PERGAMON

Computers & Industrial Engineering 40 (2001) 273–292

computers & industrial engineering

www.elsevier.com/locate/dsw

Two-sided assembly line balancing to maximize work relatedness and slackness

Tae Ok Lee[a,1], Yeongho Kim[b,2], Yeo Keun Kim[c,*]

[a]Development team 1, Korea Computer Inc., 395-65, Shindaebang-Dong Dongjak-Ku, Seoul 156-010, South Korea
[b]Department of Industrial Engineering, Seoul National University, Seoul 151-742, South Korea
[c]Department of Industrial Engineering, Chonnam National University, 300 Yongbongdong, Pukku, Kwangju 500-757, South Korea

Abstract

This paper considers two-sided (left- and right-side) assembly lines that are often used in assembling large-sized products, such as trucks and buses. A large number of exact algorithms and heuristics have been proposed to balance one-sided assembly lines. However, little attention has been paid to balancing the two-sided lines. An efficient assignment procedure is developed for two-sided assembly line balancing problems. A special emphasis is placed on maximizing work relatedness and maximizing work slackness, which are of practical significance especially in two-sided lines. We first investigate the characteristics of two-sided lines and define new measures for the balancing. Then, a group assignment procedure, which assigns a group of tasks at a time rather than a unit task, is designed. Experiments are carried out to demonstrate the performance of the proposed method. The results show that our procedure is promising in the solution quality.

Keywords: Line balancing; Two-sided assembly lines; Work relatedness; Work slackness

1. Introduction

Assembly line balancing is the problem of assigning various tasks to stations, while optimizing one or more objectives without violating any restrictions imposed on the line. Assembly lines can be categorized into one-sided lines and two-sided lines. A one-sided line is a line that uses only one side of the line, whereas a two-sided line uses both (left and right) sides of the line in parallel. A one-sided assembly line has been the basic, simple form of line balancing problems and perhaps the most widely studied one. This research deals with two-sided assembly line balancing (two-ALB) problems. A two-sided assembly

* Corresponding author. Tel.: +82-62-530-1783; fax: +82-62-530-1789.
E-mail addresses: tolee@kci.co.kr (T.O. Lee), yeongho@snu.ac.kr (Y. Kim), kimyk@chonnam.chonnam.ac.kr (Y.K. Kim).
[1] Tel.: +82-2-829-3338; fax: +82-2-829-3400.
[2] Tel.: +82-2-880-8335; fax: +82-2-889-8560.

PII: S0360-8352(01)00029-8

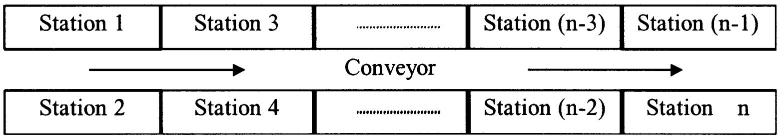

Fig. 1. Two-sided assembly line.

line is illustrated in Fig. 1. A pair of two directly facing stations, such as 1 and 2, is called a *mated-station*, and one of the two calls the other a *companion*.

Two-sided assembly lines are typically found in producing large-sized products, such as trucks and buses. Assembling these products is in some respects different from assembling small products. Some assembly operations prefer to be performed at one of the two sides (Bartholdi, 1993; Kim, Kim & Lee, 1999). Let us consider, for example, a truck assembly line. Installing a gas tank, air filter, and toolbox can be more easily achieved at the left-hand side of the line, whereas mounting a battery, air tank, and muffler prefers the right-hand side. Assembling an axle, propeller shaft, and radiator does not have any preference in their operation directions so that they can be done at any side of the line. The consideration of the preferred operation directions is important since it can greatly influence the productivity of the line, in particular when assigning tasks, laying out facilities, and placing tools and fixtures in a two-sided assembly line.

A two-sided assembly line in practice can provide several substantial advantages over a one-sided assembly line (Bartholdi, 1993). These include the following: (1) it can shorten the line length, which means that fewer workers are required, (2) it thus can reduce the amount of throughput time, (3) it can also benefit from lowered cost of tools and fixtures since they can be shared by both sides of a mated-station, and (4) it can reduce material handling, workers movement and set-up time, which otherwise may not be easily eliminated. These advantages give a good reason for utilizing two-sided lines for assembling large-sized products.

Although there is a body of literature related with assembly line balancing, little attention has been paid to two-ALB. Almost all of the previous researches have dealt with one-sided assembly line balancing problems (Baybars, 1986; Ghosh & Gagnon, 1989; Talbot, Patterson & Gehrlein, 1986; van Zante-de Fokkert & de Kok, 1997) Since the assembly line balancing problems in general fall into an NP-hard class problem (Baybars, 1986), many of the previous research have proposed heuristic approaches that can find near optimal solutions very quickly. To the best of our knowledge, Bartholdi (1993) is the first to address two-ALB problems. His major focus is placed on design and use of an interactive program that assists line managers to assign tasks, and an assignment rule is suggested for building solutions quickly. Kim et al. (1999) provide mathematical models for two-ALB problems with various objectives, such as minimization of line length, the number of stations, and workload deviations. Kim, Kim and Kim (2000) present a genetic algorithm for two-ALB.

An assignment procedure for two-ALB problems is developed in this paper. While designing the procedure, we take account of the features specific to two-sided assembly lines. As for the decision criteria, we introduce work relatedness and work slackness. However, these two criteria often conflict with traditional criteria, such as the number of stations and cycle time. Therefore, enhancing work relatedness and work slackness may sometimes need that the traditional criteria are sacrificed to some extent. This expense would be justified by the practical significance of the new criteria. The developed procedure can produce good solutions in terms of the work relatedness

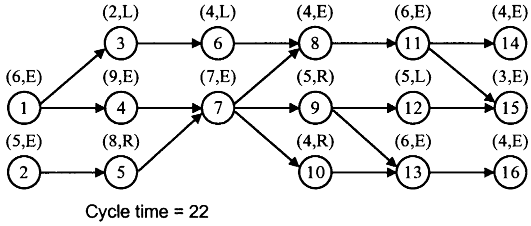

Fig. 2. Precedence diagram.

and work slackness without causing much damage to the traditional criteria. These are further described in Section 2.

2. Two-ALB problem and decision criteria

This section describes the features of two-ALB problems, which distinguish them from the traditional one-side assembly problems, and the decision criteria considered in this paper.

2.1. Two-ALB problem

A line balancing problem is usually represented by a precedence diagram as illustrated in Fig. 2. A circle indicates a task, and an arc linking two tasks represents the precedence relation between the tasks. Each task is associated with a label of (t_i, d), where t_i is the task processing time and d (= L, R or E) is the preferred operation direction. L and R, respectively, indicate that the task should be assigned to a left- and a right-side station. A task associated with E can be performed at either side of the line.

While balancing assembly lines, it is generally needed to take account of the features specific to the lines. In a one-sided assembly line, if precedence relations are considered appropriately, all the tasks assigned to a station can be carried out continuously without any interruption. However, in a two-sided assembly line, some tasks assigned to a station can be delayed by the tasks assigned to its companion (Bartholdi, 1993). In other words, idle time is sometimes unavoidable even between tasks assigned to the same station. Consider, for example, task j and its immediate predecessor i. Suppose that j is assigned to a station and i to its companion station. Task j cannot be started until task i is completed. Therefore, balancing such a two-sided assembly line, unlike a one-sided assembly line, needs to consider the sequence-dependent finish time of tasks.

This notion of sequence dependency further influences the treatment of cycle time constraint. Every task assigned to a station must be able to be completed within a predetermined cycle time. In a one-sided assembly line, this can readily be achieved by checking the total operation time of tasks assigned to a station. Therefore, a task not violating any precedence constraints can be simply added to the station if the resulting total amount of operation time does not exceed the cycle time. However, in a two-sided assembly line, due to the above sequence-dependent delay of tasks, the cycle time constraint should be more carefully examined. The amount of time required to perform tasks allocated to a station is determined by the task sequences in both sides of the mated-station as well as their operation time.

2.2. Decision criteria

We describe two decision criteria, the indices of work relatedness and work slackness. These two criteria are of practical significance in fact, and henceforth stressed in this paper.

First, related tasks are defined as two tasks that are directly connected in a precedence diagram. Allotting such related tasks to the same station is certainly preferable to splitting them into several stations. For example, consider a truck assembly line. The tasks involved in the line are often classified into several groups based on the major components, such as engine and axle. Tasks contained in different groups usually have no precedence relations so that they, in theory, can be assigned to the same station. This means that a worker has to work on several different types of tasks, such as engine-related and axle-related tasks at the same time. However, it would be desirable that his/her job belongs to as few groups as possible. This can improve work efficiency, clarify the responsibility for the completion of the related tasks, and give workers more job satisfaction. Such concept is expressed as work relatedness and is especially critical in assembling large-sized products, such as automobiles.

Although work relatedness is an important factor in task assignment, it is not easy to assess it quantitatively. Agrawal (1985) has developed an assignment rule that can take account of the relatedness by assigning immediately preceding or succeeding tasks to the same station if possible. In this research, a measure based on Agrawal's definition of work relatedness is used. Let SN_j be the number of connected networks representing the precedence relations of tasks assigned to station j, and n be the total number of stations. Then, the index of work relatedness (IWR) is defined as follows:

$$\mathrm{IWR} = \frac{n}{\sum_{j=1}^{n} \mathrm{SN}_j}. \tag{1}$$

For example, consider the precedence diagram shown in Fig. 2. If the tasks, {1,2,5}, are assigned to station 1, $\mathrm{SN}_1 = 2$ since the precedence diagram representing these tasks has two subnetworks. If the tasks, {1,3,4}, are assigned to station 1, $\mathrm{SN}_1 = 1$ since all the tasks are connected. In terms of work relatedness, assigning tasks {1,3,4} to the station is better than {1,2,5}.

The IWR takes a value in $[n/m, 1]$, where m is the total number of tasks. A larger IWR indicates a better work relatedness. The maximum is attained when, for every station, all its tasks form one subnetwork. IWR is a relative measure that needs to consider the number of stations. It tends to increase as the number of stations increases. Therefore, direct comparison of two IWR values can be meaningful when the number of stations is the same.

Although IWR is simple to use, it cannot fully capture the general meaning of work relatedness. The relatedness might mean that the tasks require a common tool or a common set-up or that a human sees some larger meaning or goal in the set of tasks. If some of these are practically important and relevant information is available, the IWR measure that tends to capture only the idea of task precedence may have to be redefined.

Another criterion is the work slackness of tasks, which is roughly defined as the amount of slack time between two related tasks. Often, in a two-sided assembly line, a task and its immediately precedent task can be assigned separately to the opposite sides of a mated-station due to the constraint of operation directions. When the precedent task is stalled due to some reasons, such as defective components or poor work, this delays the succeeding task and can further influence the following tasks in a cascading manner. If the task sequence can be adjusted so that some amount of slack time is inserted between the tasks, then this alleviates the tight operation between two tasks and the cascading delay is discontinued. We propose index of work slackness (IWS), a new measure to quantify the tightness of task sequences, as follows:

$$\mathrm{IWS} = \frac{\sum_{j=1}^{n}\left\{\sum_{i\in Q_j}\sqrt{t_{ij}^{\mathrm{s}} - t_{i'j'}^{\mathrm{f}}} + \sum_{i\in I_j - Q_j}\sqrt{\mathrm{CT}}\right\}}{m\sqrt{\mathrm{CT}}}, \tag{2}$$

where CT is the cycle time, j and j' form a mated-station, I_j is the set of tasks that are assigned to station j, Q_j is the set of tasks such that it is a subset of I_j, and for each of the tasks, one of its immediately preceding task is assigned to station j', and t_{ij}^{s} and t_{ij}^{f} are, respectively, the start time and the finish time of task i ($\in I_j$) at station j.

Let us consider a station j and all the tasks assigned to the station, i.e. I_j. Suppose that task i is an element of Q_j. Then, at least one of the immediately preceding tasks of i is assigned to the companion of station j. Let k be an immediately preceding task of i and at the same time be a task assigned to the companion of station j, i.e. $k \in I_{j'}$. Among all such tasks, the task, of which the finish time is the latest, is denoted by i'. It is on the completion of i' that task i can be started. The slackness of task i is evaluated by the square root of the difference between the start time of i and the finish time of i', i.e. $\sqrt{t_{ij}^{\mathrm{s}} - t_{i'j'}^{\mathrm{f}}}$. On the other hand, if the immediately preceding tasks of i are assigned to station j or other prior stations, i.e. $i \in I_j - Q_j$, then the immediately preceding tasks do not cause any unavoidable delay. In this case, the slackness of i is set to a constant, $\sqrt{\mathrm{CT}}$. All the task slackness values are added up over all stations and divided by the maximum possible slackness value, $m\sqrt{\mathrm{CT}}$. By taking square root, it is attempted to give preference to a sequence having more even slackness.

Notice that the index has a value in the range of (0,1). The larger the index, the better is the work slackness. IWS is equal to 1 when all the related tasks are assigned to the same side. Attempting to maximize IWS tends to put some room between two related tasks that are assigned to companion stations. This can be achieved by modifying the task sequence within a station. That is, for the tasks that do not have precedence relations, their sequence may be flexibly adjusted to improve the work slackness. Like IWR, IWS is also a relative measure that tends to increase as the number of stations increases.

3. Group assignment procedure

We develop a group assignment procedure of balancing two-sided assembly lines. The procedure assigns a group of tasks rather than a unit task, focusing on the work relatedness and slackness addressed in the Section 2. Agrawal (1985) proposed a rule for balancing one-sided assembly lines, namely the largest set rule, to improve work relatedness. This method first forms a set of task groups that can be assigned to the station under current assignment. A task group is composed of a task and all its predecessors that are not assigned yet. Such groups are repeatedly created for every task that is not assigned to any station yet. We adopted this idea in creating task groups. However, as discussed earlier, balancing a two-sided assembly line needs to consider operation directions and sequence dependency of tasks. The new procedure takes account of these two factors.

3.1. Grouping tasks

A procedure of forming task groups is described in this section. Prior to presenting the overall procedure, we describe the basic ideas and several rules that are used in the procedure.

First, the procedure begins with forming initial task groups considering the operation directions of tasks. It is disallowed for a group to contain both L- and R-tasks. Consider, for example, the precedence diagram in Fig. 2. Suppose that the first seven tasks have already been allocated to the first mated-station, and that the second mated-station is currently considered. Following Agrawal's method directly, we get nine task groups, i.e. {8}, {9}, {10}, {8,11}, {9,12}, {9,10,13}, {8,11,14}, {8,9,11,12,15}, and {9,10,13,16}. The two task groups, {9,12} and {8,9,11,12,15}, are removed from further consideration since they contain both L- and R-tasks.

Second, an operation direction is determined for each task group. A group having at least one L-task (R-task) should be allocated at a left-side (right-side) station. However, if all the tasks in a group are E-tasks, the group can be allocated to any side. The two rules presented below are used to determine the operation directions for such groups.

DR 1: Set the operation direction to the side where tasks can be started earlier.
DR 2: When the start time at both sides is the same, set the operation direction to the side where it is expected to carry out a less amount of tasks.

In the second rule, the expected amount of tasks for a side can be roughly assessed by the total task processing time that are not assigned yet, but designated to the side. For example, the task group, {8,11,14}, in the previous example, consists of all E-tasks. If workers at both the left and right stations can begin the tasks at the same time, the second rule is invoked. Since the total operation time of unassigned L-tasks is 5 and that of R-tasks is 9, the operation direction is set to the left.

Third, we need to sequence the tasks and check the cycle time constraint for each task group formed. In balancing a two-sided assembly line, the constraint is partly affected by the sequence of tasks as discussed earlier. Therefore, we need to have a method of determining the sequence of tasks. A task in a task group can be sequenced if all its precedents have already been sequenced. Often, there is a multiple of such tasks. The following two rules are proposed to choose one among them:

SR 1: Select the task whose start time is the earliest.
SR 2: Select the task (i) such that it has immediate succeeding tasks that are not contained in the task group currently considered and the operation directions of the succeeding tasks are either opposite to i's operation direction or E-type. When the number of such tasks is more than one, the task with the largest operation time is selected. (A task is selected at random to break ties.)

SR1 is first applied. This rule gives a priority to the task inducing the shortest idle time. If ties occur, then SR2 is used. This rule is intended to maximize work slackness. For example, consider the task group of {9,10,13,16} in our example. Since tasks 9 and 10 have no preceding task, either of the two can be performed earlier than the other. Suppose that their start time is the same. Then, SR2 comes into play. Task 10 and its immediate successor task 13 are contained in the same group, whereas task 9 and its immediate successor task 12 are not in the same group. Notice that task 12 should be carried out in the opposite side of task 9. Thus, SR2 directs to choose task 9. This second rule tends to reduce the potential delay of succeeding tasks.

A task is selected and sequenced using the above rules, and then the task is removed from the task group. This is repeated until all the tasks of the task group under consideration are sequenced.

For a task group to be a candidate for being assigned to the station currently considered, its last task should be able to be completed before the cycle time. Once the task sequence of a task group is determined, it can be easily checked if the group can be fit into the station. Computing the completion time of the group can do this. The computation should consider the unavoidable delay caused by the preceding tasks assigned to companion station. In our example, every group can be a candidate for the station since all of them can be finished earlier than the cycle time.

Finally, we remove the task groups that are a proper subset of others. In the example, {8}, {9}, {10}, {8,11}, and {9,10,13}, are excluded, and {8,11,14} and {9,10,13,16} remain as the final candidates.

The above procedure of forming task groups is summarized in algorithmic form. Let U denote the set of tasks that are not assigned yet, and G_i be a task group consisting of task i and all of its precedent tasks in U.

Step 1. Set TU: $= U$, and FS: $= \emptyset$.
Step 2. If TU $= \emptyset$, then go to Step 7. Otherwise, choose an arbitrary task i among the tasks in TU and have no precedent task.
Step 3. Identify G_i. If G_i contains both left tasks and right tasks, then remove task i and all its succeeding tasks from TU and go to Step 2.
Step 4. Determine the operation direction of G_i. If G_i has no R-task (L-task), set its operation direction to left (right). Otherwise, i.e. if all the tasks are E-tasks, determine its direction using DR1 and DR2.
Step 5. Determine the sequence of tasks in G_i using SR1 and SR2.
Step 6. If the last task in G_i can be completed before cycle time, update FS as FS: $=$ FS $\cup \{G_i\}$, delete task i from TU, and go to Step 2. Otherwise, remove task i and all its succeeding tasks from TU and go to Step 2.
Step 7. For every task group of FS, remove it from FS if it is a proper subset of another task group of FS.

The resulting task groups in FS become the candidates for the group assignment described in Section 3.2.

3.2. Assignment of task groups

The candidate task groups produced by the procedure in Section 3.1 do not violate precedence, cycle time, and operation direction constraints. Although any of the candidates can be assigned to the current station under consideration, the following assignment rules (AR) are proposed to select one:

AR 1: Select the task group that can be started at the earliest time.
AR 2: Select the task group that involves the minimum delay.
AR 3: Select the task group that requires the maximum operation time.

Let us consider a left- and right-side station of a mated-station. Suppose that some tasks have been already assigned to the stations. If the two stations are considerably unbalanced, it is likely that the station having less amount of tasks would involve more delay in the future assignment. Therefore, improving balance efficiency (BE) is important for not only minimizing the number of stations but also reducing chances of unavoidable delay. AR1 aims at improving the BE. This rule gives a higher priority to the station, where less amount of tasks is allocated. The rule is first applied to candidate task groups. If ties occur, then AR2 is considered. This rule chooses a group that involves the least amount of delay, which is concerned with both BE and work slackness. This rule may also result in a tie. Then, AR3, the same with Agrawal's assignment rule, is applied. The third rule tends to enhance work relatedness. This is clear considering that tasks in a group are all connected in the network representation. A tie with this rule is broken by random selection.

3.3. Overall procedure

Using the rules for grouping and assigning tasks described in previous sections, we have developed an iterative procedure to solve two-ALB problems. Let j and j', respectively, denote a left-side and right-side station of a mated-station, and S_j denote the start time at station j. D_k and T_k, respectively, denote the amount of delay and the total operation time required for performing the tasks in G_k. The overall procedure is presented below:

Step 1. Set up $j := 1$, $j' := j + 1$, $S_j := S_{j'} := 0$, and $U :=$ the set of all the tasks to be assigned.
Step 2. Run the task grouping procedure in Section 3.1, which identifies $\mathrm{FS} = \{G_1, G_2, \ldots, G_K\}$.If $\mathrm{FS} = \emptyset$, go to Step 6.
Step 3. For every G_k, $k = 1,2, \ldots, K$, compute D_k and T_k.
Step 4. Identify one task group G_r from FS using the AR in Section 3.2.
Step 5. Assign G_r to a station j (or j') according to its operation direction, and update $S_j := S_j + D_r + T_r$ (or $S_{j'} := S_{j'} + D_r + T_r$). $U := U - \{G_r\}$, and go to Step 2.
Step 6. If $U \neq \emptyset$, set $j := j' + 1$, $j' := j + 1$, $S_j := S_{j'} := 0$, and go to Step 2. Otherwise, stop.

3.4. An example

The use of the proposed procedure is demonstrated using the example shown in Fig. 2. Suppose that the cycle time is fixed to 22. The first iteration of the procedure is given below, and all the remaining iterations are summarized in Table 1, where the first column indicates the number of iteration.

Iteration 1
Step 1. $j := 1$, $j' := 2$, $S_j := S_{j'} := 0$, and $U := \{1,2,\ldots,16\}$.
Step 2. Three candidate task groups, $G_1 = \{1,3,6\}$, $G_2 = \{1,4\}$, and $G_3 = \{2,5\}$, are produced. The operation directions of groups G_1 and G_3 are already fixed to left and right, respectively. The tasks in G_2 are all E-types. Since the total operation time of unassigned L-tasks is 11 ($= t_3 + t_6 + t_{12}$) and that of R-tasks is 17 ($= t_5 + t_9 + t_{10}$), the operation direction of G_2 is set to left. For each group, a task sequence is uniquely determined by the precedence relations.
Step 3. $D_1 = D_2 = D_3 = 0$, and $T_1 = 12$, $T_2 = 15$, and $T_3 = 13$.

Table 1
List of iteration for the example problem

Iteration	Step 1	Step 2		Step 3		Step 4	Step 5	Step 6
		G_k	d_k	D_k	T_k			
1	$j := 1$ $j' := 2$ $S_1 := 0$ $S_2 := 0$ $U = \{1, 2, \ldots, 16\}$	$G_1 = \{1, 3, 6\}$ $G_2 = \{1, 4\}$ $G_3 = \{2, 5\}$	L (0) L (0) R (0)	0 0 0	12 15 13	$G_r = G_2$ (AR3)	G_2 is assigned to station 1. $S_1 := 15$ $U := U - \{1, 4\}$	–
2	–	$G_1 = \{3, 6\}$ $G_2 = \{2, 5, 7\}$	L (15) R (0)	0 2	6 20	$G_r = G_2$ (AR1)	G_2 is assigned to station 2. $S_2 := 22$ $U := U - \{2, 5, 7\}$	–
3	–	$G_1 = \{3, 6\}$	L (15)	0	6	$G_r = G_1$	G_1 is assigned to station 1. $S_1 := 21$ $U := U - \{3, 6\}$	–
4	–	No candidate	–	–	–	–	–	$j := 3$ $j' := 4$ $S_3 := 0$ $S_4 := 0$
5	–	$G_1 = \{8, 11, 14\}$ $G_2 = \{9, 10, 13, 16\}$	L (0) R (0)	0 0	14 19	$G_r = G_2$ (AR3)	G_2 is assigned to station 4. $S_4 := 19$ $U = \{8, 11, 12, 14, 15\}$	–
6	–	$G_1 = \{8, 11, 14\}$ $G_2 = \{8, 11, 12, 15\}$	L (0) L (0)	0 0	14 18	$G_r = G_2$ (AR3)	G_2 is assigned to station 3. $S_3 := 18$ $U = \{14\}$	–
7	–	$G_1 = \{14\}$	L (18)	0	4	$G_r = G_1$	G_1 is assigned to station 4 $S_4 := 22$ $U = \varnothing$	–
8	–	No candidate	–	–	–	–	–	Done

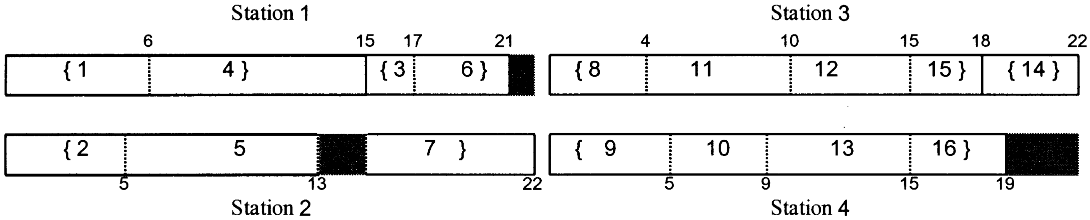

Fig. 3. Results of the proposed procedure for the example problem.

Step 4. Since all the groups can be begun at time 0 and no delay is involved, AR3 is used to select a group. G_2, which has the largest operation time, is selected.
Step 5. G_2 is assigned to station 1. $S_1 := S_1 + D_2 + T_2 := 0 + 0 + 15$. Tasks 1 and 4 are removed from U. Go to Step 2.

In the second column of Step 2, d_k indicates the operation direction of the kth task group. The directions determined by the DR rules are underlined. The number in parentheses is the start time of the group at its relevant station. In Fig. 3, the results of the above task assignments are shown in a Gantt chart. Task ID numbers are placed at their relevant positions inside the bars, and task groups are enclosed in a pair of brackets. 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. The resulting work relatedness (IWR) and work slackness (IWS) are 1.0 and 0.905, respectively.

4. Experiments

The proposed procedure is compared with Bartholdi's (1993) single-pass heuristic and several existing heuristic rules for one-sided assembly line balancing that we have modified for two-ALB. The proposed method and the compared heuristics were coded in C++ and implemented on an IBM-PC with a Pentium CPU at 166 MHz. Due to randomness involved in some of the procedures, a solution produced by a run can be different from that by others even with the same procedure. Therefore, we repeated each experiment 30 times for every problem setting, and the best average among the heuristic rules and the average resulted from the group assignment procedure are reported here. Every run consumes very little computation time (less than 3 s in CPU time) because all the procedures are single pass heuristics.

4.1. Heuristic rules

There exist several single-pass heuristic rules proposed for one-sided assembly line balancing. These include (1) MAX-DUR (Moodie & Young, 1965) that selects the task having the maximum duration (operation) time, (2) MAX-TFOL (Talbot et al., 1986) that selects the task having the maximum total number of follower tasks, (3) MAX-IFOL (Tonge, 1961) that selects the task having the maximum number of immediate follower tasks, and (4) MAX-RPW (Helgeson & Birnie, 1961) that selects the task having the maximum ranked positional weight, where the ranked positional weight of a task is the sum of the operation time of the task and all its succeeding tasks. These are generally known as good heuristic rules for minimizing the number of stations in one-sided assembly line balancing problems.

Provided below is a procedure to apply the above heuristic rules to two-ALB problems:

Step 1. Generate the initial two stations, $j := 1$ and $j' := j + 1$.
Step 2. Identify a set F whose elements are such tasks that can be assigned to either station j or j'. If $F = \emptyset$, go to Step 5.
Step 3. Select a task from F using a predetermined assignment rule.
Step 4. When the operation direction of the task selected in Step 3 is left (right), assign it to station j (j'). If the task can be performed at either side of the line, assign it to the station where it can be started earlier. If the start time at both stations is the same, assign it to the side that is expected to carry out less amount of tasks. Go to Step 2.
Step 5. If all the tasks are assigned, stop. Otherwise, create a new mated-station and increase the station number as $j := j' + 1$ and $j' := j + 1$, and go to Step 2.

In Step 2, the set, F, is composed of such tasks that their predecessors are all assigned, and at the same time, they satisfy the cycle time constraint. One of the four rules (MAX-DUR, MAX-TFOL, MAX-IFOL, and MAX-RPW) can be used in Step 3 to select a task. When two or more tasks tie in this step, one task is arbitrarily selected. Step 4 actually assigns the selected task to a left or right station according to its operation direction. When the task does not have any preferred operation direction, the current or expected workload of stations is considered. This is actually the same with the DR rules used to form task groups in Section 3.1. The procedure is repeated until all the tasks are assigned.

Bartholdi's (1993) heuristic for two-ALB is also used in the comparison. This is a modified version of first fit rule (FFR), which is a well-known heuristic for bin packing problem (Wee & Magazine, 1982).

4.2. Experimental results

Line balancing problems are generally categorized into two classes. One is a Type-I problem where cycle time is given and the objective is minimizing the number of stations. The other is a Type-II problem, of which the objective is to minimize the cycle time while the number of stations is predetermined. The experiments were carried out on three problems: B148, A65, and A205. Problem B148, which has 148 tasks, is obtained from Bartholdi (1993). In this problem, the operation time of task 79 and 108 is, respectively, changed from 281 to 111 and from 383 to 43. Since the original values are much larger than the others, it imposes a limit on the number of stations. The other two problems, A65 and A205, are practical problems obtained from a truck assembly line of AAA automobile company by the authors. The problems have 65 and 205 tasks, respectively. (These two problems, presented in Appendix A, may be useful for benchmarking test by other researchers.) The total work content of A65, B148 and A205 is 5099, 5122 and 23345, respectively. The three test-bed problems can be viewed from either Type-I or Type-II.

Table 2
Performance comparison for Type-I problems

Test-bed problems	CT	NS			NM			IWR			IWS		
		H	*G*	IR (%)	*H*	*G*	IR (%)	*H*	*G*	IR (%)	*H*	*G*	IR (%)
A65	326	17.7[a]	17.4	1.7	9.1	9.0	1.1	0.36	0.47	30.6	0.87	0.95	9.2
	381	15.7[b]	15.0	4.5	8.0	8.0	0.0	0.33	0.45	36.4	0.87	0.93	6.9
	435	14.0[c]	13.4	4.3	7.0	7.0	0.0	0.31	0.48	54.8	0.91	0.92	1.1
	490	12.1[b]	12.0	0.8	6.1	6.0	1.6	0.28	0.38	35.7	0.85	0.91	7.1
	544	11.5[a]	10.6	7.8	5.8	5.6	3.4	0.27	0.38	40.7	0.84	0.93	10.7
B148	204	27.8[a]	27.0	2.9	14.0	14.0	0.0	0.30	0.46	53.3	0.87	0.94	8.0
	255	22.0[b]	21.0	4.5	11.0	11.0	0.0	0.23	0.40	73.9	0.87	0.94	8.0
	306	19.3[d]	18.0	6.7	9.9	9.0	9.1	0.22	0.36	63.6	0.83	0.92	10.8
	357	16.0[a]	15.0	6.3	8.0	8.0	0.0	0.17	0.31	82.4	0.86	0.93	8.1
	408	14.0[b]	14.0	0.0	7.0	7.0	0.0	0.16	0.29	81.3	0.85	0.92	8.2
	459	12.1[b]	13.0	− 7.4	6.1	7.0	− 14.8	0.14	0.29	107.1	0.84	0.91	8.3
	510	12.0[a]	11.0	8.3	6.0	6.0	0.0	0.14	0.28	100.0	0.83	0.92	10.8
A205	1133	24.0[c]	23.0	4.2	12.0	12.0	0.0	0.26	0.29	11.5	0.87	0.90	3.4
	1322	21.9[d]	20.7	5.5	11.0	10.3	6.4	0.25	0.28	12.0	0.82	0.89	8.5
	1510	18.7[d]	20.0	− 7.0	9.5	10.0	− 5.3	0.23	0.26	13.0	0.80	0.88	10.0
	1699	16.7[d]	16.0	4.2	8.5	8.0	5.9	0.21	0.26	23.8	0.80	0.89	11.3
	1888	15.4[d]	16.0	− 3.9	7.8	8.0	− 2.6	0.19	0.24	26.3	0.80	0.90	12.5
	2077	14.0[d]	14.0	0.0	7.0	7.0	0.0	0.18	0.21	16.7	0.79	0.86	8.9
	2266	12.5[d]	13.0	− 4.0	6.4	7.0	− 9.4	0.18	0.21	16.7	0.78	0.84	7.7
	2454	12.0[c]	12.0	0.0	6.0	6.0	0.0	0.19	0.19	0.0	0.81	0.85	4.9
	2643	11.2[d]	12.0	− 7.1	5.7	6.0	− 5.3	0.16	0.20	25.0	0.78	0.87	11.5
	2832	10.0[d]	10.0	0.0	5.0	5.0	0.0	0.16	0.15	− 6.3	0.77	0.83	7.8

[a] MAX-RPW.
[b] MAX-TFOL.
[c] MAX-DUR.
[d] FFR.

Various lengths of cycle time were given for the Type-I problems. For each test-bed problem, the length of cycle time is determined by multiplying constants by the maximum task processing time. (The constants used are as follows: 1.2, 1.4, 1.6, 1.8, 2.0 for problem A65; 1.2, 1.5, 1.8,…,3.0 for B148; 1.2, 1.4, 1.6,…,3.0 for A205.)

The experimental results for Type-I problem are shown in Table 2. Four criteria, i.e. the number of stations (NS), the number of mated-stations (NM), the IWR, and the IWS, are used in this experiment. NS is the number of stations, where tasks are actually assigned. NM is the number of mated-stations, which in fact can be considered as the length of two-sided assembly line. Tasks may sometimes be assigned to only one side of a mated-station because of the constraints of operation directions and precedence relations, as already mentioned. Therefore, NS may not be equal to $2 \times \mathrm{NM}$. The best solution among those obtained from five heuristic rules is chosen, and this is compared with the solution from the proposed group assignment. To choose the heuristic solution, we first consider NS. If a tie occurs with the NS values, then NM, IWR, and IWS are considered in this order. In the third column, the

Table 3
Performance comparison for Type-II problems

Test-bed problems	NS	CT			BE			IWR			IWS		
		H	*G*	IR (%)	*H*	*G*	IR (%)	*H*	*G*	IR (%)	*H*	*G*	IR (%)
A65	8	682.7[a]	660.5	3.3	0.93	0.97	3.3	0.22	0.32	45.5	0.86	0.90	4.7
	10	547.2[b]	529.0	3.3	0.93	0.96	3.4	0.26	0.38	46.2	0.83	0.92	10.8
	12	459.6[a]	453.5	1.3	0.92	0.94	1.3	0.28	0.42	50.0	0.86	0.93	8.1
	14	392.8[c]	393.1	− 0.1	0.93	0.93	− 0.1	0.32	0.45	40.6	0.90	0.94	4.4
	16	343.8[a]	338.0	1.7	0.93	0.94	1.7	0.35	0.50	42.9	0.88	0.93	5.7
B148	8	667.7[b]	663.0	0.7	0.96	0.97	0.7	0.11	0.17	54.5	0.79	0.90	13.9
	10	540.6[b]	538.0	0.5	0.95	0.95	0.5	0.13	0.22	69.2	0.79	0.92	16.5
	14	388.2[c]	385.1	0.8	0.94	0.95	0.8	0.15	0.29	93.3	0.82	0.91	11.0
	16	340.4[b]	331.1	2.7	0.94	0.97	2.8	0.18	0.29	61.1	0.83	0.93	12.0
	18	304.7[b]	299.0	1.9	0.93	0.95	1.9	0.20	0.39	95.0	0.83	0.93	12.0
	20	270.8[a]	273.5	− 1.0	0.95	0.94	− 1.0	0.21	0.38	81.0	0.86	0.94	9.3
	22	242.8[c]	243.1	− 0.1	0.96	0.96	− 0.1	0.22	0.39	77.3	0.87	0.93	6.9
A205	8	3198.1[b]	3285.0	− 2.7	0.91	0.89	− 2.7	0.13	0.15	15.4	0.76	0.83	9.2
	10	2565.4[b]	2799.0	− 9.1	0.91	0.83	− 8.4	0.15	0.16	6.7	0.77	0.86	11.7
	12	2152.1[b]	2150.0	0.1	0.90	0.91	0.2	0.17	0.18	5.9	0.79	0.86	8.9
	14	1841.5[b]	1996.6	− 8.4	0.91	0.84	− 7.8	0.17	0.22	29.4	0.80	0.88	10.0
	16	1638.0[b]	1623.1	0.9	0.89	0.90	0.9	0.20	0.23	15.0	0.80	0.89	11.3
	18	1442.3[b]	1415.0	1.9	0.90	0.92	1.9	0.22	0.22	0.0	0.82	0.88	7.3
	22	1194.6[b]	1145.0	4.2	0.89	0.93	4.3	0.25	0.28	12.0	0.84	0.91	8.3
	24	1088.6[b]	1034.9	4.9	0.89	0.94	5.2	0.26	0.31	19.2	0.85	0.91	7.1
	26	1021.9[b]	996.1	2.5	0.88	0.90	2.6	0.27	0.32	18.5	0.84	0.90	7.1
	28	956.5[b]	944.6	1.2	0.87	0.88	1.3	0.28	0.34	21.4	0.85	0.91	7.1

[a] MAX-TFOL.
[b] FFR.
[c] MAX-RPW.

superscripts indicate the heuristic rule that generates the best solution. For a chosen solution, their NM, IWR, and IWS are computed. H and G stand for heuristic rules and group assignment, respectively. The improved rate (IR) for NS and NM was calculated as follows:

$$\mathrm{IR} = 100(H - G)/H.$$

The IR for IWR and IWS was calculated as follows:

$$\mathrm{IR} = 100(G - H)/H.$$

A Type-II problem is to minimize cycle time for a given number of stations. Such a problem is usually solved by an iterative use of the procedures for Type-I problems. For the given number of stations, the theoretical lower bound on cycle time is first computed. This lower bound can be derived by dividing the total amount of operation time by the number of stations. An initial iteration of assignment procedure uses this cycle time. If the resulting number of stations is greater than the given number, the cycle time is incremented by one time unit and the procedure is executed again. This is repeated until the given number of stations is attained. The results of this experimentation are shown in Table 3. The table shows the resulting CT, BE, IWR, and IWS. BE is the ratio of work content to the amount of available work time given to all the stations. The IRs in the tables are similarly computed as in Type-I problems.

The two sets of experimental results show that our group assignment method generally outperforms the heuristic rules in terms of IWR and IWS. The average IRs for IWR and IWS, respectively, amount to 40.84 and 8.35% for Type-I problems, and 40.91 and 9.24% for Type-II problems. Such a great improvement is perhaps due to the fact that the proposed group assignment procedure is mainly concerned with work relatedness and work slackness. With regard to the traditional measures of the number of stations, cycle time, and balance efficiency, the proposed procedure gives better solutions for some problem instances, but for others, the improvement in IWR and IWS requires the sacrifice of those measures. There are certainly various measures in line balancing problems, and those are not equally important. Therefore, our conclusion is that when the work relatedness and work slackness are critical, the proposed method can provide good solutions without significantly degrading the number of stations, cycle time, and balance efficiency.

5. Summary and further research issues

Two-ALB problems are considered, and a new heuristic procedure, namely group assignment, is developed. In this procedure, assignments are carried out based on task groups rather than individual tasks. We propose a method of constructing candidate groups and rules of assigning them. The experimental results demonstrate that the proposed procedure can improve the work relatedness and the work slackness with a little or no loss in cycle time and the number of stations.

The two-sided assembly lines dealt with in this paper are simple in that it considers the operation directions of tasks and the sequence dependency of task processing. The lines in the real world could be more complicated by such restrictions as symmetric tasks, synchronous tasks, task separation, and station layout. For instance, symmetric tasks are two tasks such that one task is simply the mirror image task of the other. Wiring harness, assembling springs, and mounting tires are typical symmetric tasks. Although the two tasks can be performed independently at different locations, they are usually assigned to one mated-station.

Another issue in real line balancing problems is that they usually call for simultaneous consideration of multiple objectives. Although we focus mainly on work relatedness and work slackness, this can be further extended into multi-objective problems.

Acknowledgements

We thank the anonymous referee for the constructive comments, especially on the decision criteria. The comments have greatly improved the quality of the paper.

Appendix A

Table A1. Problem A65

Task	Side	Task time	Immediate successors
1	E	49	3
2	E	49	3
3	E	71	4,23
4	E	26	5,6,7,9,11,12,25,26,27,41,45,49
5	E	42	14
6	E	30	14
7	R	167	8
8	R	91	14
9	L	52	10
10	L	153	14
11	E	68	14
12	E	52	14
13	E	135	14
14	E	54	15,18,20,22
15	E	57	16
16	L	151	17
17	L	39	31
18	R	194	19
19	R	35	21
20	E	119	21
21	E	34	31
22	E	38	31
23	E	104	24
24	E	84	31
25	L	113	31
26	R	72	31
27	R	62	28
28	R	272	50
29	L	89	50
30	L	49	50
31	E	11	32,36,51,52,53,54,55,56,58,59,60,61,62
32	E	45	33
33	E	54	34
34	E	106	35
35	R	132	50
36	E	52	37
37	E	157	38
38	E	109	39,40
39	L	32	50
40	R	32	50
41	E	52	42
42	E	193	43
43	E	34	62
44	R	34	46
45	L	97	46
46	E	37	47
47	L	25	48
48	L	89	50
49	E	27	50

(*continued*)

Task	Side	Task time	Immediate successors
50	E	50	66
51	R	46	65
52	E	46	65
53	L	55	65
54	E	118	65
55	R	47	65
56	E	164	57
57	E	113	65
58	L	69	65
59	R	30	65
60	E	25	65
61	R	106	65
62	E	23	63
63	L	118	64
64	L	155	65
65	E	65	–

Table A2. Problem A205

Task	Side	Task time	Immediate successors
1	E	692	36
2	E	42	3,4
3	R	261	5
4	L	261	5
5	E	157	7,13
6	E	90	36
7	R	54	8
8	R	67	9
9	R	30	10
10	R	106	11
11	R	32	12
12	R	62	36
13	L	54	14
14	L	67	15
15	L	30	16
16	L	106	17
17	L	32	18
18	L	62	36
19	E	56	36
20	E	67	22
21	E	86	22
22	E	37	23
23	E	41	24,34
24	E	72	26,27,28
25	R	86	28
26	L	16	35
27	R	51	35
28	R	66	29
29	R	41	30,33
30	R	72	31,32

(*continued*)

Task	Side	Task time	Immediate successors
31	R	51	35
32	R	16	35
33	R	15	35
34	L	15	35
35	E	85	36
36	E	59	37,40,41,42,62,69,72,75,83,110,111,112
37	L	23	38
38	L	13	39
39	L	19	45
40	E	108	43,54
41	E	214	92
42	E	80	43,54
43	L	37	44
44	L	84	45
45	L	18	46,48,51,53
46	L	12	47
47	L	29	92
48	L	37	49
49	L	13	50
50	L	70	92
51	L	217	52
52	L	72	92
53	L	85	92
54	R	43	55
55	R	97	56,59,61
56	R	37	57
57	R	13	58
58	R	35	92
59	R	217	60
60	R	72	92
61	R	85	92
62	E	25	63
63	E	37	64
64	E	37	65,68
65	E	103	66
66	E	140	67
67	E	49	80
68	E	35	80
69	E	51	70
70	E	88	71
71	E	53	73
72	E	144	73
73	E	337	74
74	E	107	76
75	E	371	92
76	E	97	77,78,79
77	E	166	80,82
78	L	92	80
79	R	92	80
80	E	106	81
81	E	49	84
82	E	92	92
83	E	371	92

(*continued*)

Task	Side	Task time	Immediate successors
84	E	87	85
85	E	162	86,88,90
86	E	96	87
87	E	79	92
88	E	96	89
89	E	42	92
90	R	88	91
91	R	90	92
92	R	97	93,94,95,96,97,98,99
93	R	270	135
94	E	452	135
95	R	48	113
96	E	338	113
97	E	34	100
98	E	65	100
99	E	50	100
100	E	112	101,103,105,109,130,131,134
101	E	48	102
102	E	117	113
103	E	50	104
104	R	68	113
105	L	232	106,107
106	L	122	108
107	E	151	108
108	L	31	113
109	E	97	113
110	R	308	113
111	L	116	113
112	R	312	113
113	E	34	114,115,116,117,118,119,120,121,122,123,124,161,162,163,169,171,174, 203,204,205
114	L	128	160
115	E	54	160
116	R	175	160
117	E	55	160
118	E	306	126
119	E	59	126
120	E	59	126
121	E	66	126
122	E	66	126
123	E	23	126
124	E	244	125
125	E	54	126
126	R	294	127,128,129
127	E	84	135
128	E	61	135
129	E	57	135
130	R	38	136
131	E	944	132
132	R	511	133
133	R	625	189
134	R	445	189
135	L	68	136,137,138,139,140,141,142,144,145,147,148,149,150,151,152,153,158

(*continued*)

Task	Side	Task time	Immediate successors
136	L	53	189
137	E	49	160
138	E	92	160
139	E	236	160
140	L	116	143
141	L	265	143
142	L	149	143
143	L	74	160
144	E	332	160
145	E	324	146
146	L	104	160
147	L	51	160
148	R	58	160
149	R	67	160
150	R	49	160
151	E	107	160
152	L	38	160
153	L	27	154
154	E	68	155
155	E	207	156
156	E	202	157
157	E	83	189
158	R	35	159
159	R	58	189
160	E	42	164,170,178,179,184
161	R	68	167
162	R	68	165
163	R	68	164
164	R	103	165
165	R	103	166
166	R	103	167
167	R	103	168
168	R	103	177
169	L	68	170
170	L	103	172
171	L	68	172
172	L	103	173
173	L	103	175
174	L	68	175
175	L	103	176
176	L	103	177
177	E	10	185,186,187,188,194,195
178	E	187	180
179	L	134	180
180	L	89	181,183
181	L	58	182
182	L	49	–
183	L	134	–
184	L	53	–
185	E	334	189
186	R	24	189
187	R	76	189
188	L	76	189

(*continued*)

Task	Side	Task time	Immediate successors
189	E	192	190,191,193
190	E	98	–
191	R	258	192
192	E	165	–
193	R	38	–
194	E	115	197
195	L	83	196
196	R	56	197
197	R	29	198,199,201
198	R	303	–
199	R	18	200
200	R	29	–
201	L	154	202
202	L	90	–
203	L	93	–
204	E	94	–
205	E	165	–

References

Agrawal, P. K. (1985). The related activity concept in assembly line balancing. *International Journal of Production Research*, *23*, 403–421.

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

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

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

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

Kim, Y. K., Kim, Y., & Lee, T. O. (1999). Two-sided assembly line balancing models. Technical Report. Chonnam National University, Kwangju.

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

Moodie, C. L., & Young, H. H. (1965). A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *Journal of Industrial Engineering*, *16*, 23–29.

Talbot, F. B., Patterson, J. H., & Gehrlein, W. V. (1986). A comparative evaluation of heuristic line balancing techniques. *Management Science*, *32*, 430–454.

Tonge, F. M. (1961). *A heuristic program for assembly line balancing*, Englewood Cliffs, NJ: Prentice-Hall.

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

van Zante-de Fokkert, J. I., & de Kok, T. G. (1997). The mixed and multi model line balancing problem: a comparison. *European Journal of Operational Research*, *100*, 399–412.