Balancing of mixed-model parallel U-shaped assembly lines considering model sequences

Ibrahim Kucukkoc [a b *], David Z. Zhang [a c]

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

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

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

ikucukkoc@balikesir.edu.tr, *d.z.zhang@exeter.ac.uk*, Tel: +441392723641

Abstract

As a consequence of increasing interests in customised products, mixed-model lines have become the most significant components of today's manufacturing systems to meet surging consumer demand. Also, U-shaped assembly lines have been shown as the intelligent way of producing homogeneous products in large quantities by reducing the workforce need thanks to the crossover workstations. As an innovative idea, we address the mixed-model parallel U-shaped assembly line design which combines the flexibility of mixed-model lines with the efficiency of U-shaped lines and parallel lines. The multi-line stations utilised in between two adjacent lines provide extra efficiency with the opportunity of assigning tasks into workstations in different combinations. The new line configuration is defined and characterised in details and its advantages are explained. A heuristic solution approach is proposed for solving the problem. The proposed approach considers the model sequences on the lines and seeks efficient balancing solutions for their different combinations. An explanatory example is also provided to show the sophisticated structure of the studied problem and explain the running mechanism of the proposed approach. The results of the experimental tests and their statistical analysis indicated that the proposed line design requires fewer number of workstations in comparison with independently balanced mixed-model U-lines.

Keywords: Assembly line balancing; design of production systems; production planning; U-shaped assembly lines; mixed-model assembly lines.

Corresponding author: Ibrahim Kucukkoc, ikucukkoc@balikesir.edu.tr

Balancing of mixed-model parallel U-shaped assembly lines considering model sequences

Ibrahim Kucukkoc [a b *], **David Z. Zhang** [a c]

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

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

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

ikucukkoc@balikesir.edu.tr, *d.z.zhang@exeter.ac.uk*, Tel: +441392723641

Abstract

As a consequence of increasing interests in customised products, mixed-model lines have become the most significant components of today's manufacturing systems to meet surging consumer demand. Also, U-shaped assembly lines have been shown as the intelligent way of producing homogeneous products in large quantities by reducing the workforce need thanks to the crossover workstations. As an innovative idea, we address the mixed-model parallel U-shaped assembly line design which combines the flexibility of mixed-model lines with the efficiency of U-shaped lines and parallel lines. The multi-line stations utilised in between two adjacent lines provide extra efficiency with the opportunity of assigning tasks into workstations in different combinations. The new line configuration is defined and characterised in details and its advantages are explained. A heuristic solution approach is proposed for solving the problem. The proposed approach considers the model sequences on the lines and seeks efficient balancing solutions for their different combinations. An explanatory example is also provided to show the sophisticated structure of the studied problem and explain the running mechanism of the proposed approach. The results of the experimental tests and their statistical analysis indicated that the proposed line design requires fewer number of workstations in comparison with independently balanced mixed-model U-lines.

Keywords: Assembly line balancing; design of production systems; production planning; U-shaped assembly lines; mixed-model assembly lines.

1. Introduction

The manufacturing industry has experienced crucial changes with the industrial revolution emerged in 18th century in England. Mass production techniques have been put into practice by companies to increase capacity and so productivity. Following these developments, which built a base for today's high-performance manufacturing systems, the first moving-belt was constructed by Henry Ford and his colleagues in early 20th century at Highland Park assembly plant. This was the pioneering attempt to establish an assembly line, which builds the major and the most significant parts of modern production systems in several industries, *e.g.*, automotive, electronics, home appliances *etc*. (Kucukkoc *et al.* 2015).

An assembly line is a sequence of workstations linked to each other by a conveyor or moving belt, on which homogeneous products are consecutively assembled in an efficient way. The problem of determining which task will be assembled in which workstation with the aim of minimising the total number of workstations and/or cycle time is called the *assembly line balancing problem*. Some constraints must be satisfied to obtain a feasible line balance, such as capacity constraint, precedence relationships constraint, task assignment constraint, *etc*. (Kucukkoc and Zhang 2015e). The assembly line balancing problem was first studied by Salveson (1955) in its simplest version (where a simple straight line was considered with a single commodity of product on the line) and has gained continuing interest from academics as well as practitioners since then.

In its traditional version, assembly lines have a straight structure on which a series of workstations are located sequentially, to produce only one type of product in large quantities. The problem of balancing such lines is called *simple assembly line balancing problem*, which is an NP-hard combinatorial optimisation problem, as shown by Wee and Magazine (1982). However, as shown by Thomopoulos (1967) and several studies following this, mixed-model lines (where more than one model of a base product are assembled on the same line) carry several practical advantages over single-model lines.

This paper contributes to knowledge by presenting the first and original research results on *mixed-model parallel U-shaped assembly line (MPUL shortly)* concept, which was recently introduced by Kucukkoc and Zhang (2015e). So that, as an innovative approach, the model variation flexibility is introduced to the parallel U-shaped assembly line system with the aim of obtaining well-balanced line configurations by reducing idle times. The addressed line system combines the advantages of its sub-configurations, *i.e.*, mixed-model lines, parallel lines and U-shaped lines as will be explained in the following sections. Thanks to the proposed design, the multi-line stations and crossover stations are of the key factors which provide advantages. As the leading car manufacturers set new goals for advanced flexible manufacturing systems (see for example Ford Motor Company (2015)), which can easily adapt to varying customer demand, the proposed MPUL system can replace the conventional line configurations. Thus, the MPUL system can play crucial roles in "*giving customers the features, fuel efficiency and technology they want anywhere in the world*" as stated by Ford Motor Company (2015). To solve the balancing problem on MPULs and handle its complexity, which will increase to a great extent due to the contained line configurations which are already complex alone, a heuristic solution approach is also proposed and explained, as another contribution of this research.

The remainder of this paper is organised as follows. Section 2 reports the review of literature while Section 3 introduces and defines the proposed MPUL system in details. Section 4 proposes a possible heuristic solution approach for the MPUL balancing problem and explains its running principles. Section 5 presents an explanatory example and exhibits the running principle of the heuristic algorithm. The sophisticated structure of the problem and challenging issues are also discussed in the same section. The results of the experimental tests are reported in Section 6 and the conclusions and possible industrial implications of the work are given in Section 0 followed by the future research directions.

2. Related work

During the last six decades, various types of line configurations along with several objectives and constraints have been studied by academics and practitioners. Thomopoulos (1967) proposed a mixed-model assembly line system, where a variety of product models having similar product characteristics are assembled. Mixed-model lines provide advantages over single-model lines as a variety of similar products can be assembled on the same line with no need of setup times between model changes (Kucukkoc, Karaoglan, and Yaman 2013).

Assembly lines can be divided into two main groups in terms of the line shape: *straight lines* and *U-shaped lines* (Miltenburg and Wijngaard 1994). In U-shaped lines (or U-lines, shortly), the entrance and the exit of the line system are very close to each other. Operators may handle work-pieces both on the front and back of the line thanks to the formed U-shaped line configuration. Operators located in crossover workstations can perform tasks from both front and back of the line. Thus, idle times are reduced and resource utilisation is increased thanks to the crossover stations located in between front and back of the U-line. Miltenburg and Wijngaard (1994) introduced the U-line balancing problem and a series of researchers followed them; *e.g.,* see Urban (1998), Scholl and Klein (1999), and Urban and Chiang (2006) for exact solution approaches; Erel *et al.* (2001), Gökçen *et al*. (2005), Hwang *et al*. (2008) and Sabuncuoglu *et al*. (2009) for heuristic/meta-heuristic solution approaches on simple U-line configurations. Miltenburg (2001) also introduced the embedded U-line arrangement, which considers "*a large U-line encircling a small U-line, all manned by two operators*". The difference between the *embedded U-line arrangement* (Miltenburg 2001) and the MPUL concept described in the current work will be provided in Section 3.

Almost decade ago, Gökçen *et al.* (2006) introduced the line parallelisation idea to minimise idle times by maximising the use of shared resources. Parallel line configuration provides the opportunity of building multi-line stations in which operators can perform jobs from both of the adjacent lines. This also helps obtain well-balanced line configurations as there is more chance to assign tasks in different combinations. The line parallelisation idea has been applied to mixed-model lines, where more than one model of a base product is assembled on the same line, by Ozcan *et al.* (2010). Ozcan *et al.* (2010) introduced the parallel mixed-model line system and demonstrated the requirement of considering balancing and sequencing problems simultaneously on those lines through experimental tests. In another study, Ozcan *et al*. (2010) located more than one two-sided line in parallel to each other and introduced the parallel two-sided assembly line system. A tabu search approach was also developed to find efficient solutions for the parallel two-sided assembly line balancing problem. Kucukkoc and Zhang (2015c, 2015d) developed a genetic algorithm approach and an ant colony optimisation based approach for solving the parallel two-sided assembly line balancing problem with different objectives. Kucukkoc and Zhang (2014b, 2014a) improved the parallel two-sided assembly line system in a mixed-model production environment and proposed agent-based ant colony algorithms for solving the problem efficiently. In their latter study, Kucukkoc and Zhang (2015b) enhanced the agent-based ant colony optimisation algorithm by integrating a genetic algorithm based model sequencing mechanism. As the common consequence of these researches, it was shown that locating two lines in parallel to each other helps minimise the total number of workstations.

As the pioneering research in parallelisation of the U-shaped lines, the study by Kucukkoc and Zhang (2015a) introduced the parallel U-shaped assembly line system. However, model variations across the lines have not been considered in their study. Instead, a single product model is produced on each of the U-shaped lines located in parallel to each other. Mixed-model production have been extensively studied on individual U-shaped lines. Sparling and Miltenburg (1998) introduced the mixed-model U-line (MMUL) balancing problem and presented an approximate solution approach with a numerical example. The model sequencing problem was not considered in their research. Kim *et al*. (2006), developed a new genetic approach, called endosymbiotic evolutionary algorithm, to deal with both balancing and sequencing problems in MMULs. Kara *et al*. (2007), presented a multi-objective approach (enhanced with a neighbourhood generation method) for solving the same problem with the aim of minimizing the absolute deviations of workloads, part usage rate and cost of setups. Özcan *et al*. (2011), Kazemi *et al*. (2011) and Hamzadayi and Yildiz (2012) developed genetic algorithm based approaches for balancing and sequencing MMULs. While Özcan *et al*. (2011) considered the stochastic task processing times, Kazemi *et al*. (2011) allowed the assignment of common tasks into different stations. Parallel workstations and zoning constraints were considered in Hamzadayi and Yildiz (2012). Hamzadayi and Yildiz (2013) and Kara (2008) proposed simulated annealing algorithms for balancing and sequencing MMULs. In a latter study, Kara and Tekin (2009) presented a mixed integer programming formulation which minimises the number of workstations for a given model sequence. Lian *et al*. (2012) developed a modified colonial competitive algorithm for solving the line balancing and model sequencing problems in MMULs simultaneously. The objective was to minimise the absolute deviations of workloads. The performance of the algorithm was compared with that of existing algorithms through test problems. Li *et al*. (2012) studied the problem of sequencing minimum product sets in MMUL. While the line balancing problem was not considered in their research, a branch and bound algorithm was proposed to minimise the work overload. Rabbani *et al*. (2012) addressed the mixed-model two-sided assembly lines in a multiple U-shaped layout environment. A mixed-integer programming formulation and a genetic algorithm based heuristic were developed to simultaneously minimize the cycle time and the number of workstations. In another study, Rabbani *et al.* (2012) considered only the line balancing problem on MMULs and presented a genetic algorithm approach with the aim of minimising crossover workstations considering operator travel times. Manavizadeh *et al.* (2013) developed a simulated annealing approach which assigns operators with different skill levels into workstations upon the line balance is obtained. Another simulated annealing approach was developed by Dong *et al.* (2014) to minimise the expectation of work overload time when balancing and sequencing MMULs. For the same problem, Manavizadeh *et al.* (2015) developed a multi-objective heuristic algorithm which incorporates the minimization of the cycle time, the wastages in each station and the work overload. Kucukkoc and Zhang (2015e) introduced the MPUL concept, where the variants of a base product can be assembled on each of the parallel U-lines in different model mixes.

As seen from this comprehensive survey, only the study by Kucukkoc and Zhang (2015e) incorporated model variations on parallel U-shaped lines. Kucukkoc and Zhang (2015e) brought the MPUL idea to the attention of academia. However, no solution method was proposed in their study. Also, they have not presented research results, which prove the advantages of MPULs over conventional (or independently balanced) MMULs. As a continuing research built over Kucukkoc and Zhang (2015e), this paper fills in this gap through presenting new and original research results obtained from a new solution method developed for MPULs. The individual model sequences of the lines have been considered to prevent infeasible solutions and violation of capacity limits. Furthermore, a paired-sample t-test was conducted to statistically prove the advantage of MPUL over independently balanced MMULs.

3. Problem description

3.1. Main characteristics and advantages

The parallel U-shaped assembly line system is a combination of two or more U-shaped assembly lines, represented by L_h $(h = 1, \ldots, H)$, located in parallel to each other. Two or more different product models, where each model on L_h is represented by m_{hj} ($j = 1, \dots, M_h$), are produced on each of the U-shaped assembly lines. Each product model produced on each of the lines has its own set of tasks, where a task is represented by t_{hi} ($i = 1, \dots, T_h$), performed according to predefined precedence relationships. P_{hi} represents the set of predecessors of task t_{hi} for model m_h. Each task (t_{hi}) for model m_{hj} on line L_h requires a certain amount of processing time, symbolised with pt_{hji}, to be performed. In such a parallel U-shaped assembly line system, operators located in between two adjacent U-lines will have the opportunity of performing their jobs on both of the lines. Meanwhile, operators located in the centre of the inner U-shaped line will have the flexibility of performing tasks on the front and back of the line. The proposed parallel U-shaped assembly line system is illustrated in Figure 1.

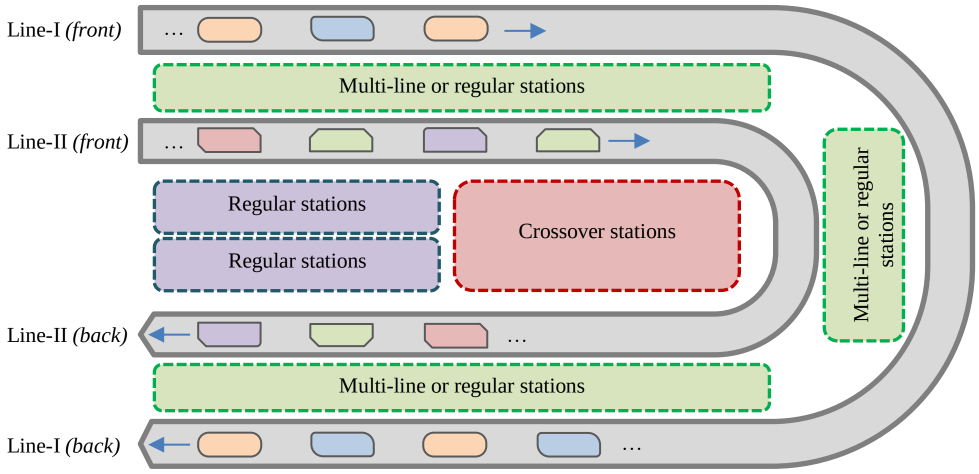


Figure 1. The proposed parallel U-shaped assembly line system.

As it can be seen from Figure 1, two U-shaped lines are located in parallel to each other. Operators located in multi-line stations are allowed to perform their jobs belonging to the models assembled on both Line-I and Line-II. This brings the opportunity of assigning tasks to the workstations in different combinations. By this way, the idle times of workstations are reduced (therefore the efficiency of the whole line system is increased) as well as the communication between workers is increased. It is also possible to utilise regular workstations instead of multi-line stations depending on the overall efficiency of the line configuration obtained.

The proposed system also carries the advantages of U-shaped lines as crossover stations are allowed to be utilised in the centre of the inner line. The operators located in these workstations can travel between the front and back of the line to help perform jobs on both branches of the U-line. Also, there are regular stations in which tasks from only one branch of the line are accomplished.

As an advantage of the proposed assembly line system, each of the parallel lines may have a different cycle time regardless of the other one. On one hand, this increases the flexibility of the system because it is possible to produce products in different throughput rates. On the other hand, the complexity of the problem of balancing this line system increases as it is needed to determine common time slots between the two lines to be able to assign tasks in multi-line stations. Gökçen *et al.* (2006) used least common multiple (LCM) based approach to make modelling easier when different cycle times are subject to consideration in such a parallel line system configuration (Zhang and Kucukkoc 2013; Kucukkoc and Zhang 2014c). This procedure will not be repeated here due to page limit.

The MPUL concept differs from the embedded U-line arrangement addressed in Miltenburg (2001) in three ways. First of all, in embedded U-line arrangement, the whole system is operated by only two operators. This requires operators walk quite a long distance across the line. Secondly, the inner line (small U-line as called in the study of Miltenburg (2001)) does not allow operators work between its two (front and back) branches. This requires other operators constantly travel back and forth between the two lines to complete all tasks. Finally, as mentioned in Miltenburg (2001), "*A bigger disadvantage (of the embedded U-line arrangement) is that the same operator is usually not able to operate the entrance and exit operations of a line*". This is caused by the different movement direction of lines in embedded U-line arrangement. Nevertheless, in our model, the lines move in the same direction. Therefore, the same worker operates on either the entrance of both lines or the exit of both lines.

3.2. Challenging issues

On two-sided assembly lines, where both sides of the line (called left and right sides) are used, some tasks can be performed on the left side of the line while some tasks belonging to the same model can be performed on the opposite (right) side of the line. There may be precedence relationships between those tasks performed on the opposite sides of the lines. This phenomenon is called interference and extra attention is needed to avoid violation of precedence relationships as this may cause infeasible solutions. In the line system proposed in this research, the situation is similar to the two-sided lines because it is allowed to perform jobs on both sides of the Line-II (see Figure 1). When a model belonging to Line-II is being assembled, two different operators – one in the multi-line station between two adjacent lines while the other one is in crossover station or regular station in the centre of Line-II – can work on the same work-piece. The challenge in this situation is to ensure precedence relationships among tasks caused by technological or organisational requirements.

Capacity constraint is another must have requirement that needs to be satisfied in assembly line balancing problems. Each workstation has a limited time, called cycle time, in which they need to complete their tasks. In a paced (synchronous) assembly line, all workstations are linked to each other via a conveyor or any other transportation system. The synchronisation is achieved by transferring semi-finished product models between stations at a pre-determined and fixed time interval. In the proposed line system, each workstation utilised on the same line has the same capacity, irrespectively whether or not it is a multi-line station. The capacity of a multi-line station is determined by the cycle time of the line on which the multi-line station is constructed. Crossover stations and regular stations adhere to the cycle time of Line-II. Obviously, when an LCM based approach is applied, both lines will be balanced using the same cycle time (but modified task times) and this makes modelling and solving the problem easier.

In some cases, there can be other constraints caused by the safety rules or allocations of the workstations, such as positive-negative zoning constraints and positional constraints. The positive zoning constraint requires the two tasks be assigned in the same workstation while in the negative zoning constraint two tasks must be assigned in different workstations. Also, in some situations, a certain task may need to be operated at a certain workstation. This is assured by positional constraints.

3.3. *Model changes*

Introducing the product model diversity to such a complex production environment makes modelling and solving the problem harder to a great extent. The most important factor that contributes to this complexity is the change in product models on the parallel lines from one production cycle to another. A production cycle is defined as a phase of the line system where there is a different mix of products in the workstations. When the cycle times of the parallel lines are different, model changes on the lines will take place at different times and this yields a quite complex situation to manage in multi-line stations. The total number of different production cycles that can appear in a parallel assembly line system depends on the product mix assembled on the lines. Formulations on how to calculate production cycles will be provided in Section 4.

To give an example, let us assume a parallel U-shaped assembly line system composed of two lines, Line-I and Line-II, where a mix of different product models are assembled on each of the U-lines, *i.e.*, models A, B, and C on Line-I and models D and E on Line-II. The cycle times of Line-I and Line-II are 10 time-units and 20 time-units, respectively. Therefore, the moving speed of Line-I is as double of that of the Line-II. If the model sequences on Line-I and Line-II are considered as $MS_1 = ABCB$ and $MS_2 = ED$, possible production cycles will be as in Figure 2.

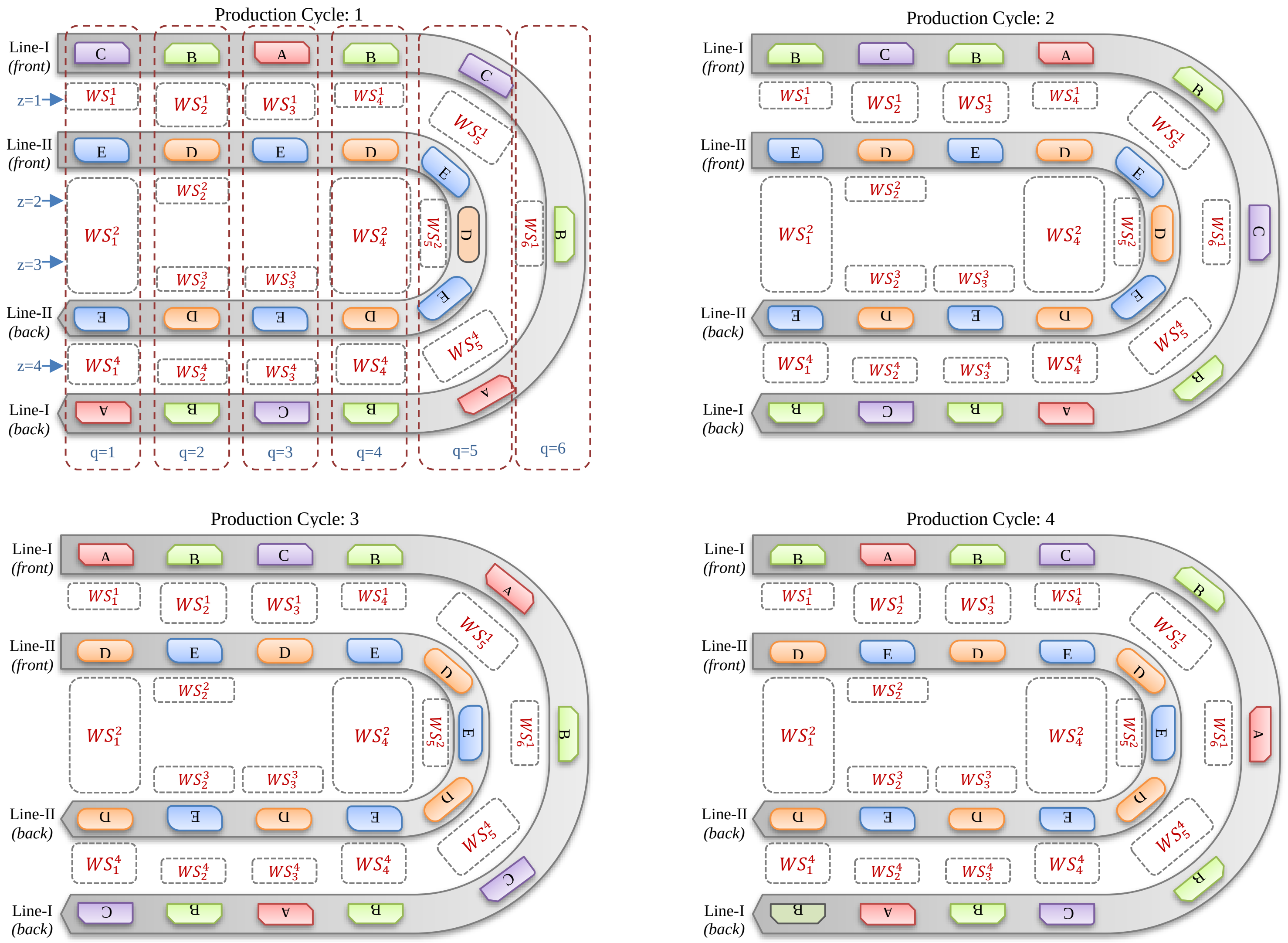


Figure 2. The changing model combinations through different production cycles.

To identify workstations (symbolised with WS_q^z), the working space is divided into different *zones* ($z = 1,2, \ldots, 4$) and *queues* ($q = 1,2, \ldots, Q$; where $Q = 6$ in this example) as seen from Figure 2. The workstations located in between two adjacent lines and assigned tasks from both of the lines are called *multi-line stations* (*e.g., see* WS_2^1, WS_3^1, WS_5^1, *etc.*) and operators located in these workstations can perform their jobs on both of the lines. Workstations located in between the front and back branches of Line-II and assigned tasks from both branches are called *crossover stations* (*e.g.,* WS_1^2 and WS_4^2) and operators located in these stations can perform their jobs on either branch. As in traditional configurations, *regular stations* (*e.g.,* WS_2^2 and WS_2^3), in which operators perform their jobs for only one specific line and/or branch are also allowed.

4. Mixed-model parallel U-line heuristic (MPUH)

This section proposes a heuristic algorithm to find balancing solutions for the MPUL system introduced in the previous section.

4.1. The outline of MPUH

The heuristic procedure proposed by Kucukkoc and Zhang (2015a) for parallel U-line systems has been improved and adapted for the MPULs. Thus, the algorithm proposed in this paper, called *mixed-model parallel U-line heuristic (or MPUH shortly)*, integrates the modifications of two well-known heuristics, the *ranked positional weight method* (Helgeson and Birnie 1961) and the *maximum number of successors* (modified from Tonge (1960)), in an MPUL production environment.

The pseudo-code of the general outline belonging to MPUH is given in Figure 3. As seen, the algorithm starts with generating all possible model sequencing combinations (PMSC) after determining the minimum part sets (MPS) in accordance with the MPS principle that will be explained in Section 4.3. Starting from the first one, each item in PMSC is tried one-by-one. To explain, the first model sequencing combination is selected and a total of '$maxBalIt$' balancing solutions is constructed using the solution building procedure that will be explained in Section 4.4. When the $maxBalIt$ number is exceeded, the best solution found for the first model sequencing combination is recorded and the algorithm moves to the second combination. Again, after constructing a total of '$maxBalIt$' balancing solutions, the best solution is recorded for this combination and accepted as the new global best solution if it is better than the current global best solution. In this way, all possible model sequencing combinations in PMSC are tried one-by-one. Finally, the algorithm is terminated and the best model sequencing combination which gives the best line balancing solution is reported.

Algorithm: MPUH procedure

```
1   Start
2   Import data
3   Generate PMSC set (see Section 4.3)
4   If ($C_1 \neq C_2$){
5     Apply LCM based approach (see Section 3.1)
6   } End
7   Calculate $\omega_{hi}^{PI}$ for all tasks, $t_{hi}$ ($h = 1,2,\dots,H;\ i = 1,\dots,T_h$)
8   For ($int\ ms = 1; ms \leq PMSC.size; ms++$){
9     Choose the $ms$th combination from PMSC
10    For ($int\ balIt = 1; balIt \leq maxBalIt; balIt++$){
11      $UTL_h \leftarrow \{1,2,\dots,T_h\}$, where $h = 1,2,\dots,H$
12      $isFrontArea = true$
13      $z = randomIntegerBetween[1,2]$
14      $WT_q^{z\varphi} \leftarrow 0;\ z = 1,2,\dots 4;\ q = 1,2,\dots Q;\ \varphi = 1,2,\dots \phi$
15      $EST_{hi}^{\varphi} \leftarrow 0; h = 1,2;\ i = 1,2,\dots T_h;\ \varphi = 1,2,\dots \phi$
16      Build a balancing solution (see Section 4.4)
17      Calculate objective function of the solution ($OBJ$)
18      If ($OBJ < OBJ^*$){
19        $OBJ^* \leftarrow OBJ$
20        Update the best solution
21      } end
22    } end
23  } end
24  Terminate
```

Figure 3. The pseudo-code of the general outline.

The main objective (*OBJ*) is to minimise the number of workstations as a primary goal and to minimise the line length (or the number of queues) as an additional goal. Therefore, if two solutions, which has the same number of workstations but different lengths are obtained, the one with the lower length is preferred. As the MPUL system is a much more complex problem in comparison with its single-model version, it requires even more sophisticated solution procedure to obtain efficient balancing solutions. Therefore, the MPUH procedure, which uses a newly introduced *task priority index* (ω_{hi}^{PI}) value for selecting tasks, comprises several improvements as the details will be explained in this section. The superscript *"PI"* is used here as an acronym for priority index. The ω_{hi}^{PI} value of a task is the multiplication of its *positional weight index* (ω_{hi}^{PW}) and the *number of successors index* (ω_{hi}^{NS}); *i.e.*, $\omega_{hi}^{PI} = \omega_{hi}^{PW} \times \omega_{hi}^{NS}$. In this equation, the positional weight index (ω_{hi}^{PW}) of a task is determined by its positional weight (PW_{hi}) which is calculated as follows; $PW_{hi} = wpt_{hji} + \sum_{j \in S_{hi}} wpt_{hji}$, where S_{hi} represents the set of successors of task t_{hi} on line L_h. The term wpt_{hji} denotes the weighted processing time of task t_{hji} and is calculated as $wpt_{hji} = \left(\sum_{j=1}^{M_h} d_{hj}\, pt_{hji}\right) / \sum_{j=1}^{M_h} d_{hj}$ where $h = 1,\dots,H$; $i = 1,\dots,T_h$ (to recapitulate, pt_{hji} is the processing time of task t_{hji}). Afterwards, on each of the lines, the tasks are sequenced in descending order based on their PW_{hi} values and the positional weight index (ω_{hi}^{PW}) of the task which has the lowest PW_{hi} value is set to '1'. The ω_{hi}^{PW} value of the task which has the second lowest PW_{hi} value is set to 2 and this is repeated until every task is assigned a ω_{hi}^{PW} value. If there are two or more tasks which have the same positional weights, the lowest numbered task gets the higher positional weight index.

To calculate the ω_{hi}^{NS} value of a task, the total number of successors (NS_{hi}) of this task is calculated as $NS_{hi} = card\{S_{hi}\}$, where $card\{S_{hi}\}$ corresponds to the length of the set of successors of task t_{hi} on line L_h. On each of the lines, tasks are sequenced in descending order based on their NS_{hi} values and the ω_{hi}^{NS} value of the task which has the lowest NS_{hi} value is set to '1'. The ω_{hi}^{NS} value of the task which has the second lowest NS_{hi} value is assigned 2, and so on. These calculations will be exemplified in Section 5.

4.2. Diversification

A stochastic assignment procedure is applied in MPUH to alternate assignment positions during the balancing process and have more diversified as well as efficient balancing solutions. For this aim, depending on the value of a newly introduced random boolean variable ($isFrontArea$), the value of z index is randomly determined after a new task is assigned. Thus, for example, if the assignment procedure is being performed on the front area of the line system ($isFrontArea = true$), z is randomly assigned 1 or 2 ($z = randomIntegerBetween[1,2]$). When there is no available task for the current assignment area (front or back), the value of $isFrontArea$ is alternated ($isFrontArea = false$), and z is assigned a random value depending on the value of $isFrontArea$ as exemplified in the pseudo-code.

4.3. MPS principle

At the beginning of the assignment procedure (when the final assignment configuration is not known), the most challenging issue is the lack of knowledge on which model will appear in workstations located at the back of the line since the total number of workstations is not known. The algorithm overcomes this problem by considering the maximum processing times of the tasks common in different models when assigning a task from the back of the line (or precedence relationships diagram). For this aim, a newly introduced term, namely deserved task time (DT_{hi}), is used as will be explained below.

When assigning tasks to the front areas of the lines, the task processing times belonging to the actual models are used by the algorithm, as different from the back of the line. The algorithm generates possible model sequences and the *minimum part set (MPS)* principle (Bard, Darel, and Shtub 1992) is used to determine model mixes on the lines. The MPS approach was used by Ozcan *et al.* (2010) for parallel mixed-model lines and Kucukkoc and Zhang (2014a) for mixed-model parallel two-sided lines. The MPS_h $(h = 1, \dots, H)$ is a vector which represents the smallest set having the same proportions of different product models as the demands. It represents the mix of product models on line L_h, such that $MPS_h = (d_{h1}, \dots, d_{hM_h})$, where $d_{hj} = D_{hj}/gcd_h$ (where $j = 1, \dots, M_h$ and $h = 1, \dots, H$). The D_{hj} denotes the demand of model m_{hj} ($j = 1, \dots, M_h$) on L_h ($h = 1, \dots, H$). The gcd_h ($h = 1, \dots, H$) is the greatest common divisor of the demands of the product models assembled on the same line (L_h, where $h = 1, \dots, H$). Obviously, the total demand is met by gcd_h times repetition of producing the MPS_h. The model sequence of line L_h is represented with MS_h and the length of MS_h for one MPS_h, which means the total number of products on line L_h for one MPS_h, is calculated as follows; $S_h = \sum_{j=1}^{M_h} d_{hj}$. Thus, the maximum number of possible model sequencing combinations for a determined model sequence pattern on two lines is calculated as $MS_{max} = (S_1 \times S_2)$. This also regulates how many different production cycles ($\varphi = 1, \dots, \phi$) the system should be split into ($\phi = MS_{max}$) (Ozcan *et al*. 2010).

4.4. Building a balancing solution

Once the model sequences are generated and all required sets and parameters are initialised (including the determination of the position in which the assignment process will begin) in accordance with the pseudo-code given in Figure 3, the first model sequence in the set of PMSC is selected. The line balancing process is performed following the steps given in Figure 4. As seen, the main principle is based on determining the available tasks for the current position and assigning the one with the highest ω_{hi}^{PI} value. The list of *available tasks* on L_h (where $h = 1, \dots, H$) is symbolised with ATL_h. When determining available tasks, the tasks in the unassigned task lists for Line-I and Line-II (UTL_1 and UTL_2) are checked one-by-one. If a task being checked (called a candidate task) is from the front of the precedence relationships graph, either the task should have no predecessors or all of its predecessors (if any) must have been assigned and completed. However, if the task is from the back of the precedence relationships graph, either the task should have no successors or all of its successors (if any) must have been assigned and completed. Also, the remaining capacity in the current workstation should be large enough to perform the task. To determine whether there is enough capacity, there are three important components which need to be considered: (i) the workload of the current workstation, (ii) the earliest starting time of the candidate task, and (iii) the deserved task time (DT_{hi}) of the candidate task. The DT_{hi} is a newly introduced term for MPUL system, because, at the beginning of the balancing process (when the final balance and so the total number of workstations are uncertain), it is not known which model will appear at the back of the line (*i.e.*, Line-I back and Line-II back). Due to this uncertainty, the maximum processing time of the candidate task among different models being produced on the same line will be considered when assigning a task from the back of the line, $DT_{hi} \leftarrow \max_{m_{hj} \in M_h} \{pt_{hji}\}$.

Algorithm: Building a balancing solution

```
Start
$q \leftarrow 1$
While ($UTL_1 \neq \emptyset$ or $UTL_2 \neq \emptyset$){
  Go to workstation $WS_q^z$
  Update $ATL_1$ and $ATL_2$
  While ($ATL_1 \neq \emptyset$ or $ATL_2 \neq \emptyset$){
    Select and assign a task ($t_{hi}$) from $ATL_1$ or $ATL_2$ based on task selection
    policy (see Section 4.4)
    Remove $t_{hi}$ from $UTL_h$
    For ($int\ \varphi = 1; \varphi \leq \phi; \varphi + +$){
      $WT_q^{z\varphi} \leftarrow DT_{hi} + max\{WT_q^{z\varphi}, EST_{hi}^{\varphi}\}$
    } end
    If (all tasks in $P_{hi}$ are assigned){
      For (int $s1 = 1; s1 \leq S_{hi}.size; s1 + +$){
        $EST_{hi}^{\varphi} \leftarrow WT_q^{z\varphi}$ where $i \in S_{hi}$ and $\varphi = 1,2, \dots \phi$
      } end
    } else if (all tasks in $S_{hi}$ are assigned){
      For (int $s2 = 1; s2 \leq P_{hi}.size; s2 + +$){
        $EST_{hi}^{\varphi} \leftarrow WT_q^{z\varphi}$ where $i \in P_{hi}$ and $\varphi = 1,2, \dots \phi$
      } end
    } end
    If ($isFrontArea = true$){
      $z = randomIntegerBetween[1,2]$
    } else if ($isFrontArea = false$){
      $z = randomIntegerBetween[3,4]$
    } end
    Update $ATL_1$ and $ATL_2$
  } end
  For ($int\ \varphi = 1; \varphi \leq \phi; \varphi + +$){
    $EST_{hi}^{\varphi} \leftarrow 0$ where $h = 1, 2;\ i = 1,2, \dots T_h;\ \varphi = 1,2, \dots \phi$
  } end
  If ($isFrontArea = true$){
    $isFrontArea = false$
  } else if ($isFrontArea = false$){
    $q \leftarrow q + 1$
    $z = randomIntegerBetween[1,2]$
  } end
} end
```

Figure 4. The pseudo-code of building a balancing solution.

However, as it is known which model will appear on the front of the lines, the task processing time of the relevant model will be used when assigning a task from the front of the precedence relationships graph, $DT_{hi} \leftarrow pt_{hji}$. Therefore, the capacity constraint will be satisfied by task t_{hji} if the following condition is fulfilled: $C \geq DT_{hi} + max\{WT_q^{z\varphi}, EST_{hi}^{\varphi}\}$ for all production cycles. The term EST_{hi}^{φ} corresponds to the earliest starting time of task t_{hi} in production cycle φ and is determined by the latest completion time of its predecessors (if the task is from the front of the line) or successors (if the task is from the back of the line).

When selecting and assigning tasks to workstations, the task which has the best priority index (ω_{hi}^{PI}) value is preferred. However, the best ω_{hi}^{PI} depends on the position of the available task. If the task is from the front of the precedence relationships graph (regardless of Line-I or Line-II), then the best priority index corresponds to the maximum ω_{hi}^{PI}. On the other hand, if the task is from the back of the precedence relationships graph (regardless of Line-I or Line-II), the best priority index corresponds to the minimum ω_{hi}^{PI}.

To increase the diversity of the solutions obtained and scan the search space more effectively, randomness is allowed when selecting tasks. Randomness is a commonly applied technique in assembly line balancing approaches, especially in priority rule-based methods, and as mentioned by Otto and Otto (2014), the quality of the solutions obtained by such methods can be improved by "*applying several passes of this priority rule with some kind of random influence*". For this aim, a random number, $r_1 \in (0,1)$, is determined by the algorithm. If $r_1 \leq RI$, where $RI \in [0,1]$ is the randomness index and determined by the user at the beginning, a random task is selected among the available ones. If $r_1 > RI$, then the task which has the best ω_{hi}^{PI} value among the available tasks is selected. Another randomly determined number, $r_2 \in (0,1)$, is used to decide on the list from which the task will be picked, *i.e.*, ATL_1 or ATL_2. If $r_2 \leq 0.5$, the task is tried to be selected from ATL_1. If $r_2 > 0.5$, the algorithm tries to assign a task from ATL_2. If any of the lists is empty, then the task is picked from the other list. When a task (t_{hi}) is assigned to a workstation (WS_q^z) on line (L_h), its workload time is increased by $DT_{hi} + max\{WT_q^{z\varphi}, EST_{hi}^{\varphi}\}$ for all production cycles, $WT_q^{z\varphi} \leftarrow DT_{hi} + max\{WT_q^{z\varphi}, EST_{hi}^{\varphi}\}$ where $\varphi = 1, \ldots, \phi$.

DT_{hi} is also used when updating EST_{hi}^{φ} values of tasks. If the task is assigned from the front of the precedence relationships graph, then EST_{hi}^{φ} values of its successors will be set to $DT_{hi} + max\{WT_q^{z\varphi}, EST_{hi}^{\varphi}\}$. On the contrary, if the task is assigned from the back of the precedence relationships graph, then EST_{hi}^{φ} values of its predecessors will be set to $DT_{hi} + max\{WT_q^{z\varphi}, EST_{hi}^{\varphi}\}$.

5. An explanatory example

A small-scale numerical example is given here to show the sophisticated structure of the MPUL balancing problem and the running principle of MPUH, explicitly.

5.1. Problem data

Let us consider two U-shaped lines, Line-I and Line-II, located in parallel to each other, as in the example given in Section 3. Two different sets of models are produced on each of the parallel lines, *i.e.*, models A and B on Line-I, and models C and D on Line-II. A common precedence relationships diagram is built among tasks belonging to the models produced on each of the parallel lines. The processing times of tasks and the precedence relationships are given in Table 1. The *IM* column presents the immediate successor(s) of the corresponding task.

The model demands are assumed $D_{1A} = 24$, $D_{1B} = 72$, $D_{2C} = 48$ and $D_{2D} = 48$ for a planning horizon of 960 time-units. Thus, the cycle times of the lines are simply calculated as $C_1 = C_2 = 10$ time-units. When the MPS principle explained in Section 4 is applied, the model mixes on Line-I and Line-II are obtained as $d_1 = (1,3)$ and $d_2 = (1,1)$, respectively, which means $S_1 = 4$ and $S_2 = 2$. This leads to a maximum of $\phi = MS_{max} = 8$ different production cycles on the lines.

Table 1. Input data for the numerical example (task times in time units).

Task No	Line-I									Line-II							
	IM	Time		Calculated Parameters						IM	Time		Calculated Parameters				
		A	B	wpt_{hi}	PW_{hi}	ω_{hi}^{PW}	ω_{hi}^{NS}	ω_{hi}^{PI}			C	D	wpt_{hi}	PW_{hi}	ω_{hi}^{PW}	ω_{hi}^{NS}	ω_{hi}^{PI}
1	3	5	5	5.00	26.00	15	15	225		3	4	3	3.50	22.50	13	13	169
2	5	4	4	4.00	22.25	13	13	169		4,5,6	2	5	3.50	37.00	14	14	196
3	4,7	1	2	1.75	21.00	12	14	168		7	2	2	2.00	19.00	11	12	132
4	7	2	1	1.25	19.25	10	12	120		7	3	3	3.00	20.00	12	11	132
5	8	3	2	2.25	18.25	9	11	99		9	3	3	3.00	18.00	10	10	100
6	9	4	3	3.25	24.25	14	10	140		8	5	4	4.50	13.5	7	8	56
7	10,12	3	4	3.75	18.00	8	9	72		9	2	2	2.00	17.00	9	9	81
8	10	4	5	4.75	16.00	7	8	56		10	5	5	5.00	9.00	6	6	36
9	11	7	6	6.25	21.00	11	7	77		11,12	1	1	1.00	15.00	8	7	56
10	13	2	0	0.50	11.25	5	6	30		13	1	1	1.00	4.00	3	5	15
11	13	4	4	4.00	14.75	6	5	30		13	3	3	3.00	6.00	4	4	16
12	14	3	3	3.00	9.75	3	4	12		14	8	8	8.00	9.00	5	3	15
13	14	4	4	4.00	10.75	4	3	12		14	2	2	2.00	3.0	2	2	4
14	15	5	5	5.00	6.75	2	2	4		-	0	2	1.00	1.0	1	1	1
15	-	1	2	1.75	1.75	1	1	1		-	-	-	-	-	-	-	-

5.2. *Steps of MPUH*

All possible model sequencing combinations are generated (as shown in Appendix A) and each combination is imported to the balancing procedure. The line balancing solutions are generated for the model sequences using the procedure explained in Section 4.4 and the solution which gives the best OBJ value is selected. To exemplify, the solution building principle of MPUH for the model sequencing combination $MS_1 = \{BBAB\}$ and $MS_2 = \{CD\}$ is given in Table 2. Note that only the first 14 steps are provided due to page limit. The task selected from the list of available tasks at each step, and the completion time of the selected task for each production cycle can be seen from the table. In the *Completion Time* column, the models that will appear in workstations located at the front of the line are also given in brackets. As the specific models that will appear in workstations located at the back of the line are unknown, they are marked with '[M]'. In this situation, the maximum task processing time among all models on the same line is considered (as explained in Section 4).

Table 2. The task selection procedure.

Step	z	q	Available Tasks		Selected Task	Completion Time (in time-units)			
			ATL_1	ATL_2		$\varphi = 1$	$\varphi = 2$	$\varphi = 3$	$\varphi = 4$
1	2	1	1,2,6	1,2,14	$ATL_2\{2\}$	2 [C]	5 [D]	2 [C]	5 [D]
2	1	1	1,2,6	1,4,5,6	$ATL_1\{1\}$	5 [B]	5 [B]	5 [A]	5 [B]
3	2	1	2,3,6	1,4,5,6,14	$ATL_2\{1\}$	6 [C]	8 [D]	6 [C]	8 [D]
4	2	1	3	3,14	$ATL_2\{3\}$	8 [C]	10 [D]	8 [C]	10 [D]
5	1	1	2,3,6	4,5,6	$ATL_1\{3\}$	7 [B]	7 [B]	6 [A]	7 [B]
6	1	1	4,6	4,5	$ATL_1\{6\}$	10 [B]	10 [B]	10 [A]	10 [B]
7	4	1	15	14	$ATL_1\{15\}$	2 [M]	2 [M]	2 [M]	2 [M]
8	3	1	14	4,5,6,14	$ATL_2\{14\}$	2 [M]	2 [M]	2 [M]	2 [M]
9	4	1	14	12,13	$ATL_1\{14\}$	7 [M]	7 [M]	7 [M]	7 [M]
10	4	1	12	13	$ATL_1\{12\}$	10 [M]	10 [M]	10 [M]	10 [M]
11	3	1	-	4,5,6,12,13	$ATL_2\{12\}$	10 [M]	10 [M]	10 [M]	10 [M]
12	2	2	2,4,9	4,5,6,13	$ATL_2\{4\}$	3 [D]	3 [C]	3 [D]	3 [C]
13	1	2	2,4,9	5,6,7	$ATL_1\{2\}$	4 [B]	4 [B]	4 [B]	4 [A]
14	1	2	4,5	5,6,7	$ATL_2\{5\}$	8 [D]	8 [C]	8 [D]	8 [C]

As seen from Table 2, the assignment procedure starts from the front area of Line-II ($z = 2$, $q = 1$). Amongst the available tasks from Line-I and Line-II, the algorithm selects task $2 \in ATL_2$, which has the maximum ω_{hi}^{PI} value among the tasks on Line-II, to assign in workstation WS_1^2. The completion times in this workstation are updated as 2, 5, 2 and 5 for production cycles 1, 2, 3 and 4, respectively. The earliest starting times of the successors of task 2 on Line-II are also updated for each production cycle. In the next step (step 2), z is assigned 1 ($z = 1$) and task $1 \in ATL_1$ is assigned to workstation WS_1^1. The completion times and earliest starting times are updated and this cycle continues until there is no available task for this queue. The algorithm proceeds to the next queue in step 12. In step 14, task $5 \in ATL_2$ is assigned to the multi-line station utilised on Line-I (WS_2^1). The operator allocated in WS_2^1 operates on both lines, *i.e.*, he/she performs operations for models A and B being assembled on Line-I and models C and D being assembled on Line-II. As seen from the table, no crossover station is utilised in the first 14 steps provided.

The algorithm consecutively determines which product model will appear on the front of the line as the new queue is utilised based on the model sequence given ($MS_1 = \{BBAB\}$ and $MS_2 = \{CD\}$). However, the final configuration of product models, including the back of the lines, is achieved once all the tasks have been assigned and so the total number of workstations has been determined. In such an environment where it is not known which model will exist in the back of the line, it is not easy to assign tasks to the workstations by ensuring that capacity constraint is satisfied while pursuing to obtain an efficient line balance.

5.3. *Final solution*

Figure 5 gives the best balancing solution obtained after MPUH algorithm was run for 50 iterations ($maxBalancingItNumber = 50$). Note that many different solutions may be obtained during these iterations thanks to the stochastic task assignment procedure of MPUH. The procedure (and the sample task assignments) given in the previous subsection can be followed when building a sample balancing solution in any iteration. However, the best solution obtained eventually (given in Figure 5) is completely different from this sample balancing solution, as expected. As seen from the figure, 10 workstations are constructed; of which four are multi-line stations (*i.e.*, WS_1^1, WS_2^2, WS_2^3 and WS_3^3), one is crossover station (*i.e.*, WS_1^2) and five are regular stations (*i.e.*, WS_1^3, WS_1^4, WS_2^1, WS_3^1 and WS_3^2). The multi-line stations and crossover stations will be influenced from the model changes during production cycles. This is exhibited in Table 3, in which the final workload times of workstations are reported across four production cycles. In accordance with the number of workstations obtained, possible production cycles are also provided in Table 3. As can be seen from the table, the model type that will exist in the workstations located in the back of the line tightly depends on the total number of workstations utilised across the lines.

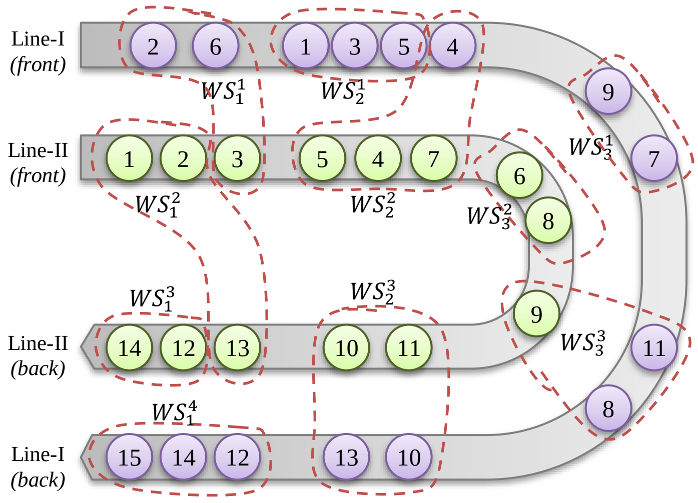


Figure 5. The best balancing solution obtained after 50 iterations.

Table 3. The workload times of workstations across production cycles.

#	Station	Production Cycle	$\varphi = 1$	$\varphi = 2$	$\varphi = 3$	$\varphi = 4$
1	WS_1^1	Assigned: Model[Tasks]	B[2,6] C[3]	B[2,6] D[3]	A[2,6] C[3]	B[2,6] D[3]
		Workload Time	9	9	10	9
2	WS_1^2	Assigned: Model[Tasks]	C[1,2] D[13]	D[1,2] C[13]	C[1,2] D[13]	D[1,2] C[13]
		Workload Time	8	10	8	10
3	WS_1^3	Assigned: Model[Tasks]	D[14,12]	C[14,12]	D[14,12]	C[14,12]
		Workload Time	10	8	10	8
4	WS_1^4	Assigned: Model[Tasks]	B[15,14,12]	B[15,14,12]	B[15,14,12]	A[15,14,12]
		Workload Time	10	10	10	9

5	WS_2^1	Assigned: Model[Tasks]	B[1,3,5]	B[1,3,5]	B[1,3,5]	A[1,3,5]
		Workload Time	9	9	9	9
6	WS_2^2	Assigned: Model[Tasks]	D[5,4,7] B[4]	C[5,4,7] B[4]	D[5,4,7] B[4]	C[5,4,7] A[4]
		Workload Time	9	9	9	10
7	WS_2^3	Assigned: Model[Tasks]	C[10,11] B[13,10]	D[10,11] B[13,10]	C[10,11] A[13,10]	D[10,11] B[13,10]
		Workload Time	8	8	10	8
8	WS_3^1	Assigned: Model[Tasks]	A[9,7]	B[9,7]	B[9,7]	B[9,7]
		Workload Time	10	10	10	10
9	WS_3^2	Assigned: Model[Tasks]	C[6,8]	D[6,8]	C[6,8]	D[6,8]
		Workload Time	10	9	10	9
10	WS_3^3	Assigned: Model[Tasks]	D[9] B[11,8]	C[9] A[11,8]	D[9] B[11,8]	C[9] B[11,8]
		Workload Time	10	9	10	10

To give an example regarding the workload variations in multi-line stations, let us consider workstation WS_1^1 given in Figure 5. In the first production cycle, model B and model C will be under operation on Line-I and Line-II, respectively. The operator working in this workstation will complete tasks 2 and 6 on model B on Line-I and he/she will then perform task 3 on model C belonging to Line-II. Thus, the workload of this workstation will be 9 time units in production cycle 1. In the next production cycle, the workload of WS_1^1 will remain the same although model D will appear on Line-II. This is because both models, C and D, require the same amount of time for task 3. However, this situation will change in the next production cycle with the launch of models A and C. The workload time of WS_1^1 will increase to 10 time units in production cycle 3. In another multi-line station, WS_2^2, the model combinations on Line-I and Line-II and so workload time of the workstation will change four times; *i.e.*, workload time will be 9, 9, 9 and 10 time units in production cycles 1, 2, 3 and 4, respectively. Similarly, there will be changes in the model combinations in the crossover station, WS_1^2. In production cycles 1 and 3, models C and D will exist on the front and back of Line-II, respectively, by requiring 8 time units of workload time. In production cycles 2 and 4, models D and C will appear and fill up the capacity of WS_1^2 with 10 time units.

If these two lines were balanced independently, a total of 11 workstations (*i.e.*, 6 workstations for Line-I and 5 workstations for Line-II) would be needed to perform tasks for all models. Therefore, it is clear that the proposed MPUL design helps save one workstation for this particular case.

6. Experimental tests

As there is no suitable comparable result reported in the literature, to show the practical benefits of the proposed MPUL system, standard test problems have been derived from the literature and solved using the proposed MPUH algorithm under two different conditions: *independent balancing (IB)* and *MPUL*. The MPUH algorithm has been coded in Java SE 7u4 environment and run on a PC with 3.1GHz Intel Core™ i5-2400 CPU. A total of 51 test cases have been formed using the test problems, *i.e.*, one test problem on Line-I and another test problem on Line-II. In *IB*, Line-I and Line-II have been considered as two separate MMUL systems and the two lines were balanced separately such that no multi-line stations were allowed. In *MPUL* condition, a MPUL system has been established where the two lines have been located in parallel to each other, as proposed. Thus, it was aimed to measure the benefit of the proposed MPUL system against existing MMUL system. Based on some preliminary tests, the randomness index parameter is assumed $RI = 0.5$ to obtain more diversified solutions, as contextualised in Section 4. The algorithm was run 100, 200, 300, 400 and 500 iterations for test cases 1-6, 7-15, 16-24, 25-38 and 39-51, respectively, and the solution which gives the minimum number of workstations (*NS*) value was taken for each test case.

Table 4 presents both the design and the results of the experimental tests. The problems considered on each line are given in Line-I and Line-II columns for each test case. The cycle times of the lines and the minimum part sets (see *MPS* column) belonging to the models produced on particular lines are presented in the table. The table comparatively reports the results of test cases solved as well. The *IB* column reports the sum of the NS values for Line-I and Line-II when the lines were balanced independently. In the *MPUL* column, the number of queues (*NQ)* and the NS values of the obtained solutions are reported for each test case. The *Diff* column denotes the relative difference in the NS values between the two solution strategies and is calculated as $Diff = (IB_{NS} - MPUL_{NS})/IB_{NS}$. The last column reports the CPU time (in seconds) consumed by MPUH algorithm for solving the corresponding test case.

Table 4. The design and results of experimental tests.

Test Case #	**Design**					**Results**				
	Line-I		**Line-II**			**IB**	**MPUL**		**Diff**	**CPU (sec.)**
	Problem	**MPS (A-B)**	**Problem**	**MPS (C-D)**	**Cycle Time**	**NS (Line I + Line II)**	**NQ**	**NS**		
1	4-task	1-1	4-task	1-1	6	6	2	5	**0.17**	78
2	4-task	1-2	4-task	2-1	7	4	1	4	0.00	35
3	4-task	1-2	4-task	1-2	5	6	2	6	0.00	110
4	5-task	1-1	5-task	1-1	4	8	2	6	**0.25**	52
5	5-task	1-2	5-task	1-2	5	6	1	4	**0.33**	109
6	5-task	1-2	5-task	2-1	6	4	1	4	0.00	114
7	12-task	1-1	12-task	1-1	9	10	3	10	0.00	49
8	12-task	1-2	12-task	1-2	11	8	2	8	0.00	227
9	12-task	1-2	12-task	2-1	13	8	2	7	**0.13**	109
10	15-task	1-2	15-task	2-1	9	14	4	13	**0.07**	370
11	15-task	2-1	15-task	2-1	10	12	3	12	0.00	225
12	15-task	1-1	15-task	1-1	12	10	3	10	0.00	119
13	16-task	2-1	16-task	1-2	14	20	5	18	**0.10**	411
14	16-task	1-1	16-task	1-1	16	18	5	17	**0.06**	160
15	16-task	1-2	16-task	1-2	18	14	4	14	0.00	356
16	25-task	1-1	25-task	1-1	11	28	7	27	**0.04**	115
17	25-task	1-2	25-task	1-2	13	22	6	21	**0.05**	305
18	25-task	2-1	25-task	1-2	14	20	5	20	0.00	307
19	27-task	2-1	27-task	2-1	7	28	8	28	0.00	368
20	27-task	1-1	27-task	1-1	8	24	6	24	0.00	198

21	27-task	1-2	27-task	2-1	10	20	5	18	**0.10**	581
22	30-task	1-1	30-task	1-1	10	20	5	19	**0.05**	253
23	30-task	1-2	30-task	2-1	11	18	5	18	0.00	652
24	30-task	1-2	30-task	1-2	9	22	6	22	0.00	951
25	45-task	1-1	45-task	1-1	55	22	6	21	**0.05**	494
26	45-task	1-1	45-task	1-1	61	20	5	19	**0.05**	460
27	45-task	1-2	45-task	2-1	58	20	5	20	0.00	1641
28	45-task	1-2	45-task	2-1	60	20	5	19	**0.05**	1620
29	45-task	1-2	45-task	2-1	63	20	5	18	**0.10**	1484
30	45-task	1-2	45-task	1-2	62	20	5	19	**0.05**	1705
31	45-task	1-2	45-task	1-2	59	20	5	19	**0.05**	1609
32	70-task	1-1	70-task	1-1	160	46	12	46	0.00	719
33	70-task	1-1	70-task	1-1	169	46	12	45	**0.02**	628
34	70-task	1-1	70-task	1-1	184	40	10	40	0.00	801
35	70-task	1-2	70-task	2-1	168	46	12	45	**0.02**	2334
36	70-task	1-2	70-task	2-1	172	44	12	44	0.00	2412
37	70-task	1-2	70-task	1-2	185	40	10	40	0.00	3282
38	70-task	1-2	70-task	1-2	176	44	11	43	**0.02**	3106
39	83-task	1-1	83-task	1-1	3700	44	11	44	0.00	948
40	83-task	1-1	83-task	1-1	3780	44	11	43	**0.02**	969
41	83-task	1-1	83-task	1-1	4350	38	10	37	**0.03**	824
42	83-task	1-2	83-task	2-1	4100	40	10	39	**0.03**	2012
43	83-task	1-2	83-task	1-2	4250	38	10	38	0.00	2103
44	83-task	1-2	83-task	1-2	4360	38	10	37	**0.03**	1989
45	83-task	1-2	83-task	1-2	4600	36	9	35	**0.03**	2237
46	111-task	1-1	111-task	1-1	6050	54	14	54	0.00	1862
47	111-task	1-1	111-task	1-1	5440	50	13	50	0.00	2006
48	111-task	1-2	111-task	2-1	6100	54	14	54	0.00	5162
49	111-task	1-2	111-task	2-1	6250	52	13	52	0.00	5460
50	111-task	1-2	111-task	1-2	6500	50	13	49	**0.02**	5093
51	111-task	1-2	111-task	1-2	6650	48	13	48	0.00	4985

As can be seen from the results, the proposed MPUL design helps reduce the number of workstations in more than half of the total number of cases solved. The largest difference (0.33 or 33%) was observed for test case 5 for which the IB and MPUL solutions were obtained 6 and 4, respectively. Thus, MPUL system helped save two workstations for this particular case. The second (0.25) and the third (0.17) best improvements have been recorded for test cases 4 and 1, respectively. For test case 4, the proposed MPUL design requires 6 workstations while 8 workstations would be needed when the lines were balanced independently. Hence, a 25% improvement has been achieved. In test case 1, the MPUL design helped save one workstation with a 17 percent improvement over the IB condition. The algorithm also helps two workstations in test cases 13, 21 and 29 but $Diff$ values are not as much high as in test cases 4 and 5. This is due to increase in NS. In addition to test case 1, the proposed design helped save one workstation for 20 other test cases (see, for example, test cases 9, 10, 14, 16, 17, etc.). For the remaining 25 test cases, MPUL heuristic requires the same number of workstations with IB solution. As seen from the results, the CPU time reasonably increases in parallel to the increase in problem size and the number of iterations. However, regardless from this, both the proposed system and the algorithm perform quite well.

A paired sample t-test was conducted to statistically analyse the experimental test results and determine whether the proposed MPUL system makes a significant improvement over IB. For this aim, the NS values obtained by the two solution strategies were used as input data. The hypotheses stated at the 95% confidence interval ($\alpha = 0.05$ significance level) are as follows:

H_0: There is no statistically significant mean difference in NS values obtained by IB and MPUL ($\mu_{MPUL} = \mu_{IB}$).

H_1: The mean difference in NS values obtained by IB and MPUL are statistically significant ($\mu_{MPUL} \neq \mu_{IB}$).

The data was analysed using Minitab® 16 statistical software package. The test results presented in Table 5 indicated that there was a significant difference in the mean NS values of MPUL (26.53 ± 15.70) and IB (27.14 ± 15.57); $t(51) = -6.52, p < 0.001$. Specifically, these results confirmed that the proposed MPUL system helps minimise the total number of workstations in comparison with independently balanced MMULs. This has clearly demonstrated that the parallelisation of MMULs provides promising advantages which need further investigation in future studies.

Table 5. The paired sample t-test results.

	Sample Size	Mean	Standard Deviation	SE Mean
MPUL	51	26.53	15.70	2.20
IB	51	27.14	15.57	2.18
Difference	51	-0.6078	0.6657	0.0932
95% CI for mean difference	: (-0.7951; -0.4206)			
t-value	: -6.52			
p-value	: < 0.001			

7. Discussion and conclusions

The main aim of this study was to introduce a flexible as well as efficient manufacturing system design which may replace traditional U-shaped lines, in future. For this aim, two MMULs were located in parallel to each other and a unique line configuration was obtained. So that, tasks belonging to the models being produced in an inter-mixed sequence on both of the lines can be performed at multi-line stations utilised in between two adjacent lines. Also, as emphasised in details in this paper, the proposed line system combines the advantages of its individual sub-configurations (*i.e.*, mixed-model lines, parallel lines and U-shaped lines) which have already been discussed widely in the literature. The MPUL balancing problem was defined and discussed by taking model sequences into consideration. It was illustratively shown that the feasibility and so the quality of a balancing solution obtained is affected by the sequences of models being assembled on the lines. The dynamism of model combinations in multi-line stations and crossover stations and their effects on the workload times of these workstations have also been exhibited clearly.

In addition to the benefits of the proposed system, such a complex system design brings some difficulties which require sophisticated solution procedures for building efficient balancing techniques. As an advantage of the U-shaped lines, the balancing process starts from both front and back areas of the lines and this provides advantages over straight lines. However, as the number of workstations that need to be utilised is uncertain at the beginning of the balancing process, it is not known which models will appear in the workstations located at the back of the line. In this environment, a heuristic algorithm was proposed to generate feasible model sequences considering minimum part sets and build balancing solutions considering these model sequences.

An explanatory example has been provided to show the running principle of MPUH and the implementation of the MPUL idea. Experimental tests have also been conducted to examine the superiority of the MPUL design against independently balanced mixed-model lines. The experimental test results and their statistical analysis have confirmed the advantages of the proposed design in minimising the total number of workstations. As a possible industrial application of the research, the techniques implemented in this study can easily be adopted by practitioners, *e.g.*, line managers in car manufacturing plants, as explained in the paper. By this way, companies will be able to produce customised products designed based on their customers' requests while reducing the need for workforce.

8. Future research directions

The outcome of this paper made it clear that the MPUL design has a promising potential to yield more efficient line systems. Therefore, one can consider the implementation of the MPUL idea for real-world cases in future studies. Developing an improved procedure to detect which model will be produced on the back of the line at a specific workstation should be considered. Also, other heuristics/metaheuristics can be developed to solve the MPUL balancing problem and their performances can be compared with that of the MPUH. In doing so, different model sequencing procedures may also be integrated into the model so that the model sequencing and the line balancing problems can be solved simultaneously.

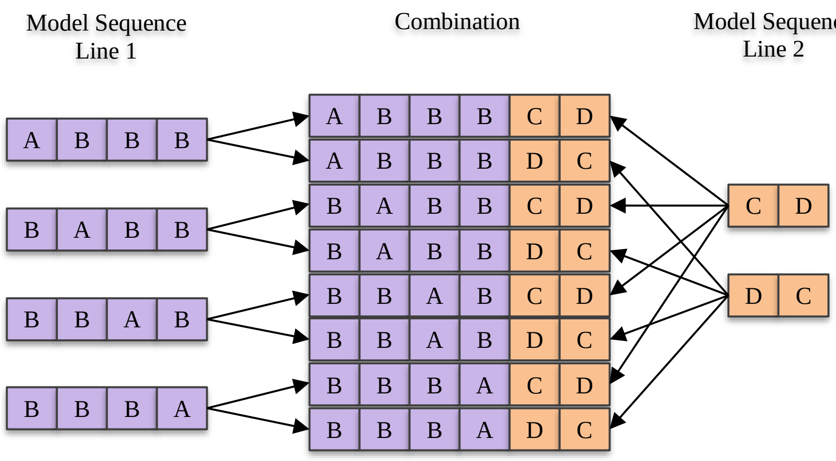
As the efficiency of a production system is strongly related to its ergonomic conditions, considering ergonomic issues when designing assembly lines is gaining even more importance in recent studies. For this reason, the current work can also be extended considering the physical strains, psychological strains, working postures and skill levels of the operators. Furthermore, some slackness may also be allowed for operators during their working period. This will have significant effects on the performance of the operators because zero idle times may result in more mistakes which yield reduced efficiency. Finally, standing on hard surfaces for a prolonged time can produce circulatory problems in the legs and feet of operators. Therefore, workplaces should allow enough room for operator movements.

Acknowledgment

The authors acknowledge the valuable comments and suggestions of anonymous referees and the editorial team of *International Journal of Production Research* which helped improve the presentation and quality of the manuscript.

Appendix A

The schematic representation of generating possible model sequencing combinations.



References

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

Dong, Jietao, Linxuan Zhang, Tianyuan Xiao, and Huachao Mao. 2014. "Balancing and sequencing of stochastic mixed-model assembly U-lines to minimise the expectation of work overload time." *International Journal of Production Research*:1-20.

Erel, E., I. Sabuncuoglu, and B. A. Aksu. 2001. "Balancing of U-type assembly systems using simulated annealing." *International Journal of Production Research* 39 (13):3003-3015.

Ford-Motor-Company. "100 Years of the Moving Assembly Line, Online http://corporate.ford.com/innovation/100-years-moving-assembly-line.html, Accessed 25 October 2015."

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

Gökçen, Hadi, Kürşat Ağpak, Cevriye Gencer, and Emel Kizilkaya. 2005. "A shortest route formulation of simple U-type assembly line balancing problem." *Applied Mathematical Modelling* 29 (4):373-380.

Hamzadayi, Alper, and Gokalp Yildiz. 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.

Hamzadayi, Alper, and Gokalp Yildiz. 2013. "A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines." *Computers & Industrial Engineering* 66 (4):1070-1084.

Helgeson, W. B., and D. P. Birnie. 1961. "Assembly line balancing using the ranked positional weight technique." *Journal of Industrial Engineering* 12 (6):394–398.

Hwang, Rea Kook, Hiroshi Katayama, and Mitsuo Gen. 2008. "U-shaped assembly line balancing problem with genetic algorithm." *International Journal of Production Research* 46 (16):4637-4649.

Kara, Yakup. 2008. "Line balancing and model sequencing to reduce work overload in mixed-model U-line production environments." *Engineering Optimization* 40 (7):669-684.

Kara, Yakup, Ugur Ozcan, and Ahmet Peker. 2007. "Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives." *Applied Mathematics and Computation* 184 (2):566-588.

Kara, Yakup, and Mahmut Tekin. 2009. "A mixed integer linear programming formulation for optimal balancing of mixed-model U-lines." *International Journal of Production Research* 47 (15):4201-4233.

Kazemi, Seyed Mahmood, Reza Ghodsi, Masoud Rabbani, and Reza Tavakkoli-Moghaddam. 2011. "A novel two-stage genetic algorithm for a mixed-model U-line balancing problem with duplicated tasks." *The International Journal of Advanced Manufacturing Technology* 55 (9-12):1111-1122.

Kim, Yeo Keun, Jae Yun Kim, and Yeongho Kim. 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.

Kucukkoc, I., A. D. Karaoglan, and R. Yaman. 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., and D. Z. Zhang. 2014a. "Mathematical Model and Agent Based Solution Approach for the Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines." *International Journal of Production Economics* 158, :314-333.

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

Kucukkoc, I., and D. Z. Zhang. 2015a. "Balancing of parallel U-shaped assembly lines." *Computers and Operations Research* 64:233–244, doi: http://dx.doi.org/10.1016/j.cor.2015.05.014.

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

Kucukkoc, I., and D. Z. Zhang. 2015c. "A mathematical model and genetic algorithm-based approach for parallel two-sided assembly line balancing problem." *Production Planning and Control* 26 (11):874–894, doi: http://dx.doi.org/10.1080/09537287.2014.994685.

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

Kucukkoc, I., and D.Z. Zhang. 2014c. "An Agent Based Ant Colony Optimisation Approach for Mixed-Model Parallel Two-Sided Assembly Line Balancing Problem." In *Pre-Prints of the Eighteenth International Working Seminar on Production Economics*, edited by R. W. Grubbström and H. H. Hinterhuber, 313-328. Innsbruck, Austria: Congress Innsbruck.

Kucukkoc, Ibrahim, Kadir Buyukozkan, Sule Itir Satoglu, and David Z. Zhang. 2015. "A mathematical model and artificial bee colony algorithm for the lexicographic bottleneck mixed-model assembly line balancing problem." *Journal of Intelligent Manufacturing*.

Kucukkoc, Ibrahim, and David Z. Zhang. 2015e. "Coping with Model Variations on Parallel U-shaped Assembly Line Configurations." *IFAC-PapersOnLine* 48 (3):2030-2035.

Li, J., J. Gao, and L. Sun. 2012. "Sequencing minimum product sets on mixed-model U-lines to minimise work overload." *International Journal of Production Research* 50 (18):4977-4993.

Lian, K., C. Zhang, L. Gao, and X. Shao. 2012. "A modified colonial competitive algorithm for the mixed-model U-line balancing and sequencing problem." *International Journal of Production Research* 50 (18):5117-5131.

Manavizadeh, N., N. S. Hosseini, M. Rabbani, and F. Jolai. 2013. "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., M. Rabbani, and F. Radmehr. 2015. "A new multi-objective approach in order to balancing and sequencing U-shaped mixed model assembly line problem: a proposed heuristic algorithm." *International Journal of Advanced Manufacturing Technology* 79 (1-4):415-425.

Miltenburg, G. J., and J. Wijngaard. 1994. "The U-Line Line Balancing Problem." *Management Science* 40 (10):1378-1388.

Miltenburg, J. 2001. "U-shaped production lines: A review of theory and practice." *International Journal of Production Economics* 70 (3):201-214.

Otto, Alena, and Christian Otto. 2014. "How to design effective priority rules: Example of simple assembly line balancing." *Computers & Industrial Engineering* 69:43-52.

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

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

Özcan, Uğur, Talip Kellegöz, and Bilal Toklu. 2011. "A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem." *International Journal of Production Research* 49 (6):1605-1626.

Rabbani, M., M. Moghaddam, and N. Manavizadeh. 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.

Rabbani, Masoud, Seyed Mahmood Kazemi, and Neda Manavizadeh. 2012. "Mixed model U-line balancing type-1 problem: A new approach." *Journal of Manufacturing Systems* 31 (2):131-138.

Sabuncuoglu, Ihsan, Erdal Erel, and Arda Alp. 2009. "Ant colony optimization for the single model U-type assembly line balancing problem." *International Journal of Production Economics* 120 (2):287-300.

Salveson, M. E. 1955. "The assembly line balancing problem." *Journal of Industrial Engineering* 6 (3):18-25.

Scholl, A., and R. Klein. 1999. "ULINO: Optimally balancing U-shaped JIT assembly lines." *International Journal of Production Research* 37 (4):721-736.

Sparling, D., and J. Miltenburg. 1998. "The mixed-model U-line balancing problem." *International Journal of Production Research* 36 (2):485-501.

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.

Urban, Timothy L. 1998. "Note. Optimal Balancing of U-Shaped Assembly Lines." *Management Science* 44 (5):738-741.

Urban, Timothy L., and Wen-Chyuan Chiang. 2006. "An optimal piecewise-linear program for the U-line balancing problem with stochastic task times." *European Journal of Operational Research* 168 (3):771-782.

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

Zhang, D. Z., and I. Kucukkoc. 2013. "Balancing Mixed-Model Parallel Two-Sided Assembly Lines." In *Proceedings of the International Conference on Industrial Engineering and Systems Management (IEEE-IESM'2013)*, edited by L. Amodeo, A. Dolgui and F. Yalaoui, 391-401. Rabat, Morocco.