www.elsevier.com/locate/ijpe

Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines

Ibrahim Kucukkoc, David Z. Zhang

PII: S0925-5273(14)00268-0
DOI: http://dx.doi.org/10.1016/j.ijpe.2014.08.010
Reference: PROECO5848

To appear in: *Int. J. Production Economics*

Received date: 27 January 2014
Revised date: 19 June 2014
Accepted date: 12 August 2014

Cite this article as: Ibrahim Kucukkoc, David Z. Zhang, Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines, *Int. J. Production Economics,* http://dx.doi.org/10.1016/j.ijpe.2014.08.010

Mathematical Model and Agent Based Solution Approach for the Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines

Ibrahim KUCUKKOC [a,b,1], David Z. ZHANG [a,c]

[a] College of Engineering, Mathematics and Physical Sciences, North Park Road, University of Exeter, EX4 4QF, Exeter, UK

[b] Department of Industrial Engineering, Faculty of Engineering and Architecture, Balikesir University, Cagis Campus, 10145, Balikesir, Turkey

[c] State Key Laboratory on Mechanical Transmission, Chongqing University, 400044, Chongqing, China

Abstract

One of the key factors of a successfully implemented mixed-model line system is considering model sequencing problem as well as the line balancing problem. In the literature, there are many studies, which consider these two tightly interrelated problems individually. However, we integrate the model sequencing problem in the line balancing procedure to obtain a more efficient solution for the problem of *Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines*. A mathematical model is developed to present the problem and a novel agent based ant colony optimisation approach is proposed as the solution method. Different agents interact with each other to find a near optimal solution for the problem. Each ant selects a random behaviour from a predefined list of heuristics and builds a solution using this behaviour as a local search rule along with the pheromone value. Different cycle times are allowed for different two-sided lines located in parallel to each other and this yields a complex problem where different production cycles need to be considered to build a feasible solution. The performance of the proposed approach is tested through a set of test cases. Experimental results indicate that considering model sequencing problem with the line balancing problem together helps minimise line length and total number of required workstations. Also, it is found that the proposed approach outperforms other three heuristics tested.

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

---

[1] Corresponding author. E-mail address: i.kucukkoc@exeter.ac.uk (I. Kucukkoc), Tel: +44(0)1392 723613
Address: College of Engineering, Mathematics and Physical Sciences, North Park Road, University of Exeter, EX4 4QF, Exeter, UK. Permanent E-mail: ikucukkoc@balikesir.edu.tr
Email address of the Second Author (D. Z. Zhang): d.z.zhang@exeter.ac.uk, Tel: +44(0)1392 723641

Abstract

One of the key factors of a successfully implemented mixed-model line system is considering model sequencing problem as well as the line balancing problem. In the literature, there are many studies, which consider these two tightly interrelated problems individually. However, we integrate the model sequencing problem in the line balancing procedure to obtain a more efficient solution for the problem of *Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines*. A mathematical model is developed to present the problem and a novel agent based ant colony optimisation approach is proposed as the solution method. Different agents interact with each other to find a near optimal solution for the problem. Each ant selects a random behaviour from a predefined list of heuristics and builds a solution using this behaviour as a local search rule along with the pheromone value. Different cycle times are allowed for different two-sided lines located in parallel to each other and this yields a complex problem where different production cycles need to be considered to build a feasible solution. The performance of the proposed approach is tested through a set of test cases. Experimental results indicate that considering model sequencing problem with the line balancing problem together helps minimise line length and total number of required workstations. Also, it is found that the proposed approach outperforms other three heuristics tested.

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

1. Introduction

Production systems and product requirements have significantly evolved since the time of famous Model-T of Henry Ford. Assembly lines, which were initially developed for the cost efficient production of a single commodity, have changed to mixed-model lines to enable highly diversified and customized products to be manufactured. Different models of a product can be produced in an intermixed sequence on a mixed-model assembly line, so that a substantial reduction in setup times and cost can be obtained with the utilisation of operators and workspace flexibly (Boysen *et al.*, 2009).

Assembly lines are flow oriented production systems divided into sequentially arranged *(work)stations* connected by a mechanical transportation mechanism, such as a moving belt or conveyor. Each station is allowed a fixed time span, called *cycle time*, to perform assigned tasks on the product unit launched down the line. The set of tasks assigned to a station constitutes its *workload* (Otto *et al.*, 2013; Sternatz, 2013). The wide spread use of mixed-model assembly lines can be attributed to the increased variety of customised product portfolio and compatibility of mixed-model lines with mass customisation in a Just-in-Time environment. So, accurately managed mixed-model assembly lines can help manufacturers balance workloads and minimise delay (Ding *et al.*, 2006), because the assignment of tasks to workstations determines the productivity of the entire manufacturing system (Sternatz, 2013).

To obtain a successfully implemented mixed-model assembly line, both model sequencing and line balancing problems must be treated together, since these two problems are tightly interrelated to each other. The line balancing problem is the problem of assigning tasks to workstations by considering certain constraints (i.e. *precedence relationships*, *capacity constraints*, etc.) and is differentiated by sought objectives and considered constraints (Morrison *et al.*, 2014). The model sequencing problem determines the production sequence of different product models assembled on the same line. The performance of an obtained line balance is affected by the sequence of produced models while the optimality of the model sequencing process depends on the results of line balancing (Kim *et al.*, 2006). However, these two problems were dealt with separately by many researchers (i.e. Askin and Zhou (1997), Gokcen and Erel

(1997), Vilarinho and Simaria (2002), McMullen and Tarasewich (2003), Haq *et al.* (2006), Ozcan and Toklu (2009), Hamta *et al.* (2013), Kucukkoc *et al.* (2013), and Manavizadeh *et al.* (2013a) for the line balancing problem; and Yano and Rachamadugu (1991), Bard *et al.* (1992), Kim *et al.* (1996), Zheng *et al.* (2011), Bautista and Cano (2011), and Xu and Li (2013) for the model sequencing problem) with different objectives ever since the mixed-model line balancing problem was first introduced by Thomopoulos (1967).

The rest of the paper is organised as follows. Following a comprehensive review of the literature in Section 2, Section 3 describes the MPTALB/S problem and presents the developed mathematical model. The solution procedure of the ABACO/S approach is depicted in Section 4. Section 5 and 6 give a numerical example and the results of computational experiments with comparisons. Finally, Section 7 gives conclusion and possible future research directions.

2. Literature Review

In the literature, few researches dealt with the two problems at the same time. A summary of the literature on simultaneous balancing and sequencing of mixed-model lines is given in Table 1. The solution approaches in those researches can be divided into two groups: (i) hierarchical solution approaches, and (ii) simultaneous solution approaches. Hierarchical approaches, which solve one problem first and then the other under the constraint of the first solution, were employed by Thomopoulos (1967), Dar-El and Nadivi (1981), Merengo *et al.* (1999), and Rekiek *et al.* (2000) (Kim *et al.*, 2006).

Regarding the simultaneous solution approaches, few studies have been carried out. Kim *et al.* (2006) proposed an endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines and demonstrated that hierarchical approaches cannot explore the solution space effectively. Kara *et al.* (2007b) addressed simultaneous balancing and sequencing problem in mixed-model U-shaped lines as contrary stations utilised on both back and front of the line are seriously affected by the model sequences on the line. That is why a task may require different processing time for different models and this issue may yield an unbalanced workload in contrary stations. They also proposed a simulated annealing approach to solve the problem with multiple conflicting objectives. Then, Ozcan *et al.* (2011) and Hamzadayi and Yildiz (2012) proposed different genetic algorithm based approaches for balancing and sequencing of U-shaped lines simultaneously; by considering stochastic times, and parallel workstations - zoning constraints, respectively.

Table 1. Summary of the literature on simultaneous balancing and sequencing of mixed-model lines

Research	Problem	Method/Approach	Performance Measure(s)										
			NS	**C**	**LE**	**WLS**	**RI**	**WIP**	**PUR**	**TUW**	**CoS**	**BC**	**LBT**
Merengo *et al*. (1999)	B/S manual MALs	3-phase methodology & simulation	●				●	●	●				
Kim *et al*. (2000a)	B/S MAL	Co-evolutionary algorithm								●			
Kim *et al*. (2000b)	B/S U-shaped MALs	Co-evolutionary algorithm			●								
Karabati and Sayin (2003)	B/S MAL	New mathematical model + heuristic		●									
Kim *et al*. (2006)	B/S U-shaped MAL	Endosymbiotic evolutionary algorithm				●							
Kara *et al*. (2007a)	B/S JIT U-shaped MAL	Simulated annealing	●			●							
Kara *et al*. (2007b)	B/S JIT U-shaped MAL	Simulated annealing				●			●		●		
Battini *et al*. (2009)	B/S MALs with finite buffer capacity	Branch and bound based step-by-step procedure			●							●	
Boysen *et al*. (2009)	MMS, car sequencing, level scheduling	Survey											
Hwang and Katayama (2010)	B/S straight and U-shaped MALs	Multi-objective evolutionary algorithm	●			●							
Ozcan *et al*. (2010a)	B/S parallel MALs	Simulated annealing			●	●							
Ozcan *et al*. (2011)	B/S stochastic U-shaped MAL	Genetic algorithm		●									
Mosadegh *et al*. (2012)	B/S MALs	Evolutionary strategies algorithm								●			●
Hamzadayi and Yildiz (2012)	B/S U-shaped MALs	Genetic algorithm based approach	●			●							
Kucukkoc and Zhang (2014b)	B/S of mixed-model parallel two-sided	Agent based ACO enhanced with ten	●		●								

assembly lines	heuristics & model sequencing agent

B/S: Balancing and sequencing, MAL: Mixed-model assembly line, MMS: Mixed-model sequencing, JIT: Just in Time, NS: Number of stations, C: Cycle time, LE: Line efficiency (minimising idle time), WLS: Workload smoothness (=absolute deviations of workloads across workstations), RI: Rate of incomplete jobs, WIP: Work in process, PUR: Part usage rate, TUW: Total utility work, CoS: Cost of setups, BC: Buffer capacity, LBT: Last best time.

Ozcan *et al.* (2010a) introduced balancing and sequencing of parallel mixed-model assembly lines and developed a simulated annealing based solution approach to the problem. Mosadegh *et al.* (2012) developed a mixed-integer linear programming model and a simulated annealing algorithm to provide exact and heuristic solutions of the problem with station-dependent assembly times in traditional mixed-model lines.

On the other hand, there is plenty of research on balancing of mixed-model lines in the context of different types of line configurations. Battaïa and Dolgui (2013) presented a taxonomy of those line balancing problems and their solution approaches. However, studies on the combination of mixed-model lines with different layouts or configurations (i.e. two-sided lines, parallel lines, U-shaped lines, etc.) are rather new as well as scarce. Two-sided lines are usually established to produce large-sized items, and both sides of the line are used to assemble the product on the line. In parallel lines, two or more lines are located in parallel to each other and operators assigned to interval stations may operate on both adjacent lines to increase productivity. Moreover, parallel two-sided lines are commonly used to produce large-sized items in many different fields; such as vehicle manufacturing industry (e.g. buses, tracks, automobiles, etc.). To combine the advantages of both parallel lines and two sided lines, Ozcan *et al.* (2010b) and Kucukkoc and Zhang (2013) dealt with *parallel two-sided assembly line balancing problem* where a single model is produced on each line.

Simaria and Vilarinho (2009), Ozcan and Toklu (2009) and Chutima and Chimklai (2012) dealt with *mixed-model two-sided line balancing problem* and developed different approaches as solution methods. Sequencing of models are not considered in those studies carried out by Simaria and Vilarinho (2009), Ozcan and Toklu (2009), and Chutima and Chimklai (2012) since there is no opportunity to utilise contrary stations (as in U-lines) or multi-line stations (as in parallel lines) and model mixes are not important unless a setup operation is required when changing from one model to another. However, Rabbani *et al.* (2012) proposed multiple U-shaped layout to deal with mixed-model two-sided assembly line balancing problem and utilised contrary workstations. Nevertheless they did not consider sequencing of the models in their study. Also, Ozcan *et al.* (2010a) addressed *balancing and sequencing of parallel mixed-model assembly lines* to increase the flexibility of parallel lines (as already mentioned above) but their study did not consider operation side variability.

The only three studies performed recently, which incorporates mixed-model lines, parallel lines, and two-sided lines, belong to Zhang and Kucukkoc (2013) and Kucukkoc and Zhang (2014a; 2014b). The *Mixed-model Parallel Two-sided Assembly Line Balancing Problem* (MPTALBP) was introduced by Zhang and Kucukkoc (2013). They defined the problem and argued that model sequencing issue is an accompanying problem to line balancing problem. But simultaneous model sequencing and line balancing problem in mixed-model parallel two-sided lines, as emphasized in Zhang and Kucukkoc (2013), was first dealt with by Kucukkoc and Zhang (2014b). They introduced the *Mixed-model Parallel Two-sided Assembly Line Balancing and Sequencing (MPTALB/S) problem* and proposed a framework of Agent Based Ant Colony Optimisation (ABACO/S) algorithm for the solution of the problem. However, the efficiency of the proposed approach was not tested and the benefits of considering model sequencing and line balancing problems together have not been examined quantitatively. Kucukkoc and Zhang (2014a) employed the framework proposed by Kucukkoc and Zhang (2014b) with some modifications to solve the MPTALBP. They also compared the performance of their method against other heuristics but their research did not consider the model sequencing problem along with the line balancing problem.

The research presented in the current work is a continuation of Kucukkoc and Zhang (2014a; 2014b). The enhanced algorithm in this research by extending the ABACO/S developed by Kucukkoc and Zhang (2014b) will also be called ABACO/S (where "S" refers to inclusion of model sequencing problem) hereafter. We explain the running principle of ABACO/S and test its performance against other heuristics through test cases. The performance of the ABACO/S is shown by comparisons and the advantages, which could be achieved in case of considering model sequencing and line balancing problems together, are examined quantitatively. Moreover, MPTALBS problem is modelled mathematically, which is only used as a means to formally describe the problem. This mathematical modelling is also the first such attempts in the literature.

3. Problem Definition

The MPTALB/S problem was recently introduced by Kucukkoc and Zhang (2014b). With the increasingly global, dynamic and customer driven structure of the world market (Zhang and Sharifi, 2007), Mixed-model Parallel Two-sided Assembly Lines (MPTALs) gained more importance and attention by academics and practitioners. As different from parallel two-sided lines (Ozcan *et al.*, 2010b), MPTALs provide the capability and advantage of responding to changing market demands. With parallel two sided assembly lines only one model is allowed to be assembled on each parallel two-sided line, whereas with mixed-model parallel two sided assembly lines more than one model of a product can be assembled at the same time on each of the MPTALs. Also, MPTALs incorporates the flexibility of multi-line workstations, different from mixed-model two-sided lines. Thus, an operator allocated in a multi-line station can perform pre-described operations on opposite sides of two adjacent lines.

The main aim of the MPTALB/S problem is (i) assigning tasks to workstations in such a way that certain constraints (such as precedence relationships caused by technological or organisational requirements, and capacity constraints, etc.) are satisfied and (ii) sequencing product models on the lines to optimize (a) pre-determined performance measure(s) or an objective function.

The idea of assembling similar large sized product models of a product family on parallel lines carries the combined practical advantages of model variations and multi-line stations. These advantages include but not limited to Kucukkoc and Zhang (2014b):

- Flexibility of producing different models with different throughput rates,
- Shorter line length than traditional lines,
- Sharing some common tools between stations,
- Reduced material handling cost and operator movement,
- Improved line efficiency/reduced operator requirement,
- Increased motivation of operators due to operation enrichment at multi-line stations between two adjacent lines,
- Improved technical and communication skills of operators.

In a MPTAL system, more than one different product model, symbolised with m_{hj} ($j = 1,\ldots,M_h$), is produced on each two-sided assembly line, where lines are symbolised with L_h ($h = 1,\ldots,H$). Each product model has its own set of tasks, t_{hji} ($i = 1,\ldots,T_{hj}$), that need to be performed according to predefined precedence relationships caused by some technological or organisational constraints. The set of predecessors of task t_{hji} for model m_{hj} on line L_h is represented by P_{hji}. Processing time of a task (pt_{hji}) may differ from one model to another and each line consists of a series of successional stations represented by W_{hkx} ($k = 1,\ldots,K_h; x = 0,1$); where x is a binary variable and “0” and “1” symbolise left side and right side of the line, respectively (Kucukkoc and Zhang, 2014b).

Precedence relationship constraints and capacity constraint are common in line balancing problems and need to be satisfied for the MPTALB/S problem as well. However, it is known that a special attention must be paid during balancing procedure in two-sided lines due to interference phenomenon. This issue was explained by Kucukkoc and Zhang (2014a).

The facility of multi-line station is one of the basic advantages of MPTALs. An illustration of multi-line station is depicted in Fig. 1. Operators assigned to multi-line stations can perform jobs on opposite sides of both two adjacent lines. Therefore, idle times and total number of required operators are minimised by utilising multi-line stations. In the figure, the operator allocated to multi-line workstation between two adjacent lines in queue 2 works on both right side of the Line I and left side of the Line II (Kucukkoc and Zhang, 2014b).

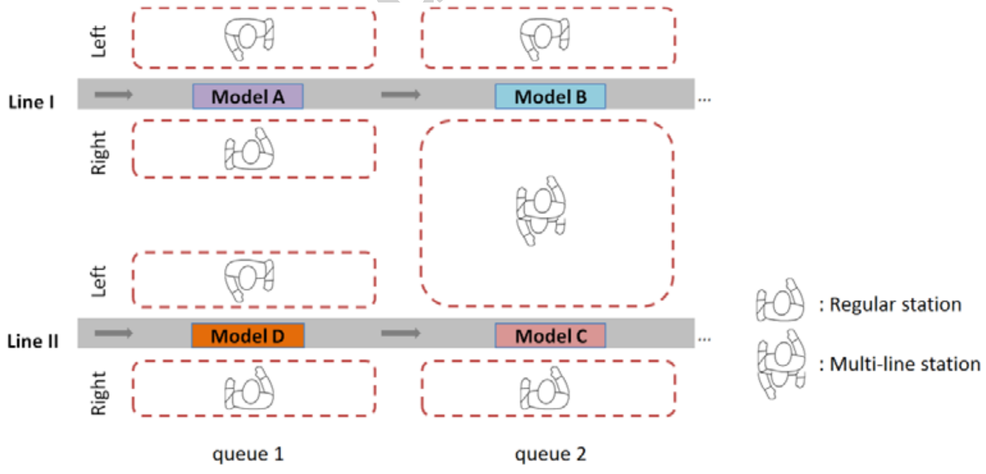


Fig. 1. Representation of typical mixed-model parallel two-sided assembly lines (Kucukkoc and Zhang, 2014b).

Another advantage of establishing MPTALs is different throughput rates of the lines. In other words, the cycle time (C_h) of each line may be different from each other. C_h is calculated according to demand over the planning horizon for each line.

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

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

When different cycle times are subject to balancing and sequencing procedure, a common cycle time should be used to tackle the complex task assignment procedure affected by model changes in each production cycle. For this aim, Least Common Multiple (LCM) of cycle times (Gökçen *et al.*, 2006) is adopted as common cycle time and task times are normalised according to the ratio of original cycle time to common cycle time. This will be exemplified in the numerical example section but please refer to Gokcen *et al.* (2006) and Kucukkoc and Zhang (2014b) for detailed explanation. The following sub-section describes the MPTALB/S problem formally. Normalised task times are provided as input to the model to keep the complexity of the model at minimum.

3.1 Mathematical Model

In the MPTALB/S problem, the model sequences of the lines are particularly important to determine the available times of the operators that are allocated to multi-line stations. This is due to different task times of models. Therefore, the problem stated above is formulated as a mixed integer programming model that also takes into account model sequences with the objectives of minimising number of utilised workstations, minimising length of the lines, and maximising workload smoothness.

3.1.1 Notation

The notation used in this study is given below. Please note that some of the notations/parameters used in this study are similar with the study of Ozcan *et al.* (2010a) and same with the study of Kucukkoc and Zhang (2014b) to provide a coherent context to the readers.

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

m_{hj} : The j^{th} product model on line L_h $(j=1,\ldots,\ M_h)$, 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 $(i=1,\ldots,T_{hj})$, 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 $(k=1,\ldots,K_h;\ x=0,1)$, where K_h is total number of workstations on line L_h,

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

φ : Production cycle $(\varphi=1,\ldots,\phi)$, where $\phi = LCM\left(S_1,\ldots,S_H\right)$; (the definition of S_h is given below).

3.1.1.1 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,

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,

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,

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

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

C : Common cycle time for all lines,

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

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

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,

PZ_{hj} : Set of pairs of tasks that must be assigned to the same workstation for model m_{hj} on line L_h (positive zoning),

NZ_{hj} : Set of pairs of tasks that must be assigned to different workstations for model m_{hj} on line L_h (negative zoning),

DL_{hj} : Set of left direction tasks for model m_{hj} on line L_h,

DR_{hj} : Set of right direction tasks for model m_{hj} on line L_h.

3.1.1.2 Decision Variables

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

,

$$\tau_{hjq}^{\varphi} = \begin{cases} 1 & \text{if model } m_{hj} \text{ is produced in queue } q \text{ on line } L_h \text{ in production cycle } \varphi \\ 0 & \text{otherwise} \end{cases},$$

st_{hji}^{φ} : Starting time of task t_{hji} of model m_{hj} on line L_h in production cycle φ,

$$U_{hkx}^{\varphi} = \begin{cases} 1 & \text{if station } W_{hkx} \text{ is utilised on side } x \text{ of line } L_h \text{ in production cycle } \varphi \\ 0 & \text{otherwise} \end{cases},$$

3.1.1.3 Intermediate Variables

q_{hkx} : Queue number in which station W_{hkx} is utilised,

LL : Maximum number of queues $(LL = \frac{\sum_{h=1}^{H} K_h}{2H})$,

$$LS_{hkx} = \begin{cases} 0 & \text{if station } W_{hkx} \text{ is utilised on left side of the first line in production cycle } \varphi \\ 1 & \text{otherwise} \end{cases},$$

K_{hq}^{φ} : Set of workstations (mated stations) that located in queue q on line L_h in production cycle φ,

σ : Variable $(\sigma = h+1, \ldots, H-1)$,

β : Variable $(\beta = 0,1)$,

$$c = \begin{cases} 1 & \text{if } x = 1 \text{ and } \beta = 1 \\ 0 & \text{otherwise} \end{cases},$$

$$\mu = \begin{cases} 1 & \text{if} (\sigma - h) = 1 \text{ and } \beta = 1 \\ 0 & \text{otherwise} \end{cases},$$

$$R_{hjrv}^{\varphi} = \begin{cases} 1 & \text{if tasks } r \text{ and } v \text{ for model } m_{hj} \text{ on line } L_h \text{ are assigned in the same queue in cycle } \varphi \\ 0 & \text{otherwise} \end{cases}$$

To consider the model sequences integrated with the line balancing problem, the Minimum Part Set (MPS) principle (Bard *et al.*, 1992) is used (Ozcan *et al.*, 2010a). According to this approach the MPS on line L_h is (MPS_h) calculated by dividing total demands of models by the greatest common divisor of these demands. Let the greatest common divisor of D_{hj} $(j=1,\ldots,M_h)$ be represented by cd_h $(h=1,\ldots,H)$. The vector $d_h=(d_{h1},\ldots,d_{hM_h})$, where $(h=1,\ldots,H)$, denotes the model mix of line L_h (Kucukkoc and Zhang, 2014b).

$$d_{hj}=\frac{D_{hj}}{cd_h},\qquad j=1,\ldots,M_h;h=1,\ldots,H\ . \tag{2}$$

MS_h represents the model sequence of line L_h which is independent from other model sequences. 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:

$$S_h=\sum_{j=1}^{M_h}d_{hj}. \tag{3}$$

The number of different model show-ups for a determined model sequence pattern on the lines depends on the lengths of the determined model sequences. This also regulates how many different production cycles $(\varphi=1,\ldots,\phi)$ the system should be split into. The maximum number of model show-ups (production cycles), $(\phi=MS_{max})$, which may appear at a cycle can be calculated as follows:

$$MS_{max}=LCM\left(S_1,\ldots,S_H\right),\qquad h=1,\ldots,H\ . \tag{4}$$

Total number of possible sequences for a mixed-model assembly line is computed using the equation: $TS_h=\left(\sum_{j=1}^{M_h}d_{hj}\right)! \Big/ \prod_{j=1}^{M_h}\left(d_{hj}!\right)$ (Manavizadeh *et al.*, 2013b).

When two mixed-model lines are taken into account, the number of sequences emerging for the system could be computed by multiplying total number of sequences belong to the lines $(TS_1\times TS_2)$.

An example of these calculations will be provided with a numerical example in Section 5.

3.1.2 Objective Function

In the literature, a large number of studies on parallel assembly line balancing problems and two-sided assembly line balancing problems consider minimisation of total number of required workstations solely as the main objective while only a few studies consider the minimisation of the line length (or number of mated workstations) as an additional objective. However, line length should also be considered in MPTALB/S problem, since different configurations of the lines are possible with the same number of workstations due to the nature of the parallel two-sided assembly lines. Utilisation of a multi-line station will affect the objective function as equally as a regular station since only one operator is allocated to those workstations.

Workload smoothness is another criterion that shows whether the lines are well balanced especially to make a distinction between two different solutions that needs the same number of workstations (as mentioned by Ozcan *et al.* (2010b)). Therefore, a new objective function is developed which considers weighted idle times of the stations (*WIT)*, which also means to minimise total number of utilised workstations, workload smoothness (*WS*), and line length (*LL*). The objective function used in this research is given in Equations 5-8.

$$Min\, Z = \gamma_1 WIT + \gamma_2 WS + \gamma_3 LL\ . \tag{5}$$

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

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

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

The main objective of the model is to minimise *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. *LL* 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 maker to decide the significance levels of the objectives.

3.1.3 Constraints

Model Occurrence Constraint:

Only up to one model m_{hj} can be produced in each queue (q), on each line (L_h), in each production cycle (φ) at a time. In other words, total number of models that produced in a queue (q) on line (L_h) in each production cycle (φ) at a time is lower than or equal to 1.

$$\sum_{j \in M_h} \tau_{hjq}^{\varphi} \le 1, \quad \forall \varphi \in \phi; \forall h \in H; \forall q \in LL . \tag{9}$$

Task Occurrence Constraint:

In a production cycles (φ), each task (t_{hji}) belonging to each model (m_{hj}) can be assigned at most once to all queues (q), sides (x), and stations (W_{hkx}).

$$\sum_{q \in LL} \sum_{x \in \{0,1\}} \sum_{k \in K_h} \tau_{hjq}^{\varphi} Y_{hjikx}^{\varphi} \le 1, \quad \forall i \in T_{hj}; \forall j \in M_h; \forall h \in H; \forall \varphi \in \phi . \tag{10}$$

Task Assignment Constraint for Demand Satisfaction:

Each task must be assigned exactly d_{hj} times in all production cycles. In other words, each task (t_{hji}) for each model (m_{hj}) must be assigned exactly d_{hj} times; in all production cycles (φ), queues (q), sides (x), and stations (W_{hkx}). It is ensured that all tasks are assigned to a station exactly once.

$$\sum_{\varphi \in \phi} \sum_{q \in LL} \sum_{x \in \{0,1\}} \sum_{k \in K_{hq}^{\varphi}} \tau_{hjq}^{\varphi} Y_{hjikx}^{\varphi} = d_{hj}, \quad \forall i \in T_{hj}; \forall j \in M_h; \forall h \in H . \tag{11}$$

Operation Direction Constraints:

A left side task $(t_{hji} \in DL_{hj})$ for each model (m_{hj}) on line $\left(L_h\right)$ must be assigned to left side stations $(x = 0)$ exactly d_{hj} times; in all production cycles (φ), queues (q), and stations (W_{hkx}).

$$\sum_{\varphi\in\phi}\sum_{q\in LL}\sum_{k\in K_h}\tau^{\varphi}_{hjq}\,Y^{\varphi}_{hjik0} = d_{hj}, \quad \forall t_{hji} \in DL_{hj}; \forall j \in M_h; \forall h \in H \; . \qquad (12a)$$

A right side task $(t_{hji} \in DR_{hj})$ for each model (m_{hj}) on line $\left(L_h\right)$ must be assigned to right side stations $(x = 1)$ exactly d_{hj} times; in all production cycles (φ), queues (q), and stations (W_{hkx}).

$$\sum_{\varphi\in\phi}\sum_{q\in LL}\sum_{k\in K_h}\tau^{\varphi}_{hjq}\,Y^{\varphi}_{hjik1} = d_{hj}, \quad \forall t_{hji} \in DR_{hj}; \forall j \in M_h; \forall h \in H \; . \qquad (12b)$$

Precedence Relationships Constraints:

Precedence relationships constraints ensure that the precedence relationships are not violated on the line L_h precedence diagram and completion time of tasks are considered to avoid interference. These constraints must be considered for each predecessor of task $(r \in P_{hjv})$, where $v \in T_{hj}$, in each production cycle (φ) on each line (L_h), for each model (m_{hj}).

Two different situations may occur during the balancing procedure: (i) tasks r and v are assigned to different queues in a cycle and (ii) tasks r and v are assigned to the same queue in the same cycle.

Following equation is active if tasks r and v are assigned to different queues in a production cycle (φ) (on each line, L_h, for each model, m_{hj}, for each predecessor of v):

$$\sum_{x\in\{0,1\}}\sum_{k\in K_h} q_{hkx}\left(Y^{\varphi}_{hjrkx} - Y^{\varphi}_{hjvkx}\right) \le 0, \quad \forall r \in P_{hjv}; \forall \varphi \in \phi; \forall h \in H; \forall j \in M_h. \; (13a)$$

Following equation is active if tasks r and v are assigned to the same queue in the same production cycle (φ) (on each line, L_h, for each model, m_{hj}, for each predecessor of v):

$$R_{hjrv}^{\varphi}\left(st_{hjr}^{\varphi}+pt_{hjr}-st_{hjv}^{\varphi}\right)\leq 0, \quad \forall r\in P_{hjv};\forall\varphi\in\phi;\forall h\in H;\forall j\in M_h. \quad (13b)$$

Capacity Constraint for Regular Stations:

Capacity constraint ensures that total workload of a workstation does not exceed pre-determined cycle time. In other words, capacity constraint assures each task is executed within the cycle time.

$$Y_{hjikx}^{\varphi}\left(st_{hji}^{\varphi}+pt_{hji}\right)\leq C, \quad \forall\varphi\in\phi;\forall h\in H;\forall x\in\{0,1\};\forall k\in K_h;\forall i\in T_{hj};$$
$$\forall j\in M_h. \qquad (14)$$

Capacity Constraint for Multi-line Stations:

If some tasks are assigned to a right side station of line L_h from left side station of its adjacent line, total workload of this multi-line station cannot exceed its capacity.

$$\sum_{j\in M_h}\sum_{i\in T_{hj}}pt_{hji}\,Y_{hjikx}^{\varphi}+LS_{hkx}\left(\sum_{j\in M_h}\sum_{i\in T_{hj}}pt_{(h+1)ji}\,Y_{(h+1)jik(x-1)}^{\varphi}\right)\leq CU_{hkx}^{\varphi}\ ,$$
$$\forall\varphi\in\phi;\forall h=1,\ldots,H-1;\forall k\in K_h;\forall x\in\{0,1\}\ . \qquad (15)$$

Assigning to Multi-line Stations Constraints:

This constraint defines whether any task is assigned to workstation W_{hkx} from its adjacent line.

$$\sum_{i\in T_{hj}}Y_{hjikx}^{\varphi}-T_{hj}U_{hkx}^{\varphi}\leq 0, \quad \forall\varphi\in\phi;\forall h\in H;\forall j\in M_h;\forall k\in K_h;\forall x\in\{0,1\}\ . \ (16)$$

Valid Zone Constraints for Multi-Line Stations:

A multi-line station can only perform tasks from its adjacent line and side. Following constraints ensure that an operator working at station W_{hkx} can only perform task(s) additionally from only one adjacent line and side; unless station W_{hkx} is not utilised on left side of the first line or on right side of the last line. For example, if an operator is located on right side of the first line $(L_h = 1, x = 1)$, that operator can perform additional tasks from only left side of the second line $(L_h = 2, x = 0)$ along with his/her main job. The operator cannot perform any job from left side of the first line $(L_h = 1, x = 0)$, or right side of the second line $(L_h = 2, x = 1)$, since it is not possible a direct communication with those tasks assigned to these stations.

Following constraint controls utilising multi-line station for the lines "$h = 1, \ldots, H-1$".

$$(1-x)\left(U^{\varphi}_{hk\beta} + U^{\varphi}_{(\sigma-\mu)kx}\right) + x\left(U^{\varphi}_{hkc} + U^{\varphi}_{\sigma kx}\right) = 1, \quad \forall \varphi \in \phi; \forall k = 1, \ldots, K_h;$$

$$\forall h = 1, \ldots, H-1; \forall x \in \{0,1\}; \forall \sigma = h+1, \ldots, H; \forall \beta \in \{0,1\}. \qquad (17a)$$

Following constraint restricts utilising multi-line station for right side of the last line $(h = H)$.

$$U^{\varphi}_{Hk0} + U^{\varphi}_{Hk1} = 1, \quad \forall \varphi \in \phi; \forall k \in K_h. \qquad (17b)$$

Zoning Constraints:

Some tasks may need to be processed in the same workstation for some specific reasons that may originate from work environment or tool requirements (positive zoning constraint). In that case, this constraint ensures that those tasks are assigned to the same workstation. PZ_{hj} is the set of pairs of tasks that must be assigned to the same workstation for model m_{hj} on line L_h.

$$\sum_{x \in \{0,1\}} \sum_{k \in K_h} k\left(Y^{\varphi}_{hjakx} - Y^{\varphi}_{hjbkx}\right) = 0, \quad \forall (a,b) \in PZ_{hj}; \forall \varphi \in \phi; \forall h \in H; \forall j \in M_h. \ (18a)$$

Some tasks must be performed in different workstations due to safety rules or processing obligations (negative zoning constraint). In that case, this constraint ensures that those tasks are assigned to different workstations. NZ_{hj} is the set of pairs of tasks that must be assigned to different workstations for model m_{hj} on line L_h.

$$\sum_{x\in\{0,1\}}\sum_{k\in K_h} k\left(Y^{\varphi}_{hjakx}-Y^{\varphi}_{hjbkx}\right)\neq 0, \quad \forall(a,b)\in NZ_{hj};\forall\varphi\in\phi;\forall h\in H;\forall j\in M_h. \quad (18b)$$

Variable Constraints:

Decision variable and indicator variable constraints are as follows:

$$Y^{\varphi}_{hjikx}\in\{0,1\}, \quad \forall\varphi\in\phi;\ \forall h\in H;\ \forall j\in M_h;\ \forall i\in T_{hj};\ \forall k\in K_h;\ \forall x\in\{0,1\}. \quad (19)$$

$$U^{\varphi}_{hkx}\in\{0,1\}, \quad \forall\varphi\in\phi;\ \forall h\in H;\ \forall k\in K_h;\ \forall x\in\{0,1\}. \quad (20)$$

$$\tau^{\varphi}_{hjq}\in\{0,1\}, \quad \forall\varphi\in\phi;\ \forall h\in H;\ \forall j\in M_h;\ \forall q\in LL. \quad (21)$$

$$st^{\varphi}_{hji}\geq 0, \quad \forall\varphi\in\phi;\ \forall h\in H;\ \forall j\in M_h;\ \forall i\in T_{hj}. \quad (22)$$

$$LS_{hkx}\in\{0,1\}, \quad \forall h\in H;\ \forall k\in K_h;\ \forall x\in\{0,1\}. \quad (23)$$

$$q_{hkx}\geq 0, \quad \forall h\in H;\ \forall k\in K_h;\ \forall x\in\{0,1\}. \quad (24)$$

$$R^{\varphi}_{hjrv}\in\{0,1\}, \quad \forall\varphi\in\phi;\ \forall r\in P_{hjv}. \quad (25)$$

$$\sigma = h+1,\ldots,H-1. \quad (26)$$

$$\beta, c, \mu\in\{0,1\}. \quad (27)$$

$$\mathrm{LL}>0. \quad (28)$$

3.2 Assumptions

The assumptions of the study are as follows (Kucukkoc and Zhang, 2014b):

- Two or more similar product models are assembled on each of the two or more parallel two-sided assembly lines.

- A common precedence diagram is used for the product models produced on the same line and it is known. By this way, common tasks between similar models on the same line are not allowed to be assigned to different stations and resource utilisation is maximised.
- Processing times of tasks for each product model are known and deterministic; while some tasks may have different processing times for different models, or some tasks may not be required for some product models (task time equals to zero).
- Deterministic and pre-determined demands are considered for product models assembled on the lines.
- Tasks can be assigned to only a predetermined operation side (L or R) or either (E) side.
- Tasks cannot be split to more than one workstation, so each task for each product model must be assigned to exactly one workstation.
- Sum of all the task times assigned to a workstation constitutes its workload time, and workload time of a station cannot exceed the cycle time.
- For a task to be assigned, all of its predecessors must have been assigned and completed.
- Operators are multi-skilled and can work at any workstation on any side of a line.
- The maximum number of operators that can be allocated to a workstation equals to one. Thus, parallel workstations are not allowed.
- No work in process inventory is allowed.

- Operator travel times are ignored and starting and finishing times are the same for all lines.

4. Solution Method

As shown by Wee and Magazine (1982), simple assembly line balancing problem is an NP-Hard class combinatorial problem. Since many complex characteristics of line configurations are involved in MPTALB/S problem along with the model sequencing problem, it is clear that MPTALB/S problem is NP-Hard as well. In the literature, researchers usually utilise heuristic, meta-heuristic, and other approaches to find approximate solutions for such complex problems. Recently, agent based solution techniques have become popular in this context. For example, Anussornnitisarn *et al.* (2005), Mes *et al.* (2007), Anosike and Zhang (2009), Bearzotti *et al.* (2012), Amini *et al.* (2012), and He *et al.* (2014) developed agent based techniques to solve problems in modelling, management, and optimisation of manufacturing and transportation processes. In the agent based systems, a network of problem solvers collaborate with each other to find solutions for problems that are beyond their individual capabilities (Goh and Zhang, 2003).

This section describes the developed ABACO/S approach to solve the MPTALB/S problem. We start explaining the ABACO/S from the most outer level and then continue with the basic programming components level by level.

Kucukkoc and Zhang (2014a) used the framework proposed by Kucukkoc and Zhang (2014b) to solve the MPTALB/S problem. However, in the algorithm developed by Kucukkoc and Zhang (2014a), the model sequencing problem, i.e. model variation in each new production cycle, was ignored. Instead, a balancing solution was sought which would be feasible for any sequence or combination of the models. For this aim, processing time of a task is assumed to be the maximum time among the models on the same line and the lines are balanced using maximum task times like single model lines.

There is no doubt that such an approach is faster in terms of processing time but yields weak solutions as a generalised solution independent from the launched sequence and model combinations in production cycles is obtained.

Different from the studies of Kucukkoc and Zhang (2014a), this research takes into account the launched model sequences and model combinations in each production cycle. Therefore, line balancing problem is dealt with according to which model is assembled on the workstation at a particular time. Moreover, the algorithm has a capability of two different model sequencing procedures; (i) combinatorial model sequencing, and (ii) random model sequencing (this will be explained later).

ABACO/S consists of four-level agents: Facilitator Agent (FA), Planning Agent (PA), Sequencing Agent (SA), and Balancing Agent (BA). These agents are programme scripts interact with each other to solve the problem collectively. The outline and multi-agent architecture of the ABACO/S are displayed in Fig. 2 and Fig. 3, respectively.

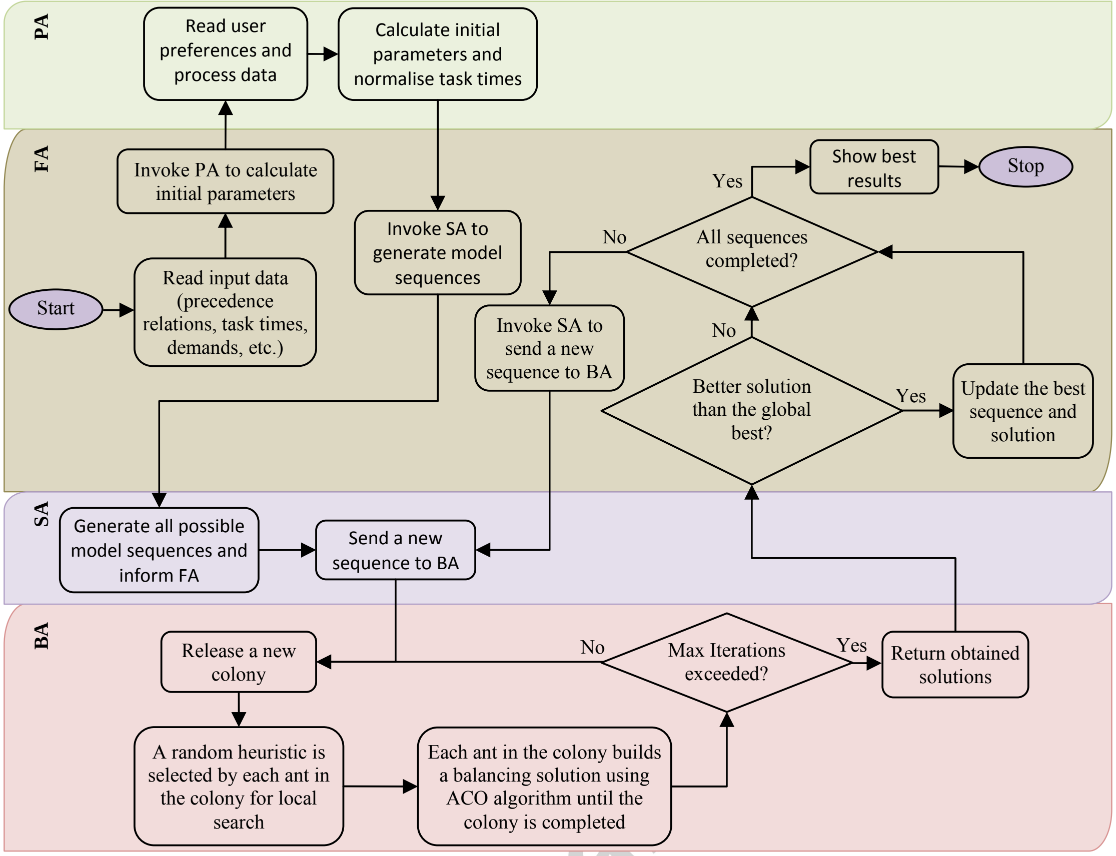
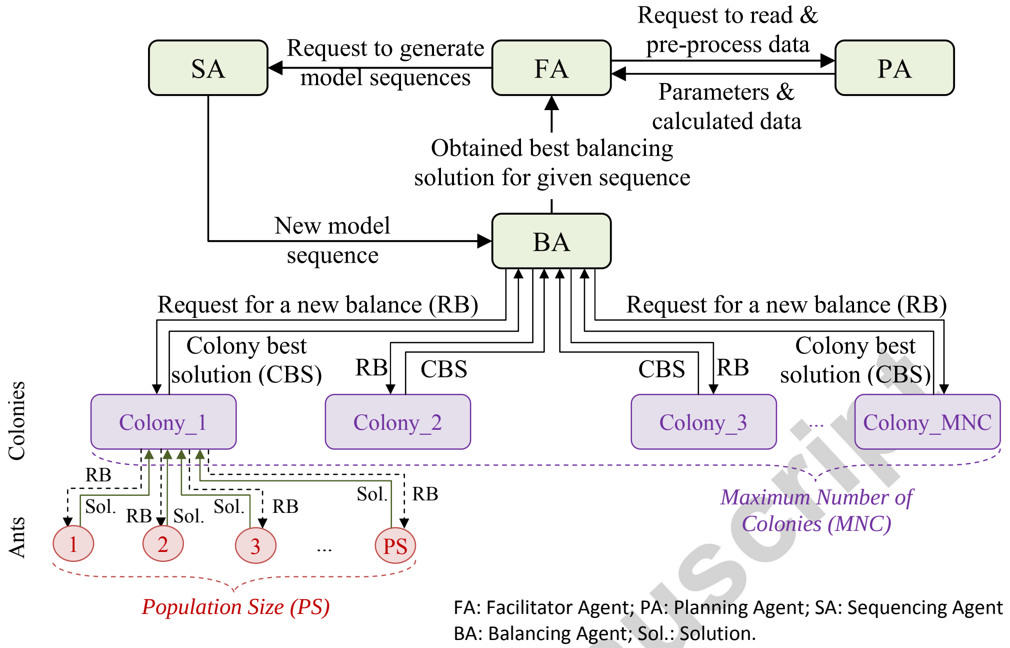


Fig. 2. The outline of the ABACO/S algorithm

BA employs a colony of ants to find solutions and ants in the colony build solutions until a maximum number of iterations is exceeded. Results are returned to FA and the global best solution is updated if a better solution is found than the current best. BA is sent another model sequence and another colony find solutions for the launched model sequence. This cycle continues until a pre-defined number of model sequences are tried. When this is achieved, the programme shows the best model sequence and the balancing solution found.

It should be mentioned that the algorithm has the capability of using combinatorial model sequencing and random model sequencing procedures. If the user prefers combinatorial model sequencing, the algorithm tries all possible model sequences generated by the SA. If random model sequencing is preferred, the algorithm tries random model sequences from possible model sequences generated by the SA until a user defined number of trials are achieved. Additionally, it is also supported to find a balancing solution for a given model sequence by the user.



FA: Facilitator Agent; PA: Planning Agent; SA: Sequencing Agent
BA: Balancing Agent; Sol.: Solution.

Fig. 3. The multi-agent architecture of the ABACO/S algorithm

The procedures of the ant colony optimisation are exhibited in Fig. 4. Each ant in the colony comes up with a solution and the performance measures are evaluated according to the quality of the obtained solution. Then, an amount of pheromone is laid on the edges of the path drawn (between task and workstation) according to the performance measures. If a solution is better than the best solution in the colony, double amount of pheromone is laid on the edges of the solution to make the path favourable to be selected by other ants. A constant amount of pheromone is evaporated from all edges and the cycle continues until the colony is completed.

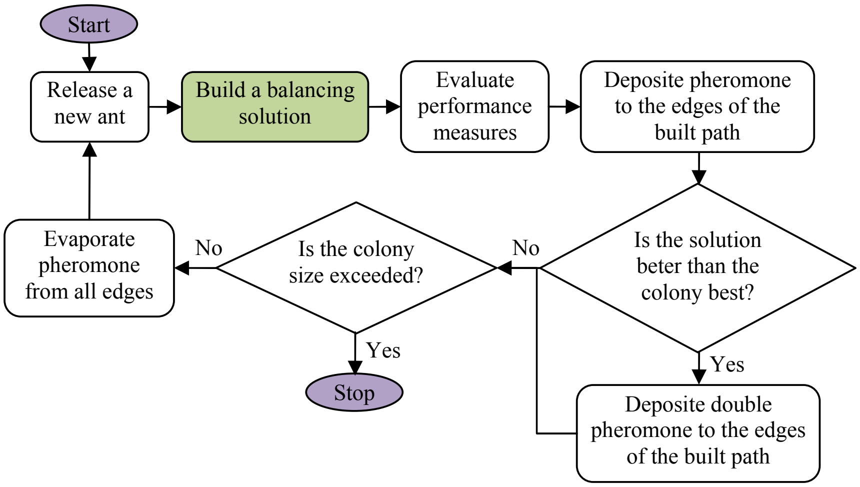


Fig. 4. Procedures of the ant colony optimisation

The pheromone update rule is (Kucukkoc and Zhang, 2014a):

$$\tau_{ik} \leftarrow (1-\rho)\tau_{ik} + \Delta\tau_{ik}, \tag{29}$$

where τ_{ik} and ρ represent the amount of virtual pheromone between task – workstation and evaporation rate, respectively; $\Delta\tau_{ik} = Q / Performance\,measure$, and Q is a user determined parameter that effects the amount of pheromone deposited.

The ant colony optimisation is enhanced with ten different heuristics, most of them are commonly used in the literature, to find better solutions for the defined complex problem as in Kucukkoc and Zhang (2014a). These heuristics are:

- COMSOAL (Arcus, 1966),
- Ranked Positional Weight Method – RPWM (Helgeson and Birnie, 1961),
- Reverse Ranked Positional Weight Method – RRPWM (produced from RPWM),
- Longest Processing Time – LPT (Talbot and Patterson, 1984),
- Shortest Processing Time – SPT (Baykasoglu, 2006),
- Smallest Task Number – STN (Arcus, 1963)

- Maximum Number of Predecessors – MNP (produced from Maximum Number of Immediate Predecessors technique proposed by Baykasoglu (2006)),
- Least Number of Predecessors – LNP, (produced from Maximum Number of Immediate Predecessors technique proposed by Baykasoglu (2006)),
- Maximum Number of Successors – MNS (produced from Maximum Number of Immediate Successors technique proposed by Tonge (1960)),
- Least Number of Successors – LNS (produced from Maximum Number of Immediate Successors technique proposed by Tonge (1960)).

The topology of the proposed ant colony, and the procedure of building a balancing solution are illustrated in Fig. 5 and Fig. 6. To provide a lean representation, only the first layer of paths and possible choices after task 1 are illustrated in the Fig. 5.

Each ant in the colony builds a balancing solution according to the given procedure in Fig. 6 (where $st(k)$ and $st(\underset{�}{k})$ mean station time of the current station and its mated station, respectively). Each ant starts from a random line and side and forwards by assigning tasks from the available tasks list to the current position. The selection probability of a task by an ant is calculated using the following equation (Kucukkoc and Zhang, 2014a):

$$p_{ik} = \frac{\left[\tau_{ik}\right]^{\alpha}\left[\eta_i\right]^{\beta}}{\sum_{y \epsilon Z_i}\left[\tau_{iy}\right]^{\alpha}\left[\eta_i\right]^{\beta}}\ , \qquad (30)$$

where i, k, and Z_i indicate task, current workstation, and list of candidate tasks when task i is selected, respectively. τ_{ik} and η_i are the amount of virtual pheromone between task – workstation, and the heuristic information of task i that comes from the randomly selected heuristic by each ant. This probability is calculated by each ant every time when a new task will be selected, and tasks have higher probability will most likely be selected and assigned by ants.

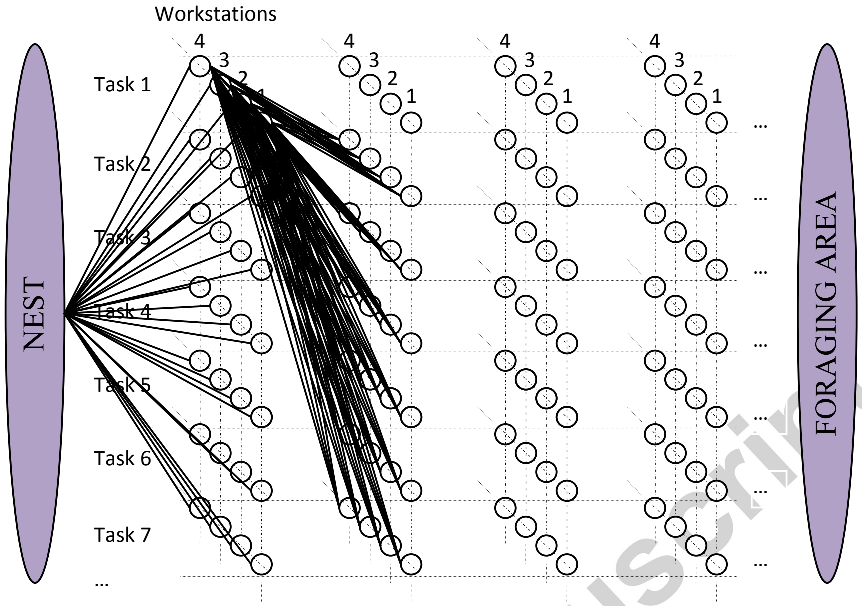


Fig. 5. The topology of the proposed ant colony optimisation

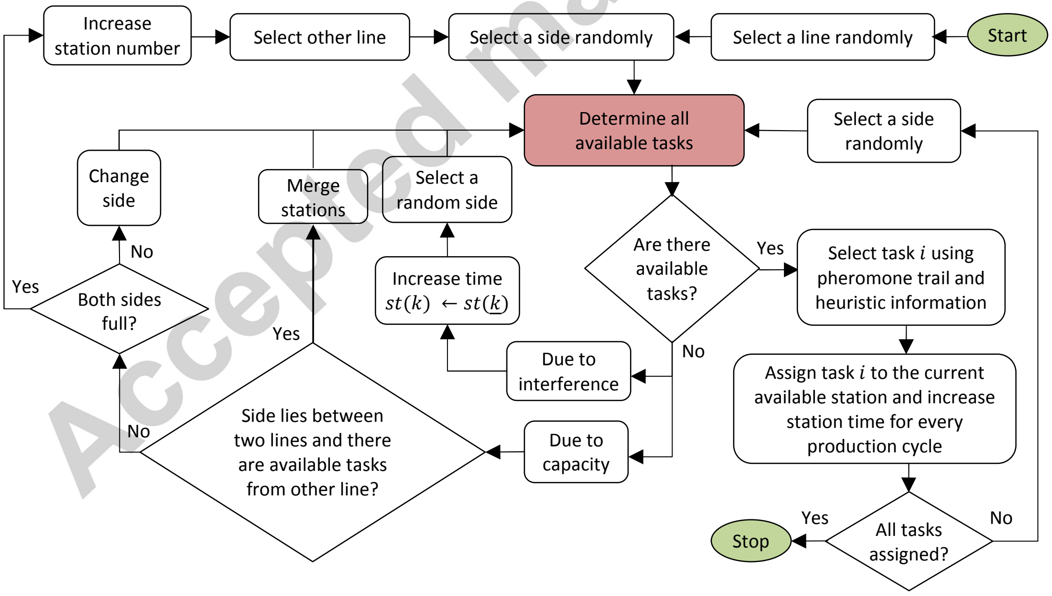


Fig. 6. Building a balancing solution procedure, adapted from (Kucukkoc and Zhang, 2014b)

To increase the possibility of obtaining well balanced solutions ants can change their sides and position at any time. But, it is not allowed to forward to another line or queue without filling the capacity of current mated stations as long as there are available tasks.

Also, it is allowed to assign tasks from the contrary side of the adjacent line if the station lies between two lines. If there is not any available task (caused by the inadequate capacity, interference, etc.), a solution is sought depending on the reason as seen in the figure.

The flowchart of determining available tasks is given in Fig. 7. This process plays a critical role in the overall balancing and sequencing system, because the solution that will be obtained at the end of the balancing and sequencing procedure must be feasible in terms of different model sequences, which change at every production cycle. That is why workloads of workstations and earliest starting times of tasks (caused by precedence relationships) must be recorded for every single production cycle. This data is used when determining whether a task is available or not, and processing time of a candidate task is considered according to the actual model at the relevant cycle. Therefore, processing time of a task for the relevant model must be equal to or less than the remaining capacity at every production cycle. Also, earliest starting times of tasks must be considered carefully as they may differ from one cycle to another caused by the tasks' processing time differences depending on the assembled model on the line.

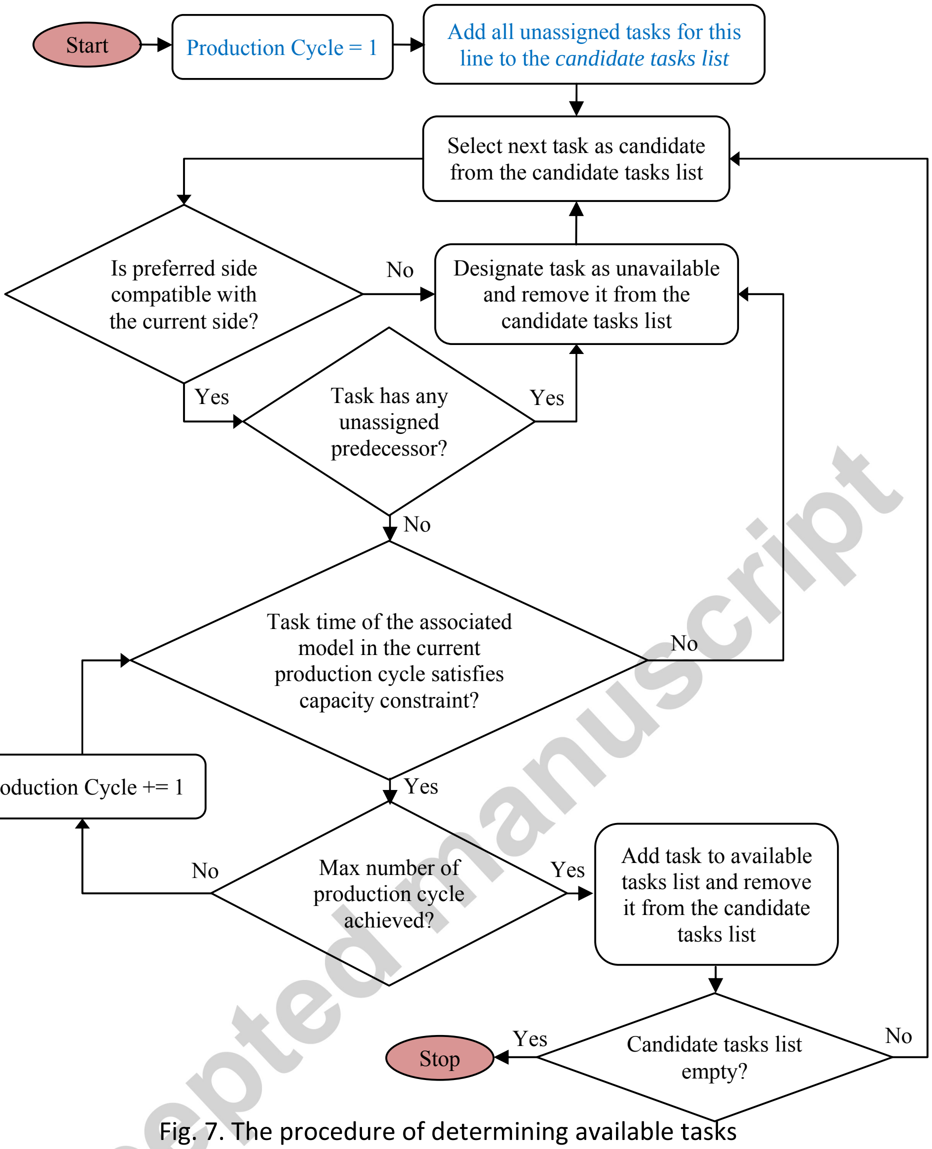


Fig. 7. The procedure of determining available tasks

5. Illustrative Example

This section briefly explains the simultaneous balancing and model sequencing procedure through a numerical example. Common precedence relationship diagrams are used between different models on the same line. Therefore, two precedence relationship diagrams, P12 and P16 are taken from Kim *et al.* (2000c) and Lee *et al.* (2001), respectively. Task times are generated randomly between zero and ten as given in Table 2 with the original preferred operation directions (where L means Left, R means Right, and E means Either side) and immediate predecessor tasks. If a task time equals to “0” time units, it means that this task is not required to be performed for this product model.

Table 2. Task times and relevant data for the numerical example

Line	Line I (P12)					Line II (P16)				
Model / Task	Model A	Model B	Model C	Side	Immediate Predecessors	Model D	Model E	Model F	Side	Immediate Predecessors
1	4	2	6	L	-	5	7	4	E	-
2	8	10	7	R	-	0	4	0	E	-
3	3	5	3	E	-	5	10	7	L	1
4	0	2	4	L	1	4	8	2	E	1
5	3	1	2	E	2	3	4	8	R	2
6	1	6	0	L	3	1	2	3	L	3
7	2	0	2	E	4, 5	7	1	6	E	4, 5
8	5	6	6	R	5	4	4	5	E	6, 7
9	4	4	2	E	5, 6	2	2	1	R	7
10	2	5	0	E	7, 8	3	3	4	R	7
11	2	9	5	E	9	5	7	4	E	8
12	3	2	1	R	11	1	6	5	L	9
13	-	-	-	-	-	4	4	6	E	9, 10
14	-	-	-	-	-	5	2	3	E	11
15	-	-	-	-	-	0	4	1	E	11, 12
16	-	-	-	-	-	5	3	5	E	13

Models A, B, and C are assembled on Line I while models D, E, and F are assembled on Line II. If demands are assumed as $D_{1A} = 8$, $D_{1B} = 8$, $D_{1C} = 16$, $D_{2D} = 8$, $D_{2E} = 8$ and $D_{2F} = 8$ for a planning horizon of 480 time units; cycle times are calculated as $C_1 = 15$ and $C_2 = 20$ time units; and minimum part sets are calculated as $MPS_1 = (1, 1, 2)$ and $MPS_2 = (1, 1, 1)$ for Line I and Line II, respectively. This means that $LCM\left(S_1, S_2\right) = LCM\left(4, 3\right) = 12$ different production cycles are subject to consideration for each model sequence.

The number of possible model sequences for Line I is ${}^{4!}\!/_{(1!\times 1!\times 2!)} = 12$ and for Line II is ${}^{3!}\!/_{(1!\times 1!\times 1!)} = 6$. Thus, $12 \times 6 = 72$ different combinations of model sequences must be tried in case of combinatorial sequencing is selected. If a random model sequencing is selected by the user, assuming it is 18, 18 different trials will be done out of 72 different combinations.

As the cycle times of the lines are different, the LCM based approach (Gökçen *et al.*, 2006; Ozcan *et al.*, 2010b) addressed in Section 3 is used. Line divisors (ld_h) are obtained as $ld_1 = LCM\left(C_1, C_2\right)/C_1 = 4$ and $ld_2 = LCM\left(C_1, C_2\right)/C_2 = 3$. Then, task times of models on Line I and Line II are multiplied by ld_1 and ld_2, respectively, and $LCM\left(C_1, C_2\right)$ is accepted as the common cycle time (C). These normalised task times and common cycle time are used while balancing the lines.

To provide more compact and easily understandable results, following objective function is used when solving the given example problem and test cases (in the Section 6):

$$Min\, Z = \gamma_1 LL + \gamma_2 NS \ , \tag{31}$$

where LL and NS indicate the *length of the lines* and the *number of utilised workstations*, respectively.

The algorithm was run for twenty random model sequences, and twenty iterations for each model sequence with ten ants in a colony for the given example. Agent behaviours and interactions between the agents are briefly shown in Fig. 8 through the multi-agent architecture of the proposed method. This sample illustration is made for only the first colony of the first model sequence $(CCBA-FDE)$ as it is not possible to show all of the steps of the solution procedure on a single figure. Facilitator agent invokes the SA until twenty model sequences are completed and solution with the best performance measure is designated as the solution of the problem at the end of this process.

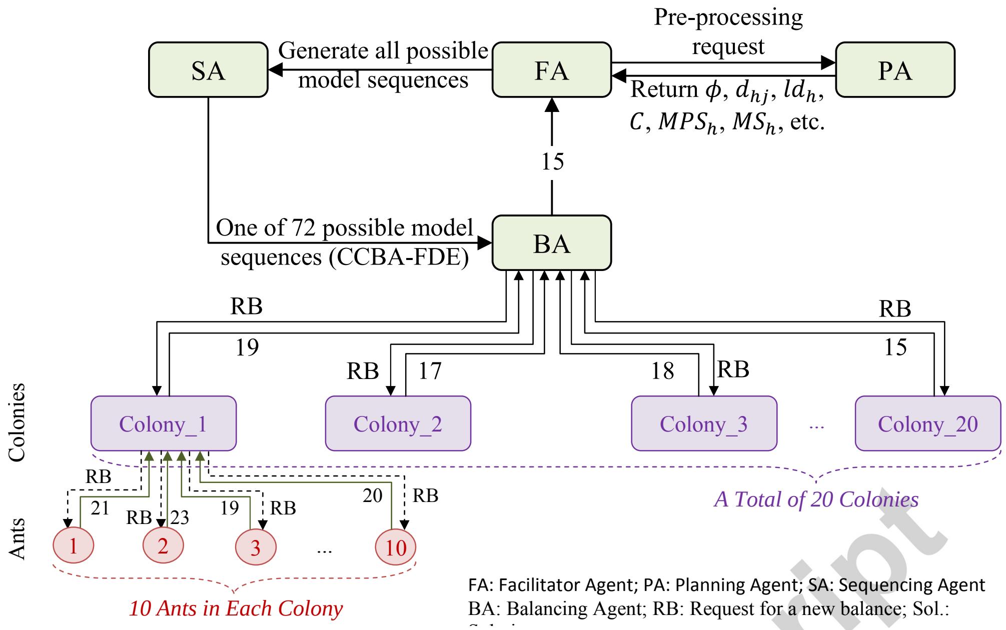


Fig. 8. Interactions between agents on the multi-agent architecture of the proposed method

Task assignment order of the algorithm is shown for the best found solution (with model sequence of Line I: $MS_1 = (ACCB)$ – Line II: $MS_2 = (EFD)$ on the lines in Fig. 9.

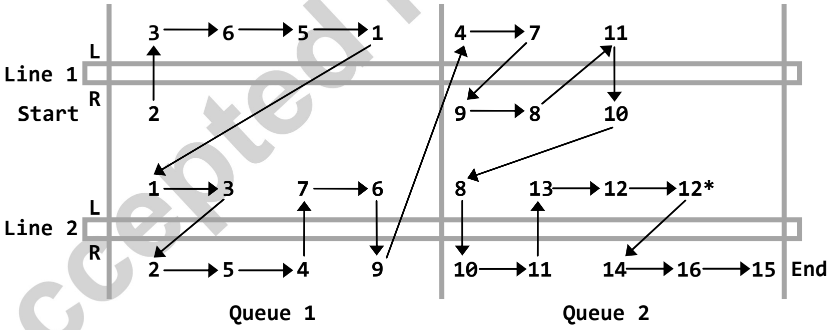


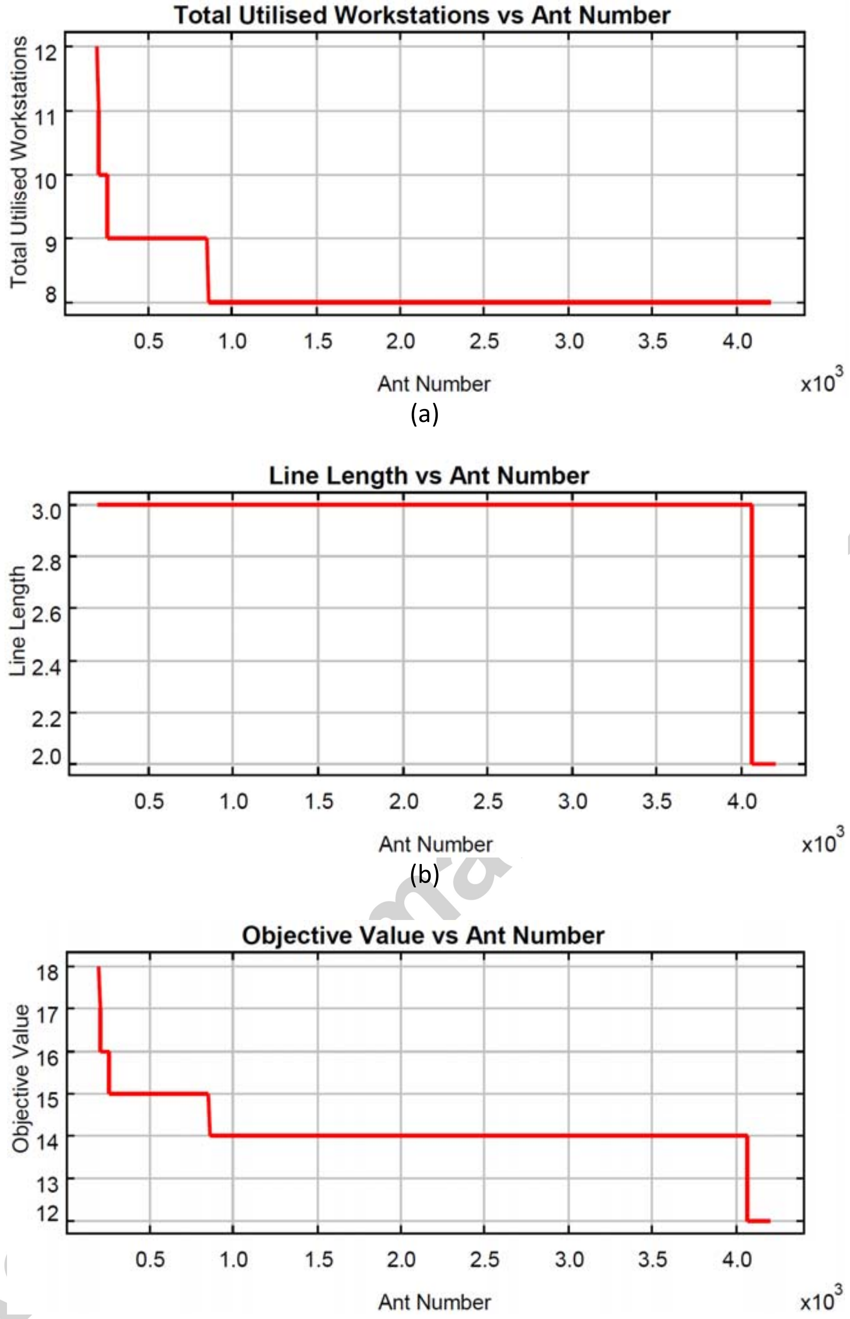
Fig. 9. Assignment order of tasks for the given example

Arrows symbolise order of assigned tasks for the example problem. An ant starts assigning tasks from a randomly selected line and side (Line 1 Side R in this example), and selects a task to assign from the available tasks list to the current position. Then, the ant stays in the current side or changes side (randomly) and selects a task to assign from the updated available tasks list. If the capacity is full or there is not any available task to assign for the current line, line is changed and tasks are assigned to the new workstations on the new line using a similar procedure. As could be seen from the figure, task 12 (highlighted with asterisk*) belonging to the models assembled on Line I is assigned to a multi-line station on Line II in the given sample solution of the problem. The obtained different solution values for different model sequences of the example problem are also given in Table 3 according to sequence. As could be seen from the table, the algorithm finds 15 as the objective value for the majority of the sequences, such as CCBA-FDE, CACB-FED, ACBC-DFE, etc. An objective value of 14 is found for the sequence CCAB-FDE and finally 12 is found for ACCB-EFD. It is obvious that better solutions could be investigated if the algorithm was run for more than twenty model sequences. However, it was thought as enough as this is just for an illustration.

Table 3. Obtained solutions with different model sequences for the given example

#	Model Sequences		Best Solution			Average Obj.	Maximum Obj.
	Line I	Line II	LL	NS	Obj.		
1	CCBA	FDE	3	9	15	17.85	23
2	CACB	FED	3	9	15	17.76	22
3	ACBC	DFE	3	9	15	17.62	22
4	CCAB	FDE	3	8	14	17.75	22
5	CABC	FED	3	9	15	18.00	22
6	ABCC	EFD	3	9	15	17.72	22
7	CABC	DFE	3	9	15	17.63	23
8	CBCA	FED	3	9	15	17.76	22
9	CBCA	FDE	3	9	15	17.73	24
10	BACC	DFE	3	9	15	17.75	22
11	BACC	FDE	3	9	15	17.72	23
12	ABCC	EDF	3	9	15	17.89	22
13	CACB	DEF	3	9	15	17.78	22
14	CBAC	FDE	3	9	15	17.73	23
15	CBAC	EDF	3	9	15	17.81	23
16	CBCA	DEF	3	9	15	17.73	23
17	CCAB	EFD	3	9	15	17.92	23
18	CABC	EFD	3	9	15	17.87	23
19	CBCA	EDF	3	9	15	17.82	22
20	**ACCB**	**EFD**	**2**	**8**	**12**	**17.63**	**23**

Fig. 10 depicts the convergence of the performance measures, i.e. total utilised workstations, line length and the objective value, for the given example (the user defined parameters are assumed as $\gamma_1 = 2$, and $\gamma_2 = 1$).



(a)

(b)

(c)

Fig. 10. The convergence of the performance measures for the given example

As could be seen from Fig. 10a, the number of total utilised workstations reduces from 12 to 9 dramatically with the fourth model sequence (CCAB-FDE) given in Table 3, and reduces to 8 consequently. However, the line length remains the same (3 units) until the last and the best model sequence (ACCB-EFD) among the obtained solutions when it reduces to 2 (Fig. 10b). The curve of the objective function is mainly affected by the number of total utilised workstations till the last model sequence and reaches the minimum with the reduction in the length of the line (Fig. 10c). These graphs exhibit the effect of model sequencing on the quality of the obtained line balance, once again.

6. Computational Results and Discussion

Test cases are solved using the developed ABACO/S algorithm to evaluate the performance of the proposed approach. As there is no ready test cases in the literature, problems are generated from the commonly used test problems combined by Kucukkoc and Zhang (2014a). Model demands and planning horizons are generated in accordance with the original cycle times in Kucukkoc and Zhang (2014a) except test cases 7 and 11, which are highlighted using (*) in the table. In these cases, the algorithm requires a huge amount of memory and time because of the exponentially increased complexity of large model sequences. Therefore, suitable data is generated and solved for these test cases as given in Table 4.

Three heuristics (COMSOAL, Ranked Positional Weight Method - RPWM, and Maximum Number of Successors - MNS), which are commonly used in the literature, are also developed to solve the same test cases with ABACO/S since no comparable result is available in the literature. These heuristics are designed here to use the same balancing/sequencing procedures with ABACO/S. The only difference is that, each heuristic applies its own rule to select and assign tasks to the workstations.

Table 4. Data for test cases

#Test Case	**Problem**		**Cycle Time**		**Demands L1**			**Demands L2**		
	Line 1	*Line 2*	*Line 1*	*Line 2*	A	B	C	D	E	F
1	P9	P9	4	7	40	20	10	20	10	10
2			6	5	20	20	10	15	30	15
3	P9	P12	5	8	40	20	20	20	20	10
4			7	6	15	15	30	20	10	40
5	P12	P12	4	5	20	10	20	10	20	10
6			6	5	20	10	20	30	15	15
7*	P12	P16	9	12	10	20	10	10	10	10

8			10	12	20	20	20	10	20	20
9	P16	P16	12	15	10	20	20	20	10	10
10			16	14	10	40	20	40	20	20
11*	P16	P24	14	16	40	20	20	40	20	10
12			16	18	15	45	30	20	40	20
13	P24	P24	15	20	20	10	10	10	10	10
14			25	20	10	20	10	10	20	20
15	P65	P65	300	480	40	20	20	20	10	20
16			420	360	15	15	30	20	40	10
17	P65	P148	405	810	10	5	5	4	4	2
18			675	540	20	10	10	10	20	20
19	P148	P148	255	510	5	10	5	2	4	4
20			425	340	20	10	10	20	20	10

A 3.1 GHz Intel Core^{TM} i5-2400 CPU computer is used to run the ABACO/S and three heuristics coded in JAVA^{TM} SE 7u4 environment. The parameters of ABACO/S are chosen through a set of experimental tests for a high quality solution and may differ from one problem to another. This is why the search space grows exponentially and the complexity increases with the increasing number of tasks. Twenty random model combinations were tried for the heuristics and the best solution was taken after the algorithm was run 20 times for the test cases 1-10; and 30 times for each model sequence for the test cases 11-20. For the ABACO/S, initial pheromone level, colony size, number of iterations, and number of random model sequences tried were increased by the increasing problem size (see Table 5).

Table 5. Parameters of the ABACO/S

#Test Case	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\rho}$	$\boldsymbol{Q}$	**Initial Pheromone**	**Colony size**	**Number of Iterations**	**Number of Sequences**
1-6	0.1	0.2	0.1	50	10	10	10	15
7-14	0.1	0.2	0.1	50	15	20	20	20
15-20	0.1	0.2	0.1	50	20	30	30	40

Table 6 exhibits the obtained results from test cases using ABACO/S and three heuristics (the user defined parameters in the objective function are considered as $\gamma_1 = 2$, and $\gamma_2 = 1$). Test cases are solved under various cycle time constraints with different model demands and obtained results are compared with respect to line lengths

(*LL*), number of workstations (*NS*), and objective values (*Obj*). As can be seen from the results table, it is possible to have balancing solutions with the same performance measures for different model sequences. For example, all four approaches find balancing solutions with the same performance measures $(LL = 3,\ NS = 9$, and $Obj = 15)$ for different model sequences in test case #2.

According to the computational results, COMSOAL finds good solutions for small-sized test cases but not large-sized ones. The overall performance of RPWM and MNS varies while MNS finds a better solution than others for the test case #17. The algorithms find similar solutions, even same for some cases such as #2, #6, #7, #8, and #9. With the increasing size of the problems, ABACO/S finds better solutions than others, see test cases #13, #15, #16, #19, and #20. Therefore, it is observed that ABACO/S find good quality solutions better than or equal to those found by the three heuristics except one test case (#17, which the best solution is investigated by the MNS for). Consequently, the results indicate that the developed algorithm has a good solution capacity for the MPTALB/S problem and outperforms the three heuristics for the experimented test cases in this research.

Table 6. Computational results

	COMSOAL				**RPWM**				**MNS**				**ABACO/S**			
#	*LL*	*NS*	*Obj*	*Sequence L1-L2*	*LL*	*NS*	*Obj*	*Sequence L1-L2*	*LL*	*NS*	*Obj*	*Sequence L1-L2*	*LL*	*NS*	*Obj*	*Sequence L1-L2*
1	**3**	**10**	**16**	**CBAAAAB-EDDF**	4	11	19	BABACAA-EDFD	3	11	17	AAABCAB-EDDF	**3**	**10**	**16**	**AABAABC-FDED**
2	3	9	15	AACBB-DEEF	3	9	15	BACAB-EDEF	3	9	15	BBAAC-EDFE	3	9	15	CAABB-EDFE
3	**3**	**8**	**14**	**BACA-EEDDF**	3	9	15	BAAC-DEEFD	**3**	**8**	**14**	**CBAA-DDFEE**	**3**	**8**	**14**	**AABC-DEFDE**
4	**2**	**8**	**12**	**BCAC-FEDFDFF**	3	9	15	CCAB-FFDEDFF	3	9	15	BACC-FDDFFEF	**2**	**8**	**12**	**BCAC-DFFFFED**
5	4	13	21	BACAC-DFEE	4	13	21	CCABA-DEEF	4	13	21	CCAAB-FEED	**4**	**12**	**20**	**CCAAB-EDEF**
6	3	10	16	ABACC-EDFD	3	10	16	CBCAA-FDDE	3	10	16	ACBAC-DFED	3	10	16	ABACC-DFDE
7*	7	17	31	BCAB-DEF	7	17	31	ABCB-DFE	7	17	31	CABB-FDE	7	17	31	ACBB-FDE
8	7	17	31	ABC-	7	17	31	CAB-	7	17	31	BCA-	7	17	31	BCA-

				FEEFD				EFDFE				FEFDE				DEFFE
9	7	23	37	ACCBB-EDDF	7	23	37	ABBCC-FDED	7	23	37	ACCBB-FEDD	7	23	37	CABCB-EFDD
10	7	21	35	ABBCCBB-EDDF	**7**	**20**	**34**	**ABCBBCB-DEDF**	7	21	35	ACBCBBB-DEDF	**7**	**20**	**34**	**CCBBABB-DEFD**
11[*]	7	22	36	BCAA-DEFDDED	6	24	36	BCAA-EEDFDDD	7	23	37	BACA-DDFDEDE	**6**	**22**	**34**	**ABAC-EFDEDDD**
12	5	18	28	BCBACB-EDFE	5	18	28	CBABBC-FDEE	**5**	**17**	**27**	**BCBCAB-FEDE**	**5**	**17**	**27**	**BBCCBA-FEDE**
13	6	20	32	ACAB-FED	6	21	33	AACB-DFE	6	20	32	CABA-DEF	**5**	**19**	**29**	**ABCA-EFD**
14	4	15	23	BACB-FEFDE	4	15	23	CBBA-FFEDE	**4**	**14**	**22**	**CABB-FDFEE**	**4**	**14**	**22**	**CABB-EFEFD**
15	12	43	67	ABAC-DFDFE	11	39	61	BACA-DFEFD	12	40	64	BCAA-DEDFF	**11**	**38**	**60**	**BACA-EFFDD**
16	10	40	60	ACCB-FEEDDEE	10	38	58	CCAB-DEDEFEE	10	38	58	CABC-FEEEDDE	**10**	**37**	**57**	**CBCA-FDEEDEE**
17	9	33	51	CAAB-DDFEE	9	32	50	AABC-EDEFD	**9**	**31**	**49**	**BACA-FDEDE**	9	32	50	AABC-EEDFD
18	10	35	55	AACB-FDFEE	**9**	**31**	**49**	**ABAC-EEFDF**	10	32	52	AABC-FFEED	**9**	**31**	**49**	**ACBA-EFFED**
19	23	72	118	CABB-EFEDF	21	69	111	ABBC-FEEFD	21	68	110	BABC-EFFED	**18**	**65**	**101**	**ABCB-EEFDF**
20	17	62	96	ABAC-DEDFE	16	60	92	BCAA-DDEEF	16	59	91	ACBA-FDEED	**15**	**58**	**88**	**BACA-DEDEF**

The effect of solving model sequencing and line balancing problems simultaneously could also be distinguished easily when the obtained results are compared with those obtained by Kucukkoc and Zhang (2014a). Table 7 gives the comparison of situations when the model sequencing problem is simultaneously considered and when it is not considered.

Table 7. Comparison of ABACO and ABACO/S

ABACO	***#Test Case***	***1***	***2***	***3***	***4***	***5***	***6***	***7***[+]	***8***	***9***	***10***
	LL	4	3	3	3	4	3	7	7	7	7
	NS	12	11	10	10	14	11	19	18	25	21
	Obj	20	17	16	16	22	17	33	32	39	35
	#Test Case	***11***[+]	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20***
	LL	7	5	7	5	14	13	11	12	24	20
	NS	25	19	24	17	49	46	40	41	85	77
	Obj	39	29	38	27	77	72	62	65	133	117
ABACO/S	***#Test Case***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
	LL	3	3	3	2	4	3	7	7	7	7
	NS	10	9	8	8	12	10	17	17	23	20
	Obj	16	15	14	12	20	16	31	31	37	34
	#Test Case	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20***
	LL	6	5	5	4	11	10	9	9	18	15
	NS	22	17	19	14	38	37	32	31	65	58
	Obj	34	27	29	22	60	57	50	49	101	88

[+]Please note that test cases #7 and #11 were solved again using ABACO with the new data generated for ABACO/S to enable a comparison under the same conditions.

In the table, the ABACO row denotes the results taken from Kucukkoc and Zhang (2014a) where only the line balancing problem is solved by utilising multi-line stations. On the other hand, the ABACO/S row reports results from the ABACO/S where the model sequencing problem was integrated to the line balancing problem by considering multi-line stations. There is no doubt that the solutions found by ABACO/S are better than those obtained by ABACO for the same test cases. Although the solutions of ABACO have more flexibility and may suit any model sequence launched, more productivity is provided and operator requirements are minimized in the solutions of ABACO/S.

As it was already mentioned in Section 2, the current work is a continuation of the study carried out by Kucukkoc and Zhang (2014b). Kucukkoc and Zhang (2014b) only introduced the problem of MPTALB/S and proposed a framework that could be used as a possible approach to solving the problem. It did not include the mathematical modelling of the problem, the solution algorithm, nor any computational study. The current work contributes to the knowledge by not only formulating the MPTALB/S problem mathematically but also conducting a comprehensive computational study for the developed ABACO/S algorithm, whose results outperform other heuristics tested within the scope of this research, over the previous work.

7. Conclusion and Future Research

A mathematical model is proposed and an agent based ant colony optimisation algorithm is developed for the recently introduced MPTALB/S problem by Kucukkoc and Zhang (2014b). The performance of the algorithm is enhanced with integrated heuristics, each of which is a commonly used individual technique in this domain. To build a complete solution, different programme scripts (called agents), collaborate with each other. Initialisation and associated calculations are carried out by facilitator agent and planning agents. Sequencing agent generates different model sequence patterns and balancing agent releases ant colonies to build solutions in accordance with the generated patterns. A numerical example is given to explain the calculation of initial parameters and solution building procedures of the algorithm. A couple of results are provided with different model sequences for the same example problem.

To compare the performance of the algorithm in solving the problem, 20 test problems are solved with ABACO/S and with three other heuristics respectively and the results are reported. The results indicate that ABACO/S outperforms other heuristics in terms of sought performance measures.

Moreover, a comparison is provided between balancing the MPTALs with and without the simultaneous model sequencing problem. Our finding is that considering the model sequencing problem along with the line balancing problem provide many advantages, such as minimising required number of operators and increasing the productivity of the line. Thus, it is demonstrated that sequence of models is a significant factor that affects the efficiency of the lines as well as task sequencing. As processing times of tasks may 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. As aforementioned, utilisation of multi-line stations is one of the major benefits of parallel lines.

Research on the development of new approaches to solving the proposed problem could be suggested for future work. The algorithm could be enhanced with a model sequencing procedure instead of generating random sequences. Calibrating the algorithm by determining the parameters using a design of experiments technique, such as Response Surface Methodology, Taguchi Design of Experiment, or 2^k Factorial Design could also be considered as an extension of the current work.

Acknowledgements

The first author gratefully acknowledges the financial support from the *Balikesir University* and the *Turkish Council of Higher Education* during his PhD at the *University of Exeter* in the UK. Both authors are grateful to anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper, and to editorial team of International Journal of Production Economics for their kind and sincere support.

Appendices

A.1 Activity Diagrams of the Used Heuristic Algorithms

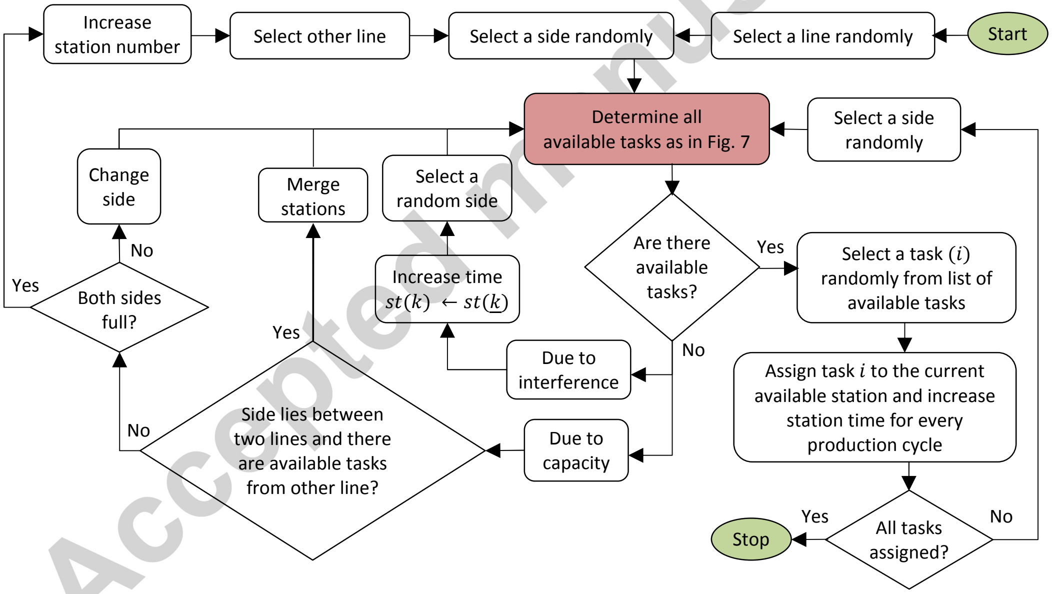
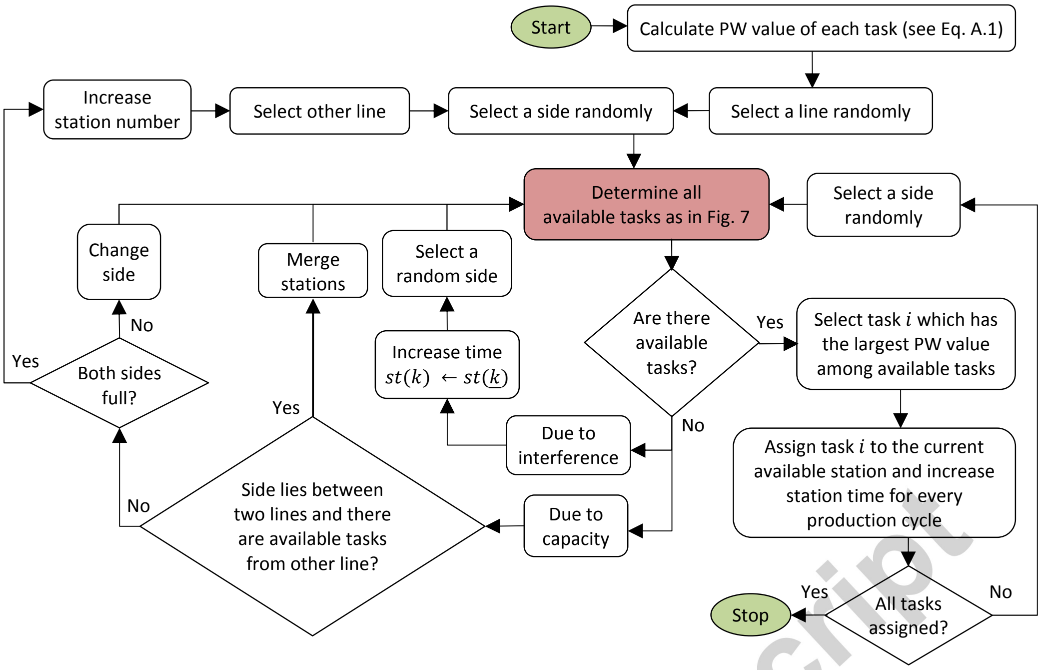
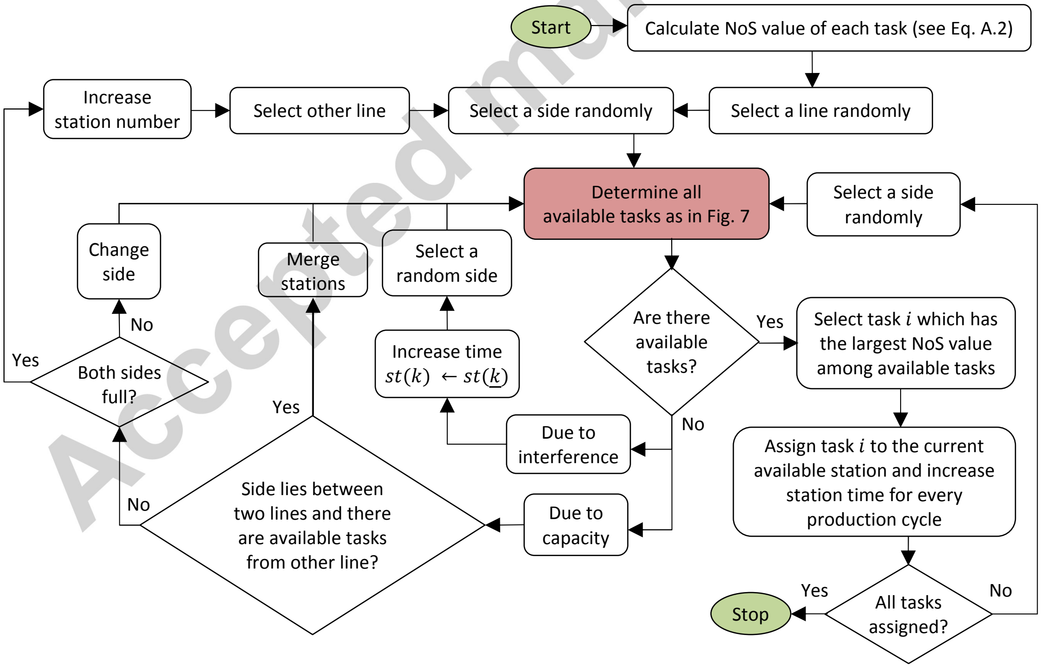


Fig. A.1. Building a line balancing solution procedure of the COMSOAL algorithm



PW: Positional Weight

Fig. A.2. **Building a line balancing solution procedure of the RPWM algorithm**



NoS: Number of Successors

Fig. A.3. Building a line balancing solution procedure of the MNS algorithm

PW value of task t_{hji} which belongs to model m_{hj} produced on line L_h is calculated as below:

$$PW_{hji} = op_{hj}\, pt_{hji} + \sum_{i \in P_{hjs}} op_{hj}\, pt_{hjs} \;, \qquad (A.1)$$

where $i \in P_{hjs}$ if task t_{hjs} is a successor of task t_{hji} .

NoS value of task t_{hji} which belongs to model m_{hj} produced on line L_h is calculated as below:

$$NoS_{hji} = \sum_{s \in T_{hj}} BS_{hjis} \,, \qquad (A.2)$$

where $i \in T_{hj}$; and BS_{hjis} is a binary variable which equals to 1 if $i \in P_{hjs}$, 0 otherwise.

A.2 Iteration Numbers of the COMSOAL Algorithm

Table A.1. Iteration numbers in which the COMSOAL algorithm found its best solutions

Test Case	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Iteration No.	4	3	6	7	9	5	11	10	9	13	15	12	9	12	14	16	11	10	17	14

References

Amini, M., Wakolbinger, T., Racer, M., Nejad, M.G., 2012. Alternative supply chain production-sales policies for new product diffusion: An agent-based modeling and simulation approach. European Journal of Operational Research, 216(2), 301-311.

Anosike, A.I., Zhang, D.Z., 2009. An agent-based approach for integrating manufacturing operations. International Journal of Production Economics, 121(2), 333-352.

Anussornnitisarn, P., Nof, S.Y., Etzion, O., 2005. Decentralized control of cooperative and autonomous agents for solving the distributed resource allocation problem. International Journal of Production Economics, 98(2), 114-128.

Arcus, A., 1966. COMSOAL, A Computer Method of Sequencing Operations for Assembly Lines. The International Journal of Production Research, 4(4), 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.

Askin, R.G., Zhou, M., 1997. A parallel station heuristic for the mixed-model production line balancing problem. International Journal of Production Research, 35(11), 3095-3105.

Bard, J.F., Darel, E., Shtub, A., 1992. An Analytic Framework for Sequencing Mixed Model Assembly Lines. International Journal of Production Research, 30(1), 35-48.

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

Battini, D., Faccio, M., Persona, A., Sgarbossa, F., 2009. Balancing-sequencing procedure for a mixed model assembly system in case of finite buffer capacity. International Journal of Advanced Manufacturing Technology, 44(3-4), 345-359.

Bautista, J., Cano, A., 2011. Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimisation and interruption rules. European Journal of Operational Research, 210(3), 495-513.

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), 217-232.

Bearzotti, L.A., Salomone, E., Chiotti, O.J., 2012. An autonomous multi-agent approach to supply chain event management. International Journal of Production Economics, 135(1), 468-478.

Boysen, N., Fliedner, M., Scholl, A., 2009. Sequencing mixed-model assembly lines: Survey, classification and model critique. European Journal of Operational Research, 192(2), 349-373.

Chutima, P., 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., Nadivi, A., 1981. A Mixed-Model Sequencing Application. International Journal of Production Research, 19(1), 69-84.

Ding, F.Y., Zhu, J., Sun, H., 2006. Comparing two weighted approaches for sequencing mixed-model assembly lines with multiple objectives. International Journal of Production Economics, 102(1), 108-131.

Goh, W.T., Zhang, Z., 2003. An intelligent and adaptive modelling and configuration approach to manufacturing systems control. Journal of Materials Processing Technology, 139(1-3), 103-109.

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

Gokcen, H., Erel, E., 1997. A goal programming approach to mixed-model assembly line balancing problem. International Journal of Production Economics, 48(2), 177-185.

Hamta, N., Ghomi, S.M.T.F., Jolai, F., Shirazi, M.A., 2013. A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. International Journal of Production Economics, 141(1), 99-111.

Hamzadayi, A., 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.

Haq, A.N., Rengarajan, K., Jayaprakash, J., 2006. A hybrid genetic algorithm approach to mixed-model assembly line balancing. International Journal of Advanced Manufacturing Technology, 28(3-4), 337-341.

He, N., Zhang, D.Z., Li, Q., 2014. Agent-based hierarchical production planning and scheduling in make-to-order manufacturing system. International Journal of Production Economics, 149(March2014), 117-130, doi: 110.1016/j.ijpe.2013.1008.1022.

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

Hwang, R., Katayama, H., 2010. Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach. International Journal of Production Research, 48(21), 6417-6441.

Kara, Y., Ozcan, U., Peker, A., 2007a. An approach for balancing and sequencing mixed-model JIT U-lines. International Journal of Advanced Manufacturing Technology, 32(11-12), 1218-1231.

Kara, Y., Ozcan, U., Peker, A., 2007b. Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives. Applied Mathematics and Computation, 184(2), 566-588.

Karabati, S., Sayin, S., 2003. Assembly line balancing in a mixed-model sequencing environment with synchronous transfers. European Journal of Operational Research, 149(2), 417-429.

Kim, Y.K., Hyun, C.J., Kim, Y., 1996. Sequencing in mixed model assembly lines: A genetic algorithm approach. Computers & Operations Research, 23(12), 1131-1145.

Kim, Y.K., Kim, J.Y., Kim, Y., 2000a. A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines. Applied Intelligence, 13(3), 247-258.

Kim, Y.K., Kim, J.Y., Kim, Y., 2006. An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines. European Journal of Operational Research, 168(3), 838-852.

Kim, Y.K., Kim, S.J., Kim, J.Y., 2000b. Balancing and sequencing mixed-model U-lines with a co-evolutionary algorithm. Production Planning & Control, 11(8), 754-764.

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

Kucukkoc, I., Karaoglan, A.D., 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), 5039-5054.

Kucukkoc, I., 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, pp. 1-16.

Kucukkoc, I., Zhang, D.Z., 2014a. An Agent Based Ant Colony Optimisation Approach for Mixed-Model Parallel Two-Sided Assembly Line Balancing Problem, in: Grubbström, R.W., Hinterhuber, H.H. (Eds.), Pre-Prints of the Eighteenth International Working Seminar on Production Economics. Congress Innsbruck, Innsbruck, Austria, pp. 313-328.

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

Lee, T.O., Kim, Y., Kim, Y.K., 2001. Two-sided assembly line balancing to maximize work relatedness and slackness. Computers & Industrial Engineering, 40(3), 273-292.

Manavizadeh, N., Hosseini, N.S., Rabbani, M., Jolai, F., 2013a. A Simulated Annealing algorithm for a mixed model assembly U-line balancing type-I problem considering human efficiency and Just-In-Time approach. Computers & Industrial Engineering, 64(2), 669-685.

Manavizadeh, N., Tavakoli, L., Rabbani, M., Jolai, F., 2013b. A multi-objective mixed-model assembly line sequencing problem in order to minimize total costs in a Make-To-Order environment, considering order priority. Journal of Manufacturing Systems, 32(1), 124-137.

McMullen, P.R., Tarasewich, P., 2003. Using ant techniques to solve the assembly line balancing problem. Iie Transactions, 35(7), 605-617.

Merengo, C., Nava, F., Pozzetti, A., 1999. Balancing and sequencing manual mixed-model assembly lines. International Journal of Production Research, 37(12), 2835-2860.

Mes, M., van der Heijden, M., van Harten, A., 2007. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. European Journal of Operational Research, 181(1), 59-75.

Morrison, D.R., Sewell, E.C., Jacobson, S.H., 2014. An application of the branch, bound, and remember algorithm to a new simple assembly line balancing dataset. European Journal of Operational Research, 236(2), 403–409.

Mosadegh, H., Zandieh, M., Ghomi, S.M.T.F., 2012. Simultaneous solving of balancing and sequencing problems with station-dependent assembly times for mixed-model assembly lines. Applied Soft Computing, 12(4), 1359-1370.

Otto, A., Otto, C., Scholl, A., 2013. Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. European Journal of Operational Research, 228(1), 33-45.

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

Ozcan, U., Kellegoz, T., Toklu, B., 2011. A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem. International Journal of Production Research, 49(6), 1605-1626.

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

Rabbani, M., Moghaddam, M., 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., Lit, P.D., Delchambre, A., 2000. Designing mixed-model assembly lines. IEEE Transactions on Robotics and Automation, 16(3), 268–280.

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

Sternatz, J., 2013. Enhanced multi-Hoffmann heuristic for efficiently solving real-world assembly line balancing problems in automotive industry. European Journal of Operational Research, DOI: 10.1016/j.ejor.2013.11.005.

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

Thomopoulos, N.T., 1967. Line Balancing-Sequencing for Mixed-Model Assembly. Management Science, 14(2), 59-75.

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

Vilarinho, P.M., 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), 1405-1420.

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

Xu, S., Li, F.M., 2013. A Modified Genetic Algorithm for Sequencing Problem in Mixed Model Assembly Lines. Advanced Materials Research, 655-657(2013), 1675-1681.

Yano, C.A., Rachamadugu, R., 1991. Sequencing to Minimize Work Overload in Assembly Lines with Product Options. Management Science, 37(5), 572-586.

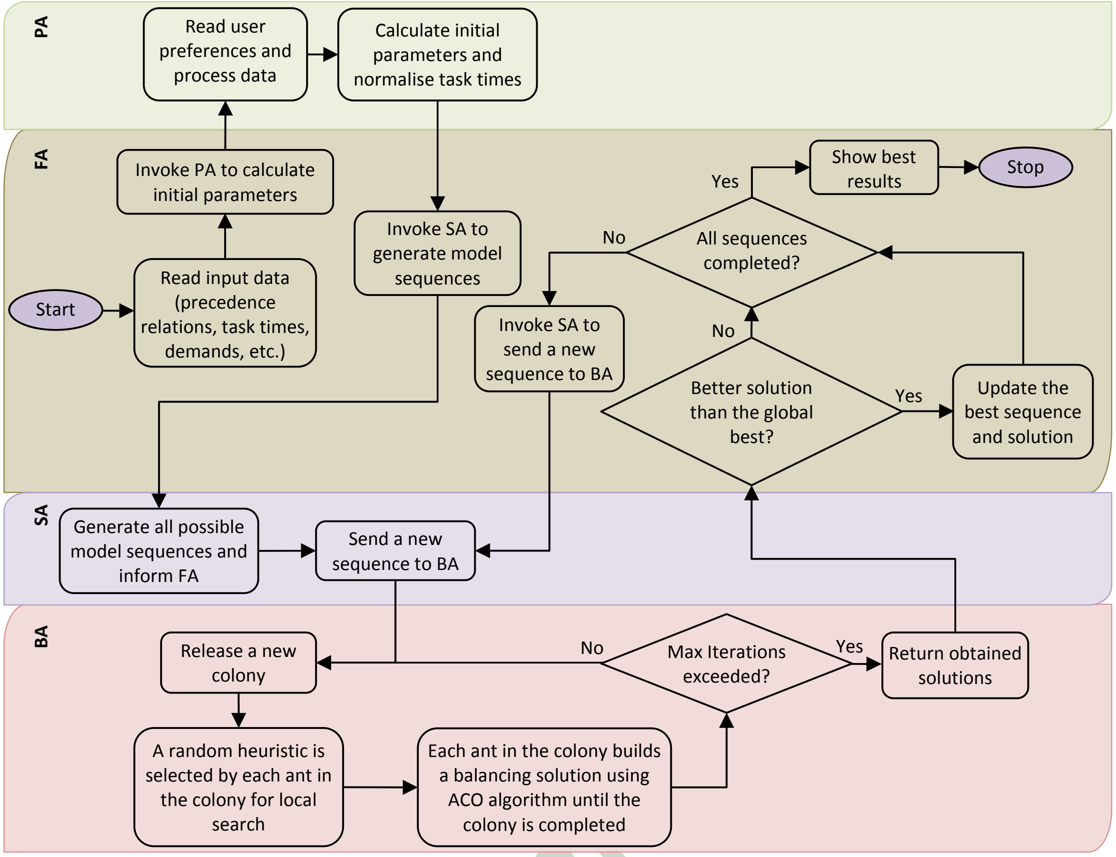
Zhang, D.Z., Kucukkoc, I., 2013. Balancing Mixed-Model Parallel Two-Sided Assembly Lines, in: Amodeo, L., Dolgui, A., Yalaoui, F. (Eds.), Proceedings of the International Conference on Industrial Engineering and Systems Management (IEEE-IESM'2013), École Mohammadia d'Ingénieurs de Rabat (EMI), International Institute for Innovation, Industrial Engineering and Entrepreneurship (I4E2), Rabat, Morocco, pp. 391-401.

Zhang, Z.W., Sharifi, H., 2007. Towards theory building in agile manufacturing strategy - A taxonomical approach. Ieee Transactions on Engineering Management, 54(2), 351-370.

Zheng, Y.Q., Wang, Y.P., Hu, B., Wang, Y.S., 2011. A sequencing approach of models in mixed-model assembly lines. Mechanika, 17(4), 418-422.

Highlights

- We model the balancing and sequencing of mixed-model parallel two sided lines.
- An agent-based ACO approach is developed to solve the recently introduced problem.
- Performance of the developed algorithm is compared against heuristics.
- Experimental results prove the performance of the ABACO/S over three heuristics.
- It is also exhibited that considering sequencing issue with balancing is essential.


PA
Read user preferences and process data
Calculate initial parameters and normalise task times
FA
Invoke PA to calculate initial parameters
Show best results
Stop
Yes
All sequences completed?
No
Invoke SA to generate model sequences
Start
Read input data (precedence relations, task times, demands, etc.)
Invoke SA to send a new sequence to BA
No
Better solution than the global best?
Yes
Update the best sequence and solution
SA
Generate all possible model sequences and inform FA
Send a new sequence to BA
BA
Release a new colony
No
Max Iterations exceeded?
Yes
Return obtained solutions
A random heuristic is selected by each ant in the colony for local search
Each ant in the colony builds a balancing solution using ACO algorithm until the colony is completed