International Journal of Production Research

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Balancing parallel two-sided assembly lines

Uğur Özcan [a] , Hadi Gökçen [b] & Bilal Toklu [b]

[a] Department of Industrial Engineering , Selçuk University , 42075 Selçuklu, Konya, Turkey

[b] Department of Industrial Engineering , Gazi University , 06570 Maltepe, Ankara, Turkey

Published online: 04 Aug 2009.

To cite this article: Uğur Özcan , Hadi Gökçen & Bilal Toklu (2010) Balancing parallel two-sided assembly lines, International Journal of Production Research, 48:16, 4767-4784, DOI: 10.1080/00207540903074991

To link to this article: http://dx.doi.org/10.1080/00207540903074991

International Journal of Production Research
Vol. 48, No. 16, 15 August 2010, 4767–4784

Taylor & Francis
Taylor & Francis Group

Balancing parallel two-sided assembly lines

Uğur Özcan[a]*, Hadi Gökçen[b] and Bilal Toklu[b]

[a]*Department of Industrial Engineering, Selçuk University, 42075 Selçuklu, Konya, Turkey;*
[b]*Department of Industrial Engineering, Gazi University, 06570 Maltepe, Ankara, Turkey*

(*Received 21 January 2009; final version received 22 May 2009*)

Two-sided assembly lines are usually designed to produce large-sized products such as automobiles, trucks and buses. In this type of production line, both left-side and right-side of the line are used. In parallel assembly lines, one or more product types are produced on two or more assembly lines located in parallel to each other. Both production lines have several serious practical advantages. For this purpose, in this paper, two or more two-sided assembly lines located in parallel to each other are considered and a tabu search algorithm which combines the advantages of both types of production lines is developed. To assess the effectiveness of the proposed algorithm, a set of test problems are solved. The proposed algorithm is illustrated with two examples, and some computational properties of the algorithm are given.

Keywords: assembly line balancing; two-sided assembly lines; parallel assembly lines; tabu search

1. Introduction

An assembly line is a manufacturing process in which components are consecutively assembled to an unfinished product depending on a set of tasks to produce a final product. A series of stations which are connected together by a conveyor belt or a similar mechanical material handling system are utilised on an assembly line. The tasks are allocated to stations according to a given precedence relationship among tasks, and they are performed in a certain time known as the task time. At each station, certain tasks are repeatedly performed in a limited time called the cycle time. The unfinished products are moved from one station to its successive station until they reach the end of the line. The problem of assigning tasks to stations in such a way that one or more objectives are optimised subject to some specific constraints is called as the assembly line balancing problem (ALBP).

Assembly lines can be classified in two general groups by means of the line configuration: traditional straight assembly lines (with single and multi/mixed products) and U-shaped assembly lines (with single and multi/mixed products). Operators are placed in a straight line along a conveying system in a serial line, while in a U-shaped assembly line they are placed in the centre of the 'U' shape. ALBP was first formulated mathematically by Salveson (1955). After this study, many studies on assembly lines including exact solution methods, heuristics and meta-heuristic approaches have been reported in the literature. The detailed reviews of such studies are given by Baybars (1986),

*Corresponding author. Email: uozcan@selcuk.edu.tr

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540903074991
http://www.informaworld.com

Ghosh and Gagnon (1989), Erel and Sarin (1998), and more recently by Scholl and Becker (2006) and Becker and Scholl (2006).

Assembly lines can also be classified as one-sided assembly lines and two-sided assembly lines. Only one side (either left-side or right-side) of the line is used in a one-sided assembly line, whereas both left-side and right-side of the line are used parallel in a two-sided assembly line. Two-sided assembly lines are usually designed to produce high-volume large-sized standardised products, such as automobiles, trucks and buses. In a two-sided assembly line some tasks may be preferred to be performed at one side of the line, while others may be performed at either side of the line. In practice, a two-sided assembly line can provide several advantages over a one-sided assembly line. These are:

(i) the assembly line length can be shorter than a one-sided assembly line;
(ii) it can reduce material handling cost, workers movement, set up time and the amount of throughput time; and
(iii) it can also reduce cost of tools and fixtures (Bartholdi 1993).

Although much research has been done on the one-sided ALBP, the two-sided assembly line balancing problem (TALBP) has been studied by considerably few researchers (Bartholdi 1993, Kim *et al*. 2000, 2009, Lee *et al*. 2001, Baykasoglu and Dereli 2008, Hu *et al*. 2008, Wu *et al*. 2008, Özcan and Toklu 2008, 2009a, b, Becker and Scholl 2009, Simaria and Vilarinho 2009).

According to Gökçen *et al*. (2006), when the demand is high enough, a possible way of increasing the efficiency of production on an assembly line is to use two or more assembly lines located in parallel to each other. Also, in any factory if products are assembled on parallel assembly lines then it can seriously improve the resources utilisation. The advantages of parallel assembly lines over a single assembly line are:

(i) it can provide to produce similar products or different models of the same product on adjacent lines,
(ii) it can reduce the idle time and increase the efficiency of the assembly lines,
(iii) it can provide to be able to make production with a different cycle time for each of the lines,
(iv) it can improve visibility and communication skills between operators, and
(v) it can also reduce operator requirements.

The concept of parallel assembly lines was first characterised by Gökçen *et al*. (2006). They developed a mathematical model and two heuristic procedures for the problem of balancing parallel assembly lines. After this study, Benzer *et al*. (2007) proposed a network model for solving the problem. Scholl and Boysen (2009) presented a detailed PALB problem description and modelled the problem as a binary linear programming formulation. They also proposed an exact solution approach based on SALOME. Kara *et al*. (2009) proposed two goal programming approaches to balance parallel assembly lines with precise and fuzzy goals.

In the mathematical complexity, one-sided ALBP is NP-hard class of combinatorial optimisation problems (Gutjahr and Nemhauser 1964). The combinatorial structure of this problem makes it difficult to obtain an optimal solution when the problem size increases. TALBP and parallel assembly line balancing problem (PALBP) are also NP-hard class. In addition to the mathematical complexity of the one-sided ALBP, TALBP also has an additional level of complexity, since the tasks have restrictions on the operation direction (Bartholdi 1993). In PALBP, two or more assembly lines located in

parallel to each other are balanced simultaneously. It increases the level of complexity of the problem.

Balancing simultaneously two or more two-sided assembly lines where the lines are located in parallel to each other provides the advantages of both parallel assembly lines and two-sided assembly lines, and reduces the idle time of stations while increasing the efficiency of the lines. In order to assure this, we focused on the idle time of stations. Idle times are sometimes unavoidable, even between tasks assigned to the same station, when balancing a two-sided assembly line. On the contrary, when balancing two or more two-sided assembly lines together, idle times can be reduced. In this study, we consider the balancing problem of parallel two-sided assembly lines (PTALBP). Thus, the purpose of this paper is to introduce and characterise PTALBP. For this purpose, a tabu search algorithm is proposed to solve PTALBP which aims to minimise the number of stations utilised on two-sided lines.

This paper is organised as follows: in Section 2 the concepts of two-sided assembly lines and parallel assembly lines are given and PTALBP is introduced and characterised. In Section 3, the proposed tabu search algorithm for solving PTALBP is presented. Two explanatory numerical example problems are solved by the proposed algorithm in Section 4. The performance of the proposed algorithm is discussed in Section 5. Finally, the conclusions and some directions for future researches are given in Section 6.

2. Parallel two-sided assembly lines

2.1 *Two-sided assembly lines*

Two-sided assembly lines are typically found in assembling large-sized high-volume products such as cars, trucks and buses. Each line has two sides, left-side and right-side, and each position has a pair of stations directly opposite each other, so that each component is performed by two operators simultaneously. Some tasks are forced to be assigned to only one side of the line, i.e. left-side (L-type tasks) or right-side (R-type tasks), and the remainders can be assigned to either side (E-type tasks) of the line. In order to explain TALBP clearly, consider the 12-task problem from Kim *et al*. (2000) depicted in Figure 1. The numbers in the nodes represent the tasks, the labels (t_j, d_j) below the nodes represent completion time of task j and preferred operation direction of task j. L/R indicates task j should be assigned to a left-side/a right-side station. E indicates task j can be performed at either side of the line. The directed arrow between node i and node j implies that task i immediately precedes task j. Figure 2 illustrates a solution of a two-sided

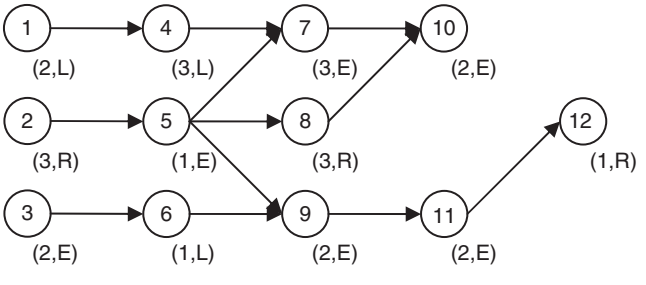

Figure 1. Data of the 12-task problem.

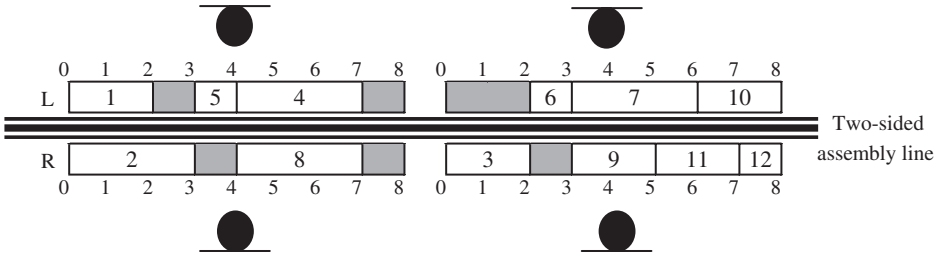

Figure 2. Task assignments.

assembly line assuming a cycle time of 8 time units. A pair of two directly facing stations called a mated station and one of them calls the other a companion. While balancing two-sided assembly lines, idle time is sometimes unavoidable even between tasks assigned to the same station. In Figure 2, for example, task 2 is assigned to the right-side of mated-station I and tasks 1 and 5 are assigned to the left-side of the same mated-station. Task 5 cannot be started at the finishing time of task 1, since task 2 is the immediate predecessor of task 5. So the start time of task 5 is equal to the finish time of task 2. Therefore, the sequence-depended finish time of tasks are taken into account in a two-sided assembly line, unlike a one-sided assembly line.

2.2 *Parallel assembly lines*

When the demand is high enough, a possible way of increasing the efficiency of the production on the assembly line is to use two or more assembly lines located in parallel to each other. Also, if products are produced on more than one assembly line and the lines are parallel to each other in any factory, then it can seriously improve the resources utilisation. In order to explain the parallel assembly line concept, two classical assembly line balancing problems well known in the literature are used. These are Jaeschke's 9-task problem given in Figure 3(a) and the Jackson's 11-task problem given in Figure 3(b). Cycle time is assumed as 18-time units. At each assembly line different product models are produced.

Figure 4(a) and (b) depicts an optimal solution of the 9-task problem and the 11-task problem. Both of them are balanced with three stations (the tasks in each line are allocated to three stations). The total number of stations required is six. Figure 4(c) also depicts an optimal solution of the problems when they are considered as parallel assembly lines. In this solution, the operator at Station I first completes tasks 1 and 2 on Line I and then completes tasks 1, 2 and 5 on Line II. Operators working in parallel assembly lines may work on two different components within a cycle time. The obtained number of stations is five. The stations which open both Line I and Line II are called common stations (e.g. Station I), and the other stations are called separate stations (e.g. Station V).

2.3 *Balancing parallel two-sided assembly lines*

Parallel two-sided assembly lines can be designed to carry out one or more product models with similar production characteristics in which a set of two-sided assembly lines ($h = 1, \ldots, H$) located in parallel to each other. Each two-sided assembly line is used to

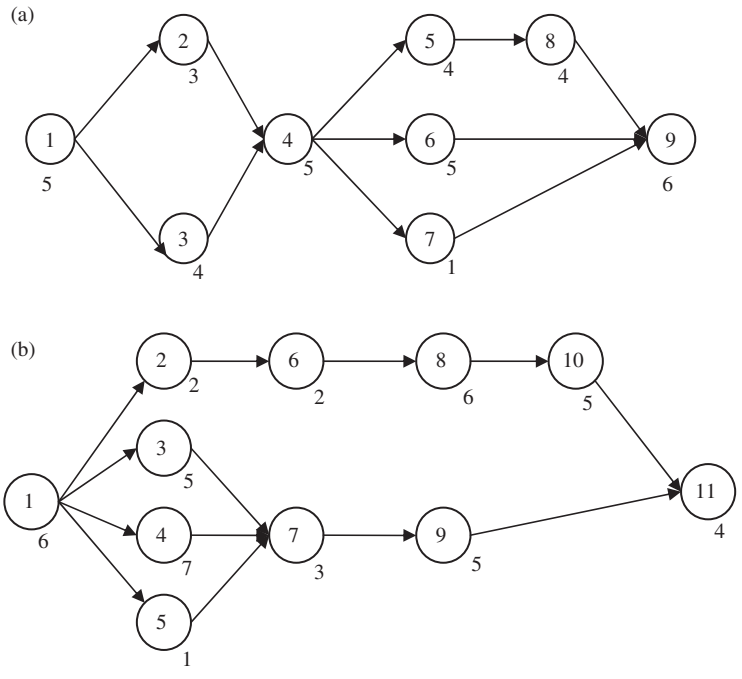

Figure 3. Data of (a) the 9-task problem and (b) the 11-task problem.

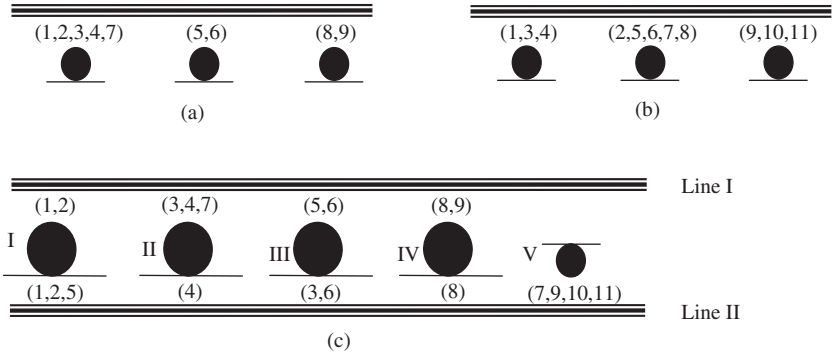

Figure 4. The task assignments of (a) the 9-task problem, (b) the 11-task problem and (c) a solution of parallel assembly line with a cycle time of 18-time units.

produce only one product model which is pre-determined by the line manager, and it does not change in the production process. So, each line has its own set of tasks ($i = 1, \ldots, n_h$). The tasks performed in line h are assigned to a station utilised on line h according to a given precedence relationship among tasks. P_{ih} is the set of predecessors of task i on line h. The tasks produced on different two-sided lines are performed simultaneously and the tasks assembled in line h are performed in a certain time (t_{ih}). A series of stations ($k = 1, \ldots, m$) are utilised on parallel two-sided assembly lines (a station can be utilised either on only one two-sided line or on two adjacent two-sided lines) and each two-sided assembly line may have a different cycle time (C_h). Operators perform tasks on a set of stations which are

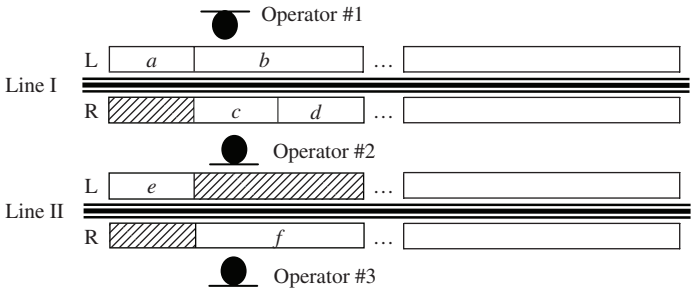

Figure 5. Configuration of a parallel two-sided assembly line.

utilised on only one two-sided line or on two adjacent two-sided lines. The configuration of a parallel two-sided assembly line is illustrated in Figure 5. As it can be seen in Figure 5, two two-sided assembly lines are located in parallel to each other. The precedence relations among tasks are assumed as task $a \in P_{b1}$ and P_{c1}, task $c \in P_{d1}$, and task $e \in P_{f2}$. Operator 2 first completes task e at the left-side station of Line II and then he/she completes tasks c and d at the right-side station of Line I. Operators 1 and 3 perform their tasks only at the left-side station of Line I and the right-side station of Line II, respectively.

It is assumed that PTALBP considered in this paper work under the following conditions:

(i) There are two or more two-sided assembly lines located in parallel to each other.
(ii) Each two-sided assembly line has two sides, left-side and right-side.
(iii) One or more product models with similar production characteristics are produced, and exactly one product model is produced on a two-sided assembly line.
(iv) Precedence diagrams for each product model are known.
(v) Task completion times of each product model are deterministic and known.
(vi) Each two-sided assembly line may have different cycle time.
(vii) Operators working at each station of each two-sided assembly line are multi-skilled (flexible workers).
(viii) For separate stations, operators must be worked at either left-side or right-side of any parallel two-sided line.
(ix) For common stations, operators must be worked both right-side and left-side which is close to each other of two adjacent two-sided line.
(x) Operator travel times are ignored.
(xi) Parallel tasks and parallel stations are not allowed.

For the case of the products produced on parallel assembly lines having different cycle times, it is especially difficult to control the cycle time constraint for the common stations of parallel assembly lines. Therefore, a common cycle time (C) should be used. Gökçen *et al.* (2006) used an approach based on the least common multiple (LCM) of the cycle times for the different cycle time situation. In this study, LCM approach is also used. The steps of the LCM approach for two different assembly lines are given below as an example:

(i) Find LCM of the cycle times.
(ii) Obtain the D_1 and D_2 values by dividing both cycle times to LCM value.

(iii) Constitute two precedence diagrams with different task times by multiplying the task times in each diagram with D_1 and D_2 values, separately.
(iv) Select the LCM as cycle time (C).

3. Proposed tabu search algorithm

Tabu search algorithm, defined and developed primarily by Glover (Glover 1989, 1990, Glover and Laguna 1997), is one of the most effective methods using local search techniques to find possible optimal or near-optimal solutions of many combinatorial optimisation problems. For more details on tabu search refer to Glover and Laguna (1993, 1997), Gendreau (2003) and in the context of the ALBP to Scholl and Voß (1996), Chiang (1998) and Lapierre *et al.* (2006).

The proposed tabu search starts with a feasible initial solution (S_0) and stores it as the current solution (S_c) and the best solution (S_b); the cost of this solution ($f(S_0)$) becomes the current value of objective function ($f(S_c)$) and the best value of objective function ($f(S_b)$). The neighbourhood solutions ($N(S_c)$) of the S_c are then generated by a move (m). These are candidate solutions. They are evaluated by the objective function and a candidate solution (S'_c) which is the best not tabu or satisfies the aspiration criterion is selected as new S_c. This selection is called a move and added to the tabu list, another (the oldest one) move is removed from the tabu list if it is overloaded. If the new S_c is better than the S_b, it is stored as new S_b. Otherwise, S_b remains unchanged. This searching process is repeated until the iteration counter ($iter$) is equal to the given number of iterations ($iter_{\max}$). Figure 6 shows the proposed algorithm. In the remainder of this section, a detailed description of the proposed tabu search algorithm is given.

3.1 *Initial solution*

In the proposed algorithm, in order to construct an initial solution we randomly generate a sequence includes the tasks on each two-sided assembly line and priority index of each task. The tasks are assigned to the stations sequentially by the priority value of tasks. The advantage of assigning tasks to stations according to a sequence is that it allows consideration of the sequence-depended finish time of tasks and ordering the task operations within a station. The position of a priority list (P list) represents a task, and the value of the position represents the priority value of that task. The length of a P list is equal to $n = n_1 + n_2 + \cdots + n_H$. Each priority value take values between 1 to n^*H. In order to obtain a feasible solution at each iteration of the tabu search algorithm, the tasks are assigned to stations as follows:

(1) Initialisation step: $h = 0$ and $k = 0$.
(2) $h = h + 1$ until $h = H - 1$.
(3) For all tasks $i \in Line_h$, determine a set of assignable tasks (A list): ($A = \{i \mid$ all $p \in P_{ih}$ have already been assigned to a station or $P_i = \emptyset$, and task i has not yet been assigned to a station$\}$).
(4) If $A = \emptyset$, then calculate objective function and Stop.
(5) Sort the tasks in A list in decreasing order of the priority value of tasks in P list.
(6) Select the first task j in $Line_h$.
(7) Open a new station on $Line_h$ according to the preferred operation direction of the selected task j: Set $k = k + 1$, $d_k = 0$.

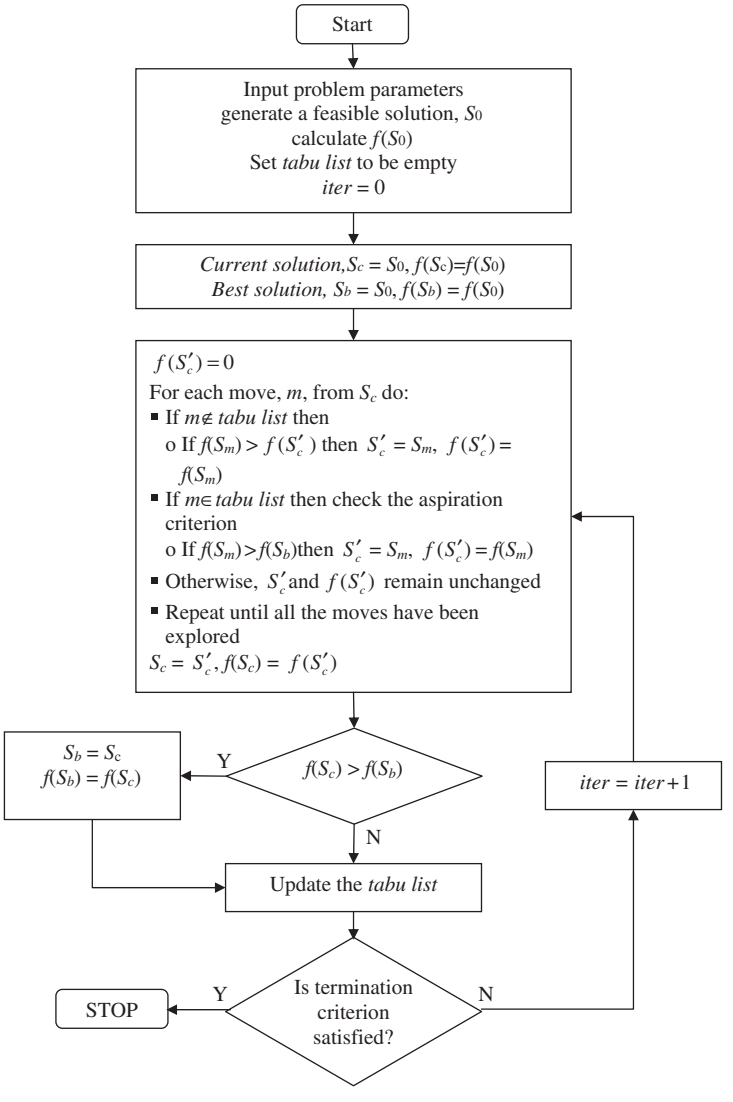

Figure 6. Flowchart of the proposed tabu search algorithm.

(If selected task j in $Line_h$ is an L-type/R-type task then, a new station is opened on left-side/right-side of $Line_h$. If the selected task j in $Line_h$ is an E-type task, then a side is selected randomly. In the next steps, if a selected task $i \in Line_h$ is an L-type/E-type task, or if task $i \in Line_{h-1}$ is an R-type/E-type task, then the selected task i can be assigned to station k which is opened on the left-side of $Line_h$. If station k is opened on the right-side of $Line_h$, then a selected task i can be assigned to station k; if task $i \in Line_h$ is an R-type/E-type task, or task $i \in Line_{h+1}$ is an L-type/E-type task.)

(8) Assign task j to station k on $Line_h$. Set $t^f_{jh} = t_{jh}$, $d_k = t^f_{jh}$.

(9) For all tasks $i \in \{Line_{h-1}, Line_h, Line_{h+1}\}$, determine a set of assignable tasks (B list): ($B = \{i \mid$ all $p \in P_{ih}$ have already been assigned to a station or $P_i = \emptyset$, and task i has not yet been assigned to a station$\}$).

(10) If $B = \emptyset$, then go to Step 3.
(11) Sort the tasks in B list in decreasing order of the priority value of tasks in P list.
(12) Assign the first task j in B for which;

(12.1) If station k is opened on the left-side of $Line_h$, then
(12.1.1) If task $j \in Line_h$ and it is an L-type task, then
If $t_{jh} + d_k \le C$ and $t_{jh} + t^f_{rh} \le C$ ($t^f_{rh} = \max\{t^f_{ph} \mid p \in P_j$ have already been assigned to the right-side of $Line_h\}$), then assign task j to station k. Set $t^f_{jh} = \max\{(t_{jh} + d_k), (t_{jh} + t^f_{rh})\}$, set $d_k = t^f_{jh}$, go to Step 9. Otherwise go to Step 13.
(12.1.2) If task $j \in Line_{h-1}$ and it is an R-type task, then
If $t_{jh-1} + d_k \le C$ and $t_{jh-1} + t^f_{rh-1} \le C$ ($t^f_{rh-1} = \max\{t^f_{ph-1} \mid p \in P_j$ have already been assigned to the left-side of $Line_{h-1}\}$), then assign task j to station k. Set $t^f_{jh-1} = \max\{(t_{jh-1} + d_k), (t_{jh-1} + t^f_{rh-1})\}$, set $d_k = t^f_{jh-1}$, go to Step 9. Otherwise go to Step 13.
(12.2) If station k is opened on the right-side of $Line_h$, then
(12.2.1) If task $j \in Line_h$ and it is an R-type task, then
If $t_{jh} + d_k \le C$ and $t_{jh} + t^f_{rh} \le C$ ($t^f_{rh} = \max\{t^f_{ph} \mid p \in P_j$ have already been assigned to the left-side of $Line_h\}$), then assign task j to station k. Set $t^f_{jh} = \max\{(t_{jh} + d_k), (t_{jh} + t^f_{rh})\}$, set $d_k = t^f_{jh}$, go to Step 9. Otherwise go to Step 13.
(12.2.2) If task $j \in Line_{h+1}$ and it is an L-type task, then
If $t_{jh+1} + d_k \le C$ and $t_{jh+1} + t^f_{rh+1} \le C$ ($t^f_{rh+1} = \max\{t^f_{ph+1} \mid p \in P_j$ have already been assigned to the right-side of $Line_{h+1}\}$), then assign task j to station k. Set $t^f_{jh+1} = \max\{(t_{jh+1} + d_k), (t_{jh+1} + t^f_{rh+1})\}$, set $d_k = t^f_{jh+1}$, go to Step 9. Otherwise go to Step 13.
(12.3) If selected task j is an E-type task then randomly select a side.

(13) If none of these tasks in B could be assigned to station k then go to Step 3.

Where, d_k is workload of station k, t^f_{ih} is the finishing time of task i in $Line_h$.

3.2 *Neighbourhood generation*

A new solution which is obtained from a current solution by a move is called a neighbour solution. In this paper, the swap operator is used to generate the neighbour solutions. This neighbourhood contains all those permutations S'_c obtained from S_c by swapping the priority index of tasks in positions p and q, i.e.,

$$S_c = \langle x_1, \ldots, x_p, \ldots, x_q, \ldots, x_n \rangle$$

$$S'_c = \langle x_1, \ldots, x_q, \ldots, x_p, \ldots, x_n \rangle; \quad \forall p, q \in \{1, \ldots, n\},\ p \neq q.$$

The complete swap neighbourhood of a given sequence of priority index of tasks contains all combinations of total number of tasks on parallel lines taken two at a time. Consequently, there are $n(n-1)/2$ swaps in the neighbourhood ($N(S_c)$). Let H be the number of parallel assembly lines, and n_h be the number of tasks produced in parallel assembly line h. Then $n = \sum_{h=1}^{H} n_h$.

3.3 *Objective function*

The type I ALBP consists of finding an assignment of tasks to stations such that the required number of stations is minimised. When the minimisation of the number of stations directly uses as an objective function, we cannot obtain a strong distinction between the solutions. Because, the number of stations used in a given solution is not sufficient to evaluate its potential for improvement or to compare the quality of two competing solutions (Lapierre *et al.* 2006). In order to get a more diversified evaluation of a solution than by using the number of stations several approaches have been presented (Scholl and Becker 2006). In addition to overcome this difficulty Chiang (1998) proposed a nonlinear objective function which is given in Equation (1) to solve type I ALBPs.

$$\max \sum_{k=1}^{m} \left(\sum_{j \in S_k} t_j \right)^2 \tag{1}$$

where, k is the station number ($k = 1, \ldots, m$), j is the task number ($j = 1, 2, \ldots, n$), t_j is the performance time of task j, and S_k is the set of tasks which are assigned to station k.

In this study, we modified and used this nonlinear objective function to obtain the minimum required number of stations. Maximising this objective function helps to reduce the number of stations (Chiang 1998), and it also gives more information about the searching process. The following Equation (2) is modified and proposed to compute the nonlinear objective function for PTALBP.

$$\max \sum_{k=1}^{m} \left(\sum_{j \in S_k \wedge j \in h} t_{jh} \right)^2 \tag{2}$$

3.4 *Tabu list and aspiration criterion*

In this study, the length of the tabu list is fixed and after some experimentation has been set to $\sqrt{\sum_{h=1}^{H} n_h}$. Tabu list consisting of a two-dimensional array is used to check if a move from a solution to its neighbourhood is forbidden or allowed. Whenever a pair of tasks is declared as tabu, the tasks remain tabu for the tabu length iterations. If these tasks are not tabu then the priority index of tasks are free to swap. The tabu restriction may be overridden if the move will produce a solution that is better than that found in the past. This rule is called the aspiration criterion.

3.5 *Termination criterion*

The proposed tabu search algorithm stops when the iteration counter (*iter*) reaches the iteration limit ($iter_{\max}$). In our study, the iteration limit is fixed and after some experimentation has been set to $\sum_{h=1}^{H} n_h$. And also the proposed procedure terminates when the theoretical minimum number of stations is reached. Hu *et al.* (2008) presented the simple lower bound of TALBP. The calculation of the lower bound of TALBP is

as follows.

$$Max = \max\left\{\left\lceil\frac{LTotal}{C}\right\rceil, \left\lceil\frac{RTotal}{C}\right\rceil\right\} \tag{3}$$

$$LB = 2 \times Max + \max\left\{0, \left\lceil\frac{ETotal - (Max \times C - LTotal) - (Max \times C - RTotal)}{C}\right\rceil\right\} \tag{4}$$

where *LTotal*, *RTotal* and *ETotal* are the total task time of the L, R and E directional tasks, respectively, and C is the cycle time. In order to obtain a simple lower bound for PTALBP, we modified Equations (3) and (4). In this study, the modified lower bound given in Equation (7) is used as another termination criterion.

$$Max_h = \max\left\{\left\lceil\frac{LTotal_h}{C_h}\right\rceil, \left\lceil\frac{RTotal_h}{C_h}\right\rceil\right\} \quad h = 1, \ldots, H \tag{5}$$

$$LB_h = 2 \times Max_h + \max\left\{0, \left\lceil\frac{ETotal_h - (\mathrm{Max}_h \times C_h - LTotal_h) - (\mathrm{Max}_h \times C_h - RTotal_h)}{C_h}\right\rceil\right\} \quad h = 1, \ldots, H \tag{6}$$

$$LB = \left\lceil\sum_{h=1}^{H} LB_h\right\rceil \tag{7}$$

where C_h is the cycle time of parallel two-sided line h, and $LTotal_h$, $RTotal_h$ and $ETotal_h$ are the total task time of the L, R and E directional tasks of parallel two-sided line h, respectively.

4. Illustrative examples

To clarify the solution procedure, two numerical examples are used. In illustrative examples, the same and different cycle times are used for parallel two-sided lines, respectively.

4.1 *Example I*

Here, the example problem given in Figure 1 is used. There are two parallel two-sided assembly lines and the same product has been produced on each line. Cycle time for each line is the same and it is assumed as 8-time units. The lower bound (theoretical minimum number of stations) of this problem is calculated as 7 by using Equation (7). The length of the tabu list is ($\sqrt{12+12}$ =) 5, and the iteration limit is (12 + 12=) 24. At each iteration, 276 neighbourhood solutions are generated by swap neighbourhood.

Initial solution

The proposed algorithm starts with a feasible initial solution. The initial solution is obtained from a sequence which includes priority index of all tasks produced in parallel

Task [line]	1[1]	2[1]	3[1]	4[1]	5[1]	6[1]	7[1]	8[1]	9[1]	10[1]	11[1]	12[1]	1[2]	2[2]	3[2]	4[2]	5[2]	6[2]	7[2]	8[2]	9[2]	10[2]	11[2]	12[2]
Priority index	1	3	14	24	4	12	2	22	7	8	15	20	6	21	17	10	19	11	18	16	23	5	9	13

Figure 7. Initial priority list.

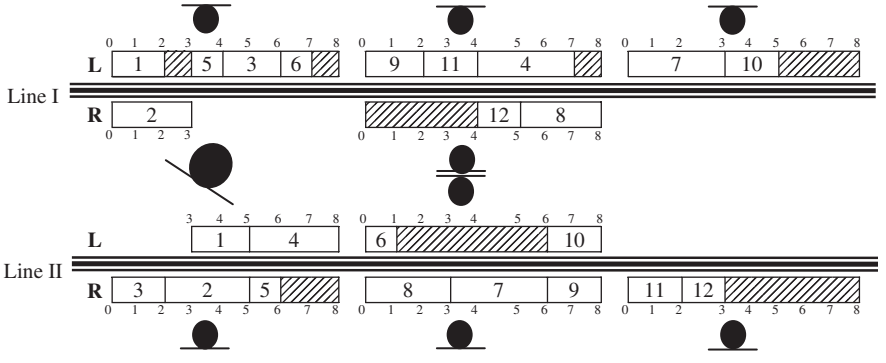

Figure 8. Initial feasible solution.

two-sided lines. The priority index of a task, demonstrating the assignment priority of this task, is randomly generated between one and 24, and different from each other. Figure 7 shows the initial priority index of tasks and Figure 8 shows a feasible initial solution with nine stations. The objective function value of this solution is $(6^2+8^2+6^2+7^2+4^2+3^2+8^2+5^2+3^2=)$ 308. The start time and the finish time of the tasks are also given in Figure 8.

The Gantt chart given in Figure 8 represents the initial parallel two-sided line balance. Task numbers are placed at their relevant positions inside the bars. For every task, its start time and finish time are shown alongside the bars. Shaded rectangles indicate either unavoidable delay between two consecutive tasks or idle time at the end of cycle time.

New solution generation

Using swap operator, the neighbourhoods are generated from initial priority index sequence. For each of the newly generated neighbourhoods objective function is calculated and neighbourhood solution and maximum value of objective function is taken as the current solution for the next iteration. For illustration purposes, only the top five candidates which have a maximum value of objective function are shown in Table 1. Among these moves, move (2, 8) is taken as the first best move and to avoid cycling, this move is classified as tabu. This move is forbidden for the next five iterations. The tabu structure for this iteration is shown in Figure 9. In this structure each cell contains the number of iterations remaining until the corresponding moves are allowed to exchange position again.

The new priority list and the parallel assembly line balance with eight stations are given in Figures 10 and 11, respectively. The value of objective function is 332. As shown is Figure 11, all stations are separate stations.

Table 1. Top five candidate solutions at the first iteration.

Swap		
Row i	Row j	Objective function
1	4	314
1	21	318
2	8	332*
8	19	332
8	20	332

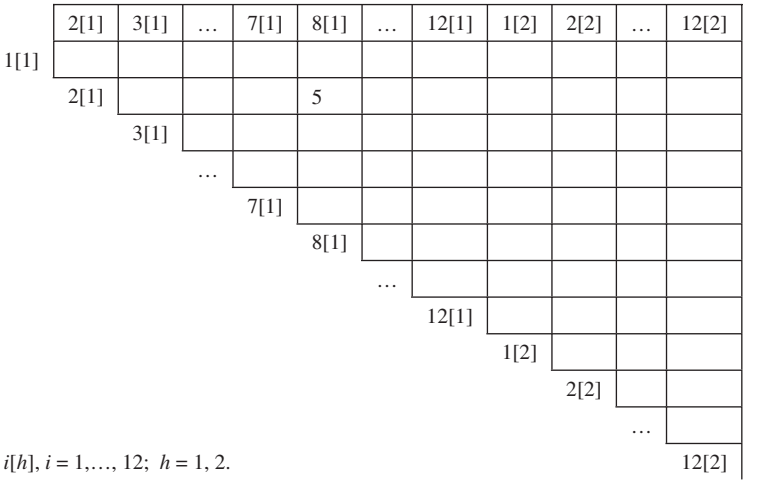

Figure 9. Tabu structure.

Task [line]	1[1]	2[1]	3[1]	4[1]	5[1]	6[1]	7[1]	8[1]	9[1]	10[1]	11[1]	12[1]	1[2]	2[2]	3[2]	4[2]	5[2]	6[2]	7[2]	8[2]	9[2]	10[2]	11[2]	12[2]
Priority index	1	22	14	24	4	12	2	3	7	8	15	20	6	21	17	10	19	11	18	16	23	5	9	13

Figure 10. New priority list.

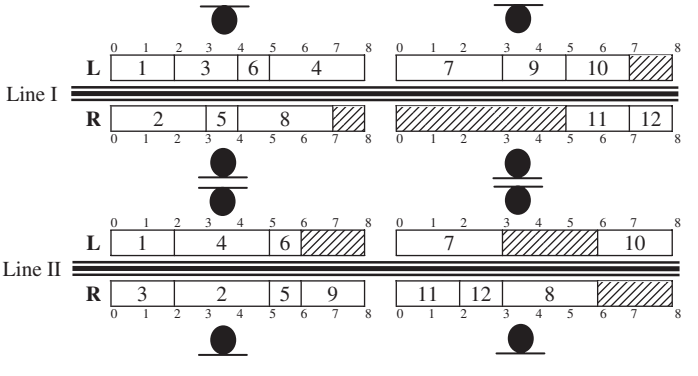

Figure 11. New solution.

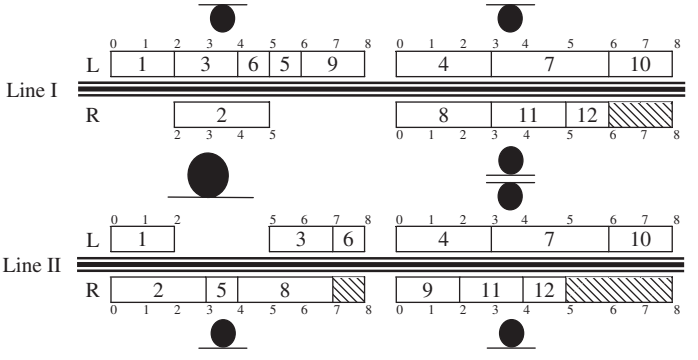

Figure 12. Final solution.

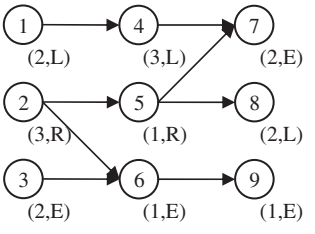

Figure 13. Data of the 9-task problem.

Final solution

The searching process is stopped, since the theoretical number of stations is found. The final solution with seven stations is shown in Figure 12. The value of objective function is 366. In this solution, the operator at common station first completes task 1 in Line II and then completes task 2 in Line I, finally completes tasks 3 and 6 in Line II. It can be noted that the number of stations obtained from this solution is one station less than the independent solutions of two-sided assembly lines (4 + 4) given in Figure 2.

4.2 *Example II*

Precedence diagrams of an example for two different product models are illustrated in Figure 1 and Figure 13. The problem data given in Figure 13 is taken from Kim *et al.* (2000). The first product which is produced in two-sided line I has nine operations, and the second product produced in two-sided line II has 12 operations. The operations of products are similar but not the same. It is assumed that the cycle time of Line I and Line II are 4 and 8, respectively. For these cycle time values, LCM value is obtained as 8. Then D_1 and D_2 values are calculated as 2 and 1, respectively. D_1 value is multiplied by the task times in precedence diagram (in Figure 13) for product 1. In the same way, D_2 value is also multiplied by the task times in precedence diagram (in Figure 1) for product 2. The lower bound value of this problem is calculated as 8. The length of the tabu list is 5, and the iteration limit is equal to 21. At each iteration, 210 neighbourhood solutions are generated by swap operator.

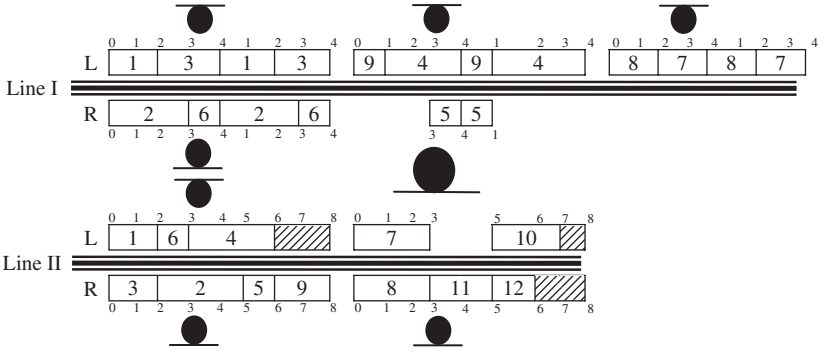

Figure 14. Final solution of problem II.

When the proposed algorithm is applied for cycle time of 8-time units, it can be reached to the balance given in Figure 14. In the new obtained two-sided line balance, two and one unit products have been produced for each cycle time of 8-time units in Line I and Line II, respectively. Therefore, the number of stations required for the new line balance is eight. The obtained result is equal to the lower bound of this problem. If the lines described above are balanced independently for different cycle times, the number of stations required will be nine $(5+4)$.

5. Computational results

In this section, the computational results of the proposed algorithm are presented. As a basis for generating test problems for PTALBP, we have used the well-known test problems for TALBP. We have combined the seven original test problems (P9, P12, P16, P24, A65, B148 and A205). The test problems of P9, P12 and P24 are taken from Kim *et al*. (2000), P16, A65 and A205 are taken from Lee *et al*. (2001) and B148 is taken from Bartholdi (1993). B148 was then modified by Lee *et al*. (2001). The test problems are solved with various cycle times which include the same cycle time situation and the different cycle time situation.

The number of stations obtained from the proposed algorithm is compared with the theoretical minimum number of stations, *LB*, and independent line balance of the two-sided assembly lines collected from literature. The results are given in Table 2.

As seen from Table 2, in six of the 32 test problems, the number of stations required for the proposed algorithm is less than the independent line balance solutions. In the remaining 26 test problems, results of the proposed algorithm and independent line balance are the same.

The optimal number of stations is not less than the theoretical minimum number of stations. That is, if the number of stations obtained from the procedure is equal to the theoretical minimum number of stations, then the result is optimal. If it is greater than the theoretical minimum number of stations, then the result may be optimal. For the values of the number of stations which is different from theoretical minimum number of stations in Table 2, it is not possible to say anything about whether the values of the number of stations are optimal or not. This comparison may only give an idea about the performance of the procedure. As seen from Table 2, there were nine problems where the optimal

Table 2. Computational results.

No.	Test problems (Line 1 -Line 2)	No. of task (Line 1 -Line 2)	Cycle time (Line 1 -Line 2)	No. of station for independent balance of Line 1 + Line 2	Theoretical min. no of station (LB)	Proposed approach No. of station
1	P9-P9	9-9	3-3	6 + 6	12	12
2	P9-P9	9-9	4-5	5 + 4	8	8
3	P9-P12	9-12	6-6	3 + 5	7	8
4	P9-P12	9-12	4-7	5 + 4	8	9
5	P12-P12	12-12	5-5	6 + 6	10	11
6	P12-P12	12-12	6-7	5 + 4	8	9
7	P12-P16	12-16	7-16	4 + 6	9	10
8	P12-P16	12-16	8-21	4 + 5	8	8
9	P16-P16	16-16	16-16	6 + 6	11	11
10	P16-P16	16-16	19-21	5 + 5	9	10
11	P16-P24	16-24	19-35	5 + 4	9	9
12	P16-P24	16-24	22-40	4 + 4	8	8
13	P24-P24	24-24	18-18	8 + 8	16	16
14	P24-P24	24-24	20-24	8 + 6	13	14
15	P24-A65	24-65	30-490	5 + 11	16	16
16	P24-A65	24-65	20-544	8 + 10	17	18
17	A65-A65	65-65	381-381	15 + 15	27	29
18	A65-A65	65-65	435-435	13 + 13	24	25
19	A65-A65	65-65	490-544	11 + 10	20	21
20	A65-B148	65-148	381-408	15 + 13	26	28
21	A65-B148	65-148	490-459	11 + 12	22	23
22	A65-B148	65-148	544-510	10 + 11	20	21
23	B148-B148	148-148	408-408	13 + 13	26	26
24	B148-B148	148-148	306-357	18 + 15	32	33
25	B148-B148	148-148	459-510	12 + 11	22	23
26	B148-A205	148-205	306-1888	18 + 15	30	33
27	B148-A205	148-205	510-2832	11 + 10	19	21
28	B148-A205	148-205	255-1510	21 + 18	36	39
29	A205-A205	205-205	1510-1510	18 + 18	31	36
30	A205-A205	205-205	2832-2832	10 + 10	17	20
31	A205-A205	205-205	2077-2266	14 + 12	22	26
32	A205-A205	205-205	2454-2643	12 + 11	19	23

solutions are known exactly from the 32 problems. In these problems, the number of stations obtained from the procedure are equal to the lower bound. In the remaining test problems, the proposed algorithm mostly produces one or more stations. As a result, it can be seen that the performance of the procedure is sufficient.

6. Conclusion and future research directions

The main objective of this paper is to introduce and characterise the problem of balancing parallel two-sided assembly lines. To the best knowledge of the authors, there is no published study concerning this new problem. For this purpose, a tabu search algorithm is proposed to solve the problem which aims to minimise the number of stations and it is illustrated with two numerical examples for the same cycle time and the different cycle

time situations. The proposed tabu search algorithm combines the advantages of both parallel and two-sided assembly lines. Also, several well-known test problems in the literature are solved by the proposed algorithm, and obtained results are compared with the theoretical minimum number of stations and independent two-sided assembly line balance results obtained from the other techniques in the literature. Comparison results show that the performance of the proposed algorithm is sufficient. The proposed algorithm provides a significant improvement in two-sided assembly line efficiency when the lines are located in parallel to each other.

This paper represents a good starting point for future studies. First, a mathematical formulation of the problem or an optimal solution methodology such as branch-and-bound algorithms and dynamic programming may be developed. Second, more efficient heuristics or meta-heuristic approaches such as genetic algorithms, simulated annealing and ant colony optimisation, may be developed to solve the problem. Third, a number of realistic constraints such as positional constraints, synchronous tasks constraints and zoning constraints may be considered. In addition, balancing and model sequencing of parallel mixed-model two-sided assembly lines may be considered as future research.

References

Bartholdi, J.J., 1993. Balancing two-sided assembly lines: a case study. *International Journal of Production Research*, 31 (10), 2447–2461.

Baybars, I., 1986. A survey of exact algorithms for the simple line balancing problem. *Management Science*, 32 (8), 909–932.

Baykasoglu, A. and Dereli, T., 2008. Two-sided assembly line balancing using an ant-colony-based heuristic. *International Journal of Advanced Manufacturing Technology*, 36 (5–6), 582–588.

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

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

Benzer, R., *et al*. 2007. A network model for parallel line balancing problem. *Mathematical Problems in Engineering*, Art. No. 10106.

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

Chiang, W.C., 1998. The application of a tabu search metaheuristic to the assembly line balancing problem. *Annals of Operations Research*, 77, 209–227.

Erel, E. and Sarin, S.C., 1998. A survey of the assembly line balancing procedures. *Production Planning and Control*, 9 (5), 414–434.

Gendreau, M., 2003. An introduction to tabu search, *In*: F. Glover and G.A. Kochenberger, eds. *Handbook of Metaheuristics*. Boston/Dordrecht/London: Kluwer Academic Publishers, 37–54.

Ghosh, S. and Gagnon, J., 1989. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, 27 (4), 637–670.

Glover, F. and Laguna, M., 1993. Tabu search, *In*: C.R. Reeves, ed. *Modern Heuristic Techniques for Combinatorial Problems*. London/Edinburgh/Boston/Melbourne/Paris/Berlin/Wien: Blackwell, 70–150.

Glover, F. and Laguna, M., 1997. *Tabu search*. Boston/Dordrecht/London: Kluwer Academic Publishers.

Glover, F., 1989. Tabu search – Part I. *ORSA Journal on Computing*, 1 (3), 190–206.
Glover, F., 1990. Tabu search – Part II. *ORSA Journal on Computing*, 2 (1), 4–32.
Gökçen, H., Ağpak, K., and Benzer, R., 2006. Balancing of parallel assembly lines. *International Journal of Production Economics*, 103 (2), 600–609.
Gutjahr, A.L. and Nemhauser, G.L., 1964. An algorithm for the line balancing problem. *Management Science*, 11 (2), 308–315.
Hu, X.-F., Wu, E.-W., and Jin, Y., 2008. A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research*, 186 (1), 435–440.
Kara, Y., Gökçen, H., and Atasagun, Y., 2009. Balancing parallel assembly lines with precise and fuzzy goals. *International Journal of Production Research*, DOI: 10.1080/00207540802534715.
Kim, Y.K., Kim, Y., and Kim, Y.J., 2000. Two-sided assembly line balancing: a genetic algorithm approach. *Production Planning and Control*, 11 (1), 44–53.
Kim, Y.K., Song, W.S., and Kim, J.H., 2009. A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers and Operations Research*, 36 (3), 853–865.
Lapierre, S.D., Ruiz, A., and Soriano, P., 2006. Balancing assembly lines with tabu search. *European Journal of Operational Research*, 168 (3), 826–837.
Lee, T.O., Kim, Y., and Kim, Y.K., 2001. Two-sided assembly line balancing to maximise work relatedness and slackness. *Computers and Industrial Engineering*, 40 (3), 273–292.
Özcan, U. and Toklu, B., 2008. A tabu search algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, DOI: 10.1007/s00170-008-1753-5.
Özcan, U. and Toklu, B., 2009a. Multiple-criteria decision-making in two-sided assembly line balancing: a goal programming and a fuzzy goal programming models. *Computers and Operations Research*, 36 (6), 1955–1965.
Özcan, U. and Toklu, B., 2009b. Balancing of mixed-model two-sided assembly lines. *Computers and Industrial Engineering*, 57 (1), 217–227.
Salveson, M.E., 1955. The assembly line balancing problem. *Journal of Industrial Engineering*, 6 (3), 18–25.
Scholl, A. and Voß, S., 1996. Simple assembly line balancing Heuristic approaches. *Journal of Heuristics*, 2 (3), 217–244.
Scholl, A. and Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168 (3), 666–693.
Scholl, A. and Boysen, N., 2009. Designing parallel assembly lines with split workplaces: model and optimisation procedure. *International Journal of Production Economics*, 119 (1), 90–100.
Simaria, A.S. and Vilarinho, P.M., 2009. 2–ANTBAL: an ant colony optimisation algorithm for balancing two-sided assembly lines. *Computers and Industrial Engineering*, 56 (2), 489–506.
Wu, E.-F., *et al.*, 2008. A branch-and-bound algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, 39 (9–10), 1009–1015.