Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines

Ibrahim Kucukkoc [ab] **David Z. Zhang** [a]

[a] College of Engineering Mathematics and Physical Sciences, University of Exeter, Exeter, UK

[b] Department of Industrial Engineering, Balikesir University, Balikesir, Turkey

Email: *I.Kucukkoc@exeter.ac.uk* *Tel: +44(0)1392 723613*

Email: *D.Z.Zhang@exeter.ac.uk* Tel: *+44(0)1392 723641*

Abstract: Growing interests from customers in customised products and increasing competitions among peers necessitate companies to configure their manufacturing systems more effectively than ever before. We propose a new assembly line system configuration for companies that need intelligent solutions to satisfy customised demands on time with existing resources. A mixed-model parallel two-sided assembly line system is introduced based on the parallel two-sided assembly line system previously proposed by Özcan et al. (Balancing parallel two-sided assembly lines, International Journal of Production Research, 48 (16), 4767-4784, 2010). The mixed-model parallel two-sided assembly line balancing problem is illustrated with examples from the perspective of simultaneous balancing and sequencing. An agent based ant colony optimisation algorithm is proposed to solve the problem. This algorithm is the first attempt in the literature to solve an assembly line balancing problem with an agent based ant colony optimisation approach. The algorithm is illustrated with an example and its operational procedures and principles explained and discussed.

Keywords: mixed-model parallel two-sided assembly lines; assembly line balancing; agent based ant colony optimisation; meta-heuristics; artificial intelligence

Corresponding author: Ibrahim Kucukkoc, I.Kucukkoc@exeter.ac.uk

Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines

Growing interests from customers in customised products and increasing competitions among peers necessitate companies to configure their manufacturing systems more effectively than ever before. We propose a new assembly line system configuration for companies that need intelligent solutions to satisfy customised demands on time with existing resources. A mixed-model parallel two-sided assembly line system is introduced based on the parallel two-sided assembly line system previously proposed by Ozcan et al. (Balancing parallel two-sided assembly lines, International Journal of Production Research, 48 (16), 4767-4784, 2010). The mixed-model parallel two-sided assembly line balancing problem is illustrated with examples from the perspective of simultaneous balancing and sequencing. An agent based ant colony optimisation algorithm is proposed to solve the problem. This algorithm is the first attempt in the literature to solve an assembly line balancing problem with an agent based ant colony optimisation approach. The algorithm is illustrated with an example and its operational procedures and principles explained and discussed.

Keywords: mixed-model parallel two-sided assembly lines; assembly line balancing; agent based ant colony optimisation; meta-heuristics; artificial intelligence

1. Introduction

Assembly lines are the most crucial constituents of mass production systems and provide improved labour productivity especially for companies which have to produce high volume products in a cost effective manner, within a reasonable time (Kara *et al.* 2010). The throughput level of a line is one of the key factors which determines the response time of an entire manufacturing system. Assembly line balancing problem is to assign tasks to an ordered sequence of workstations optimally by satisfying specific constraints. It is one of the most important problems in designing and managing assembly lines (Ozbakir and Tapkan 2011, Kucukkoc and Yaman 2013).

The initial serious attempt to increase productivity by using carefully designed manufacturing operations, that comprised machine-assisted specialised labour, emerged in the 18^{th} century in England. With the industrial revolution (1750-1900), manufacturing industry experienced crucial structural changes and companies started to adopt mass production techniques to increase capacity and productivity. Henry Ford and his colleagues constructed first moving-belt conveyor to assemble flywheel magnetos, in 1913 (Tanenbaum and Holstein 2012). Although the early aim of Ford was to produce only a "horseless carrier", as a common idea (Ford 2009), sales of model T passed 250 thousand in 1914, through the efficiency of the assembly line.

Since then, assembly line balancing problems have been considered by a large number of researchers from both academia and industry. Many types of assembly lines have been studied and various types of solution approaches suggested to solve these complex problems (Kucukkoc *et al.* 2013a).

Workers who perform partial *tasks* through an assembly line can assemble complex products (Gunasekaran and Cecille 1998, Becker and Scholl 2006). Line balancing is to divide the total workload of assembly line into several *workstations* and to determine which task will be performed at each workstation (while each task is allocated only once). Generally, these workstations are linked together by a transport system whose primary mission is to move products among serially constructed workstations (Bautista and Pereira 2011). A set of tasks is performed at each workstation and each task has its own *processing time*. Due to technological and organisational conditions, *precedence constraints* that are usually represented as a network must be satisfied in the assignment process (Sarker and Shanthikumar 1983, Simaria 2006). *Workload* (or *station time*) of a workstation, the time required to perform the set of tasks assigned to the workstation, cannot exceed *cycle time* determined by the designer or manager of the line. Hence, production rate of the system is determined by cycle time (Darel and Cother 1975, Simaria 2006). The main objective of assembly line balancing is to minimise the sum of the differences between the cycle time and individual workloads, so minimise total idle time of the line, by minimising the number of required stations, and/or the cycle time.

Assembly lines can be classified into two groups based on the operation side utilisation of the lines: (i) one sided assembly lines and (ii) two-sided assembly lines. Two-sided assembly lines are chiefly used in the production of large-sized products and workers at each pair of opposite stations work in parallel on different tasks but on the same individual item (Bartholdi 1993). The main difference between this kind of systems and one sided system is that some tasks are required to be performed on a specific side (Left-L or Right-R) of the line and some on both sides (Either-E) simultaneously (Kucukkoc *et al.* 2013b). Two-sided assembly lines are more practical for large-sized products (i.e. trucks) than for small ones (i.e. electrical drills) because of the *interference* phenomenon (Kim *et al.* 2000, Lee *et al.* 2001). Interference will be explained in detail in the problem definition section.

Two-sided assembly line balancing problem has been studied by various researchers. Heuristic approaches were proposed by Bartholdi (1993), Lee *et al.* (2001), Hu *et al.* (2008), Ozcan and Toklu (2010), Yegul *et al.* (2010), Yin *et al.* (2011) and exact solution approaches by Wu *et al.* (2008) and Hu *et al.* (2010). Meta-heuristics have also been presented to address two-sided assembly line balancing problem, i.e. Baykasoglu and Dereli (2008), Simaria and Vilarinho (2009), Ozcan and Toklu (2009b), Ozbakir and Tapkan (2010), Ozcan (2010), Ozbakir and Tapkan (2011), Chutima and Chimklai (2012). Among these meta-heuristics, the studies of Kim *et al.* (2000, 2009), Taha *et al.* (2011), Purnomo *et al.* (2013), and Rabbani *et al.* (2012) employ different variations of genetic algorithm to balance two-sided lines. As can be comprehended from these studies, there exist numerous genetic algorithm approaches in the literature for two-sided assembly line balancing problems.

In addition to the two types of assembly lines based on line configuration, there is another type called parallel assembly lines. Although the literature on assembly line balancing problems is rather extensive, the studies on parallel assembly line balancing problem (PALBP) are quite limited. Table 1 summarises the main contributions regarding parallel assembly line balancing problems and lists solution methods developed till now. The parallel line configuration idea was first proposed by Suer and Dagli (1994). They proposed a heuristic procedure which aims at determining the number of lines and workstations by considering assigning different models of a product to the lines. However, in the study, the precedence constraints were not considered and it was assumed that the entire job can be divided into any number of operations. Since then, Suer (1998) has proposed alternative line configuration strategies for a single product.

Table 1. Summary of the literature on parallel assembly line balancing problems

However, the real PALBP, balancing of more than one assembly lines with a common set of resources, was presented by Gokcen *et al.* (2006) (Ozcan *et al.* 2009). Since then, Benzer *et al.* (2007) proposed a new shortest path approach based model for PALBP and illustrated the performance of the model on a numerical example. Lusa (2008) presented a detailed survey on multiple or parallel assembly line balancing problems and described the main literature contributions briefly. Baykasoglu *et al.* (2009) proposed a novel ant colony optimisation based algorithm for PALBP. They compared their test results with three other existing approaches from the literature to prove the efficiency of the proposed algorithm. Cercioglu *et al.* (2009) proposed a simulated annealing approach to solve PALBP and compared obtained results with the results of existing heuristic algorithm proposed by Gokcen *et al.* (2006). Ozcan *et al.* (2009) developed the first multi-objective tabu search algorithm for PALBP and tested the performance of the algorithm on a set of well-known problems in the literature. Scholl and Boysen (2009) modelled the PALBP mathematically and proposed an exact solution procedure. Kara *et al.* (2010) suggested a fuzzy goal programming model that can be used for balancing parallel assembly lines. Ozcan *et al.* (2010a) addressed parallel mixed model assembly line balancing and sequencing problem with a simulated annealing approach. The approach maximises the line efficiency and considers workload smoothness among workstations. Ozbakir *et al.* (2011) developed a novel multiple-colony ant algorithm for balancing bi-objective parallel assembly lines. The work is one of the first attempts for PALBP with swarm intelligence based meta-heuristics (Kucukkoc *et al.* 2013b).

Another classification scheme of assembly line balancing problems is based on the variety of products assembled on the line: (i) single model assembly lines, (ii) mixed-model assembly lines, and (ii) multi-model assembly lines. To explain briefly (Rekiek *et al.* 2002, Rekiek and Delchambre 2006, Boysen *et al.* 2008, Hamzadayi and Yildiz 2012):

- Single model assembly lines are used to assemble a single homogenous product in large quantities;
- Mixed model assembly lines are utilised to assemble a set of different models of the same product simultaneously in an intermixed sequence;
- Multi-model assembly lines are used to assemble batches of similar models with intermediate setup operations.

Assembly lines were used for high-volume production of a single commodity in its traditional form. Simple assembly line balancing problem, the most employed form of line balancing problems, assumes the single-model production, and was considered by a vast number of publications such as Baybars (1986), Saltzman and Baybars (1987), Hoffmann (1992), Rubinovitz and Levitin (1995), Klein and Scholl (1996), Sprecher (1999), Peeters and Degraeve (2006), Gokcen *et al.* (2005), Zhang *et al.* (2007), Liu *et al.* (2008), Nourmohammadi and Zandieh (2011).

However, with the change of global market, companies converted single model lines into mixed-model lines in order to provide diversity and meet customised customer demands on time in an intelligent way. An advantage of mixed-model lines over multi-model lines is that setup process is not required between model changes. Multi-model lines are used rarely since they require setup times between passes from one model to another. They have been studied by few researchers such as Berger *et al.* (1992) Pastor *et al.* (2002) Eryuruk *et al.* (2008, 2011).

Table 2 gives a summary of the main contributions in the literature on mixed-model assembly line balancing problem from 2007 to 2013. As can be observed from the summary, studies on both parallel lines and two-sided lines are quite new as well as scarce. Few researchers carried out studies in the literature. Ozcan *et al.* (2010b) proposed parallel two-sided assembly line configuration to combine the advantages of parallel lines and two sided lines. They developed a tabu search algorithm to balance two or more two-sided assembly lines located in parallel to each other. Kucukkoc *et al*. (2013b) proposed an Ant Colony Optimisation (ACO) algorithm which considers line length as well as total number of required workstations. Kucukkoc *et al*. (2013c) enhanced previously developed ACO algorithm with ranked positional weight method (RPWM) heuristic.

Table 2. Detailed summary of the main contributions in the literature on mixed-model assembly line balancing problems (2007-2013)

On the other hand, Ozcan and Toklu (2009a) introduced mixed-model two-sided assembly line balancing problem and proposed a simulated annealing algorithm to deal with the problem. Other meta-heuristics, which are ant colony optimisation and particle swarm optimisation algorithms, have been developed by Simaria and Vilarinho (2009), and Chutima and Chimklai (2012), respectively, for mixed-model two-sided assembly lines. Rabbani *et al*. (2012) addressed two-sided U-shaped line balancing problem and proposed a genetic algorithm approach which considers operator travel times as well. Nevertheless, parallel lines are not incorporated in these studies again.

The only study, which addresses model variations on parallel assembly lines, belongs to Ozcan *et al*. (2010a). However, there is no study which addresses parallel two-sided assembly line system with model variations which is introduced in this paper. Although mixed-model parallel two-sided assembly lines are encountered in producing large-sized high volume products in industry, none of the researchers has considered this issue so far. Mixed-model parallel two-sided assembly lines offer many benefits to companies by combining the advantages of both parallel lines and two-sided lines with model variation flexibility. Based on this motivation, mixed-model parallel two-sided assembly line balancing and sequencing problem is introduced, illustrated, and explored with numerical examples, in this research.

The remainder of this paper is organised as follows. In Section 2, we introduce mixed-model parallel two-sided assembly lines. Section 3 describes the problem of simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines with an explanatory example with different production cycles. The proposed agent based ant colony optimisation (called as ABACO hereafter) methodology for solving the problem, and a test example, are given in Sections 4 and 5, respectively. Section 6 discusses some issues about the proposed problem and method. Finally, we present conclusions with limitations and industrial implications, and describe future work in Section 7.

2. Mixed-model parallel two-sided assembly lines

Parallel two-sided assembly line balancing problem, which aims at balancing more than one two-sided assembly lines constructed in parallel simultaneously, was introduced by Ozcan *et al.* (2010b). Parallel two-sided assembly lines are widely used in the production of one or more similar product models that have similar production processes in a set of two-sided assembly lines constructed in parallel to each other. However, only one model is allowed to be assembled on each line at a time as in Ozcan et al (2010b).

Mixed-model assembly lines provide more flexibility and capability of responding to different market demands to satisfy customised customer demands on time and to reach global markets in today's highly competitive business environment. However, companies need to construct their production systems in an intelligent way to deal with undesirable costs caused by customisation of products.

With the solution of producing more than one model on each adjacent line of parallel two-sided lines, a new competitive line system called *mixed-model parallel two-sided assembly lines* can be obtained. Balancing of these lines can be called *mixed-model parallel two-sided assembly line balancing problem*.

The mixed-model parallel two-sided assembly line balancing problem is balancing more than one mixed-model two-sided assembly lines constructed in parallel. The main objective is allocating tasks to the workstations optimally by considering technological priorities, capacity constraints, and some other constraints like zoning or positional constraints. As will be explained in this section, with the integration of simultaneous model sequencing procedure with mixed-model parallel two-sided assembly line balancing problem, the problem becomes more complex and turns into *mixed-model parallel two-sided assembly line balancing and sequencing problem (MPTALB/S)*.

The idea of constructing mixed-model parallel two-sided assembly lines is a completely new topic. It provides the flexibility of producing similar large sized product models on parallel lines. This new type of configuration carries the combined practical advantages of mixed-model assembly lines, parallel assembly lines, and two-sided assembly lines. These advantages include:

- Shorter line length than traditional assembly lines,
- Shared use of common tools,
- Flexibility of producing different models with different throughput rates,
- Less material handling cost and operator movement requirements,
- Increased line efficiency with reduced operator requirement,

- Increased motivation of operators due to operation enrichment at combined workstations between two lines,
- Increased skill levels of operators,
- Improved communication skills among operators.

The precedence relationships among tasks should be considered carefully since tasks, which have precedence relationships with each other and are performed on both sides of each line, must be assigned with the consideration of completion time of previously assigned tasks. Let us consider P_{1A9} as the set of predecessors of task 9 on Line I for model A. If the precedence relationships among tasks are assumed as task $4 \in P_{1A9}$ and task $8 \in P_{1A9}$, task 9 can be initialised after the completion of tasks 4 and 8, which may be performed on the other side of the line. This phenomenon is called *interference* in the literature.

The workstations can be utilised either on only one or on both adjacent two-sided lines. The common stations constructed for both adjacent lines are called *"multi-line stations"* (Battaïa and Dolgui 2013). A similar version of this structure, split workplaces, has been used by Scholl and Boysen (2009) in defining common stations on parallel assembly lines. The utilisation of multi-line stations is one of the basic advantages of parallel assembly lines since multi-line stations help minimise the total number of required operators and thus minimise idle times. Figure 1 shows a typical configuration of two adjacent mixed-model parallel two-sided assembly lines with regular and multi-line workstations. As can be seen from the figure, seven operators are allocated to perform tasks for all models (A, B, C, and D) in two queues. The operator allocated at multi-line workstation, which is utilised between two adjacent lines in queue 2, works on both right side of the Line I and left side of the Line II.

Figure 1. Representation of regular stations and multi-line stations on mixed-model parallel two-sided assembly lines

More than one different product model, m_{hj} $(j = 1,\ldots,M_h)$, is produced on each two-sided assembly line L_h $(h = 1,\ldots,H)$. As can be seen in Figure 1, product models A and B are assembled on Line I while C and D are performed on Line II. Each product model has its own set of tasks, t_{hji} $(i = 1,\ldots,T_{hj})$, performed according to predefined precedence relationships. P_{hji} represents the set of predecessor tasks of task t_{hji} for model m_{hj} on line L_h. Each task (t_{hji}) for model m_{hj} on line L_h, requires a certain amount of processing time (pt_{hji}) to be processed; and each line consists of a series of workstations, W_{hkx} $(k = 1,\ldots,K_h;\ x = 0,1)$.

The cycle time of each line (C_h) may be different from each other and it is calculated as follows:

$$C_h = \frac{P}{\sum_{j=1}^{M_h} D_{hj}} \qquad h = 1, \ldots, H \qquad (1)$$

where D_{hj} represents demand for model m_{hj} on line L_h over a planning period (P).

Parallel two-sided assembly lines consist of a number of two-sided serial assembly lines arranged in a parallel form. Each line may have a different cycle time. In that case, a common time should be used to assign tasks in each cycle. Gokcen *et al.* (2006) used a least common multiple (LCM) based approach for different cycle time situations (Ozcan *et al.* 2010b). In this approach, common cycle time of two lines with different cycle times is calculated as follows (Gökçen *et al.* 2006, Kucukkoc and Zhang 2013):

- *least common multiple of the cycle times is found,*
- *LD1 and LD2 are obtained through dividing both cycle times by the LCM value,*
- *two precedence diagrams are constituted with different task times by multiplying task times in each diagram with LD1 and LD2 values, respectively,*
- *LCM is determined as common cycle time of all lines.*

The model sequences of lines are important in determining the available times of operators that are allocated to multi-line stations , as the availability of an operator allocated between two adjacent lines depends on the sequence of models assembled on the lines. This issue will be explained with an example in the following subsections.

Minimum part set (MPS) principle (Bard *et al.* 1992) is used in the study to consider the model sequences integrated with balancing problem (Ozcan *et al.* 2010a). Let the greatest common divisor of D_{hj} $(j = 1, \ldots, M_h)$ is represented by cd_h $(h = 1, \ldots, H)$. The minimum part set on line L_h is represented by MPS_h, and calculated by dividing total demands of models by the greatest common divisor of these demands. The vector $d_h = (d_{h1}, \ldots, d_{hM_h})$, where $(h = 1, \ldots, H)$, represents the model mix of line L_h while MS_h represents model sequence of line L_h which is independent from the sequence of other lines.

$$d_{hj} = \frac{D_{hj}}{cd_h} \qquad j = 1, \ldots, M_h \qquad h = 1, \ldots, H \qquad (2)$$

The length of MS_h for one MPS_h, which means total number of products on line L_h for one MPS_h, is calculated as follows:

$$S_h = \sum_{j=1}^{M_h} d_{hj} \qquad (3)$$

More explanations and an example about minimum part set principle will be given in the following subsections. The maximum number of model combinations, which may appear at a cycle can be calculated as follows:

$$MS_{max} = LCM(S_1, \dots, S_H) \qquad (h = 1, \dots, H) \tag{4}$$

Since there may exist various model combinations as explained before, the system should be split into different production cycles ($\varphi = 1, \dots, \phi$), and each model combination should be of interest to balancing and sequencing in each cycle ($\phi = MS_{max}$).

2.1. Nomenclature

Following expression is considered as decision variable of the problem:

$$Y^{\varphi}_{hjikx} = \begin{cases} 1 & \text{if task } t_{hji} \text{ of model } m_{hj} \text{ is assigned to station } W_{hkx} \text{ on side } x \text{ of line } L_h \text{ in } \varphi \\ 0 & \text{otherwise} \end{cases}.$$

All other notation and parameters are summarised as follow:

2.1.1. Notation

L_h : The h^{th} line ($h = 1, \dots, H$),

m_{hj} : The j^{th} product model on line L_h ($j = 1, \dots, M_h$), where M_h is the number of product models made on line L_h,

t_{hji} : The i^{th} task for model m_{hj} on line L_h ($i = 1, \dots, T_{hj}$), where T_{hj} is total number of tasks for model m_{hj} on line L_h,

W_{hkx} : The k^{th} workstation on line L_h ($k = 1, \dots, K_h$; $x = 0, 1$), where K_h is total number of workstations on line L_h,

x : Side of the line, $= \begin{cases} 0 & \text{indicates left side of relevant line} \\ 1 & \text{indicates right side of relevant line} \end{cases}$,

φ : Production cycle ($\varphi = 1, \dots, \phi$), where $\phi = LCM(S_1, \dots, S_H)$.

2.1.2. Parameters

P : A pre-specified planning period,

P_{hji} : Set of predecessors of task t_{hji} for model m_{hj} on line L_h,

D_{hj} : Demand, over the planning period, for model m_{hj} produced on line L_h,

cd_h : Greatest common divisor of product model demands (D_{hj}) for line L_h,

MPS_h : Minimum part set or model mix of line L_h ($d_h = d_{h1},\ldots,d_{hM_h}$),

MS_h : Model sequence of line L_h,

d_{hj} : Normalised demand for model m_{hj} in model mix of line L_h, where a normalised demand for a product model is defined as the demand in terms of greatest common divisor of the relevant line,

S_h : Total number of product models on line L_h for one MPS_h (the length of MS_h for one MPS_h), $\left(S_h = \sum_{j=1}^{M_h} d_{hj}\right)$,

$LCM(S_1,\ldots,S_H)$: Least common multiple of S_h values $(h = 1,\ldots,H)$,

C_h : Cycle time of line L_h; $C_h = \frac{P}{\sum_{j=1}^{M_h} D_{hj}}$,

C : Common cycle time for all lines,

op_{hj} : Overall proportion of assembled product model m_{hj};

$op_{hj} = \frac{D_{hj}}{\sum_{j=1}^{M_h} D_{hj}}, (h = 1,\ldots,H)$,

pt_{hji} : Processing time of task t_{hji} of model m_{hj} on line L_h,

$\gamma_1, \gamma_2, \gamma_3$: User defined weighting factors to determine the significance of performance measures, i.e. the weight associated with each objective function.

2.2. Objective function

As mentioned above, the main objective of the proposed problem in this study is minimising total number of required workstations. The objective function used in this study is given in Equations 5-8.

$$Min\ Z = \gamma_1 WIT + \gamma_2 WS + \gamma_3 Q \tag{5}$$

$$WIT = \sum_{\varphi=1}^{\phi} \sum_{h=1}^{H} \sum_{k=1}^{K_h} \sum_{x\in\{0,1\}} \left(C - \sum_{j=1}^{M_h} \sum_{i=1}^{T_{hj}} op_{hj} pt_{hji} Y_{hjikx}^{\varphi} \right) \tag{6}$$

$$WS = \sum_{\varphi=1}^{\phi} \sum_{k=1}^{K_h} \sum_{x\in\{0,1\}} \frac{\sum_{h=1}^{H} \left(\sum_{j=1}^{M_h} \sum_{i=1}^{T_{hj}} op_{hj} pt_{hji} Y_{hjikx}^{\varphi} - C \right)^2}{\sum_{h=1}^{H} K_h} \tag{7}$$

$$Q = \frac{\sum_{h=1}^{H} K_h}{2H} \tag{8}$$

The main objective of the model is to minimise weighted idle times of the stations (WIT), which also means to minimise total number of utilised workstations, as well as to ensure a smooth workload (WS) among the stations from cycle to cycle. Length of the line (Q) is also considered as additional objective in the proposed model. γ_1, γ_2, and γ_3 are user defined weighting factors which allow decision makers to decide the significance levels of objectives.

2.3. Assumptions and constraints

The assumptions considered in this study are as follow:

- More than one similar product model $(j = 1, \dots, M_h)$ is assembled on each of the two or more parallel two-sided assembly lines.
- Task times (pt_{hji}) of each product model are known and deterministic.
- Cycle time is calculated according to demand over the planning horizon and can be different for different lines.
- Demand is known and deterministic for product models assembled on each line.
- Each product model has its own precedence relationships diagram and it is known.
- Common tasks between similar models must be allocated to the same workstation. Some tasks may have different processing times for different models, or the processing times may equal to zero.
- Tasks can be assigned to only a predetermined side (Left-L or Right-R) or either (E) side.
- Each task for each product model must be assigned to exactly one workstation (W_{hkx}), in other words tasks cannot be split to more than one workstation.
- Sum of the all task times assigned to a workstation constitutes its workload, and workload of a station cannot exceed the predetermined cycle time (C_h) of the relevant line.
- A task can only be assigned if all of its predecessors (P_{hji}) have been completed. That can be achieved in two alternative ways:
  - all predecessors are completed before the current queue, or
  - if some of the predecessor tasks are assigned to the current queue, then all predecessors are completed before the initialisation of the task.
- Operators are multi-skilled and can work at each side of a line.
- Only one operator is assigned to a workstation.

- Operator travel times are ignored.
- No work in process inventory is allowed.
- Starting and finishing times are same for all lines.

3. An explanatory example of MPTALB/S problem

In this section, an illustrative example is provided to elaborate the problem. Assume that there is a line system, which consists of two mixed-model parallel two-sided assembly lines, as depicted in

Figure 2. As can be seen from the figure, two models (A and B) are executed on Line I while remaining two models (C and D) are produced on Line II, simultaneously.

Figure 2. A schematic view of mixed-model parallel two-sided assembly lines

Eleven workstations are utilised as illustrated in the figure, where one of the operators performs on both adjacent lines. Operator 6 performs tasks on Line I and on Line II as well. Line balance and sequence of the models affect the workload of a station in a cycle, because different model mixes may exist at multi-line stations, which are utilised on two adjacent lines.

Let us consider demands are 10, 30, 20, and 20 for the models A, B, C, and D respectively ($D_{1A} = 10, D_{1B} = 30, D_{2C} = 20, D_{2D} = 20$) over the pre-specified planning horizon, 480 time units. The cycle times of the lines are calculated easily using Equation 1 ($C_1 = C_2 = 480\ time\ units/40\ items = 12\ time\ units/item$).

As described above, minimum part set of each line (MPS_h) is calculated by dividing total demands (10, 30, 20, 20) of models (A, B, C, D) by the greatest common divisor of these demands for each line. While the greatest common divisor (cd_1) of D_{1j} ($j = 1, \ldots, M_h$) is 10 for Line I, cd_2 is calculated as 20 for Line II. So, model mix of Line I (d_1) can be calculated for MPS_1 as follows:

$$d_1 = (D_{1A}/\ cd_1,\ D_{1B}/\ cd_1) = (1,3).$$

Similarly, model mix of Line II is $d_2 = (1,1)$. Consequently, the total number of products in line h for one MPS_h is; $S_1 = 4$ for L_1, and $S_2 = 2$ for L_2.

If the model sequences are considered as $MS_1 = ABBB$ and $MS_2 = CD$ for Line I and Line II respectively, possible model mixes of the given example can be represented as in Table 3. Three different model mixes appear at multi-line station, station 6, and same combinations repeat by cycle 5. Therefore, there exist four different model mixes for the sequence of models on two adjacent lines. This situation can be illustrated as in Figure 3 for four production cycles.

Table 3. Possible mixes of product models for given example ($MS_1 = ABBB$ and $MS_2 = CD$)

Figure 3. Model-mixes of the problem when $MS_1 = ABBB$ and $MS_2 = CD$

Based on this illustration, it is obvious that model combinations will change in workstation 6 in case of consideration of different model mixes on the lines rather than $MS_1 = ABBB$ and $MS_2 = CD$. Accordingly, workload and availability of the operator who performs at this station will be affected by that change. Consequently, model-sequencing problem on the lines must also be taken into account with balancing problem, simultaneously.

If the model sequences are considered as $MS_1 = BBAB$ and $MS_2 = DC$, all possible model mixes that may appear on the lines can be represented as in Table 4.

Table 4. Possible mixes of models for another model combination ($MS_1 = BBAB$ and $MS_2 = DC$)

4. Solution approach

The newly proposed line system and problem definition based on this system have been explained in previous sections. This section first addresses how natural ant systems work briefly. Then, it describes framework of a proposed agent based ant colony optimisation algorithm for solving this problem and illustrates its operational principles step by step through an example.

4.1. Ant colony optimisation

Ant colony optimisation is inspired from the collective behaviour of ants and is one of the most efficient meta-heuristics in solving combinatorial optimisation problems. Ant algorithms, initially proposed by Dorigo et al. (1996), belong to the category of nature inspired algorithms. The initial form of ant colony optimisation techniques, called the ant system, was developed to solve small-sized travelling salesman problem with up to 75 cities. Since then, several researchers carried out a substantial amount of research in ant colony optimisation and have developed algorithms which demonstrate better performance than the ant system. Ant colony optimisation algorithms mimic real ant colonies in the nature and their capability of finding the shortest path between the nest and food sources, where each ant represents a complete solution. Foraging behaviour of ants help them find the shortest path by depositing a substance called pheromone on the ground while they are walking. In this way, a pheromone trail is formed and ants smell pheromone to choose their way in probability. Paths involve strong pheromone levels have more chance to be selected by ants (Dorigo et al. 1999). When a set of possible paths are given to the ants, each ant chooses one path randomly, and apparently some ants picking the shortest path will return faster. Then, there will be more pheromone on the shortest path, influencing later ants to follow this path, after their completion of one tour. By time, the path which has high level of pheromone will be most often selected and considered as the shortest route (Leung et al. 2010).

4.2. ABACO framework

There is an increasing interest in agent based methodologies in solving complex problems that may be too large for centralised approaches. In agent based methodologies, a network of problem solvers collaborate with each other to find solutions for problems that are beyond their individual capabilities.

Our algorithm for solving the defined problem is referred to as ABACO, which is developed in a Java programming environment and has four levels of computational systems. The architecture of ABACO comprises different classes specialised in carrying out different objectives. Initialisation and planning processes are performed at the first level by Facilitator Agent (FA). Customer demands of products are considered in calculating cycle times of independent lines; and MPS_h values of each line is calculated by dividing total demands of models by the greatest common divisor of these demands for each line. At the second level, MPS_h values are sent to Sequencing Agent (SA) and sequencing procedure is invoked. All possible model sequences based on minimum part sets are generated and returned one by one to FA to be used later by Balancing Agent (BA) at the third level. At the same time, different production cycles are also computed by SA and returned to BA. The precedence relationships between tasks are read by BA and a new colony is released with a predefined number of ants. Each ant in the colony builds a balancing solution (as will be explained later in this section) and the best solution in each colony is returned to the BA. BA iterates this procedure until a predefined number of colonies have been used and returns the best solution from each colony to FA. FA sends a new sequence to another BA until all sequences are processed. Then, solutions obtained by BAs for different sequences are conveyed to FA and evaluated by FA to present the best solution as output. The four-level ABACO system constructed for MPTALB/S problem is outlined in Figure 4.

Figure 4. Proposed ABACO platform

4.2.1. Procedures

The colonies of ants, where each ant represents a potential solution, perform balancing procedures based on model sequences and guidance received from BA. When a new solution is built by an ant (as will be explained in Section 4.2.2), an amount of pheromone, which represents a temporary signal that later ants may follow it, is laid on the edges of found solution (edges represents tasks selected in succession). Laid pheromone amount is calculated by considering the quality measure of the solution. To help the algorithm converge, double amount of pheromone is deposited on edges of the best solution for all iterations of each colony. The equations and explanations related to these calculations will be given later in this section.

Figure 5 depicts the outline of ant colony algorithm procedure. The algorithm starts with initialisation of pheromones. A new sub-colony is released and different solutions (paths) are obtained by each ant in the colony using the given solution building procedure in Figure 6. The basic idea is selection of tasks to be added to the current workstation by artificial ants. Pheromone level determines the probability of a task being selected by an ant. Pheromone amount, a measure of each path's relative desirability, is calculated according to the quality of the drawn path by each ant.

Figure 5. Outline of ant colony algorithm procedure

In the algorithm, pheromone is released between task and position of the workstation by each ant according to the quality of drawn path. For this aim, a task-workstation matrix that holds pheromone levels between those entities is employed. A constant value of pheromone is evaporated after each tour. When a sub-colony has completed their tour, global best solution is updated if a better solution is found and double pheromone is laid to the edges of global best solution. The algorithm continues until all colonies complete their tours and stops when a predetermined maximum sub-colony number is exceeded. Task selection probability, pheromone deposition and evaporation strategies are given below:

The probability of selection task i for ant n in workstation k is:

$$p_{ik}^{n} = \frac{[\tau_{ik}]^{\alpha}[\eta_i]^{\beta}}{\sum_{y \epsilon Z_i^n}[\tau_{iy}]^{\alpha}[\eta_i]^{\beta}} \qquad (9)$$

where Z_i^n indicates candidate task list for ant n after selection of task i; τ_{ik} is the amount of virtual pheromone between *task – workstation position;* and η_i is the heuristic information of task i that comes from *ranked positional weight method (RPWM)*[1].

The pheromone update rule is:

[1] RPWM is a well-known heuristic widely used in solving assembly line balancing problems. It calculates positional weights of tasks and ranks them to be prioritised for assignment. Each task has its own weight and the weight is computed by summing all the successor tasks' times. Tasks with the highest positional weights are selected and assigned to earlier stations to allow assignment of successor tasks. Please see Helgeson and Birnie (1961) for detailed information about RPWM.

$$\tau_{ik} \leftarrow (1-\rho)\tau_{ik} + \Delta\tau_{ik} \quad (10)$$

$$\Delta\tau_{ik} = \frac{1}{Objective\ Function\ Value} \quad (11)$$

4.2.2. Building a balancing solution

Ants employed in each colony at level 4 build solutions and make operational decisions in a decentralized manner. The flowchart of the procedures for building a balancing solution is represented in Figure 6. A similar version of this procedure has also been used by Simaria and Vilarinho (2009). A balancing solution is generated by each ant in the colony using the procedure given in Figure 6. At the beginning, all tasks are grouped according to line and preferred operation direction data. Then, the procedure starts with randomly selecting a line and operation side to start assigning tasks. Available tasks that satisfy capacity constraints of the current station, have no predecessors or all of their predecessor tasks already completed, and do not violate interference rule are determined for designated line and operation direction. Among assignable tasks, a task is selected by benefiting from pheromone value and heuristic information for the relevant task.

Figure 6. Constructing a balancing solution procedure by each ant

A timeline is used by the algorithm in the balancing procedure. If the current time of the side is lower than the current time of the opposite side ($st(k) < st(\underline{k})$), assignment continues on the same side. Otherwise, side is changed.

If assignable tasks list is empty at any time, an action is taken based on the reason investigated. If there is no capacity to assign any task and the workstation lies between two adjacent lines, it could be merged with adjacent station on the other line. Otherwise, either the side is changed; or the line is changed if both sides are full and then station number is increased. If there are tasks whose predecessors have been assigned to the opposite side but will be finished in a forward time (this phenomenon is called interference as mentioned earlier), the current time is forwarded to the current time of opposite side ($st(k) \leftarrow st(\underline{k})$). The procedure continues with a randomly selected side.

Task side incompatibility occurs when there are no tasks that can be assigned to the current side. This may be caused by one of the following reasons:

- If the current time of the current side is inferior from the current time of the opposite side ($st(k) < st(\underline{k})$): The current time is forwarded to the current time of the opposite side and then a random side is selected to continue.

- If the current time of the current side is equal or greater than the opposite side's current time ($st(k) \geq st(\underline{k})$): The assignment procedure continues on the opposite side.

During the task allocation process, if the current side of a line lies between two lines (i.e. Line I side right, or Line II side left, called interval side) and the efficiency of the current workstation is lower than 75% ($steff < 75\%$), the current station is merged with the adjacent workstation in the other line so that some tasks can be performed from other line.

5. An example problem

To explain the proposed model sequencing and line balancing procedures, a numerical example is presented in this section. Computations of minimum part sets and possible model mixes have already been explained in Section 3. In this example, negotiations among Facilitator Agent, Sequencing Agent, and Balancing Agent; and assigning tasks to workstations are illustrated visually.

Two different problems (P12 and P9) are taken from the literature (Kim *et al.* 2000) and combined to explain the simultaneous sequencing and balancing procedure of the proposed method. Precedence relationships and preferred operation directions of the problems are depicted in Figure 7. Arcs from node(s) to node(s) represent precedence relationships and letters over each node symbolise preferred operation directions, where L means Left, R means Right, and E means Either side. Task times for each product model are generated randomly as a number between zero and the predetermined cycle time value, and are given in Table 5.

Figure 7. Precedence relationship diagrams for the illustrative example: (a) P12 and (b) P9, adapted from (Kim et al. 2000)

Table 5. Task times for product models

Three models from P12 (models A, B, and C) are assigned to Line I while two models from P9 (D, and E) are assigned to Line II. Demands for models are assumed to be 10, 10, and 10 for models A, B, and C on Line I; and 20, and 10 for models D, and E on Line II, respectively. Based on model demands, minimum part sets are calculated as $MPS_1 = (1,1,1)$ and $MPS_2 = (2,1)$ for Line I and Line II, respectively.

For a planning horizon of 150 time units, cycle time is calculated as 5 time units for both lines where production starts and finishes at the same time. Cycle times are same for this example but each line may have a different cycle time. In that case, a common time should be used to assign tasks in each cycle (as explained in Section 2). Please refer to studies of Gokcen et al. (2006) and Ozcan et al. (2010b) to find more about least common multiple (LCM) based approach for situations involving different cycle times. The overall proportions of the number of units of product models are the same ($q_A = q_B = q_C$) for Line I while model D doubles model E ($q_D = 2q_E$) for Line II.

Minimum part sets are computed by FA and possible model sequences are requested from SA. Then, obtained model sequences are sent to BAs to produce balancing solutions using ACO algorithm given in Figure 5 and Figure 6. A new colony is charged by BA to produce a solution for each different model sequence. Finally, obtained solutions are evaluated by FA and the solution which has the best objective value is determined as the solution of the problem. This communication process between agents for model sequencing and line balancing procedures are represented as in Figure 8.

Figure 8. Simultaneous model sequencing - line balancing procedure

To represent a sample output of model sequencing - line balancing procedure, a general solution we have obtained is given in Figure 9. Please see Figure A1 in appendices section for detailed balancing results cycle-by-cycle. Based on the input data given above for the example problem, model sequences are assumed as $MS_1 = C, B, A$ for Line I and $MS_2 = D, D, E$ for Line II. As already explained above, total number of production cycles subject to consideration in this example can be computed as:

$$MS_{max} = LCM(S_1, S_2) = LCM(3,3) = 3.$$

Figure 9. Representation of a balancing solution for given example

As can be seen from Figure 9, product models A, B, and C are assembled on Line I while D and E are assembled on Line II. Different colours symbolise different product models and some tasks may require varied processing time for different models. If a task requires "0" time units, it means this task is not required for this product model and those tasks are not shown on the diagram. Task times are given in horizontal bars where lengths of bars correspond to processing times of related tasks. Idle times are represented by grey shaded bars.

Although it looks like ten workstations are utilised, in fact nine operators are needed for this balance, because the workstation utilised between two adjacent lines in queue 3 is considered as a multi-line workstation and only one operator is enough to perform tasks in this workstation for both lines. This issue can be comprehended in Figure A1. Nevertheless, sequence of models is a significant factor that affects the efficiency of the lines as well as task sequencing. Since task times vary from one model to another, the sequence of models on the line influences the availability of the operators, who perform their jobs in multi-line workstations. The operator works in queue 3 in multi-line workstation performs jobs on models C-D, B-D, and A-E in production cycles 1, 2, and 3, respectively (see Figure A1). Although grey shades exist in balancing solution in this workstation, there is no idle time since one operator works on both adjacent lines. As aforementioned, utilisation of multi-line stations is one of the major benefits of parallel lines.

6. Discussion

In this research, an experimental study has not been carried since the main objective is to introduce the problem rather than to demonstrate the performance of proposed method. Proposed method provides an insight to solve the problem for further researchers. So, experimental studies and related statistical tests to prove its superiority are left to further research. However, an example is given in order to illustrate simultaneous model sequencing and balancing of a simple mixed-model parallel two-sided assembly line system.

As it has already been stated above, the main benefit of the proposed assembly line system is its flexibility to produce more than one product model on the same line with less workforce, because constructing multi-line workstations on more than one assembly line minimises operator requirements. However, the complexity of the problem increases dramatically with the consideration of various product models, which have different precedence relationships, task times, and sequences on the lines. Wee and Magazine (1982) showed that simple assembly line balancing problem is an NP-hard class of combinatorial problem. Since MPTALB/S problem is a much more complex version of simple assembly line balancing problem, it is also NP-Hard, which means that obtaining an optimal solution when the problem size increases becomes difficult, because, the solution space will grow exponentially as the number of tasks increases (Wu *et al*. 2008). It is the major reason why; (i) a considerable amount of researches in the literature strives to develop heuristics and meta-heuristics instead of exact algorithms to solve assembly line balancing problems, and (ii) an agent based ant colony optimisation algorithm is proposed in this study for MPTALB/S problem.

Assumptions made in Section 2 could be considered as limitations of the work. Relaxation of some of these assumptions may lead to an increased balancing solution which is more efficient or realistic. For example, common tasks can be assigned to different workstations and/or separate precedence diagrams can be employed instead of a combined precedence diagram for different product models. However, assigning common tasks of different models to different workstations may cause additional equipment costs. Dynamic demand is also another challenging issue which manufacturers face with in real world applications.

In case of consideration different cycle times for parallel lines, using classical LCM approach used by Gokcen et al. (2006) may not be sufficient to satisfy capacity constraints in multi-line stations for different model mixes. This is why, production cycles will change in different time slots for different lines. That means calculated model combinations for multi-line stations will change as well. Therefore, a comprehensive study is needed to calculate all possible model combinations in this case.

Industrial implication of proposed line system is that it is feasible for such systems to produce large-sized products. Such systems also enable the satisfaction of customised demands in a cost effective manner with shared use of common tools and the flexibility of producing different models with different throughput rates. The systems also reduce operator requirement so that line efficiencies can be improved. Due to these advantages and those explained in Section 2, some of the companies have already utilised mixed-model parallel two-sided assembly lines though there is no academic research on this topic yet.

7. Conclusions and future research directions

The mixed-model parallel two-sided assembly line system has been introduced along with its characteristics. The operation of the system has been illustrated through examples with changing model combinations cycle-by-cycle. Although the proposed line system is frequently used in producing large sized products like automobiles, and trucks in industry, it has not been studied in an academic manner in the literature. Based on this motivation, this paper addresses a new type of line balancing problem in the literature. The major objective of the study is to introduce the problem of simultaneous sequencing and balancing of mixed-model parallel two-sided assembly lines and to initiate future research in this field. Moreover, an agent based ant colony optimisation algorithm is proposed to show how this kind of problems can be solved using an agent based ant colony optimisation algorithm. To the best knowledge of the authors, the proposed algorithm in this study is the first agent based approach for parallel two-sided assembly line balancing problems. It is used to minimise the number of stations by minimising idle times and to combine the advantages of both parallel and two-sided assembly lines by benefiting from model variation flexibilities.

An example is generated using two different test problems in the literature to demonstrate the solution procedure of the problem, visually. Outline of proposed ABACO framework; flowcharts of ant colony optimisation and procedures for building balancing solutions; and communications between agents are depicted in figures. An output of these procedures, in accordance with varied model sequences and combinations from one production cycle to another, is also exhibited in appendices and explored in text.

Experimental studies to assess the efficiency and performance of the proposed algorithm are left to future research since the main objective of this paper is to introduce a new problem, as aforementioned. Another reason is that, test problems must be newly generated or adapted from previous test problems for computational experiments, since there is not yet test problem set for this type of line configuration problems. Further research is being carried out to construct test problems, based on which the efficiency and performance of both proposed line system and the solution algorithm can be compared with other similar line systems like parallel two-sided assembly lines with no model variations, and/or mixed-model two-sided assembly lines without parallelisation.

Undoubtedly, more powerful solution approaches are needed. The reason is that, including model sequencing problem into balancing of a complex line system increases the complexity of the entire problem dramatically and requires more and more CPU time. Other meta-heuristics (evolutionary algorithms, tabu search algorithm, simulated annealing, etc.) or their combinations might also be proposed to increase the solution capacity of algorithm; or exact solution procedures and mathematical formulations may be developed to solve the problem. Furthermore, some constraints that reflect more realistic conditions in real applications (i.e. zoning constraints, task synchronisation constraints, and positional constraints) may be of interest for future studies.

Acknowledgment

Second author gratefully acknowledge financial support from Balikesir University and Turkish Council of Higher Education during his PhD research at the University of Exeter, UK. Both authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Appendices

Figure A1. Detailed result of the model sequencing – line balancing procedure for $MS_1 = C, B, A$ and $MS_2 = D, D, E$; (a) production cycle 1, (b) production cycle 2, (c) production cycle 3

References

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), 449-457.

Bard, J.F., Darel, E. and Shtub, A., 1992. An analytic framework for sequencing mixed model assembly lines. *International Journal of Production Research,* 30 (1), 35-48.

Bartholdi, J.J., 1993. Balancing 2-sided assembly lines - a case-study. *International Journal of Production Research,* 31 (10), 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), 259-277.

Bautista, J. and Pereira, J., 2011. Procedures for the time and space constrained assembly line balancing problem. *European Journal of Operational Research,* 212 (3), 473-481.

Baybars, I., 1986. An efficient heuristic method for the simple assembly line balancing problem. *International Journal of Production Research,* 24 (1), 149-166.

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), 582-588.

Baykasoglu, A., Ozbakir, L., Gorkemli, L. and Gorkemli, B., 2009. Balancing parallel assembly lines via ant colony optimization. *CIE: 2009 International Conference on Computers and Industrial Engineering, Vols 1-3*, 506-511.

Becker, C. and Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research,* 168 (3), 694-715.

Benzer, R., Gokcen, H., Cetinyokus, T. and Cercioglu, H., 2007. A network model for parallel line balancing problem. *Mathematical Problems in Engineering,* doi:10.1155/2007/10106.

Berger, I., Bourjolly, J.M. and Laporte, G., 1992. Branch-and-bound algorithms for the multiproduct assembly line balancing problem. *European Journal of Operational Research,* 58 (2), 215-222.

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

Cercioglu, H., Ozcan, U., Gokcen, H. and Toklu, B., 2009. A simulated annealing approach for parallel assembly line balancing problem. *Journal of the Faculty of Engineering and Architecture of Gazi University,* 24 (2), 331-341.

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), 39-55.

Darel, E.M. and Cother, R.F., 1975. Assembly line sequencing for model mix. *International Journal of Production Research,* 13 (5), 463-477.

Dorigo, M., Di Caro, G. and Gambardella, L.M., 1999. Ant algorithms for discrete optimization. *Artificial Life,* 5, 137–172.

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), 29-41.

Eryuruk, S.H., Baskak, M. and Kalaoglu, F., 2011. Assembly line balancing by using statistical method in clothing production. *Tekstil Ve Konfeksiyon,* 21 (1), 65-71.

Eryuruk, S.H., Kalaoglu, F. and Baskak, M., 2008. Assembly line balancing in a clothing company. *Fibres & Textiles in Eastern Europe,* 16 (1), 93-98.

Ford, H., 2009. *My life and work - an autobiography of henry ford* New York: Classic House Books.

Gökçen, H., Agpak, K. and Benzer, R., 2006. Balancing of parallel assembly lines. *International Journal of Production Economics,* 103 (2), 600-609.

Gökçen, H., Ağpak, K., Gencer, C. and Kizilkaya, E., 2005. A shortest route formulation of simple u-type assembly line balancing problem. *Applied Mathematical Modelling,* 29 (4), 373-380.
Gunasekaran, A. and Cecille, P., 1998. Implementation of productivity improvement strategies in a small company. *Technovation,* 18 (5), 311-320.
Hamzadayi, A. and Yildiz, G., 2012. A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model u-lines with parallel workstations and zoning constraints. *Computers & Industrial Engineering,* 62 (1), 206-215.
Hoffmann, T.R., 1992. Eureka - a hybrid system for assembly line balancing. *Management Science,* 38 (1), 39-47.
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), 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), 435-440.
Kara, Y., Gokcen, H. and Atasagun, Y., 2010. Balancing parallel assembly lines with precise and fuzzy goals. *International Journal of Production Research,* 48 (6), 1685-1703.
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), 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), 853-865.
Klein, R. and Scholl, A., 1996. Maximizing the production rate in simple assembly line balancing - a branch and bound procedure. *European Journal of Operational Research,* 91 (2), 367-385.
Kucukkoc, I., Karaoglan, A.D. and Yaman, R., 2013a. Using response surface design to determine the optimal parameters of genetic algorithm and a case study. *International Journal of Production Research,* 51 (17), 5039-5054.
Kucukkoc, I. and Yaman, R., 2013. A new hybrid genetic algorithm to solve more realistic mixed-model assembly line balancing problem. *International Journal of Logistics Systems and Management,* 14 (4), 405-425.
Kucukkoc, I. and Zhang, D.Z., 2013. Balancing parallel two-sided assembly lines via a genetic algorithm based approach. *Proceedings of the 43rd International Conference on Computers and Industrial Engineering (CIE43).* The University of Hong Kong, Hong Kong, 1-16.
Kucukkoc, I., Zhang, D.Z. and Keedwell, E.C., 2013b. Balancing parallel two-sided assembly lines with ant colony optimisation algorithm. *Proceedings of the 2nd Symposium on Nature-Inspired Computing and Applications (NICA) at Artificial Intelligence and the Simulation of Behaviour (AISB) 2013 Convention* University of Exeter, Exeter, UK, 21-28.
Kucukkoc, I., Zhang, D.Z., Keedwell, E.C. and Pakgohar, A., 2013c. An improved ant colony optimisation algorithm for type-i parallel two-sided assembly line balancing problem. *The OR Society - YOR18 Biennial Conference.* University of Exeter, Exeter, UK.
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), 273-292.

Leung, C.W., Wong, T.N., Mak, K.L. and Fung, R.Y.K., 2010. Integrated process planning and scheduling by an agent-based ant colony optimization. *Computers & Industrial Engineering,* 59 (1), 166-180.

Liu, S.B., Ng, K.M. and Ong, H.L., 2008. Branch-and-bound algorithms for simple assembly line balancing problem. *International Journal of Advanced Manufacturing Technology,* 36 (1-2), 169-177.

Lusa, A., 2008. A survey of the literature on the multiple or parallel assembly line balancing problem. *European Journal of Industrial Engineering,* 2 (1), 50-72.

Nourmohammadi, A. and Zandieh, M., 2011. Assembly line balancing by a new multi-objective differential evolution algorithm based on topsis. *International Journal of Production Research,* 49 (10), 2833-2855.

Ozbakir, L., Baykasoglu, A., Gorkemli, B. and Gorkemli, L., 2011. Multiple-colony ant algorithm for parallel assembly line balancing problem. *Applied Soft Computing,* 11 (3), 3186-3198.

Ozbakir, L. and Tapkan, P., 2010. Balancing fuzzy multi-objective two-sided assembly lines via bees algorithm. *Journal of Intelligent & Fuzzy Systems,* 21 (5), 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), 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), 81-97.

Ozcan, U., Cercioglu, H., Gokcen, H. and Toklu, B., 2009. A tabu search algorithm for the parallel assembly line balancing problem. *Gazi University Journal of Science,* 22 (4), 313-323.

Ozcan, U., Cercioglu, H., Gokcen, H. and Toklu, B., 2010a. Balancing and sequencing of parallel mixed-model assembly lines. *International Journal of Production Research,* 48 (17), 5089-5113.

Ozcan, U., Gokcen, H. and Toklu, B., 2010b. Balancing parallel two-sided assembly lines. *International Journal of Production Research,* 48 (16), 4767-4784.

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

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

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

Pastor, R., Andres, C., Duran, A. and Perez, M., 2002. Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *Journal of the Operational Research Society,* 53 (12), 1317-1323.

Peeters, M. and Degraeve, Z., 2006. A linear programming based lower bound for the simple assembly line balancing problem. *European Journal of Operational Research,* 168 (3), 716-731.

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), 189-199.

Rabbani, M., Moghaddam, M. and Manavizadeh, N., 2012. Balancing of mixed-model two-sided assembly lines with multiple u-shaped layout. *International Journal of Advanced Manufacturing Technology,* 59 (9-12), 1191-1210.

Rekiek, B. and Delchambre, A., 2006. *Assembly line design: The balancing of mixed-model hybrid assembly lines with genetic algorithms* London: Springer.

Rekiek, B., Dolgui, A., Delchambre, A. and Bratcu, A., 2002. State of art of optimization methods for assembly line design. *Annual Reviews in Control,* 26, 163-174.

Rubinovitz, J. and Levitin, G., 1995. Genetic algorithm for assembly line balancing. *International Journal of Production Economics,* 41 (1-3), 343-354.

Saltzman, M.J. and Baybars, I., 1987. A 2-process implicit enumeration algorithm for the simple assembly line balancing problem. *European Journal of Operational Research,* 32 (1), 118-129.

Sarker, B.R. and Shanthikumar, J.G., 1983. A generalized-approach for serial or parallel line balancing. *International Journal of Production Research,* 21 (1), 109-133.

Scholl, A. and Boysen, N., 2009. Designing parallel assembly lines with split workplaces: Model and optimization procedure. *International Journal of Production Economics,* 119 (1), 90-100.

Simaria, A.S., 2006. *Assembly line balancing - new perspectives and procedures.* Universidade de Aveiro.

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), 489-506.

Sprecher, A., 1999. A competitive branch-and-bound algorithm for the simple assembly line balancing problem. *International Journal of Production Research,* 37 (8), 1787-1816.

Suer, G.A., 1998. Designing parallel assembly lines. *Computers & Industrial Engineering,* 35 (3-4), 467-470.

Suer, G.A. and Dagli, C., 1994. A knowledge-based system for selection of resource allocation rules and algorithms. *In* Mital, A. and Anand, S. eds. *Handbook of expert system applications in manufacturing; structures and rules.* London: Chapman and Hall, 108-147.

Taha, R.B., El-Kharbotly, A.K., Sadek, Y.M. and Afia, N.H., 2011. A genetic algorithm for solving two-sided assembly line balancing problems. *Ain Shams Engineering Journal,* 2 (3-4), 227-240.

Tanenbaum, M. and Holstein, W.K., 2012. *Mass production* [online]. Britannica Academic Edition, Science and Technology. Available from: www.britannica.com/EBchecked/topic/368270/mass-production [Accessed Access Date 2012].

Wee, T.S. and Magazine, M.J., 1982. Assembly line balancing as generalized bin packing. *Operations Research Letters,* 1 (2), 56-58.

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), 1009-1015.

Yegul, M.F., Agpak, K. and Yavuz, M., 2010. A new algorithm for u-shaped two-sided assembly line balancing. *Transactions of the Canadian Society for Mechanical Engineering,* 34 (2), 225-241.

Yin, F., Sun, J. and Wu, A., Year. Two-sided with multi-parallel stations assembly line balancing based on heuristic algorithm *In*: *IEEE 18th International Conference* *on Industrial Engineering and Engineering Management (IE&EM)*, Changchun International Convention Center, China, 996-998.

Zhang, Z.Q., Cheng, W.M., Tang, L.S. and Zhong, B., 2007. Ant algorithm with summation rules for assembly line balancing problem. *Proceedings of 14th International Conference on Management Science & Engineering,* Vols 1-3, 369-374.

List of Figures

List of Tables

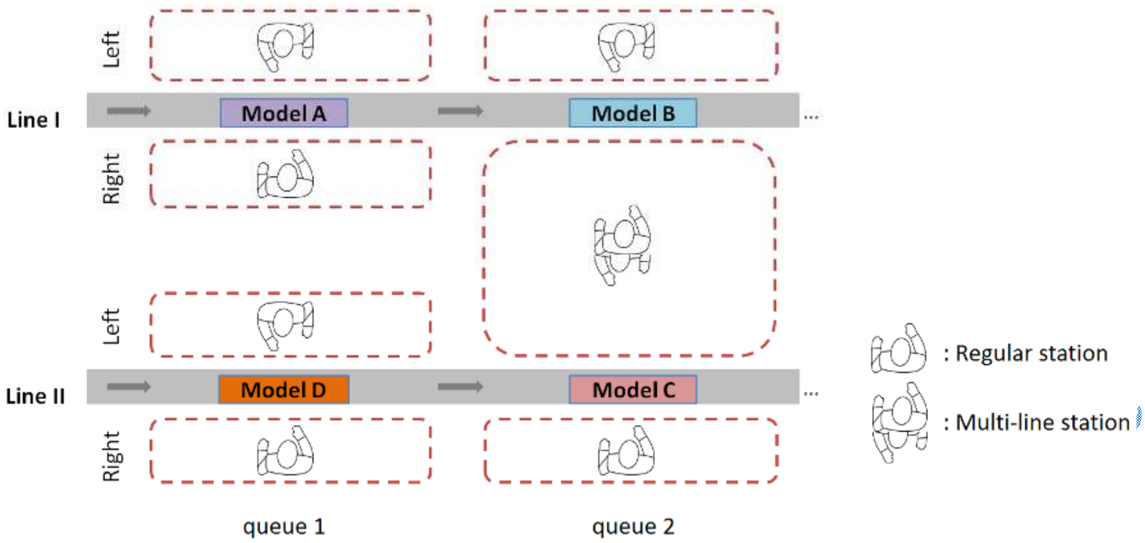


Figure 1. Representation of regular stations and multi-line stations on mixed-model parallel two-sided assembly lines

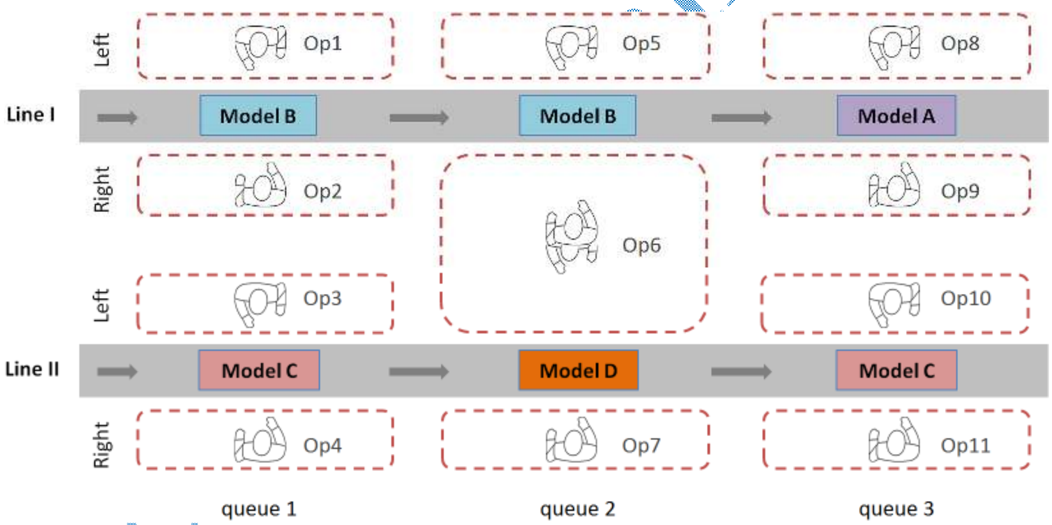


Figure 2. A schematic view of mixed-model parallel two-sided assembly lines

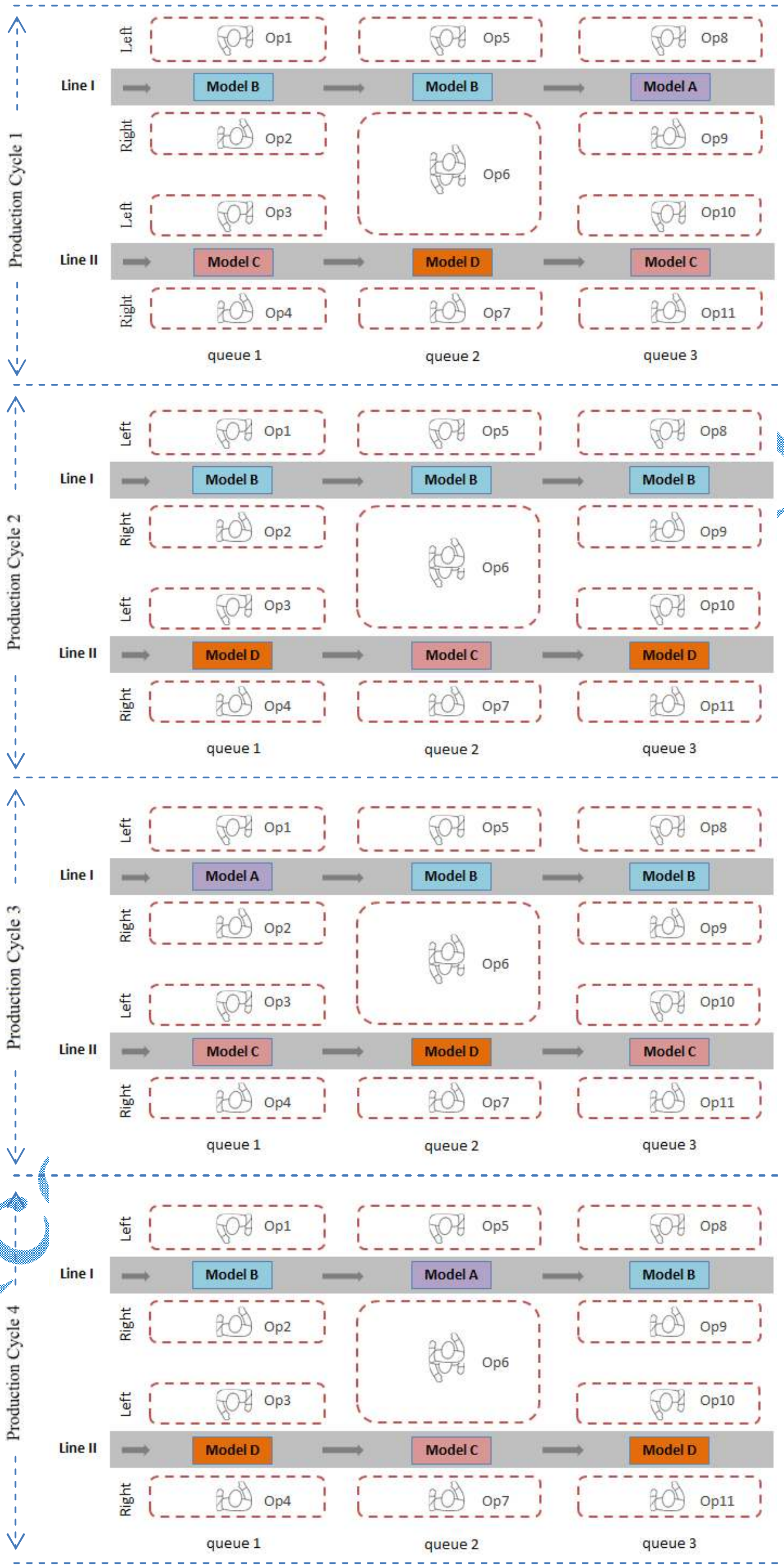


Figure 3. Model-mixes of the problem in the case $MS_1 = ABBB$ and $MS_2 = CD$

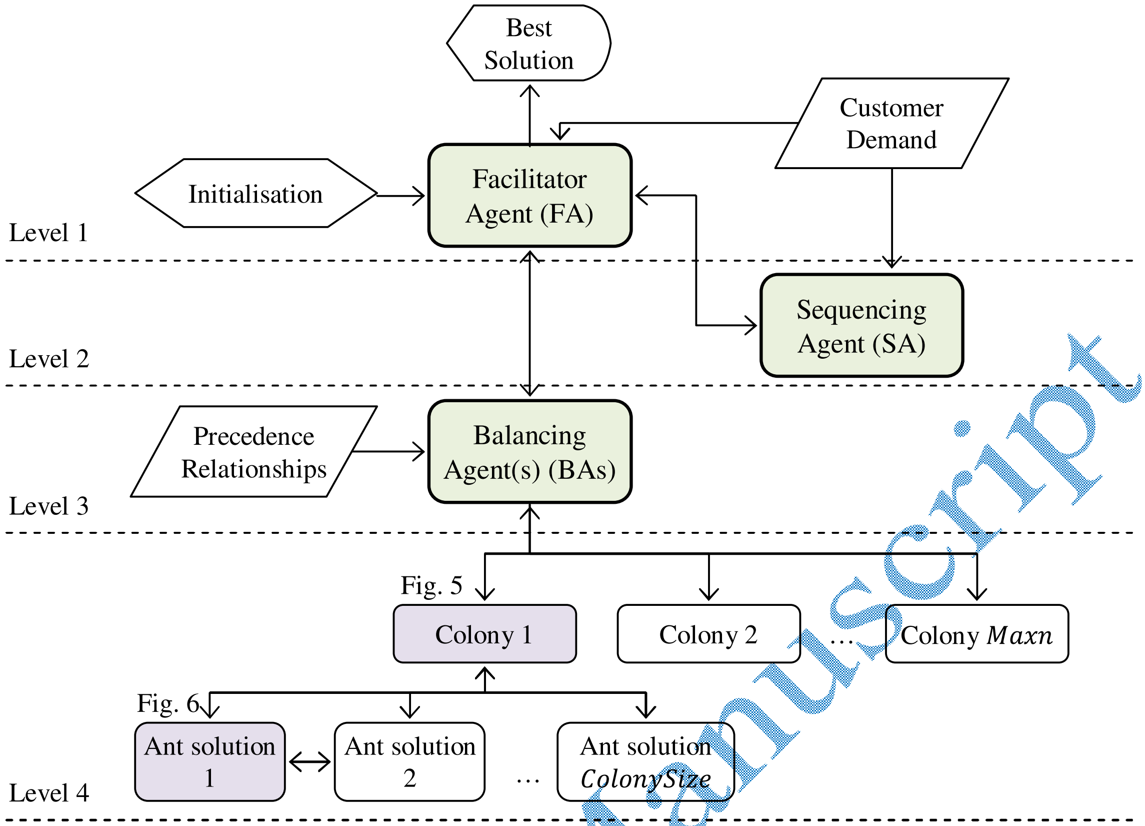


Figure 4. Proposed ABACO platform

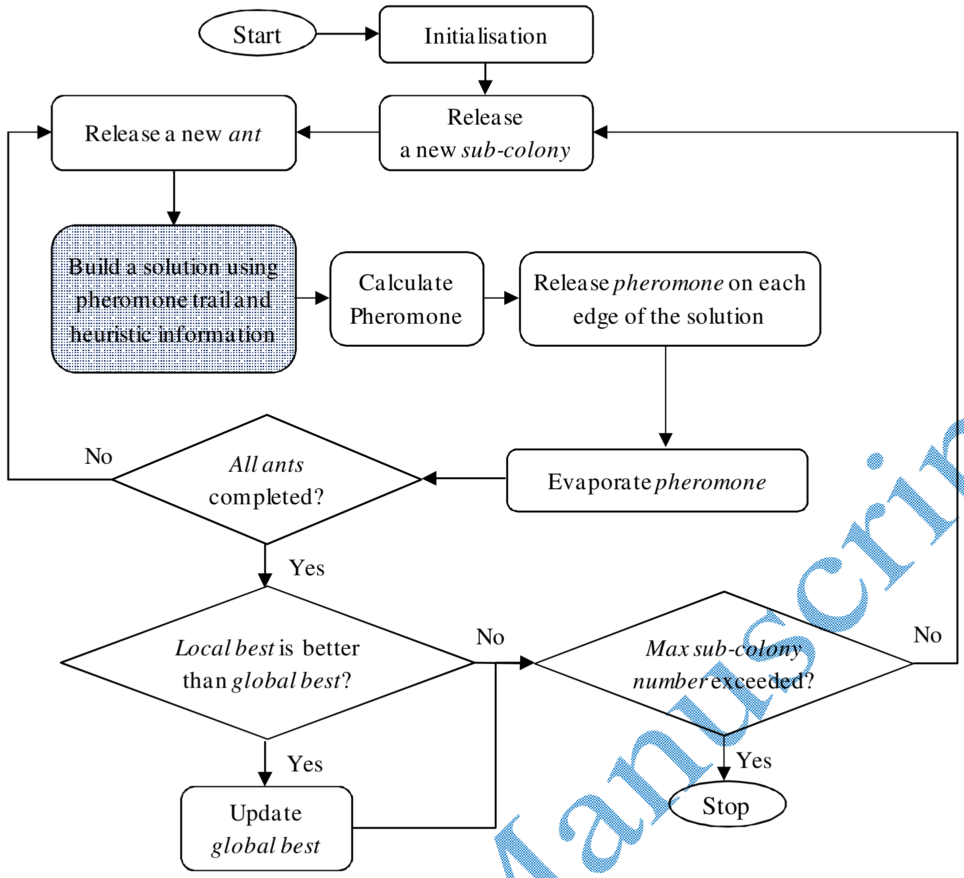


Figure 5. Outline of ant colony algorithm procedure

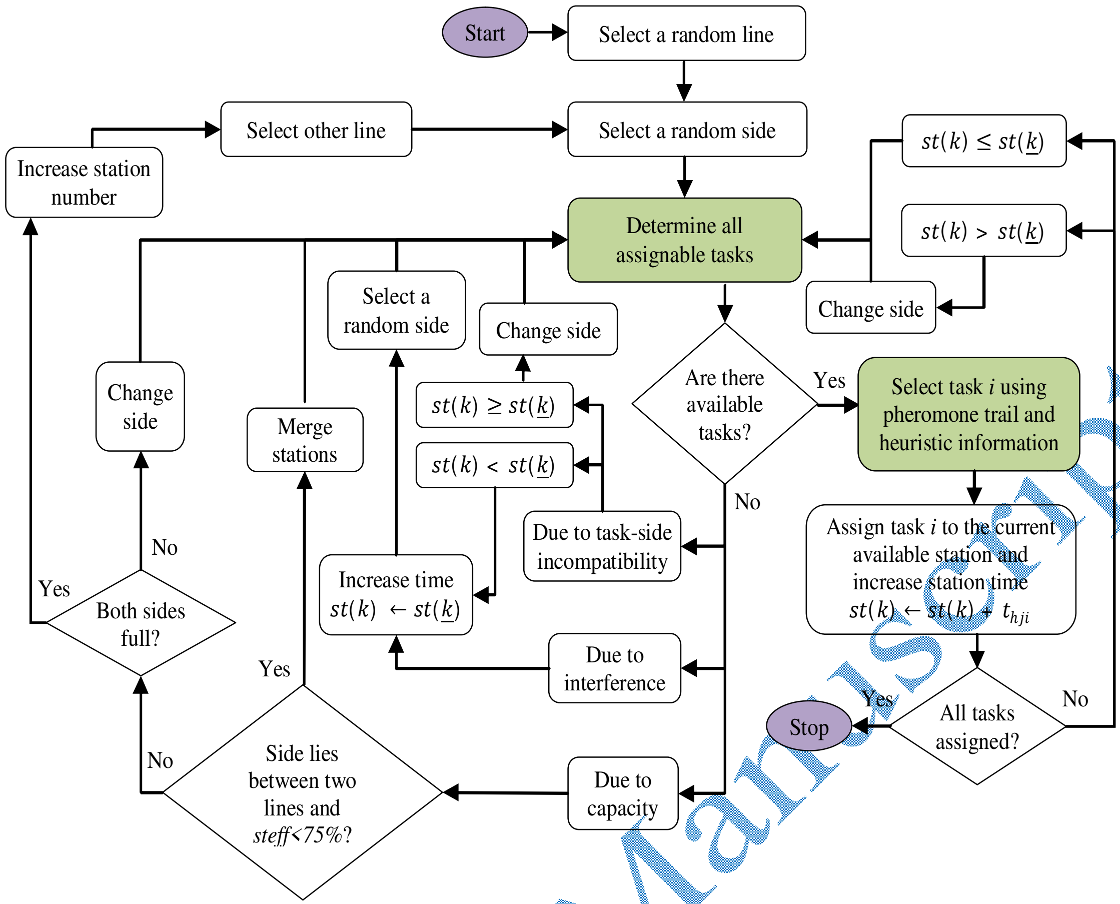


Figure 6. Constructing a balancing solution procedure by each ant

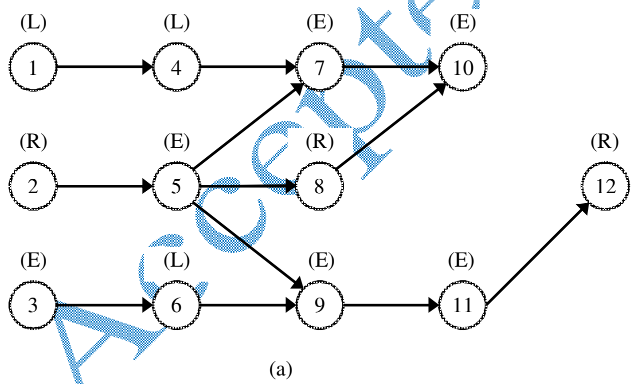


Figure 7. Precedence relationship diagrams for illustrative example: (a) P12 and (b) P9, adapted from (Kim *et al.* 2000)

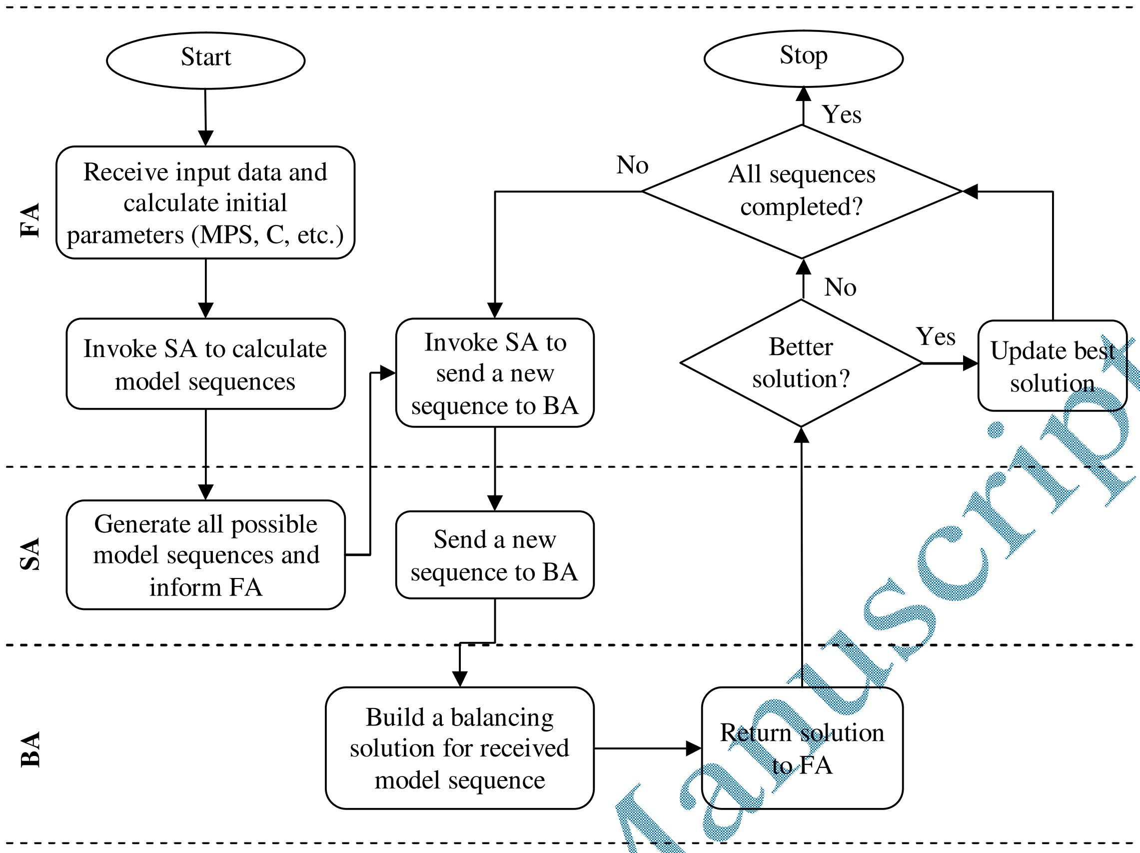


Figure 8. Simultaneous model sequencing - line balancing procedure

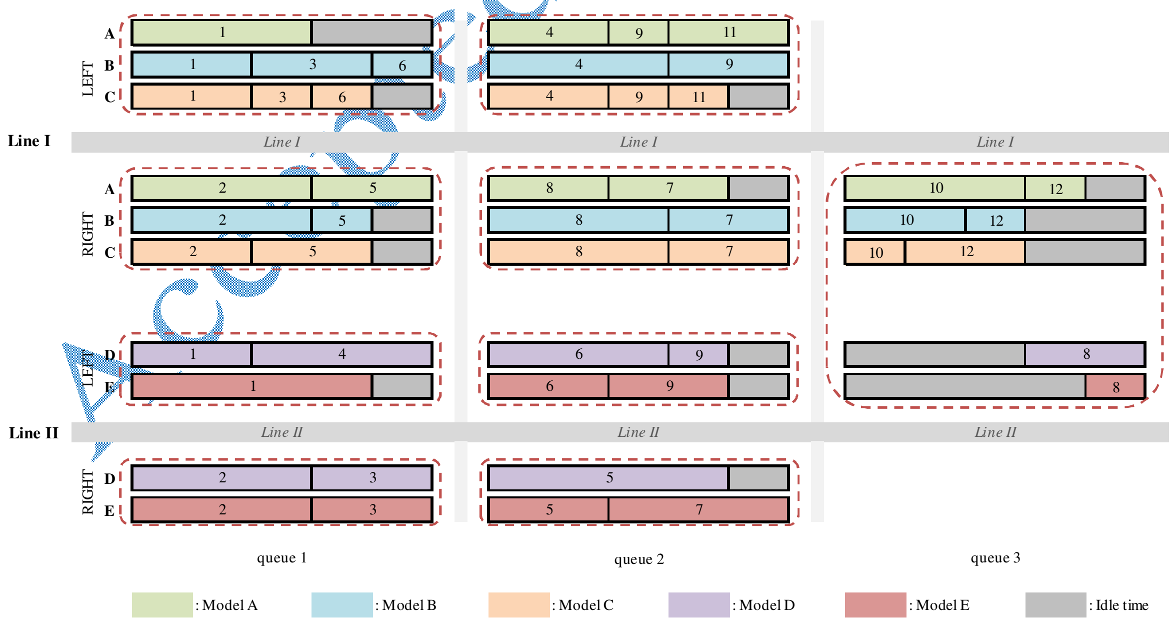
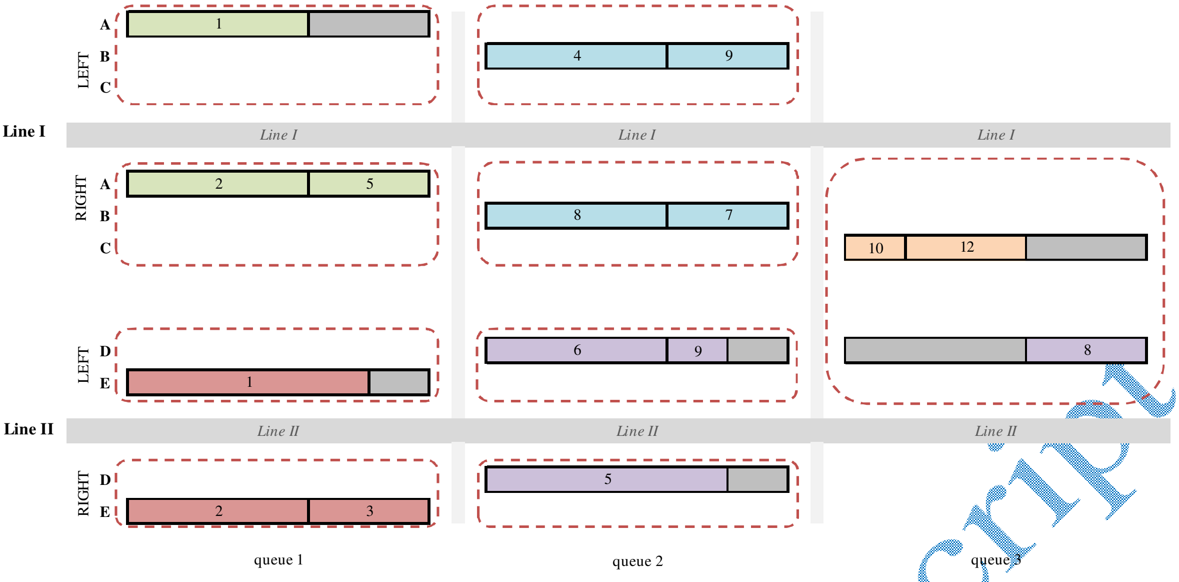
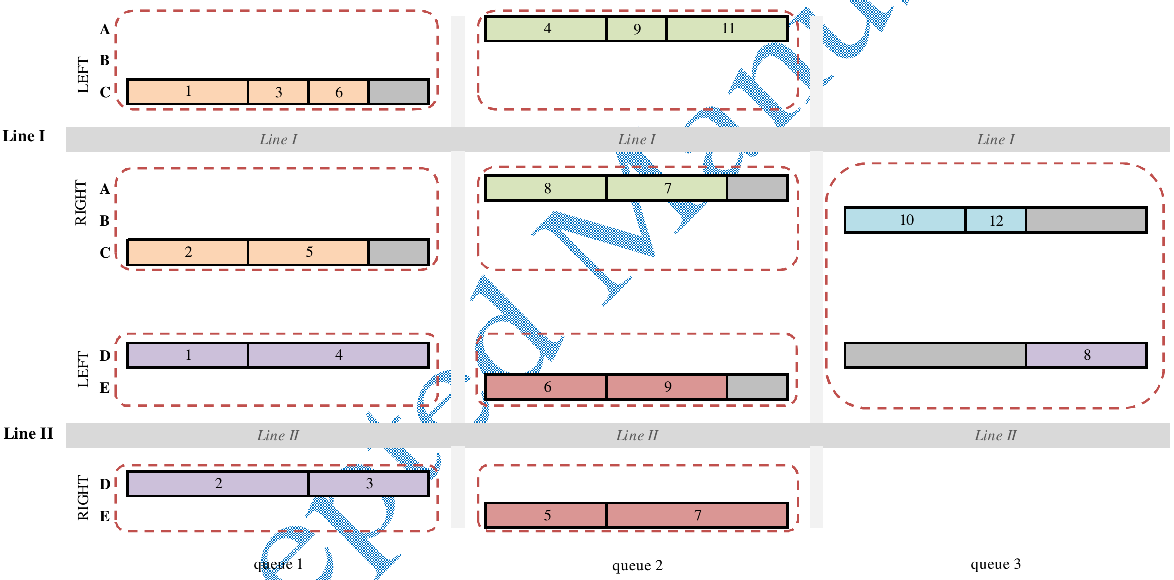


Figure 9. Representation of a balancing solution for given example

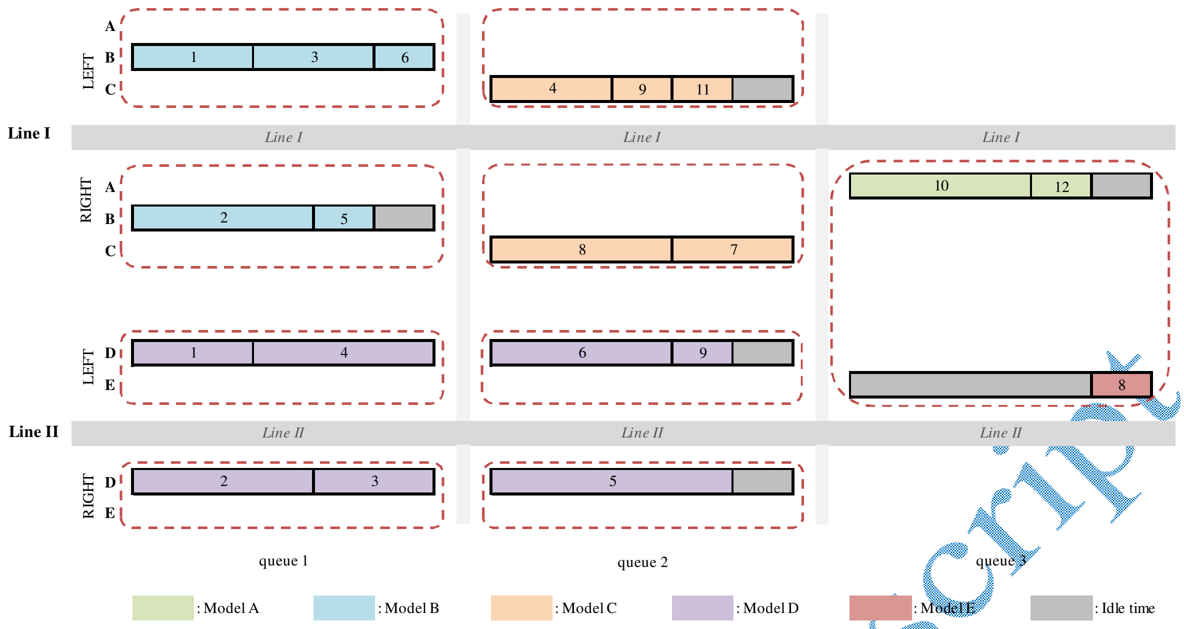


(a)



(b)

Final version available online at: http://www.tandfonline.com/doi/full/10.1080/00207543.2013.879618



(c)

Figure A1. Detailed result of the model sequencing – line balancing procedure for MS_1 = C, B, A and $MS_2 = D, D, E$; (a) production cycle 1, (b) production cycle 2, (c) production cycle 3

Table 1. Summary of the literature on parallel assembly line balancing problems

Research	Method / approach	Obj. (min)				Additional constraints/features
		N	L	C	O	
Suer and Dagli (1994)	Heuristic procedure	●	●			Dynamic number of lines
Suer (1998)	3-phase heuristic with IP and MILP model	●	●			Dynamic number of lines
Gokcen *et al.* (2006)	Heuristic procedures and a mathematical programming model	●				Fixed number of lines
Benzer *et al.* (2007)	A network model	●				Fixed number of lines
Lusa (2008)	Survey					
Baykasoglu *et al.* (2009)	Ant colony optimisation	●				Fixed number of lines
Cercioglu *et al.* (2009)	Simulated annealing based approach	●				Fixed number of lines
Ozcan *et al.* (2009)	Tabu search algorithm	●				Fixed number of lines, workload balance between workstations
Scholl and Boysen (2009)	Binary linear programme and Salome based exact solution procedure	●			●	Fixed lines, product-line assignment considered
Kara *et al.* (2010)	Two goal programming approaches	●		●		Fixed lines, three conflicting goals, task loads of workstations
Ozcan *et al.* (2010a)	Simulated annealing algorithm	●				Fixed lines, mixed-models and model sequencing, workload variance between workstations
Ozcan *et al.* (2010b)	Tabu search algorithm	●				Fixed parallel two-sided lines
Kucukkoc *et al.* (2013b)	Ant colony optimisation	●				Line length, Two-sided lines
Kucukkoc *et al.* (2013c)	Ant colony optimisation algorithm with RPWM	●				Line length, Two-sided lines

N: Number of stations, L: Number of lines, C: Cycle time, O: Number of operators, IP: Integer programming, MILP: Mixed-integer linear programming, RPWM: Ranked positional weight method

Table 2. Detailed summary of the main contributions in the literature on mixed-model assembly line balancing problems (2007-2013)

Research	Line Config.				Parallel Stations	Main Obj. (min)				Additional Const.				Sequencing	Setup times	Methodology	Additional Objectives/Features
	S.Straight	U-Shaped	Parallel L.	Two-Sided		N	C	M	O	Zoning	Positional	Synch. Tasks	Space				
Kara *et al*. (2007)		●							●						●	Simulated annealing, new neighbourhood generation	JIT, WS, constant rate of parts consumption
Battini *et al*. (2007)	Multi-turn circular transfer						●									Heuristic approach	Multi-stations
Simaria and Vilarinho (2009)				●		●			●	●		●				Math. model, ACO	WS
Choi (2009)	●								●							Goal programming model (0-1)	Processing time, PW
Kara and Tekin (2009)		●				●										MIP formulation, COMSOAL based heuristic	Model mixes, operator travel times in crossover stations
Ozcan and Toklu (2009a)				●		●		●		●	●	●				Mathematical model, SA	WS
Emde *et al*. (2010)	●					*Computational evaluation*										*Computational evaluation*	Evaluation of different WS strategies
Ozcan *et al*. (2010a)			●			●			●					●		Simulated annealing	WS
Ozturk *et al*. (2010)	●								●				●	●	●	MIP and Constraint Programming	Minimising the maximum completion time of tasks
Ozcan *et al*. (2011)		●					●							●		Genetic algorithm	Stochastic environment
Xu and Xiao (2011)	●					●										Robust GA	Uncertain times and *changing demands*
Yagmahan (2011)	●					●										Multi-objective ACO	WS
Akpinar and Bayhan (2011)	●				●	●			●	●						Hybrid genetic algorithm	WS
Hamzadayi and Yildiz (2012)		●			●	●			●	●				●		Priority based GA (PGA), SA based fitness eval. app	WS
Rabbani *et al*. (2012)		●		●		●			●							Genetic algorithm	Min number of crossover stations, op.
Chutima and Chimklai (2012)				●		●		●	●							Multi objective PSONK	WR, WS
Liao *et al*. (2012)	●					●			●							Multi agent based framework,	WS
Manavizadeh *et al*. (2012)	●					●	●			●						Multi objective GA	MTO environment
Mosadegh *et al*. (2012)	●								●					●		Simulated annealing	Minimising total utility work, station
Tiacci (2012)	●	●			●	*Simulation*										Object oriented simulation	Stochastic times, buffers
Akpinar *et al*. (2013)	●				●	●				●					●	Hybrid ACO + GA	
Kucukkoc *et al*. (2013)	●				●					●						Hybrid GA	
Manavizadeh *et al*. (2013)		●				●										Simulated annealing	Human Eff, WS, Kanban sys.

N: Number of workstations, C: Cycle time, O: Other special objectives, M: Mated stations, WS: Workload smoothness, WR: Work relatedness, PW: Physical workload, JIT: Just in time, MTO: Make to order, ACO: Ant colony optimisation, GA: Genetic algorithm, PSONK: Particle swarm optimisation with negative knowledge.

Table 3. Possible mixes of product models for given example ($MS_1 = ABBB$ and $MS_2 = CD$)

Station No	1	2	3	4	5	6		7	8	9	10	11
Cycle/Line	$\alpha_{1,1}^{\varphi}$	$\alpha_{1,2}^{\varphi}$	$\alpha_{2,3}^{\varphi}$	$\alpha_{2,4}^{\varphi}$	$\alpha_{1,5}^{\varphi}$	$\alpha_{1,6}^{\varphi}$	$\alpha_{2,6}^{\varphi}$	$\alpha_{2,7}^{\varphi}$	$\alpha_{1,8}^{\varphi}$	$\alpha_{1,9}^{\varphi}$	$\alpha_{2,10}^{\varphi}$	$\alpha_{2,11}^{\varphi}$
1	B	B	C	C	B	<u>***B***</u>	<u>***D***</u>	D	A	A	C	C
2	B	B	D	D	B	<u>***B***</u>	<u>***C***</u>	C	B	B	D	D
3	A	A	C	C	B	B	D	D	B	B	C	C
4	B	B	D	D	A	<u>***A***</u>	<u>***C***</u>	C	B	B	D	D
5	B	B	C	C	B	B	D	D	A	A	C	C
6	B	B	D	D	B	B	C	C	B	B	D	D
7	A	A	C	C	B	B	D	D	B	B	C	C
8	B	B	D	D	A	A	C	C	B	B	D	D

$\alpha_{h,k}^{\varphi}$: The product model that is produced on line h at station k in production cycle φ.

Table 4. Possible mixes of models for another model combination ($MS_1 = BBAB$ and $MS_2 = DC$)

Station No	1	2	3	4	5	6		7	8	9	10	11
Cycle/Line	$\alpha_{1,1}^{\varphi}$	$\alpha_{1,2}^{\varphi}$	$\alpha_{2,3}^{\varphi}$	$\alpha_{2,4}^{\varphi}$	$\alpha_{1,5}^{\varphi}$	$\alpha_{1,6}^{\varphi}$	$\alpha_{2,6}^{\varphi}$	$\alpha_{2,7}^{\varphi}$	$\alpha_{1,8}^{\varphi}$	$\alpha_{1,9}^{\varphi}$	$\alpha_{2,10}^{\varphi}$	$\alpha_{2,11}^{\varphi}$
1	A	A	D	D	B	<u>***B***</u>	<u>***C***</u>	C	B	B	D	D
2	B	B	C	C	A	<u>***A***</u>	<u>***D***</u>	D	B	B	C	C
3	B	B	D	D	B	B	C	C	A	A	D	D
4	B	B	C	C	B	<u>***B***</u>	<u>***D***</u>	D	B	B	C	C
5	A	A	D	D	B	B	C	C	B	B	D	D
6	B	B	C	C	A	A	D	D	B	B	C	C
7	B	B	D	D	B	B	C	C	A	A	D	D
8	B	B	C	C	B	B	D	D	B	B	C	C

$\alpha_{h,k}^{\varphi}$: The product model that produced on line h at station k in production cycle φ.

Table 5. Task times for product models

	Line I (P12)			**Line II (P9)**	
Task No/Model	Model A	Model B	Model C	Model D	Model E
1	3	2	2	2	4
2	3	3	2	3	3
3	0	2	1	2	2
4	2	3	2	3	0
5	2	1	2	4	2
6	0	1	1	3	2
7	2	2	2	0	3
8	2	3	3	2	1
9	1	2	1	1	2
10	3	2	1		
11	2	0	1		
12	1	1	2		

Final version available online at: http://www.tandfonline.com/doi/full/10.1080/00207543.2013.879618