tasks need to be completed prior to initializing others), capacity constraints (caused by the cycle time) and task occurrence constraints (every task need to be assigned to exactly one station) (Battaïa and Dolgui, 2013).

Since the launch of the first assembly line at Highland Park by Henry Ford and his colleagues (Ford, 2009), the assembly lines have faced major changes. Among those, the most significant one might be the transition from single-model lines to mixed-model lines. With the development of more consumer-centric production strategies, companies have converted their single-model lines into mixed-model lines on which more than one product model can be produced with no need for set up between model changes (Kucukkoc and Zhang, 2014b, Boysen et al., 2008). This is because the mixed-model lines enable producing customized products and such a strategy plays a key role in meeting rapidly changing customer demands in today's highly competitive global market (Ozcan and Toklu, 2009b).

The mixed-model assembly line balancing problem was brought to the attention of the academia by Thomopoulos (Thomopoulos, 1967) and it has been extended in several aspects later on. For example, Vilarinho and Simaria (2002), Akpinar and Bayhan (2011) and Battini et al. (2007) proposed approximate solution algorithms for solving the mixed-model assembly line balancing problem considering parallel workstations; while Akpinar et al. (2013) considered sequence-dependent setup times. Emde et al. (2010) investigated a computational evaluation of the objectives used for smoothing workload distributions. Kucukkoc et al. (2015) introduced and modeled the lexicographic objectives for hierarchical workload smoothing on mixed-model lines and Buyukozkan et al. (2016) proposed artificial bee colony and tabu search approaches with optimized parameters for solving the same problem. Stochastic task times have been taken into account by McMullen and Frazier (1997), Manavizadeh et al. (2012) and Ozcan et al. (2011) .

Assembly lines are classified as paced and unpaced lines depending on the usage of the material handling system across the workstations. Unpaced lines are also divided into two groups, synchronous and asynchronous lines (Öner-Közen et al., 2017). This paper focuses on the paced lines, where an automated transportation system (such as conveyor or moving belt) is utilized to move work-pieces to downstream workstations at a constant speed.

In terms of the configuration of the workstations across the line, assembly lines are also classified as one-sided lines and two-sided lines (see Boysen et al. (2007) and Battaïa and Dolgui (2013) for a comprehensive taxonomy of assembly line balancing problems). In a one-sided assembly line, only one side of the line (left or right) is utilized. Whereas, both left and right sides of the line are utilized in two-sided assembly lines (Li et al., 2016). Two-sided lines are mainly used for producing large-sized products (such as trucks and buses) in mass quantities. The two-sided assembly line balancing problem was introduced by Bartholdi (1993). While a great deal of research has been conducted on the two-sided assembly line balancing problem with single model production, very little has focused on the mixed-model two-sided lines despite their practical relevance. See, for example, Lee *et al*. (2001), Hu *et al*. (2008), Ozcan and Toklu (2010), Kim *et al*. (2000, 2009), Baykasoglu and Dereli (2008), Simaria and Vilarinho (2009), Ozcan and Toklu (2009a), Ozcan (2010), Ozbakir and Tapkan (2010, 2011), Chutima and Chimklai (2012), Purnomo *et al*. (2013), Li et al. (2017b, 2016) for heuristics and meta-heuristics; and Wu *et al*. (2008) and Hu *et al*. (2010) for exact solution approaches on single model two-sided lines.

The pioneering studies on balancing the mixed-model two-sided assembly line have been conducted by Simaria and Vilarinho (2009) and Ozcan and Toklu (2009b). Simaria and Vilarinho (2009) modeled the problem mathematically and proposed an ant colony optimization algorithm, of which the performance has been tested using benchmark problems. Özcan and Toklu (2009b) aimed to minimize the number of mated-stations (as the primary goal) and the number of stations (as the secondary goal) in their model. They also proposed a particle swarm optimization algorithm which employs two performance criteria to measure the solution quality: *(i)* maximizing the weighted line efficiency, and *(ii)* minimizing the weighted smoothness index. Another particle swarm optimization algorithm, which adopts negative knowledge, was also proposed by Chutima and Chimklai (2012) for the associated problem. Kucukkoc and Zhang (2014a, 2015a, 2016, 2014b) applied the line parallelization idea to mixed-model two-sided lines. However, none of those works considered the underground stations. One can refer to Make et al. (2016) for a comprehensive review of the literature on two-sided assembly line balancing problems (Kucukkoc and Zhang, 2016).

Kellegöz (2017) proposed another classification scheme based on the utilization of the workers in the workstations: (i) assembly lines with one worker for each workpiece, (ii) assembly lines with at most two workers for each workpiece, and (ii) assembly lines allowing multi workers for each workpiece. In that scheme, the first type refers to the traditional assembly line balancing problem while the second one is referred to as the two-sided assembly line balancing problem.

It should be noted here that multi-maned assembly line balancing problem has also been studied in the literature. In a multi-maned workstation, more than one worker can work on the same work-piece. This structure can be corresponded to the third problem type in the scheme proposed by Kellegöz (2017). As mentioned by Dimitriadis (2006), to ensure that the workers do not block each other during the work, the product should be of sufficient size. Dimitriadis (2006) proposed a heuristic procedure to balance the assembly line incorporating group working while Fattahi et al. (2011) proposed a mathematical model and ant colony optimization algorithm for the same problem. In Fattahi et al. (2011), it was attempted to minimize the total number of workers on the line as the first objective and the number of opened multi-manned workstations as the second one. Roshani and Giglio (2017) developed a simulated annealing algorithm to minimize the cycle time in multi-maned assembly lines. Roshani and Nezami (2017) extended the problem considering mixed-models and proposed a mathematical model and a simulated annealing approach to solve the problem.

It is clear from this survey that there is a lack of research on underground stations, which is the basic motivation of this work. While the work of Becker and Scholl (2009) analyzed the utilization of variable parallel workplaces across the line, the underground workstation structure was not addressed in that paper. To the best of authors' knowledge, all studies on two-sided lines (including Becker and Scholl (2009)) assume the composition of serially linked workstations located on both left and right sides of the line. Nevertheless, this is not the case in most practical production environments where some workstations are located under the physical line layout, referred to as underground workstations. For instance, in the car assembly industry, the tasks on motor-vehicle chassis must be operated underground. The main reason of utilizing underground workstations is that overturning the products is uneconomical or even technically infeasible. This situation can be observed in many factories assembling large-sized high-volume products, such as cars and trucks.

Underground stations allow the opportunity of performing tasks underneath the semi-product being assembled on the line simultaneously with the left and right-side tasks. The novelty of this paper is to bring the underground workstation concept to the attention of the academia and present the first attempt for solving mixed-model two-sided assembly line balancing problem with underground stations (MTALB-US). The first mathematical model, which aims at minimizing the number of mated stations and the number of stations, is developed and an ant colony optimization (ACO) algorithm is proposed for solving this new and practical problem. Test problems are derived from the literature and lower bounds are calculated using a modified formulation. A case study is also provided regarding the industrial implication of the problem studied.

The remainder of this paper is organized as follows. Section 2 defines the problem using illustrations and presents its mixed-integer linear programming model. A numerical example is also given together with its optimum solution obtained by the model presented. Section 3 provides the details of a meta-heuristic algorithm (developed for solving especially the large-sized problems) and the procedure used for parameter optimization. Section 4 reports the results of computational experiments and a case study whose data have been derived from a real manufacturing system. Section 5 concludes the paper and provides future research directions.

2. Problem statement

The concept of mixed-model two-sided assembly line system introduced in this research allows the utilization of stations located under the physical line, called underground stations. Figure 1 shows a typical representation of the two-sided assembly line with underground stations. As the utilization of underground workstations addresses to a specific physical feature, this problem is a specific case of general assembly line balancing problem. The utilization of underground stations enables the opportunity of performing tasks underneath the semi-product being assembled on the line. So that, the underneath tasks can be performed concurrently with the left and right side tasks which will eventually result in a shorter assembly line. Otherwise, in traditional approach studied in the literature extensively, the product model need to be lifted to overhead (or tilted) to allow operators work under the product. Due to the weight and volume of the products assembled, this is usually uneconomical or even technically infeasible in two-sided lines. Furthermore, lifting or tilting the products completely (or partially) obstructs other operators who can work on the left and/or right side tasks concurrently and requires more advanced material transportation equipment.

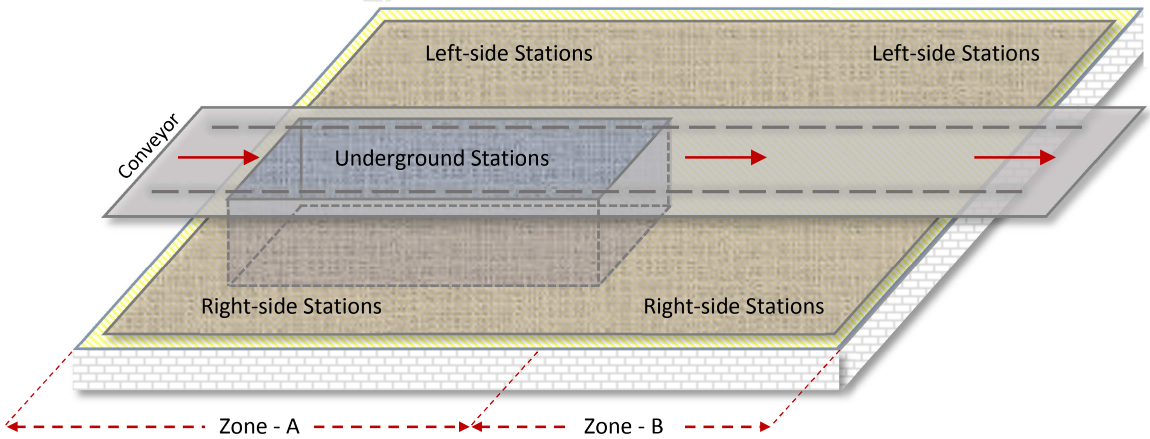


Figure 1. Illustration of a two-sided line with underground stations

The MTALB-US concept incorporates producing a total of nm large-sized product models, $M = \{1,2, \ldots, m, \ldots, nm\}$, in an inter-mixed sequence. A model mix determined in the sequencing phase based on the customer demand can be produced with no need for set-up between model changes, which is the main advantage of mixed-model lines (note that the sequencing problem is out of the scope of this paper). A set of nt tasks, $I = \{1,2, \ldots, i, \ldots, nt\}$, needs to be completed in a total of nj mated-stations, ($J = \{1,2, \ldots, j, \ldots, nj\}$. A mated-station may be composed of a left-side station, a right-side station and an underground station. Nevertheless, not all mated-stations may support the underground works due to the physical conditions of the line layout. In those mated-stations, only left and right side stations can be utilized. As seen from Figure 1, mated stations that will be located in Zone-A are able to handle underground tasks because there is physical space under the conveyor dedicated for underground workstations. Nevertheless, the workstations located in Zone-B are able to perform left and right tasks but not the underground ones. Each mated-station completes tasks assigned to it within the cycle time (CT) and the product moves to the downstream mated-station. The finishing time of any task assigned to any type of workstation (left, right or under) cannot exceed CT, which is determined by the line manager based on the planning horizon (P) and the total demand of the product models assembled on the line ($CT = P/(\sum_{m \in M} D_m)$, where D_m is the demand for model m). The tasks must be assigned to workstations considering the precedence relationships between them. For example, if tasks 2 and 4 are the predecessors of task 9 (i.e. $P_a(9) = \{2,4\}$, where $P_a(i)$ represents the set of predecessors of task i), task 9 can only be initialized after both of the tasks 2 and 4 are completed. This constraint is valid for all workstations regardless of their type. Thus, a task assigned to an underground station may have precedence relationships with those assigned to other type of stations (left and/or right side) as well as the underground stations. Therefore, different from the two-sided lines with no underground stations, the finishing times of the tasks performed in underground stations need to be considered when assigning tasks to left/right stations. Similarly, it is also required to check the finishing times of the tasks assigned to left/right stations when assigning tasks to underground stations. The violation of this rule causes infeasible solutions and is called interference.

The decision of determining where the underground stations will be located is important. The correct decision should be given by the line manager when the line is first utilized. Otherwise, some additional constructions may be needed in future if significant changes in the line layout are planned. Therefore, in type-I ALB problems (i.e. the number of workstations is minimized given the cycle time) it could be assumed that all mated-stations can support underground work. However, in type-II problems (i.e. the cycle time is minimized given the number of workstations) which are mostly encountered when rebalancing an existing line, some workstations have already been configured to support the underground works due to the construction at its initial phase. Therefore, the underground tasks must be allocated to those workstations which already have an underground working zone. In some occasions, due to the physical restrictions encountered in real world, the underground stations can be located to specific areas and the underground tasks can be restricted to be assigned to these areas. For example, assume a physical system composed of eight mated-stations. Only the zone that the second, third and fourth mated-stations to be located support the underground tasks. In this case, the underground tasks must be assigned to one of those three workstations.

2.1. Assumptions

The assumptions considered in the mathematical model which will be given in Section 2.2 are as follows:

- Operators can perform tasks in parallel at both sides of the line in two-sided mated-stations.
- Operators can perform tasks in parallel at three sides of the line in three-sided mated-stations.

- L-type tasks are required to be performed at left side, R-type tasks are required to be performed at right side and E-type tasks are required to be performed at left side or right side of the line.
- U-type tasks are required to be performed at the underground stations.
- Some mated-stations contain an underground station, while others do not have an underground station. It is known that which mated-stations contain an underground station.
- U-type tasks must be allocated to the mated-stations which contain an underground station.
- Product models, which do not require set-up time in between model changes, are produced on the same two-sided assembly line.
- Precedence diagrams of different models are known. The combined precedence diagram is employed.
- Task times are deterministic and independent of the assigned station.
- Common tasks among different models exist. A task completion time may differ from one model to another, and also it may be equal to zero.
- The travel times of operators are ignored and there are enough number of operators.
- Parallel tasks and parallel stations are not allowed, there exists only one operator working in a workstation.
- No work-in-process inventory is allowed.

The advantage of using a combined precedence diagram is to assign common tasks between different models to the same workstation, in parallel to the common tendency in real world applications and the studies in literature. This also helps to shrink the mathematical model to be developed in Section 2.2. Otherwise, every task would be represented specific for the models produced, which would increase the number of decision variables. On the other hand, the use of deterministic task times and ignoring the travel times of operators may limit the representation of real-world conditions but these assumptions need to be done due to the trade-off between the complexity and representability of the studied problem.

2.2. Mathematical model

The mixed-integer linear programming (MILP) model for the problem described in the previous subsection is developed based on the model formulated by Ozcan and Toklu (2009b). As the model proposed here incorporates a new workstation concept, called underground workstation (different from Ozcan and Toklu (2009b)); the parameters, variables, objective function and constraints are modified/expanded properly to fulfil this specification. For example, a new index is employed to represent the new side (under) and a new task set (AU) is constructed to hold tasks which should be performed at an underground station. The objective function and the constraints are modified to consider all sides while ensuring feasibility.

2.2.1. Notation

Indices:

i, h, p : The task index.

m : The product model index.

j, g : The mated-station index.

k, f : The side of the line; $k, f = \begin{cases} 1 & \textit{if the side is left} \\ 2 & \textit{if the side is right} \\ 3 & \textit{if the side is under} \end{cases}$.

(j, k) : The k side station of the mated-station j.

Parameters:

I : The set of tasks in the combined precedence diagram, $I = \{1,2, \dots, i, \dots, nt\}$.

J : The set of mated-stations, $J = \{1,2, \dots, j, \dots, nj\}$.

U : The set of mated-stations which allow underground tasks.

M : The set of product models, $M = \{1,2, \dots, m, \dots, nm\}$.

AL : The set of tasks which should be performed at a left side station, $AL \subseteq I$.

AR : The set of tasks which should be performed at a right side station, $AR \subseteq I$.

AE : The set of tasks which can be performed on left or right side of a mated-station, $AE \subseteq I$.

AU : The set of tasks which should be performed at an underground station, $AU \subseteq I$.

P_0 : The set of tasks that have no immediate predecessors.

$P_a(i)$: The set of all predecessors of task i.

$P(i)$: The set of immediate predecessors of task i.

$S_a(i)$: The set of all successors of task i.

$S(i)$: The set of immediate successors of task i.

$C(i)$: The set of tasks whose operation directions are opposite to that of task i,

$$C(i) = \begin{cases} AL \cup AU & if\ i \in AR \\ AR \cup AU & if\ i \in AL \\ AU & if\ i \in AE \\ AL \cup AR \cup AE & if\ i \in AU \end{cases}.$$

$K(i)$: The set of integers which indicate the preferred operation direction of task i,

$$K(i) = \begin{cases} \{1\} & if\ i \in AR \\ \{2\} & if\ i \in AL \\ \{1,2\} & if\ i \in AE \\ \{3\} & if\ i \in AU \end{cases}.$$

t_{im} : The processing time of task i for model m.

ψ : A large positive number.

CT : Cycle time.

PC : The set of pairs of tasks and predetermined stations for positional constraint.

PZ : The set of pairs of tasks for positive zoning constraint.

NZ : The set of pairs of tasks for negative zoning constraint.

SC : The set of pair of tasks for synchronism constraint.

w_1 : The weight of an opened mated-station.

w_2 : The weight of an opened workstation.

Decision variables:

x_{ijk} : 1, if task i is assigned to side k of mated-station j; 0, otherwise (for $\forall i \in I, j \in J, k \in K(i)$).

t_{im}^{f} : Continuous variable, the finishing time of task i for model m.

V_{jk} : 1, if station (j, k) is utilized for a model; 0, otherwise.

G_j : 1, if only one side of mated-station j is utilized; 0, otherwise.

F_j : 1, if both sides of mated-station j are utilized; 0, otherwise.

H_j : 1, if three sides of mated-station j are utilized; 0, otherwise.

Indicator variables:

$$z_{ip} = \begin{cases} 1 & if\ task\ i\ is\ assigned\ earlier\ than\ task\ p\ in\ the\ same\ station \\ 0 & if\ task\ p\ is\ assigned\ earlier\ than\ task\ i\ in\ the\ same\ station \end{cases}.$$

2.2.2. Objective function

The model aims to minimize the number of mated stations (which corresponds to the length of the line) as well as the number of stations. This is to maximize the utilization of earlier mated-stations as much as possible.

$$\text{Minimize } w_1 \cdot \sum_{j \in J} (H_j + F_j + G_j) + w_2 \cdot \sum_{j \in J} \sum_{k=1,2,3} V_{jk}. \quad (1)$$

2.2.3. Constraints

One of the essential constraints of an assembly line balancing problem is that each task must be assigned to exactly one side of a mated-station. This is ensured by Constraint (2).

$$\sum_{j\in J}\sum_{k\in K(i)} x_{ijk} = 1 \;\; \forall i \in I. \tag{2}$$

As explained, some mated-stations may not contain an underground station. Therefore, Constraint (3) guarantees that the U-type tasks to be allocated to a mated-station containing an underground station.

$$\sum_{j\in U}\sum_{k\in K(i)} x_{ijk} = 1 \;\; \forall i \in AU. \tag{3}$$

Constraints (4) and (5) correspond to that every task must be completed within the cycle time.

$$t_{im}^{f} \le CT \;\; \forall i \in I, m \in M. \tag{4}$$

$$t_{im}^{f} \ge t_{im} \;\; \forall i \in I, m \in M. \tag{5}$$

Constraints (6) – (9) deal with the precedence relationship constraint between tasks, which is another must have constraint in assembly line balancing problems. Assume two tasks, h and i. Constraint (6) relates to the situation that $h \in P(i)$ and these two tasks are assigned to different mated-stations.

$$\sum_{g\in J}\sum_{k\in K(h)} g\cdot x_{hgk} \le \sum_{j\in J}\sum_{k\in K(i)} j\cdot x_{ijk} \;\; \forall i \in I - P_0, h \in P(i). \tag{6}$$

Constraint (7) deals with the situation that $h \in P(i)$ and these two tasks are assigned to the same mated-station.

$$t_{im}^{f} - t_{hm}^{f} + \psi\left(1 - \sum_{k\in K(h)} x_{hjk}\right) + \psi\left(1 - \sum_{k\in K(i)} x_{ijk}\right) \ge t_{hm}\,, \quad \forall i \in I - P_0, h \in P(i), j \in J, m \in M. \tag{7}$$

Constraints (8) and (9) deal with the situation that these two tasks have no precedence relationship in between (i.e. $h \notin P(i)$ and $h \notin S(i)$). Please refer to (2009b) for detailed explanation.

$$t_{pm}^{f} - t_{im}^{f} + \psi\left(1 - x_{ijk}\right) + \psi\left(1 - x_{pjk}\right) + \psi\left(1 - z_{ip}\right) \ge t_{pm} \;\; \forall i \in I, m \in M, \quad p \in \{r \mid r \in I - \left(P_a(i) \cup S_a(i) \cup C(i)\right) \text{ and } i < r\}, j \in J, k \in K(i) \cap K(p). \tag{8}$$

$$t_{im}^{f} - t_{pm}^{f} + \psi(1 - x_{ijk}) + \psi(1 - x_{pjk}) + \psi \cdot z_{ip} \geq t_{im} \ \ \forall i \in I, m \in M, \quad (9)$$
$$p \in \{r | r \in I - (P_a(i) \cup S_a(i) \cup C(i)) \text{ and } i < r\}, j \in J, k \in K(i) \cap K(p).$$

Constraints (10) and (11) determine the numbers of workstations, two-sided mated-stations and three-sided mated-stations.

$$\sum_{i \in I} x_{ijk} - \psi \cdot V_{jk} \leq 0 \ \ \forall j \in J, k \in \{1,2,3\}. \quad (10)$$

$$\sum_{k \in \{1,2,3\}} V_{jk} - 3 \cdot H_j - 2 \cdot F_j - G_j = 0 \ \ \forall j \in J. \quad (11)$$

If a task must be assigned to a specific side of a specific mated-station (j, k), this is ensured by Constraint (12), which is called positional constraint.

$$x_{ijk} = 1 \ \ \forall (i, (j, k)) \in PC, k \in K(i). \quad (12)$$

If any two tasks have positive or negative constraints, this is guaranteed by Constraints (13) and (14).

$$x_{ijk} - x_{hjk} = 0 \ \ \forall (i, h) \in PZ, k \in K(i) \cap K(h). \quad (13)$$

$$\sum_{k \in K(i)} x_{ijk} + \sum_{k \in K(h)} x_{hjk} \leq 0 \ \ \forall (i, h) \in NZ. \quad (14)$$

Constraints (15-16) deal with the synchronism constraint. By this way, it is ensured that any two tasks which need to be done synchronously are assigned to the same mated-station and initialized at the same time.

$$x_{ijf} - x_{hjk} = 0 \ \ \forall (i, h) \in SC, k \in K(h), f \in K(i), k \neq f. \quad (15)$$

$$t_{im}^{f} - t_{im} = t_{hm}^{f} - t_{hm} \ \ \forall (i, h) \in SC, m \in M. \quad (16)$$

2.3. A numerical example

A numerical example is given here to represent the characteristics of the MTALB-US. A mixed-model two-sided line with two different product models (namely, A and B with equal demands) assembled on it is assumed. The first mated-station does not support underground works while the others do. The cycle time is restricted to 24 time-units/item. The combined precedence diagram of the problem (taken from Kim et al. (2000)) is composed of 24 tasks. Table 1 provides the data required. The execution times of tasks (in time-units) for model A and model B are given in the second and third columns. As seen, some tasks may not be needed for some models. For example, tasks 7 and 11 are not needed for model A while tasks 1 and 8 are not for model B. The processing times of these tasks are given as ‘0’ in the table. The forth column gives the operation side in which the corresponding task must be assigned, where ‘L’ and ‘R’ denote the left and right sides of the line, respectively. ‘E’ states either left or right side while ‘U’ corresponds to tasks those need to be assigned to and underground station. For instance, tasks 6 and 8 can be assigned to either left or right side while tasks 18, 19 and 22 can only be assigned to an underground station.

Table 1. The input data for the numerical example

Task No	Processing Time (time-units)		Operation Side	Immediate Successor(s)
	A	B		
1	3	0	L	11
2	7	9	L	5,6
3	7	9	R	6,7
4	5	7	R	15
5	4	6	L	8
6	3	4	E	9
7	0	4	R	10
8	3	0	E	12
9	6	9	E	12, 13, 14
10	4	0	E	14
11	0	4	L	16
12	3	8	L	17
13	3	8	E	18, 19
14	9	5	R	19
15	5	0	R	20
16	9	7	L	21
17	2	5	E	21
18	7	4	U	22
19	9	8	U	23
20	9	4	R	23, 24
21	8	3	L	-
22	0	8	U	-
23	9	7	R	-
24	9	7	E	-

The MILP model given in the previous subsection is coded in GAMS which provides a CPLEX solver platform embedded into it. The numerical example was solved by the model executed on an Intel Xeon CPU E3-1270 PC running at 3.5 GHz with 16 GB of RAM and the optimum solution was found within 127 seconds (see Table 2). As seen from the table, three mated-stations (composed of a total of six stations) are utilized to perform 24 tasks. In the first two mated-stations, only the left and right sides of the line are used. However, in the third mated-station, the underground station is utilized (to perform tasks 18, 19 and 22) as well as the right-side station. This is because those underground tasks can only be initialized after all of their predecessors have been executed.

Table 2. The assignment solution for the numerical example

	Mated-station 1	Mated-station 2	Mated-station 3
Left	2, 5, 9, 8, 10	1, 11, 12, 16, 21	
Right	3, 6, 4, 15, 7	13, 14, 17, 20	24, 23
Under			19, 18, 22

The detailed balancing configuration of tasks is also depicted in Figure 2. The length of each bar relates to the duration of task number given inside it while the numbers given over bars represent the finishing times. As seen from the figure, the left side of the first mated-station is fully utilized for both models. Therefore, this workstation has the highest workload time across the line. The smallest workload time belongs to the right side of the last mated-station. Due to the deviation in the task execution times between the two models, some workstations may be full for one model but not for the other one. In other words, it is possible to have idle time differences between the two models. It should be noted that task 23 can only be initialized after the completion of task 19 (at the underground station), which is one of the predecessors of task 23 (see the right side of the mated-station 3). Therefore, one time-unit of idle time is unavoidable for model B before initializing task 23.

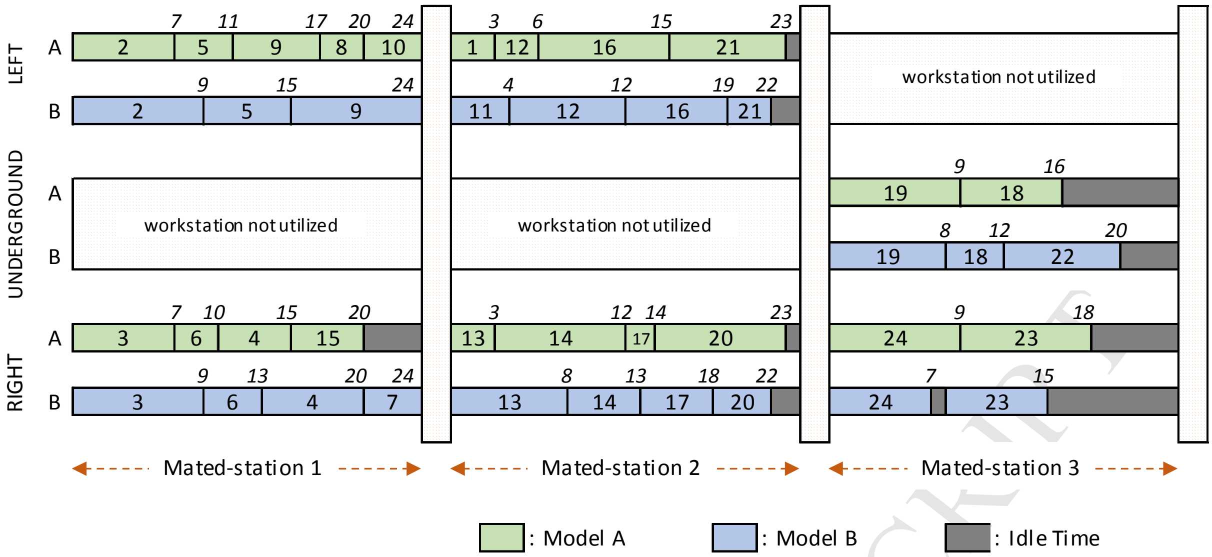


Figure 2: The detailed configuration of tasks for the optimum solution

3. Proposed approach

3.1. ACO algorithm

The mathematical model presented in Section 2.2 is coded in GAMS environment for solving the small and medium-sized test problems optimally (the results will be provided in Section 4). However, due to the NP-hard nature of the studied problem, it is not practical or sometimes impossible to solve the large-sized problems (even some medium-sized ones) using CPLEX by GAMS. Therefore, an ACO based algorithm is also developed and illustratively explored in this section. A total of 10 commonly used line balancing rules (information on which will be provided in Table 3) has been embedded into the ACO for providing heuristic information while selecting tasks. Thus, each ant has the opportunity of selecting a random behavior.

ACO is a nature-inspired algorithm which mimics the food-search mechanism of ants in nature. The ant system, which is considered to be the primitive version of ACO, was proposed by Dorigo et al. (1996) and since then ant algorithms have been applied to an extensive number of problems. Please see Dorigo and Blum (2005) for a comprehensive survey on ACO optimization theory and Kucukkoc and Zhang (2013) for their applications on ALB problems.

As in the formulation given for the mathematical model in Section 2.2, the ACO algorithm proposed in this paper aims to minimize the number of mated stations (NM) and the number of stations (NS) as the primary goal. Further on this, the algorithm incorporates another performance measure, smoothness index (SI), as a secondary goal. The objective function is composed of a weighted summation of NM and NS: $minimize\ Obj = w_1 \cdot NM + w_2 \cdot NS$, where w_1 and w_2 are the weighting parameters which control the significance of NM and NS.

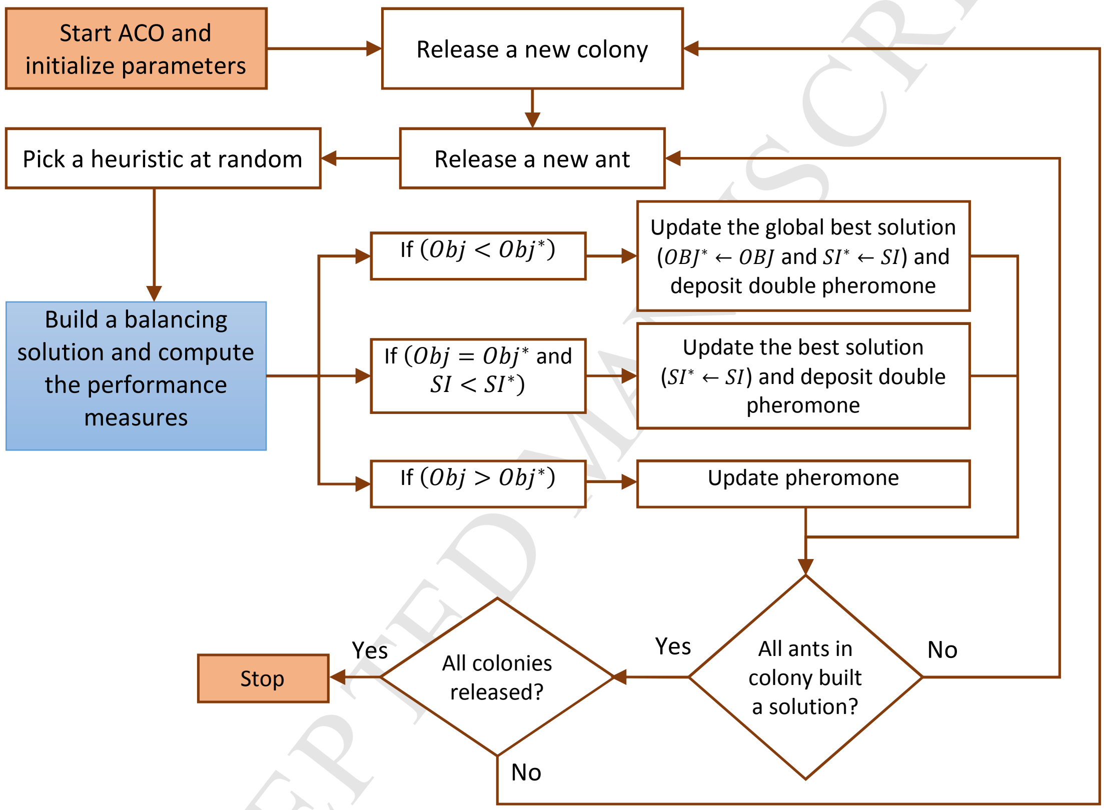


Figure 3. The outline of the proposed ACO algorithm

If any two solutions have the same objective function value, then the solution which has the smaller SI value is favored, $SI = 100 \times \left(\sqrt{\sum_{m \in M} d_m \sum_{j \in J} \sum_{k \in \{1,2,3\}} \left(MaxWT_m - WT_{m(j,k)}\right)^2}\right) / (CT \times NS)$, where $WT_{m(j,k)}$ is the workload time for model m at side j of mated-station k and $MaxWT_m$ is the maximum workload time for model m among workstations. If $WT_{m(j,k)}$ equals zero for all models at side j of mated-station k, this workstation is not considered in the calculation.

The outline of the proposed ACO algorithm is shown in Figure 3. As seen in the figure, the algorithm starts with initializing all parameters and releasing a new colony. The colony is composed of a limited number of ants (called colony size) released one-by-one to build balancing solutions. The common practice in most studies is to provide heuristic information from only one rule. However, this study provides each ant the opportunity of picking a rule randomly from Table 3 to search the solution space more efficiently. This decision is made regardless of the pheromone information of task and each heuristic has equal chance (1/10) to be selected. A balancing solution is obtained by each ant according to the procedure given in Figure 4 and the performance measure of the solution is computed. If the objective function value of the newly obtained solution (Obj) is worse than the current best solution ($Obj > Obj^*$) the pheromone on the edges of the solution obtained is updated.

Table 3. The heuristic rules employed by ACO

Rule	Selection Criterion
COMSOAL (Arcus, 1966)	Random $i \in AT_{jk}$
Smallest Task Number (Arcus, 1963)	$\min_{i \in AT_{jk}}\{i\}$
Shortest Processing Time (Baykasoglu, 2006)	$\min_{i \in AT_{jk}}\{t_{im}\}$
Largest Processing Time (Talbot and Patterson, 1984)	$\max_{i \in AT_{jk}}\{t_{im}\}$
Maximum Number of Predecessors	$\max_{i \in AT_{jk}}\{card(Pa(i))\}$
Least Number of Predecessors	$\min_{i \in AT_{jk}}\{card(Pa(i))\}$
Maximum Number of Successors (1960)	$\max_{i \in AT_{jk}}\{card(Sa(i))\}$
Least Number of Successors	$\min_{i \in AT_{jk}}\{card(Sa(i))\}$
Ranked Positional Weight (Helgeson and Birnie, 1961)	$\max_{i \in AT_{jk}}\{\sum_{m \in M} d_m t_{im} + \sum_{h \in \boldsymbol{Sa(i)}} \sum_{\boldsymbol{m \in M}} d_m t_{hm}\}$
Reversed Ranked Positional Weight	$\min_{i \in AT_{jk}}\{\sum_{m \in M} d_m t_{im} + \sum_{h \in \boldsymbol{Sa(i)}} \sum_{\boldsymbol{m \in M}} d_m t_{hm}\}$

AT_{jk} is the set of available tasks for position (j,k); d_m is the proportional demand of model m, $d_m = D_m/(\sum_{m \in M} D_m)$; and $card(X)$ denotes the cardinality (number of elements) of set X.

The pheromone is deposited between the tasks and the workstations in which they are assigned according to the formulation given in Eq. (17). If the newly obtained solution is better than the current best, it is accepted as the current best solution (if $Obj < Obj^*$: $Obj^* \leftarrow Obj$ and $SI^* \leftarrow SI$; if $Obj = Obj^*$ but $SI < SI^*$: $SI^* \leftarrow SI$) and double amount of pheromone is released.

$$\tau_{i(j,k)} \leftarrow (1-\rho)\tau_{i(j,k)} + \Delta\tau_{i(j,k)} \quad (17)$$

where $\Delta\tau_{i(j,k)} = Q/Obj$; ρ and Q denote the evaporation rate and a user-determined parameter, respectively.

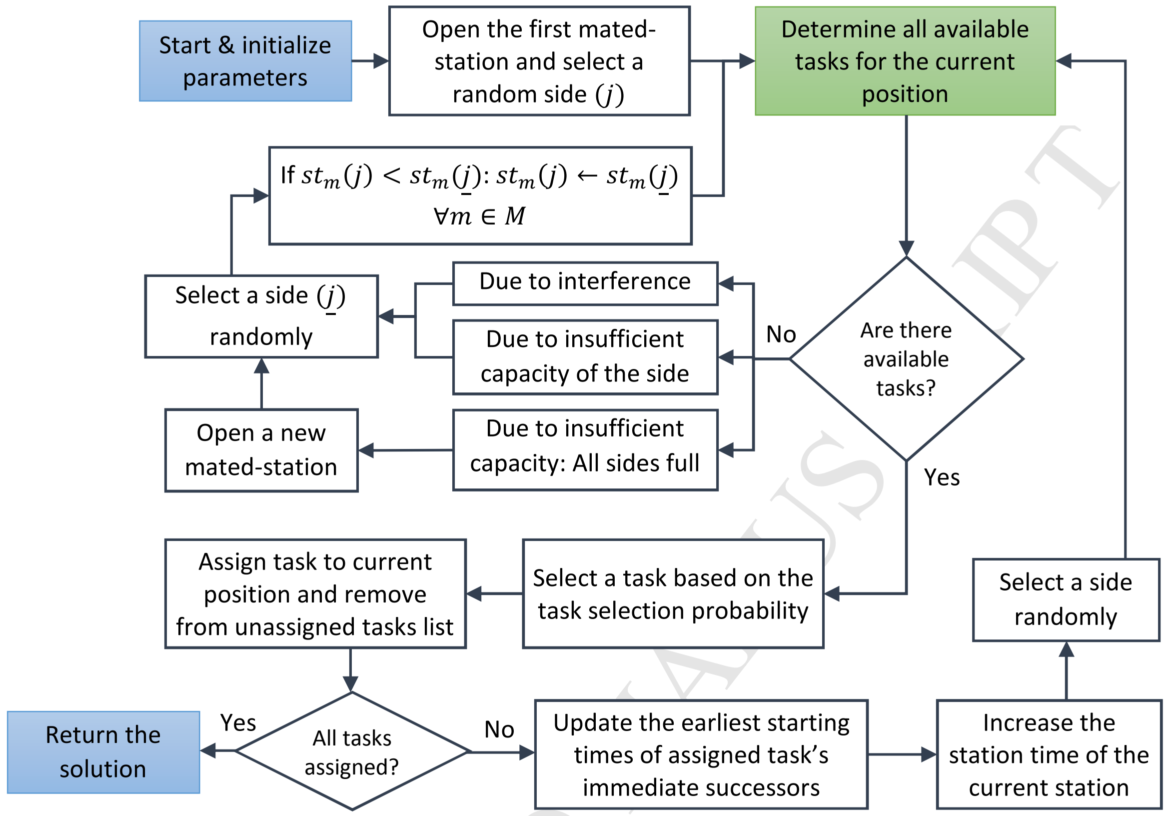


Figure 4. The flow-chart of the procedure used for building a balancing solution

A new ant, which builds another new balancing solution, is released. If a better solution is found, the best solution is updated and this cycle continues until all ants in the colony come up with a solution. When the whole colony completes its tour, a new colony is released and all ants in the new colony find solutions by updating best solutions (if any). This cycle continues until all colonies complete their tour and the algorithm is terminated reporting the best solution investigated.

The performance of a heuristic/meta-heuristic algorithm mainly depends on its decomposition method as well as other factors, such as parameters values, termination criterion and the representation of the solution. The solution building procedure used in the ACO proposed here opens the first mated-station and selects a random side (left, right or under). This decision is made giving all sides equal chance (1/3) to increase the diversity of the individuals created. Available tasks are determined for the current position (the side of the mated-station) based on the following criteria: (i) Operation side of the task is compatible, (ii) remaining capacity is enough to perform the task for all models and (iii) all predecessors of the task have been assigned and completed for all models prior to the initialization of this task. Thus, it is ensured that the solutions generated by the algorithm are always feasible. The selection probability of task i for the position (j,k) is calculated using Equation (18) (Dorigo and Stutzle, 2010) making use of the heuristic information obtained from 10 heuristics commonly used in the line balancing domain (see Table 3). Basically, the task which has the higher $p_{i(j,k)}$ value has the higher selection probability for the corresponding position.

$$p_{i(j,k)} = \frac{\left[\tau_{i(j,k)}\right]^{\alpha}[\eta_i]^{\beta}}{\sum_{h\epsilon Z_i}\left[\tau_{h(j,k)}\right]^{\alpha}[\eta_h]^{\beta}} \tag{18}$$

where α and β are weighting parameters which determine the relative importance of pheromone and heuristic information in the task selection process, respectively. Z_i is the list of candidate tasks when selecting task i. $\tau_{i(j,k)}$ is the pheromone amount existing between task i and position (j,k), and η_i is the heuristic information of task i that comes from the heuristic selected randomly (Kucukkoc and Zhang, 2016).

Regarding building a balancing solution given in Figure 4, when task i is assigned to a workstation, the earliest starting times of its successors are set to the finishing time of task i for each model, t_{im}^{f}. By this way, it is ensured that no task can be initialized before the completion of all its predecessors. The station time of the current workstation is also set to t_{im}^{f} and a random side is selected (left, right or under). If there is no available task for the current side (j), a new side ($\underline{j}$) is selected randomly as long as there is an available side whose capacity is not full. If the station time of the new side for model m ($st_m(\underline{j})$) is larger than that of the current side ($st_m(j)$), $st_m(j)$ is set to $st_m(\underline{j})$. When the capacities of all sides for the current mated-station come up short and there is no available task for this reason, a new mated station is opened and the assignment continues in this way until there is no task unassigned.

3.2. Parameter optimization with response surface methodology

Response surface methodology (RSM), a well-known design of experiment technique proposed by Box and Wilson (1951), is used for optimizing the parameters of the ACO. It is a combination of statistical and mathematical techniques used for modeling the mathematical relations between the inputs (called factors) and outputs (called response) of a process (Kucukkoc et al., 2013). Please refer to Box and Wilson (1951) and Khuri and Mukhopadhyay (2010) for more information on RSM. Table 4 presents the factors (ACO parameters) and their levels used for the experimental design. The factors and levels have been determined in accordance with the similar studies and common practice (see Kucukkoc and Zhang (2015b) and Buyukozkan et al. (2016)). The test problem #36, of which the details will be given in Section 4, is selected for the experiments and two responses are aimed to be optimized, namely mean NS (Y_{NS}) and mean SI (Y_{SI}) in parallel with the objective used for the ACO algorithm. The Obj value is not considered as a direct response as it is the minimum of all solutions obtained during a number of iterations (or colonies) given in the table (see x_4 row).

Table 4. Factors and levels

Factor	Symbol	Level		
		Lower	Medium	Higher
Alpha	x_1	0.1	0.5	0.9
Beta	x_2	0.1	0.5	0.9
Evaporation Rate	x_3	0.05	0.275	0.5
Number of Colonies	x_4	20	260	500

Minitab-16 statistical software was used to create an experimental design for the factors according to the central composite design (given in Table 5) and establish mathematical models for optimization of the ACO parameters. The ACO algorithm was run for solving test problem #36 using the parameters in each experiment listed in the design and the responses (Y_{NS} and Y_{SI}) are reported in Table 5.

When the results are analyzed, the full quadratic regression equations of Y_{NS} and Y_{SI} for the experimental design given in Table 5 are obtained as follows.

$$Y_{NS} = 11.8569 - 0.2286x_1 + 0.0338x_2 - 0.1483x_3 - 0.0032x_4 + 0.0425x_1^2 + 0.0737x_2^2 + 0.0355x_3^2 + 0.0000041x_4^2 + 0.0039x_1x_2 + 0.3402x_1x_3 + 0.0000065x_1x_4 + 0.0763x_2x_3 - 0.000058x_2x_4 - 0.000011x_3x_4. \quad (19)$$

$$Y_{SI} = 30.1796 - 0.7417x_1 - 0.2199x_2 - 0.4366x_3 - 0.0077x_4 + 0.0419x_1^2 + 0.6044x_2^2 + 0.4288x_3^2 + 0.000010x_4^2 + 0.1875x_1x_2 + 0.9861x_1x_3 + 0.0002x_1x_4 + 0.0417x_2x_3 - 0.0003x_2x_4 - 0.0003x_3x_4. \quad (20)$$

Table 5. Experimental design with uncoded values of factors

Run	Factors				Responses	
	x_1	x_2	x_3	x_4	Y_{NS}	Y_{SI}
1	0.1	0.1	0.05	20	11.75	29.95
2	0.9	0.1	0.05	20	11.65	29.38
3	0.1	0.9	0.05	20	11.91	30.31
4	0.9	0.9	0.05	20	11.70	29.92
5	0.1	0.1	0.5	20	11.75	29.94
6	0.9	0.1	0.5	20	11.73	29.83
7	0.1	0.9	0.5	20	11.82	30.19
8	0.9	0.9	0.5	20	11.86	30.20
9	0.1	0.1	0.05	500	11.23	28.55
10	0.9	0.1	0.05	500	11.13	28.28
11	0.1	0.9	0.05	500	11.29	28.80
12	0.9	0.9	0.05	500	11.17	28.40
13	0.1	0.1	0.5	500	11.22	28.53
14	0.9	0.1	0.5	500	11.15	28.32
15	0.1	0.9	0.5	500	11.29	28.73
16	0.9	0.9	0.5	500	11.30	28.83
17	0.1	0.5	0.275	260	11.27	28.64
18	0.9	0.5	0.275	260	11.20	28.41
19	0.5	0.1	0.275	260	11.19	28.42
20	0.5	0.9	0.275	260	11.29	28.81
21	0.5	0.5	0.05	260	11.22	28.53
22	0.5	0.5	0.5	260	11.24	28.55
23	0.5	0.5	0.275	20	11.70	29.55
24	0.5	0.5	0.275	500	11.23	28.60
25	0.5	0.5	0.275	260	11.24	28.60
26	0.5	0.5	0.275	260	11.24	28.63
27	0.5	0.5	0.275	260	11.25	28.60
28	0.5	0.5	0.275	260	11.23	28.50
29	0.5	0.5	0.275	260	11.25	28.64
30	0.5	0.5	0.275	260	11.27	28.69
31	0.5	0.5	0.275	260	11.25	28.66

According to the ANOVA results, the p-value of the full quadratic model for Y_{NS} has been obtained as 0.000 ($p = 0.000 < 0.05$), where $R^2 = 99.27$. The Minitab analysis report also denoted that the value of R^2 is 95.22% ($R^2_{pred} = 95.22$) for predicting response of a new experiment except those reported in Table 4. On the other hand, the values for the same items have been computed as $p = 0.000 < 0.05$, $R^2 = 98.30\%$ and $R^2_{pred} = 91.84\%$ for the full quadratic model belonging to Y_{SI}.

The fitted values and prediction errors of the responses are given in Table A 1 (see Appendix). The optimization results indicated that the optimum parameters for ACO are $Alpha = 0.9$, $Beta = 0.1081 \cong 0.11$, $Evaporation\ Rate = 0.05$, and $Number\ of\ Colonies = 394$. The optimization plot is also presented in Figure A 1 (see Appendix).

4. Computational experiments and case study

4.1. Lower bound calculation

For small and medium-sized test problems, of which the optimum solutions have been found through CPLEX, the performance of the proposed ACO has been compared to those obtained from CPLEX. However, this is not the case for large-sized test problems or even some medium-sized problems. Therefore, the lower bounds for NM and NS have been calculated for each test problem using the formulation given in Equations (21) – (28), adapted from Özcan and Toklu (2009b). Thus, although the optimum solutions are not known, a relatively fair comparison can be made for those problems.

The formulation of Özcan and Toklu (2009b) is not used as in its original form because Özcan and Toklu (2009b) calculated the lower bounds using the weighted task times and lower bound calculation method as shown in Hu et al. (2008). The formulation of Özcan and Toklu (2009b) suits the weighted task time situation and is not efficient for non-weighted situation where each model must be finished within a cycle time, such as the current study. Also, a new side (under) is added over the traditional two-sided balancing practice. As a result of these, this paper presents a modified lower bound calculation for mixed-model two-sided assembly line balancing problem with underground stations.

$$S_m^L = \lceil \textstyle\sum_{i \in AL} t_{im} / CT \rceil^+ \quad (21)$$

$$S_m^R = \lceil \textstyle\sum_{i \in AR} t_{im} / CT \rceil^+ \quad (22)$$

$$S_m^U = \lceil \textstyle\sum_{i \in AU} t_{im} / CT \rceil^+ \quad (23)$$

$$S_m^E = \left\lceil max\left(\left(\textstyle\sum_{i \in AE} t_{im} - \left((S_m^L + S_m^R) \cdot CT - \textstyle\sum_{i \in AL} t_{im} - \textstyle\sum_{i \in AR} t_{im}\right)\right), 0\right) / CT \right\rceil^+ \quad (24)$$

$$LB_m^{Ns} = S_m^L + S_m^R + S_m^E + S_m^U \quad (25)$$

$$LB_m^{Nm} = max\left(S_m^L + \left\lceil max\left((S_m^E - |S_m^L - S_m^R|), 0\right)/2\right\rceil^+, S_m^R + \left\lceil max\left((S_m^E - |S_m^L - S_m^R|), 0\right)/2\right\rceil^+, S_m^U\right) \quad (26)$$

$$LB^{Ns} = \max_m\{LB_m^{Ns}\} \quad (27)$$

$$LB^{Nm} = \max_m\{LB_m^{Nm}\} \quad (28)$$

where, S_m^L, S_m^R and S_m^U are calculated using the processing times of tasks for left, right and underground tasks for each model (see Equations (21)-(23)). S_m^E is then calculated in Equation (24) using S_m^L and S_m^R as well for model m. These can be assumed to be the theoretical minimum number of workstations for each side for each model. The expression $\lceil X\rceil^+$ denotes the least integer number equals or greater than X. LB_m^{Nm} and LB_m^{Ns} are the lower bounds on the number of mated-stations and the number of workstations for model m, calculated using the values obtained in the previous steps. Eventually, LB^{Nm} and LB^{Ns}, respectively the lower bounds on the number of mated-stations and the number of stations for the whole problem, are determined in Equations (27)-(28).

4.2. Computational tests

Computational tests have been performed by solving the modified test problems majorly derived from the literature (see Table 6) using the MILP model coded in GAMS and the ACO algorithm coded in Java (with optimum parameters obtained in Section 3.2).

The aim here is to provide the optimum solutions of the problems as long as the hardware/software capabilities allow and to test the performance of the developed ACO algorithm. All experiments have been conducted on the same PC, an Intel Xeon CPU E3-1270 PC running at 3.5 GHz with 16 GB of RAM. The weighting parameters, w_1 and w_2, were assumed to be 100 and 1, respectively, for all test problems. The initial pheromone is assumed 30 for all problems; while the colony size is assumed 20, 30 and 50 for small, medium and large-sized problems, respectively. All workstations are assumed to support the underground works. Demands for models are assumed to be $D_a = 40$, $D_b = 30$ and $D_c = 30$ for models A, B and C, respectively. These values have been used by ACO in determining the heuristic information of tasks when calculating the task selection probability (which has been explained in Section 3.1).

Table 6. The source of data used for test problems

Size	Problem	Precedence Relationships	Task Times	Operation Sides
Small	P9/1	Kim et al. (2000)	Adapted from Özcan and Toklu (2009)	Adapted from Kim et al. (2000)
	P9/2		Newly generated	
	P12/1		Adapted from Özcan and Toklu (2009)	
	P12/2		Newly generated	
	P16/1	Lee et al. (2001)	Adapted from Özcan and Toklu (2009)	Adapted from Lee et al. (2001)
	P16/2		Newly generated	
Medium	P20	Kucukkoc and Zhang (2015a)	Kucukkoc and Zhang (2015a)	Adapted from Kucukkoc and Zhang (2015a)
	P24/1	Kim et al. (2000)	Adapted from Özcan and Toklu (2009)	Adapted from Kim et al. (2000)
	P24/2		Newly generated	
	P36	Kucukkoc and Zhang (2015a)	Kucukkoc and Zhang (2015a)	Adapted from Kucukkoc and Zhang (2015a)
Large	P65/1	Lee et al. (2001)	Adapted from Özcan and Toklu (2009)	Adapted from Lee et al. (2001)
	P65/2		Newly generated	
	P148/1	Bartholdi (1993)	Adapted from Özcan and Toklu (2009)	Adapted from Bartholdi (1993)
	P148/2		Newly generated	
	P205/1	Lee et al. (2001)	Adapted from Yuan et al. (2015)	Adapted from Lee et al. (2001)
	P205/2		Adapted from Delice et al. (2014)	

As seen from Table 6, the benchmarks have been divided into three groups based on their size (i.e. the number of tasks they include): (i) small-sized problems (variants of P9, P12 and P16), medium-sized problems (variants of P20, P24 and P36) and large-sized problems (variants of P65, P148 and P205). For P9/1, P9/2, P12/1, P12/2, P24/1 and P24/2; the precedence relations were taken from Kim et al. (2000) and the operation directions were adapted from Kim et al. (2000). For P16/1, P16/2, P65/1, P65/2, P205/1 and P205/2; the precedence relations and operation directions were taken and adapted from Lee et al. (2001). The precedence relations and task times of P20 and P36 were taken from Kucukkoc and Zhang (2015a) while their operation directions were adapted from the same study. The precedence relations and operation directions of P148/1 and P148/2 were taken from and adapted from Bartholdi (1993). The task times of P9/1, P12/1, P16/1, P24/1, P65/1 and P148/1 were adapted from Özcan and Toklu (2009b) while those of P9/2, P12/2, P16/2, P24/2, P65/2 and P148/2 were newly generated randomly. The task times for P205/1 and P205/2 were adapted from Yuan et al. (2015) and Delice et al. (2014), respectively.

ACO was run 2, 4 and 6 times for each of the small, medium and large-sized problems, respectively, and the best results have been reported in Table 7. As seen, problems (given in the second column) have been solved under various cycle time constraints (see the CT column) and the optimum results have been reported in the *OPT by CPLEX* column (in $NM(NS)$ format). The column nj *(CPLEX)* presents the total number of mated-stations allowed (as indicated in the mathematical model in Section 2.2) to run the GAMS code. The results obtained from the proposed ACO algorithm have been reported in the *ACO NM(NS)* column. The lower bounds on the number of mated-stations (LB^{Nm}) and the number of workstations (LB^{Ns}) have also been computed for each problem using Equations (21) – (28) coded in Java and reported in the $LB^{Nm}(LB^{Ns})$ column. The final two columns correspond to the average and standard deviation of the CPU time (s) needed to run ACO, which seem quite reasonable.

Table 7. Results of the computational tests for small and medium-sized test problems

#	Problem	CT	$LB^{Nm}(LB^{Ns})$	nj (CPLEX)	OPT by CPLEX $NM(NS)$	ACO $NM(NS)$	ACO CPU Time (s)	
							Average	Std. Dev.
1	P9/1	4	2(4)	4	**3(5)**	**3(5)**	11.30	0.20
2		5	2(4)		**2(4)**	**2(4)**	11.30	0.10
3		6	1(3)		**2(4)**	**2(4)**	10.55	0.25
4		7	1(3)		**1(3)**	**1(3)**	11.25	0.05
5		8	1(3)		**1(3)**	**1(3)**	11.50	0.10
6	P9/2	5	2(5)	4	**3(5)**	**3(5)**	11.65	0.25
7		6	2(4)		**2(5)**	**2(5)**	11.25	0.25
8		7	2(4)		**2(4)**	**2(4)**	11.30	0.10
9		8	1(3)		**2(4)**	**2(4)**	11.50	0.10
10		9	1(3)		**1(3)**	**1(3)**	11.55	0.05
11	P12/1	5	2(5)	4	**3(6)**	**3(6)**	11.25	0.25
12		6	2(5)		**3(5)**	**3(5)**	11.65	0.15
13		7	2(4)		**2(4)**	**2(4)**	11.60	0.30
14		8	1(3)		**2(3)**	**2(3)**	11.40	0.30
15	P12/2	5	2(5)	4	**3(6)**	**3(6)**	10.65	0.15
16		6	2(5)		**3(5)**	**3(5)**	10.85	0.45
17		7	2(4)		**2(4)**	**2(4)**	11.10	0.50
18		8	1(3)		**2(3)**	**2(3)**	11.30	0.20
19	P16/1	13	3(7)	5	5(8)	5(9)	12.60	0.10
20		14	3(6)		5(8)	5(9)	11.40	0.30
21		15	3(6)		**4(7)**	**4(7)**	11.65	0.45
22		16	3(6)		**3(7)**	**3(7)**	12.10	0.30
23		17	2(5)		**3(7)**	**3(7)**	12.60	0.30
24	P16/2	16	3(7)	5	**5(9)**	**5(9)**	12.50	0.10
25		17	3(7)		**4(8)**	**4(8)**	12.00	0.40
26		18	3(6)		**4(8)**	**4(8)**	11.20	0.00

27		19	3(6)		**4(8)**	**4(8)**	12.55	0.05
28		21	3(6)		**4(7)**	**4(7)**	12.35	0.35
29	P20	18	3(6)	5	4(7)	4(8)	14.38	0.29
30		19	3(6)		**4(6)**	**4(6)**	14.23	0.59
31		20	3(6)		**4(6)**	**4(6)**	16.13	0.33
32		21	2(5)		**3(6)**	**3(6)**	13.25	0.65
33		22	2(5)		**3(6)**	**3(6)**	14.33	0.24
34	P24/1	18	4(9)	5	**4(10)**	**4(10)**	14.55	0.33
35		20	4(9)		**4(9)**	**4(9)**	14.05	0.39
36		21	3(8)		**4(8)**	**4(8)**	13.15	0.40
37		22	3(8)		**3(8)**	**3(8)**	14.43	0.25
38		23	3(8)		**3(8)**	**3(8)**	13.30	0.31
39	P24/2	16	4(9)	5	4(11)	5(11)	14.33	0.34
40		17	4(9)		**4(9)**	**4(9)**	14.25	0.22
41		18	4(9)		**4(9)**	**4(9)**	13.50	0.46
42		19	3(8)		**3(8)**	**3(8)**	14.33	0.24
43		20	3(8)		**3(8)**	**3(8)**	14.63	0.36
44	P36	18	5(10)	-	-	5(12)	48.65	1.07
45		19	4(9)	-	-	5(11)	49.40	0.68
46		20	4(9)	-	-	5(11)	49.25	0.49
47		21	4(9)	-	-	5(11)	50.40	0.43
48		22	4(8)	-	-	4(10)	51.30	0.65

As seen from Table 7, the optimal solutions of the first 43 test problems have been reported. As expected the optimum solutions of the problems usually differ from the lower bound, especially when the problem size gets bigger. This is due to the precedence relationships among tasks and the irregular distribution of processing times between tasks/models, which have not been considered in the theoretical lower bound calculation. Therefore, the performance of the ACO algorithm will be compared to OPT result where applicable. The comparison to lower bound value will be used as an indicator of the performance of the ACO where the OPT result is not available. As reported in the table, the ACO algorithm performs quite well for the small and medium-sized test problems. The optimum results have been investigated by ACO for 39 small and medium-sized test problems out of 43 (see boldface results given in the table). Although ACO performs very well for the majority of the problems, it was unable to find the optimum solutions of #19, #20, #29 and #39. The solutions found by ACO for the three of these problems (namely #19, #20 and #29) require one more workstation than that of the optimum solution. For #39, although the number of workstations is the same for both ACO and OPT, the line is longer in the ACO solution as it requires one more mated-station. For test problems #44-#48 (variants of P36), due to the size and so the complexity of the problem, CPLEX fell short in obtaining the optimum solution within 48 hours of computational time. When the ACO results found for these problems (#44-#48) are compared to the lower bound values, it is clear that ACO solutions are of optimum or near optimum. This interpretation is also supported by the difference between the lower bound values and the optimum results for even the small-sized problems.

The results of the computational tests for the large-sized problems are reported in Table 8. As it is not practical to obtain the optimum solutions of these problems using the mathematical modeling, the lower bound values of these problems have been calculated and provided in the $LB^{Nm}(LB^{Ns})$ column. Since the lower bounds are usually smaller than the optimum solutions even for the small and medium-sized problems, it is expected to have higher variation for large-sized instances. Therefore, it does not provide an exact outcome to compare the ACO results with lower bounds. However, this comparison can provide an idea regarding the quality of the solutions obtained. While the gap between the lower bounds and the ACO results are relatively small in variants of P65/1 and P65/2, it gets larger in those of P148/1, P148/2, P205/1 and 205/2.

Table 8. Results of the computational tests for large-sized problems

#	Problem	CT	$LB^{Nm}(LB^{Ns})$	ACO $NM(NS)$	CPU Time (s)	
					Average	Std. Dev.
49	P65/1	252	5(11)	6(14)	52.30	0.96
50		260	5(11)	6(13)	51.10	0.62
51		271	5(10)	6(12)	55.07	0.25
52		286	5(10)	6(12)	50.40	0.86
53		292	5(10)	6(12)	48.07	1.06
54	P65/2	290	9(20)	12(26)	53.60	1.10
55		320	8(19)	11(24)	51.73	1.33
56		371	7(16)	9(21)	53.70	1.77
57		402	7(15)	8(18)	55.57	1.14
58		430	6(14)	8(17)	56.10	0.62
59	P148/1	146	9(21)	11(27)	123.67	1.39
60		150	9(19)	11(25)	122.77	1.38
61		166	8(18)	10(24)	120.10	1.02
62		178	8(18)	9(22)	117.07	0.75
63		192	7(16)	9(21)	118.43	1.03
64	P148/2	370	12(26)	14(33)	124.37	0.74
65		400	11(25)	13(33)	131.37	1.47
66		422	11(24)	13(30)	124.17	1.62
67		490	9(21)	11(26)	116.80	2.01
68		525	9(19)	10(25)	119.47	2.88
69	P205/1	940	7(16)	10(23)	355.73	6.34
70		1020	6(15)	9(22)	348.40	6.62
71		1115	6(14)	9(20)	330.73	4.58
72		1128	6(13)	9(20)	332.37	1.77
73		1250	5(12)	8(18)	325.63	3.92

74	P205/2	585	10(23)	14(31)	372.20	4.17
75		600	10(22)	14(31)	385.20	2.67
76		635	9(21)	13(29)	787.33	9.45
77		720	8(19)	13(26)	357.13	2.22
78		830	7(17)	10(24)	352.60	1.07

These results might be considered fairly reasonable and indicate that ACO performs well when the gaps between the lower bounds and the optimum solutions are considered even for the small and medium-sized problems (e.g., see #1, #9, #11, #19, #20, #23, #24, #27, etc.). For example, the optimum solution of #19 was obtained as 5(8) while the lower bound was calculated as 3(7). Similar situations were also observed for other test problems, e.g. see #24. While the lower bound was calculated as 3(7) for the corresponding problem, the optimum solution was found as 5(9). It is reasonable that the gap between the lower bounds and the optimum solutions gets bigger when the problem size increases. Therefore, it can be concluded that the ACO algorithm has promising capacity for solving even the large-sized problems to optimum or near optimum.

As one of the main limitations of the work, the parameter optimization has been conducted solving a medium size test problem, i.e. #36, and the parameters obtained eventually have been used to solve other test cases, including the case study with 175 tasks in the next section. Although this approach limits the capacity of the ACO algorithm, it is preferred to show the main idea of the methodology while keeping the paper at a reasonable size. It is also worthy to mention that the optimized parameters have been obtained under the stochastic search behavior of the ACO algorithm, which influences the performance of the algorithm. This is because the algorithm may behave different in every iteration due to the randomness in task selection process and the use of randomly selected heuristic rule, as explained in Section 3.1. While these features help algorithm avoid falling into local minima, they have a certain effect on the parameter optimization process in that sense.

4.3. Case study

This section provides the results of a case study, which has been conducted at a safety cabin (called a model hereafter) manufacturer company serving in automotive industry. The case study concerns the final assembly, where accessories and functional units produced locally or obtained from the suppliers are assembled successively. The final assembly is made along a straight mixed-model line across which left, right and underground stations are located, followed by a set of final testing operations which have not been considered here. The two mostly produced models (called M1 and M2) have been considered within the scope of this study. Please note that all data (including the task processing times and cycle time) have been given in coded values to keep privacy. Although the line was operating during the study, it will be considered as if a new line to be utilized to fit the scope of this research. Hence, in parallel to the type-I problem introduced in this study, the current line balance will not be provided here. Thus, the aim is to minimize the number of mated-stations and the number of stations (as explained in Section 3.1, where $w_1 = 100$ and $w_2 = 1$) given the cycle time. Table A 2 (given in Appendix) presents the processing times, operation sides and precedence relationships of 175 tasks for the final assembly of M1 and M2, except final testing operations. The models have equal demand rates and the cycle time is 60 time-units/item. The letters 'L', 'R', 'E' and 'U' given in the table relates to left, right, either and under, respectively.

Table 9. The summary of the balancing results for the case study

Cycle Time	46	48	50	52	54	56	58	60	62	64
$LB^{Nm}(LB^{Ns})$	20(42)	19(41)	18(39)	18(38)	17(37)	16(35)	16(34)	15(33)	15(32)	14(31)
ACO NM(NS)	21(46)	20(44)	20(43)	19(41)	18(39)	17(38)	17(37)	16(35)	16(35)	15(33)
WLE (%)	88.54	88.71	87.14	87.88	88.96	88.04	87.30	89.21	86.34	88.71

The problem has been solved under various cycle time constraints, ranging from 46 time-units/item to 64 time-units/item. ACO was run three times for each case using the optimal parameters found in the parameter optimization process, and the results have been presented in Table 9. As seen, the results are very close to lower bounds, reported in the second row. The weighted line efficiency ($WLE(\%) = 100 \times (\sum_{m\in M} d_m \sum_{i\in I} t_{im})/(C \times NS)$) of each solution is reported in the table, which indicates that the highest efficiency (89.21%) was obtained when $CT = 60$. For a better presentation, the WLE values obtained under different CT constraints have been plotted in Figure 5.

The detailed balancing configuration of the solution which has the highest weighted line efficiency (when $CT = 60$) is also given in Table 10.

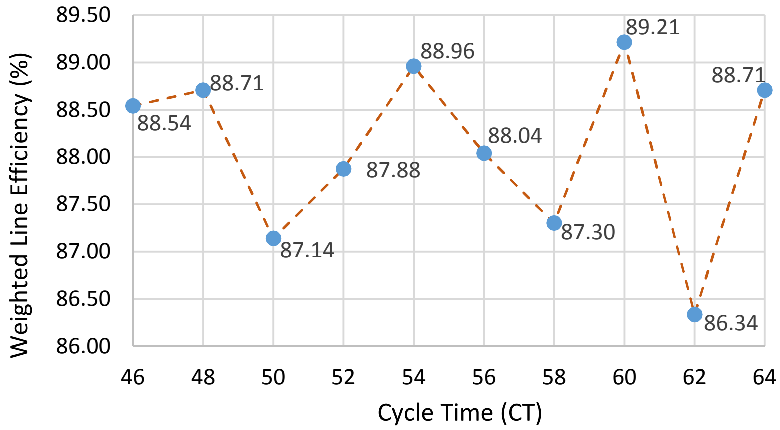


Figure 5: WLE values of solutions obtained under different cycle time constraints

Table 10. The detailed assignment configuration of the solution obtained when cycle time is 60 time-units/item

Mated-station	Side	Assigned Tasks	Station Workloads	
			Model A	Model B
1	Left	[76, 88, 89, 7, 172]	47	59
	Right	[1, 110, 142, 53, 157]	59	58
	Underground	[]	-	-
2	Left	[6, 90, 62, 91, 147, 5]	43	59
	Right	[152, 4, 2, 8, 63, 154]	49	55
	Underground	[]	-	-
3	Left	[65, 92, 160, 12, 13, 148]	60	60
	Right	[141, 9, 10, 64, 66, 3, 34]	55	55
	Underground	[]	-	-
4	Left	[39, 35, 11, 14, 51]	57	57
	Right	[168, 144, 46, 145, 143]	60	49
	Underground	[]	-	-
5	Left	[150, 151, 52, 38]	59	59
	Right	[36, 37, 146, 47, 54]	58	58
	Underground	[]	-	-
6	Left	[75, 78, 15, 80]	53	57
	Right	[48, 61, 55, 70, 79]	44	56
	Underground	[]	-	-
7	Left	[71, 86, 99, 81, 56]	60	60
	Right	[68, 26, 96, 40, 162]	50	50
	Underground	[16, 17, 18, 19, 20]	60	45
8	Left	[149, 41, 170, 58, 97, 84]	58	56
	Right	[87, 57, 69, 164, 161, 43]	57	52
	Underground	[21, 31, 29, 22, 30]	53	60
9	Left	[23, 24, 45, 27]	57	57
	Right	[77, 82, 59, 100, 125, 42, 126, 127]	60	59
	Underground	[]	-	-
10	Left	[174, 175, 44, 60]	58	58
	Right	[49, 98, 163, 28]	37	60
	Underground	[32, 33]	38	26

11	Left	[129, 101, 135, 121, 133, 67]	49	59
	Right	[83, 120, 139, 128]	58	58
	Underground	[]	-	-
12	Left	[123, 136, 124, 165, 169]	55	60
	Right	[72, 73, 50, 74]	52	52
	Underground	[]	-	-
13	Left	[166, 25, 102, 167, 106]	59	59
	Right	[85, 93, 94]	57	57
	Underground	[]	-	-
14	Left	[95, 111, 113, 105, 114]	41	59
	Right	[109, 103, 104, 155, 134, 132, 158]	57	57
	Underground	[]	-	-
15	Left	[138, 171, 137, 117, 118, 140]	56	56
	Right	[153, 159, 122, 156]	30	41
	Underground	[]	-	-
16	Left	[131, 173, 130, 107, 108]	55	55
	Right	[119, 112, 115, 116]	30	38
	Underground	[]	-	-

As seen from Table 10, a total of 16 mated stations (composed of a total of 35 left, right and underground stations) have been utilized for the execution of 175 tasks. Only 3 underground stations were needed for 12 underground operations (see mated-stations 7, 8 and 10). Therefore, the underground side stations have not been utilized in mated-stations 1-6, 9, and 11-16, for which the station workload values have been given as '-' in the table. The heaviest workload belongs to the left sides of mated-stations 3 and 7, which have been fully utilized with 60 time-units for both models. On the other hand, the smallest workload belongs to the underground station of mated-station 10 with 38 time-units and 26 time-units of workload for models M1 and M2, respectively, which corresponds to 32 time-units of weighted workload. The weighted efficiency of the line can simply be calculated as $WLE(\%) = 100 \times (\sum_{m \in M} d_m \sum_{i \in I} t_{im})/(C \times NS) = 1873.5/(46 \times 60) = 89.21\%$. Based on the common practice of considering a solution successful if its efficiency is about 90%, the solution obtained here is fairly good especially taking into account the size of the problem studied.

5. Conclusions and future research

In today's modern manufacturing environment, complex assembly line structures (lifts, robots, etc.) are used to move products across the line and lift/tilt them for performing underneath operations. However, this is not convenient and/or safe for large or very large-sized products (e.g. trailers and buses) and not affordable for most manufacturers. Therefore, even though it can be possible to find some alternative ways for performing underneath operations without underground stations, in some situations underground stations are essential. This study introduced the problem of balancing mixed-model two-sided assembly lines with underground workstations, which allow performing tasks underneath the product concurrently with the left and right side tasks. The problem was modeled mathematically and coded in GAMS. A numerical example and its optimum solution (obtained from CPLEX solver) have been provided with the aim of defining the characteristics of the problem. Test problems have been derived from the literature, adapted to the studied concept and attempted to be solved using the CPLEX solver available by GAMS. The optimum solutions of the first 43 test problems (small and medium in problem size) have been obtained by CPLEX. The NP-hard complexity of the problem makes it unpractical to find the optimum solution of the large-sized problems using an exact solution method. Therefore, for solving the large-sized problems, an ACO algorithm was developed to solve problems where CPLEX fell short. RSM was used to optimize the ACO parameters and all test problems were solved using ACO. For small and medium-sized problems (except P36), the performance of the ACO was compared to optimum solutions. For large-sized problems and P36, the comparison was made against lower bounds, calculated using a new formulation considering the underground stations and exact processing times of tasks.

The results of the computational tests indicated that ACO performs well. A real world problem which contains 175 tasks and incorporates underground operations have been solved using ACO under various cycle time constraints and the solution which has almost %90 weighted line efficiency was presented. In terms of the industrial implications and managerial aspects of the research, the concept studied in this paper can easily be adopted by line managers of high volume products. Thus, resources (workforce and facilities) can be saved with minimizing the idle times by incorporating underground operations concurrently with the left and right-side operations when the line to be constructed for the first time. By this way, line managers can also put in practice shortening their lines.

The problem may be extended by solving it considering different objectives (e.g. minimizing the cycle time or total utility works, or maximizing the work-relatedness, etc.) and the performance of the proposed approach can be evaluated. One can consider rebalancing of MTALB-US, referred to as type-II assembly line balancing problem. So that, decision makers may consider rebalancing their existing lines making use of the methodology proposed. It is also worthy to examine the line concept introduced in this paper in terms of ergonomic aspects (Otto and Scholl, 2011). The influence of working in an underground station for a prolonged time on the physiological and psychological health of the operators might be investigated. Job rotation can be a good precaution for such effects. Operators can be allocated to different workstations (and so tasks assigned to these workstations) from time to time which also helps advance their skills.

Appendix

Table A 1. Fitted values and prediction errors of the responses

Run	Y_{NS}	$\hat{Y}_{NS}$	PE%	Y_{SI}	$\hat{Y}_{SI}$	PE%
1	11.75	11.77	0.17	29.95	29.92	0.08
2	11.65	11.64	0.12	29.38	29.42	0.14
3	11.91	11.86	0.43	30.31	30.24	0.22
4	11.70	11.73	0.23	29.92	29.86	0.19
5	11.75	11.73	0.16	29.94	29.88	0.21
6	11.73	11.72	0.10	29.83	29.73	0.33
7	11.82	11.85	0.23	30.19	30.21	0.08
8	11.86	11.84	0.19	30.20	30.19	0.05
9	11.23	11.24	0.10	28.55	28.61	0.22
10	11.13	11.11	0.19	28.28	28.18	0.35
11	11.29	11.31	0.16	28.80	28.82	0.08
12	11.17	11.18	0.07	28.40	28.51	0.39
13	11.22	11.20	0.18	28.53	28.51	0.06
14	11.15	11.19	0.36	28.32	28.43	0.40
15	11.29	11.29	0.03	28.73	28.74	0.02
16	11.30	11.29	0.13	28.83	28.78	0.18
17	11.27	11.28	0.10	28.64	28.69	0.19
18	11.20	11.21	0.09	28.41	28.46	0.19
19	11.19	11.20	0.13	28.42	28.50	0.29
20	11.29	11.30	0.06	28.81	28.83	0.08
21	11.22	11.22	0.03	28.53	28.54	0.03
22	11.24	11.26	0.15	28.55	28.65	0.34
23	11.70	11.75	0.38	29.55	29.81	0.86
24	11.23	11.21	0.22	28.60	28.45	0.53
25	11.24	11.24	0.01	28.60	28.57	0.10
26	11.24	11.24	0.01	28.63	28.57	0.20
27	11.25	11.24	0.10	28.60	28.57	0.10
28	11.23	11.24	0.07	28.50	28.57	0.25
29	11.25	11.24	0.10	28.64	28.57	0.24
30	11.27	11.24	0.28	28.69	28.57	0.41
31	11.25	11.24	0.10	28.66	28.57	0.31

Please note that $\hat{Y}_{NS}$ and $\hat{Y}_{SI}$ columns present the fitted values of Y_{NS} and Y_{SI}, respectively. PE% is the prediction error in percentage.

Table A 2. Data used for the case study

Task No	Task Time		Side	Immediate Predecessor(s)	Task No	Task Time		Side	Immediate Predecessor(s)
	M1	M2				M1	M2		
1	29	22	E	-	55	8	0	E	54
2	16	16	R	7	56	9	9	E	55
3	10	10	R	7	57	5	0	E	56
4	4	4	E	-	58	11	11	E	3, 17, 54
5	18	18	E	4	59	4	4	E	57, 58
6	15	15	E	-	60	6	6	L	30
7	9	9	E	-	61	18	28	E	47
8	11	11	R	7	62	5	5	E	-
9	7	7	E	6	63	4	4	R	62
10	4	4	E	-	64	3	3	E	62
11	13	13	E	10,9,2	65	3	3	E	62
12	6	6	E	5	66	4	4	E	62
13	13	13	E	12	67	3	3	E	28, 57, 64, 66
14	8	8	E	13	68	5	5	R	8, 15, 39
15	25	29	E	14	69	7	7	R	68
16	24	0	U	15	70	2	2	E	3, 8, 11, 15
17	14	9	U	16	71	21	21	E	70
18	6	20	U	17	72	11	11	E	67
19	8	8	U	18	73	18	18	E	72
20	8	8	U	18	74	12	12	E	71, 73
21	16	16	U	19, 20	75	4	4	E	38, 39
22	5	5	U	21	76	8	8	E	-
23	6	6	E	22	77	6	6	E	21
24	5	5	E	23	78	11	11	L	75
25	11	11	L	23	79	6	6	E	76, 78
26	7	7	E	15	80	13	13	E	38, 39
27	31	31	E	11, 21, 26	81	6	6	L	80
28	7	6	E	33	82	8	8	E	77, 80
29	14	14	U	21	83	17	17	E	28, 37, 43, 75,
30	9	9	U	29	84	8	8	E	80
31	9	16	U	21	85	14	14	E	65, 74
32	7	6	U	27	86	18	18	E	79
33	31	20	U	32	87	18	18	E	79
34	22	22	E	5	88	10	22	E	76
35	24	24	E	34	89	10	10	E	76
36	14	14	E	35	90	0	8	E	76
37	15	15	E	36	91	0	8	E	76
38	17	17	E	36	92	13	13	E	1
39	4	4	L	34	93	19	19	E	85, 86, 87, 92
40	28	28	E	39	94	24	24	E	93
41	7	7	E	40	95	10	10	E	93
42	9	9	E	29, 40	96	4	4	R	71
43	17	17	E	41	97	20	0	E	41
44	16	16	E	42	98	16	20	E	42
45	15	15	E	11	99	6	6	E	80
46	16	5	R	39	100	9	8	E	43, 78, 99
47	17	17	R	46	101	0	10	E	83
48	10	20	R	47	102	24	24	E	85
49	10	30	R	48	103	6	6	E	93
50	11	11	R	49	104	7	7	E	103
51	8	8	L	11	105	4	4	E	103

52	10	10	E	51	106	4	4	E	102
53	11	17	E	-	107	7	7	L	102
54	10	10	E	52, 53	108	13	13	L	107

Table A 2. Data used for the case study (continued)

Task No	Task Time		Side	Immediate Predecessor(s)	Task No	Task Time		Side	Immediate Predecessor(s)
	M1	M2				M1	M2		
109	7	7	E	93	143	6	6	R	145
110	6	6	E	-	144	26	26	R	141, 142
111	12	30	E	93	145	6	6	R	144
112	11	11	R	111	146	2	2	R	145
113	11	11	L	111	147	5	5	L	-
114	4	4	E	113	148	10	10	L	-
115	10	10	E	110, 112, 113	149	12	14	L	15, 148
116	0	8	E	115	150	26	26	L	147, 148
117	9	9	L	103	151	6	6	L	150
118	17	17	E	117	152	6	12	E	-
119	9	9	E	83, 117	153	7	12	E	45, 103
120	15	15	E	79, 83	154	8	8	E	-
121	11	11	E	120	155	7	7	E	45, 103
122	7	7	E	117	156	9	12	E	152, 153, 155
123	14	14	E	121	157	3	3	E	-
124	5	10	E	123	158	12	12	E	155, 157
125	11	11	E	45	159	7	10	E	154, 158
126	5	5	E	125	160	15	15	E	147
127	8	8	E	125	161	3	3	R	69
128	12	12	R	129	162	6	6	R	71
129	16	16	E	127	163	4	4	R	69
130	8	8	E	132	164	7	7	R	69, 71
131	9	9	E	132	165	6	6	L	124, 139
132	12	12	E	134	166	9	9	L	162, 163, 164, 165
133	7	7	E	129	167	11	11	L	166
134	6	6	E	103	168	6	6	E	4
135	12	12	L	129	169	18	18	E	83
136	12	12	L	135	170	0	16	L	162
137	9	9	E	132	171	12	12	L	25, 151
138	6	6	E	132	172	10	10	E	1
139	14	14	E	120	173	18	18	E	128, 136, 156, 160
140	3	3	L	132	174	30	30	E	43
141	5	5	R	-	175	6	6	E	174
142	10	10	R	-	-	-	-	-	-

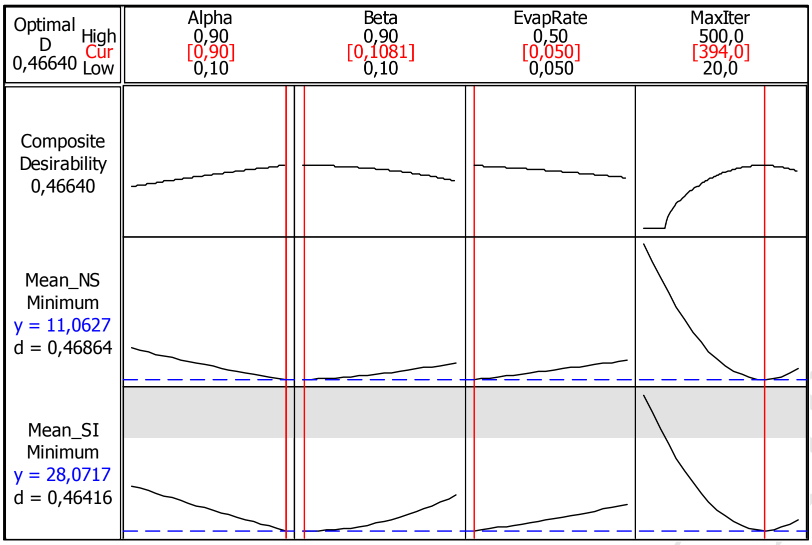


Figure A 1. The optimization plot

References

Abdolreza, R. and Farnaz Ghazi, N. (2017) 'Mixed-model multi-manned assembly line balancing problem: a mathematical model and a simulated annealing approach', *Assembly Automation,* 37(1), pp. 34-50.

Abdullah Make, M. R., Ab. Rashid, M. F. F. and Razali, M. M. (2016) 'A review of two-sided assembly line balancing problem', *The International Journal of Advanced Manufacturing Technology*.

Akpinar, S. and Bayhan, G. M. (2011) 'A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints', *Engineering Applications of Artificial Intelligence,* 24(3), pp. 449-457.

Akpinar, S., Bayhan, G. M. and Baykasoglu, A. (2013) 'Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks', *Applied Soft Computing,* 13(1), pp. 574-589.

Arcus, A. (1966) 'COMSOAL, A Computer Method of Sequencing Operations for Assembly Lines', *The International Journal of Production Research,* 4(4), pp. 259-277.

Arcus, A. L. (1963) *An analysis of a computer method of sequencing assembly line operations.* Ph.D. Dissertation, University of California, Berkeley, USA.

Bartholdi, J. J. (1993) 'Balancing 2-Sided Assembly Lines - a Case-Study', *International Journal of Production Research,* 31(10), pp. 2447-2461.

Battaïa, O. and Dolgui, A. (2013) 'A taxonomy of line balancing problems and their solution approaches', *International Journal of Production Economics,* 142(2), pp. 259-277.

Battini, D., Faccio, M., Ferrari, E., Persona, A. and Sgarbossa, F. (2007) 'Design configuration for a mixed-model assembly system in case of low product demand', *International Journal of Advanced Manufacturing Technology,* 34(1-2), pp. 188-200.

Baybars, I. (1986) 'A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem', *Management Science,* 32(8), pp. 909-932.

Baykasoglu, A. (2006) 'Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems', *Journal of Intelligent Manufacturing,* 17(2), pp. 217-232.

Baykasoglu, A. and Dereli, T. (2008) 'Two-sided assembly line balancing using an ant-colony-based heuristic', *International Journal of Advanced Manufacturing Technology,* 36(5-6), pp. 582-588.

Becker, C. and Scholl, A. (2009) 'Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure', *European Journal of Operational Research,* 199(2), pp. 359-374.

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

Boysen, N., Fliedner, M. and Scholl, A. (2008) 'Assembly line balancing: Which model to use when?', *International Journal of Production Economics,* 111(2), pp. 509-528.

Buyukozkan, K., Kucukkoc, I., Satoglu, S. I. and Zhang, D. Z. (2016) 'Lexicographic bottleneck mixed-model assembly line balancing problem: Artificial bee colony and tabu search approaches with optimised parameters', *Expert Systems with Applications,* 50, pp. 151-166.

Chutima, P. and Chimklai, P. (2012) 'Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge', *Computers & Industrial Engineering,* 62(1), pp. 39-55.

Delice, Y., Aydoğan, E. K., Özcan, U. and İlkay, M. S. (2014) 'A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing', *Journal of Intelligent Manufacturing*, pp. 1-14.

Dimitriadis, S. G. (2006) 'Assembly line balancing and group working: A heuristic procedure for workers’ groups operating on the same product and workstation', *Computers & Operations Research,* 33(9), pp. 2757-2774.

Dorigo, M. and Blum, C. (2005) 'Ant colony optimization theory: A survey', *Theoretical Computer Science,* 344(2-3), pp. 243-278.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) 'Ant system: Optimization by a colony of cooperating agents', *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* 26(1), pp. 29-41.

Dorigo, M. and Stutzle, T. (2010) 'Ant Colony Optimization: Overview and Recent Advances', *Handbook of Metaheuristics, Second Edition,* 146, pp. 227-263.

Emde, S., Boysen, N. and Scholl, A. (2010) 'Balancing mixed-model assembly lines: a computational evaluation of objectives to smoothen workload', *International Journal of Production Research,* 48(11), pp. 3173-3191.

Fattahi, P., Roshani, A. and Roshani, A. (2011) 'A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem', *The International Journal of Advanced Manufacturing Technology,* 53(1), pp. 363-378.

Ford, H. (2009) *My Life and Work - An Autobiography of Henry Ford.* New York: Classic House Books.

Gurevsky, E., Hazır, Ö., Battaïa, O. and Dolgui, A. (2012) 'Robust balancing of straight assembly lines with interval task times', *Journal of the Operational Research Society,* 64(11), pp. 1607-1613.

Helgeson, W. B. and Birnie, D. P. (1961) 'Assembly Line Balancing Using the Ranked Positional Weight Technique', *The Journal of Industrial Engineering,* 12(6), pp. 394-398.

Hu, X. F., Wu, E. F., Bao, J. S. and Jin, Y. (2010) 'A branch-and-bound algorithm to minimize the line length of a two-sided assembly line', *European Journal of Operational Research,* 206(3), pp. 703-707.

Hu, X. F., Wu, E. F. and Jin, Y. (2008) 'A station-oriented enumerative algorithm for two-sided assembly line balancing', *European Journal of Operational Research,* 186(1), pp. 435-440.

Kellegöz, T. (2017) 'Assembly line balancing problems with multi-manned stations: a new mathematical formulation and Gantt based heuristic method', *Annals of Operations Research,* 253(1), pp. 377-404.

Khuri, A. I. and Mukhopadhyay, S. (2010) 'Response surface methodology', *Wiley Interdisciplinary Reviews: Computational Statistics,* 2(2), pp. 128-149.

Kim, Y. K., Kim, Y. H. and Kim, Y. J. (2000) 'Two-sided assembly line balancing: a genetic algorithm approach', *Production Planning & Control,* 11(1), pp. 44-53.

Kim, Y. K., Song, W. S. and Kim, J. H. (2009) 'A mathematical model and a genetic algorithm for two-sided assembly line balancing', *Computers & Operations Research,* 36(3), pp. 853-865.

Kucukkoc, I., Buyukozkan, K., Satoglu, S. I. and Zhang, D. Z. (2015) 'A mathematical model and artificial bee colony algorithm for the lexicographic bottleneck mixed-model assembly line balancing problem', *Journal of Intelligent Manufacturing*.

Kucukkoc, I., Karaoglan, A. D. and Yaman, R. (2013) 'Using response surface design to determine the optimal parameters of genetic algorithm and a case study', *International Journal of Production Research,* 51(17), pp. 5039-5054, doi: http://dx.doi.org/10.1080/00207543.2013.784411.

Kucukkoc, I. and Zhang, D. Z. 'On applications of ant colony optimisation techniques in solving assembly line balancing problems'. *55th Annual Conference of the Operational Research Society 2013, OR55*, Exeter, 45-51.

Kucukkoc, I. and Zhang, D. Z. (2014a) '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,* 158, pp. 314-333, doi: http://dx.doi.org/10.1016/j.ijpe.2014.08.010.

Kucukkoc, I. and Zhang, D. Z. (2014b) '*Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines*', *International Journal of Production Research,* 52(12), pp. 3665-3687.

Kucukkoc, I. and Zhang, D. Z. (2015a) 'Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines', *The International Journal of Advanced Manufacturing Technology*, pp. doi: http://dx.doi.org/10.1007/s00170-015-7320-y.

Kucukkoc, I. and Zhang, D. Z. (2015b) 'Type-E parallel two-sided assembly line balancing problem: Mathematical model and ant colony optimisation based approach with optimised parameters', *Computers and Industrial Engineering,* 84, pp. 56–69, doi: http://dx.doi.org/10.1016/j.cie.2014.12.037.

Kucukkoc, I. and Zhang, D. Z. (2016) 'Mixed-model parallel two-sided assembly line balancing problem: A flexible agent-based ant colony optimization approach', *Computers & Industrial Engineering,* 97, pp. 58-72.

Lee, T. O., Kim, Y. and Kim, Y. K. (2001) 'Two-sided assembly line balancing to maximize work relatedness and slackness', *Computers & Industrial Engineering,* 40(3), pp. 273-292.

Li, Z., Kucukkoc, I. and Nilakantan, J. M. (2017a) 'Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem', *Computers & Operations Research,* 84, pp. 146-161.

Li, Z., Tang, Q. and Zhang, L. (2016) 'Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm', *Journal of Cleaner Production,* 135, pp. 508-522.

Li, Z., Tang, Q. and Zhang, L. (2017b) 'Two-sided assembly line balancing problem of type I: Improvements, a simple algorithm and a comprehensive study', *Computers & Operations Research,* 79, pp. 78-93.

Manavizadeh, N., Rabbani, M., Moshtaghi, D. and Jolai, F. (2012) 'Mixed-model assembly line balancing in the make-to-order and stochastic environment using multi-objective evolutionary algorithms', *Expert Systems with Applications,* 39(15), pp. 12026-12031.

McMullen, P. R. and Frazier, G. V. (1997) 'A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations', *International Journal of Production Economics,* 51(3), pp. 177-190.

Otto, A. and Scholl, A. (2011) 'Incorporating ergonomic risks into assembly line balancing', *European Journal of Operational Research,* 212(2), pp. 277-286.

Ozbakir, L. and Tapkan, P. (2010) 'Balancing fuzzy multi-objective two-sided assembly lines via Bees Algorithm', *Journal of Intelligent & Fuzzy Systems,* 21(5), pp. 317-329.

Ozbakir, L. and Tapkan, P. (2011) 'Bee colony intelligence in zone constrained two-sided assembly line balancing problem', *Expert Systems with Applications,* 38(9), pp. 11947-11957.

Ozcan, U. (2010) 'Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm', *European Journal of Operational Research,* 205(1), pp. 81-97.

Ozcan, U., Kellegoz, T. and Toklu, B. (2011) 'A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem', *International Journal of Production Research,* 49(6), pp. 1605-1626.

Ozcan, U. and Toklu, B. (2009a) 'A tabu search algorithm for two-sided assembly line balancing', *International Journal of Advanced Manufacturing Technology,* 43(7-8), pp. 822-829.

Ozcan, U. and Toklu, B. (2009b) 'Balancing of mixed-model two-sided assembly lines', *Computers & Industrial Engineering,* 57(1), pp. 217-227.

Ozcan, U. and Toklu, B. (2010) 'Balancing two-sided assembly lines with sequence-dependent setup times', *International Journal of Production Research,* 48(18), pp. 5363-5383.

Purnomo, H. D., Wee, H. M. and Rau, H. (2013) 'Two-sided assembly lines balancing with assignment restrictions', *Mathematical and Computer Modelling,* 57(1-2), pp. 189-199.

Roshani, A. and Giglio, D. (2017) 'Simulated annealing algorithms for the multi-manned assembly line balancing problem: minimising cycle time', *International Journal of Production Research,* 55(10), pp. 2731-2751.

Simaria, A. S. and Vilarinho, P. M. (2009) '2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines', *Computers & Industrial Engineering,* 56(2), pp. 489-506.

Talbot, F. B. and Patterson, J. H. (1984) 'An Integer Programming Algorithm with Network Cuts for Solving the Assembly Line Balancing Problem', *Management Science,* 30(1), pp. 85-99.

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,* 97, pp. 146-156.

Thomopoulos, N. T. (1967) 'Line Balancing-Sequencing for Mixed-Model Assembly', *Management Science,* 14(2), pp. B-59-B-75.

Tonge, F. M. (1960) 'Summary of a Heuristic Line Balancing Procedure', *Management Science,* 7(1), pp. 21-42.

Vilarinho, P. M. and Simaria, A. S. (2002) 'A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations', *International Journal of Production Research,* 40(6), pp. 1405-1420.

Wu, E. F., Jin, Y., Bao, J. S. and Hu, X. F. (2008) 'A branch-and-bound algorithm for two-sided assembly line balancing', *International Journal of Advanced Manufacturing Technology,* 39(9-10), pp. 1009-1015.

Yuan, B., Zhang, C., Shao, X. and Jiang, Z. (2015) 'An effective hybrid honey bee mating optimization algorithm for balancing mixed-model two-sided assembly lines', *Computers & Operations Research,* 53, pp. 32-41.

Öner-Közen, M., Minner, S. and Steinthaler, F. (2017) 'Efficiency of paced and unpaced assembly lines under consideration of worker variability – A simulation study', *Computers & Industrial Engineering,* 111, pp. 516-526.

Özcan, U. and Toklu, B. (2009) 'Balancing of mixed-model two-sided assembly lines', *Computers & Industrial Engineering,* 57(1), pp. 217-227.