Science and Technology

RMUTT Journal

Vol.5 No.2 (2015) : 96-117
www.sci.rmutt.ac.th/stj
Online ISSN 2229-1547

Double Assembly Line Balancing Algorithms on Real-world Instances of Producing Digital Rice Cookers and Digital Hot Pots

Patawee Phiphatsomporn[1] and Pisut Pongchairerks[1*]

[1] Faculty of Engineering
Thai-Nichi Institute of Technology, Suan Luang, Bangkok 10250
*E-mail: pisut@tni.ac.th

Abstract

This paper considers actual assembly line balancing problem's instances of producing digital rice cookers and digital hot pots by a well-known Japanese manufacturer. To efficiently solve these instances, this paper proposes four assembly line balancing algorithms. Each proposed algorithm arranges the given work elements into the workstations by using the original precedence diagram as well as the reversed precedence diagram, through a particular simple heuristic. For any instance, each proposed algorithm thus provides two solutions, i.e., a solution for the instance with the original precedence diagram and a solution for the same instance with the reversed precedence diagram. The solution for the reversed precedence diagram is then modified to be usable for the original precedence diagram. The final solution of each proposed algorithm is the best solution between the two given solutions.

Keywords: Largest candidate rule, Ranked positional weight, Most following task, Reversed ranked positional weight, Assembly line balancing problem, ALBP, Assembly line balancing algorithm

Received: Nov 17, 2015
Revised: Dec 24, 2015
Accepted: Dec 22, 2015

1. Introduction

An assembly line is a sequence of workstations sequentially connected by a material handling system, e.g., a conveyor system. All work elements which are necessary for assembling a product unit must be assigned into these workstations without violating the precedence constraints between the work elements. The most important property of the assembly line is that the production rate of the assembly line always equals the production rate of the bottleneck workstation, i.e., the workstation consuming the longest service time among the service times of all workstations in the assembly line. In order to cope with the bottleneck problem, the engineer must rearrange the work elements into the workstations to make them have equal or almost equal service times. This duty of the engineer is so-called the assembly line balance. The decision problem of arranging work elements into workstations for optimally balancing the service times of all workstations is known as the assembly line balancing problem (ALBP).

The research shown in this paper considers the actual ALBP instances given by a famous Japanese manufacturer producing digital rice cookers and digital hot pots, located in Thailand. The engineers of this manufacturer currently cope with ALBP by using an ad hoc procedure. To do so, they first identify which workstation is the bottleneck workstation of the assembly line and then move some work elements from this workstation into the nearby workstations. However, the balance efficiencies for the assembly line given by this ad hoc procedure have no longer been satisfied by the management. This above statement guides this paper to develop the systematic algorithms providing more efficient solutions to the ALBP instances.

The remaining parts of this paper are organized as follows. Section 2 presents the notations and terminologies as well as a literature review corresponding to this paper. Section 3 presents the details of the proposed assembly line balancing algorithms. Section 4 presents the ALBP instances generated from the actual data of the assembly lines producing digital rice cookers and digital hot pots. Section 5 evaluates the performances of the proposed assembly line balancing algorithms on the ALBP instances. Section 6 finally presents the conclusions. This article is a revised and expanded version of a paper presented in the *Proceedings of the International Conference on Science and Technology*, Pathum Thani, Thailand, 2015 [1].

2. Preliminaries

Section 2.1 provides the notations and terminologies about ALBP, which are used in this paper. Section 2.2 later presents the literature corresponding to the research in this paper.

2.1 Notations and Terminologies

The notations and terminologies used in this paper are based on the assumption that, in each assembly line being considered here, each workstation is operated by a single worker. These notations and terminologies are similar to those used in [2] and [3] as follows.

- T_{ek} is the work element time of the work element k. It is the performing time for work element k (k = 1, 2,…, n_e) where n_e is the number of all work elements divided from the total work content. Note that a work element refers to a series of work activities grouped logically based on their unified function, and the total work content refers to the total amount of work required to produce a product unit.
- T_c is the most preferred cycle time of the assembly line, which is the most preferred time-interval between two successive product units exiting the assembly line. The value of T_c is usually assigned by the middle-level management in order to balance between supply and demand. The assembly line thus should operate at the same pace as T_c; however, it is usually allowed to operate at a slightly faster pace than T_c.
- T_r is the repositioning time, i.e., the time required in each cycle to reposition the worker or the work unit or both. In this paper, the repositioning times of all workstations in the same assembly line are equal.
- T_{si} is the service time of the workstation i, i.e., the time to perform all assigned work elements at the workstation i. It is the sum of all work element times assigned to the workstation i (i = 1, 2,…, n) where n is the total number of workstations. $T_{si} = \sum_{k \in i} T_{ek}$.
- T_s is the maximum available service time. It is calculated by subtracting the repositioning time from the most preferred cycle time, i.e., $T_s = T_c - T_r$. The service time of every workstation T_{si} (i = 1, 2,…, n) is not allowed to exceed T_s. In case that every T_{si} (i = 1, 2,…, n) is less than T_s, the assembly line will operate at a faster pace than T_c.
- T_{wc} is the total work content time, i.e., the time to perform the total work content. $T_{wc} = \sum_{k=1}^{n_e} T_{ek} = \sum_{i=1}^{n} T_{si}$.
- E_b is the balance efficiency of the assembly line. $E_b = T_{wc} \div n\text{Max}\{T_{si}\}$. The perfect line balance returns E_b = 100%.
- An original problem refers to an ALBP instance with original precedence constraints. An original precedence diagram is a visual representation of the original precedence constraints.
- A reversed problem refers to an ALBP instance with reversed precedence constraints. A reversed precedence diagram is a visual representation of the reversed precedence constraints. The reversed precedence diagram can be constructed by reversing the directions of all arrows in the original diagram. See [3] for more details.

2.2 ALBP Description and Algorithms in Literature

An assembly line is a sequence of workstations connected altogether with a specific material handling system (e.g., a belt conveyor system) in order to assemble a number of all required parts together to be a finished product unit. Before using the assembly line production, the total work content must be divided into n_e work elements. These n_e work elements are then assigned into the workstations in the order that does not violate the precedence constraints.

To assemble product units, the base parts are launched onto the beginning of the assembly line. Each base part will then be transported from the workstation 1 through the workstation n respectively via the material handling system. Any base part will successfully become a finished product unit, once it has passed through all workstations and reached at the end of the assembly line. The production rate of the assembly line equals to the production rate of the bottleneck workstation, i.e., the workstation consuming the highest service time. Therefore, if the engineer needs to improve the production rate of the assembly line, he must improve the production rate of the bottleneck workstation. To do so, the engineer has to balance the workloads of all workstations in the assembly line by reassigning the work elements into the workstations so that the service times of all workstations are fairly equal. However, this is not so easy, because there are the precedence constraints which are the restrictions on the order in which the work elements can be performed.

The assembly line balancing problem (ALBP) is to decide the proper number of workstations required for the assembly line as well as to assign the work elements into these workstations in order that all constraints are satisfied. The objective of ALBP used in this paper is to maximize the balance efficiency. The constraints of ALBP are as follows: (1) the service time plus the repositioning time of every workstation cannot be higher than the most preferred cycle time (T_c) predefined by the factory, and (2) the precedence constraints cannot be violated. The review articles of ALBP are found in [2-8].

ALBP is related in many fields, e.g., operations research, industrial engineering, supply chain and logistics management. It is an important problem which belongs to the class of NP-hard problems, as mentioned in [9]. The exact methods for ALBP are well surveyed in [4]. The most popular heuristic for ALBP is the *L*argest *C*andidate *R*ule method (LCR) presented in [2, 10]. The procedure of LCR method is shown in the given steps below.

Step 1: List all n_e work elements from left to right in descending order of T_{ek}.

Step 2: Let $i = 1$.

Step 3: Select the leftmost work element in the list which satisfies precedence constraints and does not cause the sum of the work element times of all assigned work elements at the workstation i to exceed T_s. Then, assign this selected work element into the workstation i, and delete this selected work element from the list written in Step 1. If there are no more work elements which can be selected and assigned into the workstation i, go to Step 4; otherwise, repeat Step 3 again.

Step 4: If all n_e work elements have been selected and assigned, stop; otherwise, increase i by 1 and repeat from Step 3.

The *R*anked *P*ositional *W*eight method (RPW) presented in [2, 3, 11] is another well-known simple algorithm for ALBP. The procedure of RPW is similar to that of LCR presented above with only the difference in Step 1. Step 1 for RPW is to list all n_e work elements in descending order of RPW_k (instead of T_{ek}). The RPW_k is the sum of T_{ek} and all other work element times for work elements that follow the work element k in the arrow chain of the precedence diagram. This method has a variant called the *R*eversed *R*anked *P*ositional *W*eight method (R-RPW) shown in [3, 11]. R-RPW is the RPW which is applied to the reversed problem; after the solution for the reversed problem is successfully constructed, it is then un-reversed to become a solution for the original problem.

The *M*ost *F*ollowing *T*ask method (MFT) presented in [12-17] is a variant of LCR method which uses FWE_k (instead of T_{ek}) to assign the priorities for the work elements. The FWE_k is the number of all work elements following the work element k in the arrow chain of the precedence diagram. The procedure of MFT is similar to that of LCR above mentioned with only the difference in Step 1. The MFT's Step 1 is to list all n_e work elements from left to right in descending order of FWE_k. Note that, the word 'task' in the name of the most following *task* method (MFT) shown in [12-17] means 'work element' in this paper. This paper thus renames the most following *task* method (MFT) to the most following *work element* method (MFWE).

Nowadays, meta-heuristic algorithms become popularly used for solving combinatorial problems including ALBP. For example, some meta-heuristic algorithms for ALBP are the genetic algorithms [3, 18, 19], tabu search algorithm [20], and ant colony optimization algorithm [21].

3. Proposed Algorithms

This paper proposes four algorithms as follows: the *D*ouble *L*argest *C*andidate *R*ule method (D-LCR), *D*ouble *R*anked *P*ositional *W*eight method (D-RPW), *D*ouble *M*ost *F*ollowing *W*ork *E*lement method using *L*ongest work element time rule as a secondary criterion (D-MFWEL), and *D*ouble *M*ost *F*ollowing *W*ork *E*lement method using *S*hortest work element time rule as a secondary criterion (D-MFWES). The explanations for these methods are given here in Sections 3.1 through 3.4, respectively.

3.1 Double Largest Candidate Rule Method

The double largest candidate rule method (D-LCR) runs the original largest candidate rule method (LCR) as well as running the reversed largest candidate rule method (R-LCR). The final solution of D-LCR is the best solution between the solutions from LCR and R-LCR. LCR [2] is the largest candidate rule method with the original precedence constraints. The procedure of LCR was already given in Section 2.2. R-LCR is the method which runs LCR for the being considered ALBP instance with the reversed precedence constraints, called the reversed problem. After receiving the solution for the reversed problem, it then transforms this solution into a solution for the original problem by assigning all work elements of the workstation *i* of the reversed problem's solution into the workstation $n - i + 1$ of the original problem's solution. Moreover, for any two work elements assigned in the same workstation, the work element assigned before in the workstation *i* of the reversed problem's solution will be assigned after in the workstation $n - i + 1$ of the original problem's solution. The procedure of D-LCR is given in Algorithm 1.

Algorithm 1 Procedure of D-LCR

Step 1: Let S_1 be the solution of LCR for the being considered instance with the original precedence diagram and S_2 be the solution of LCR for the being considered instance with the reversed precedence diagram. Let S_2^u be the solution which is un-reversed from S_2 to be usable for the original problem. Let S^* be the final solution of D-LCR. Let PD_1 be the original precedence diagram and PD_2 be the reversed precedence diagram. Let *L* be the list of all as-yet-unassigned work elements in descending order of their priorities. Let *t* be a binary parameter to indicate whether the current state is LCR ($t = 1$) or R-LCR ($t = 2$). Now, let $t = 1$.

Step 2: Create the list *L* by listing all n_e work elements from left to right in descending order of T_{ek}.

Step 3: Create the solution S_t by using these following steps:

- Step 3.1: Let $i = 1$.
- Step 3.2: Select the leftmost work element among all existing work elements in the list *L* which satisfy the precedence constraints in PD_t and do not cause the sum of the work element times of all work elements assigned for the workstation *i* to exceed T_s. Then, assign this selected work element into the workstation *i*, and update *L* by deleting this selected work element from the list. If there are no more work elements which can be selected and assigned into the workstation *i*, go to Step 3.3; otherwise, repeat Step 3.2 again.
- Step 3.3: If all n_e work elements have already been selected and assigned (i.e., the list *L* is now empty), the solution S_t is now successfully constructed with the number of all workstations equal *i* (i.e., $n = i$) and then go to Step 4; otherwise, increase *i* by 1 and repeat from Step 3.2.

Step 4: Transform the solution to be usable for the original problem (if necessary) by following these conditions:

- If $t = 1$, the transformation is unnecessary; let $t = 2$ and repeat from Step 2.
- If $t = 2$, transform the solution S_2 into the solution S_2^u by assigning all work elements of the workstation *i* ($i = 1, 2, \ldots, n$) of the solution S_2 into the workstation $n - i + 1$ of the solution S_2^u. Moreover, for any two work elements assigned in the same workstation, the work element assigned before in the workstation *i* of the solution S_2 will be assigned after in the workstation $n - i + 1$ of the solution S_2^u. After S_2^u has been completely constructed, go to Step 5.

Step 5: The final solution S^* is the best solution between S_1 and S_2^u based on their E_b values.

3.2 Double Ranked Positional Weight Method

The double ranked positional weight method (D-RPW) runs both RPW and R-RPW algorithms. The final solution of D-RPW is the best solution between the solutions taken from RPW and R-RPW. In other words, D-RPW is the variant of D-LCR method which replaces LCR by RPW as well as replacing R-LCR by R-RPW. Note that, RPW and R-RPW are found in [3, 11]. The procedure of RPW was already explained in Section 2.2. R-RPW is the method which runs RPW for the being considered ALBP instance with the reversed precedence constraints, called the reversed problem; it then transforms the solution for the reversed problem into a solution for the original problem by using the same procedure of R-LCR.

The procedure of D-RPW is similar to the procedure of D-LCR given in Algorithm 1 on Section 3.1. The differences between the procedures of D-LCR and D-RPW are only in Steps 1 and 2. Steps 3 through 5 of D-RPW are identical to those of D-LCR. Algorithm 2 provides the mentioned procedure of D-RPW.

Algorithm 2 Procedure of D-RPW

Step 1: Let S_1 be the solution of RPW for the being considered instance with the original precedence diagram and S_2 be the solution of RPW for the being considered instance with the reversed precedence diagram. Let S_2^u be the solution which is un-reversed from S_2 to be usable for the original problem. Let S^* be the final solution of D-RPW. Let PD_1 be the original precedence diagram and PD_2 be the reversed precedence diagram. Let L be the list of all as-yet-unassigned work elements in descending order of their priorities. Let t be a binary parameter to indicate whether the current state is RPW (t = 1) or R-RPW (t = 2). Now, let t = 1.

Step 2: Create the list L by listing all n_e work elements from left to right in descending order of RPW_k (k = 1, 2,…, n_e), where RPW_k is the sum of T_{ek} and all other work element times for work elements that follow the work element k in the arrow chain of PD_t.

Step 3: Create the solution S_t by using these following steps:

Step 3.1: Let i = 1.

Step 3.2: Select the leftmost work element among all existing work elements in the list L which satisfy the precedence constraints in PD_t and do not cause the sum of the work element times of all work elements assigned for the workstation i to exceed T_s. Then, assign this selected work element into the workstation i, and update L by deleting this selected work element from the list. If there are no more work elements which can be selected and assigned into the workstation i, go to Step 3.3; otherwise, repeat Step 3.2 again.

Step 3.3: If all n_e work elements have already been selected and assigned (i.e., the list L is now empty), the solution S_t is now successfully constructed with the number of all workstations equal i (i.e., $n = i$) and then go to Step 4; otherwise, increase i by 1 and repeat from Step 3.2.

Step 4: Transform the solution to be usable for the original problem (if necessary) by following these conditions:

- If t = 1, the transformation is unnecessary; let t = 2 and repeat from Step 2.

- If $t = 2$, transform the solution S_2 into the solution S_2^u by assigning all work elements of the workstation i ($i = 1, 2,\ldots, n$) of the solution S_2 into the workstation $n - i + 1$ of the solution S_2^u. Moreover, for any two work elements assigned in the same workstation, the work element assigned before in the workstation i of the solution S_2 will be assigned after in the workstation $n - i + 1$ of the solution S_2^u. After S_2^u has been completely constructed, go to Step 5.

Step 5: The final solution S^* is the best solution between S_1 and S_2^u based on their E_b values.

3.3 Double Most Following Work Element Method Using Longest Work Element Time Rule as Secondary Criterion

The procedure of MFWEL [14-16] is developed by changing the Step 1 of the procedure of LCR shown in Section 2.2. The MFWEL procedure's Step 1 is to list all n_e work elements in descending order of FWE_k. The FWE_k is the number of all work elements following the work element k in the arrow chain of the precedence diagram. Moreover, if two or more work elements have the equal numbers of all following work elements, the work element with the longer work element time must be listed before. The "*L*" in MFWEL refers to the use of the *longest* work element time rule as its secondary criterion. The MFWEL can be found in [14-16]; however, those articles call the longest work element time rule with different names, such as the longest task time rule or the longest operation time rule. The Reversed MFWEL (R-MFWEL) is the method which runs MFWEL for the reversed problem; it then transforms the received solution into the solution for the original problem via the same procedure of R-LCR. The procedure of D-MFWEL is written by only changing Steps 1 and 2 of the procedures of D-LCR or D-RPW, as shown in Algorithm 3.

Algorithm 3 Procedure of D-MFWEL

Step 1: Let S_1 be the solution of MFWEL for the being considered instance with the original precedence diagram and S_2 be the solution of MFWEL for the being considered instance with the reversed precedence diagram. Let S_2^u be the solution which is un-reversed from S_2 to be usable for the original problem. Let S^* be the final solution of D-MFWEL. Let PD_1 be the original precedence diagram and PD_2 be the reversed precedence diagram. Let L be the list of all as-yet-unassigned work elements in descending order of their priorities. Let t be a binary parameter to indicate whether the current state is MFWEL ($t = 1$) or R-MFWEL ($t = 2$). Now, let $t = 1$.

Step 2: Create the list L by listing all n_e work elements from left to right in descending order of FWE_k ($k = 1, 2,\ldots, n_e$), where FWE_k is the number of all work elements following the work element k in the arrow chain of PD_t; in case that two or more work elements have the equal numbers of all following work elements in the arrow chain of PD_t, the work element with longer work element time will be listed before in the list L.

Step 3: Create the solution S_t by using these following steps:

Step 3.1: Let $i = 1$.

Step 3.2: Select the leftmost work element among all existing work elements in the list L which satisfy the precedence constraints in PD_t and do not cause the sum of the work element times of all work elements assigned for the workstation i to exceed T_s. Then, assign this selected work element into the workstation i, and update L by deleting this selected work element from the list. If there are no more work elements which can be selected and assigned into the workstation i, go to Step 3.3; otherwise, repeat Step 3.2 again.

Step 3.3: If all n_e work elements have already been selected and assigned (i.e., the list L is now empty), the solution S_t is now successfully constructed with the number of all workstations equal i (i.e., $n = i$) and then go to Step 4; otherwise, increase i by 1 and repeat from Step 3.2.

Step 4: Transform the solution to be usable for the original problem (if necessary) by following these conditions:

- If $t = 1$, the transformation is unnecessary; let $t = 2$ and repeat from Step 2.
- If $t = 2$, transform the solution S_2 into the solution S_2^u by assigning all work elements of the workstation i ($i = 1, 2,\ldots, n$) of the solution S_2 into the workstation $n - i + 1$ of the solution S_2^u. Moreover, for any two work elements assigned in the same workstation, the work element assigned before in the workstation i of the solution S_2 will be assigned after in the workstation $n - i + 1$ of the solution S_2^u. After S_2^u has been completely constructed, go to Step 5.

Step 5: The final solution S^* is the best solution between S_1 and S_2^u based on their E_b values.

3.4 Double Most Following Work Element Method Using Shortest Work Element Time Rule as Secondary Criterion

The *M*ost *F*ollowing *W*ork *E*lement method using *S*hortest work element time rule as a secondary criterion (MFWES) is the variant of MFWEL that uses the shortest work element time rule as the secondary criterion instead of the longest work element time rule. The shortest work element time rule can be found in many articles under the name of the shortest task time rule [12, 17]. The *R*eversed MFWES (R-MFWES) is similar to the R-MFWEL; however, it uses the shortest work element time rule as its secondary criterion instead of the longest work element time rule.

Double most following work element method using the shortest work element time rule as its secondary criterion (D-MFWES) is the algorithm that runs both MFWES and R-MFWES; the final result of D-MFWES is the best solution between the solutions from MFWES and R-MFWES. Algorithm 4 provides the procedure of D-MFWES.

Algorithm 4 Procedure of D-MFWES

Step 1: Let S_1 be the solution of MFWES for the being considered instance with the original precedence diagram and S_2 be the solution of MFWES for the being considered instance with the reversed precedence diagram. Let S_2^u be the solution which is un-reversed from S_2 to be usable for the original problem. Let S^* be the final solution of D-MFWES. Let PD_1 be the original precedence diagram and PD_2 be the reversed precedence diagram. Let L be the list of all as-yet-unassigned work elements in descending order of their priorities. Let t be a binary parameter to indicate whether the current state is MFWES ($t = 1$) or R-MFWES ($t = 2$). Now, let $t = 1$.

Step 2: Create the list L by listing all n_e work elements from left to right in descending order of FWE_k ($k = 1, 2,\ldots, n_e$), where FWE_k is the number of all work elements following the work element k in the arrow chain of PD_t; in case that two or more work elements have the equal numbers of all following work elements in the arrow chain of PD_t, the work element with shorter work element time will be listed before in the list L.

Step 3: Create the solution S_t by using these following steps:

Step 3.1: Let $i = 1$.

Step 3.2: Select the leftmost work element among all existing work elements in the list L which satisfy the precedence constraints in PD_t and do not cause the sum of the work element times of all work elements assigned for the workstation i to exceed T_s. Then, assign this selected work element into the workstation i, and update L by deleting this selected work element from the list. If there are no more work elements which can be selected and assigned into the workstation i, go to Step 3.3; otherwise, repeat Step 3.2 again.

Step 3.3: If all n_e work elements have already been selected and assigned (i.e., the list L is now empty), the solution S_t is now successfully constructed with the number of all workstations equal i (i.e., $n = i$) and then go to Step 4; otherwise, increase i by 1 and repeat from Step 3.2.

Step 4: Transform the solution to be usable for the original problem (if necessary) by following these conditions:

- If $t = 1$, the transformation is unnecessary; let $t = 2$ and repeat from Step 2.
- If $t = 2$, transform the solution S_2 into the solution S_2^u by assigning all work elements of the workstation i ($i = 1, 2,\ldots, n$) of the solution S_2 into the workstation $n - i + 1$ of the solution S_2^u. Moreover, for any two work elements assigned in the same workstation, the work element assigned before in the workstation i of the solution S_2 will be assigned after in the workstation $n - i + 1$ of the solution S_2^u. After S_2^u has been completely constructed, go to Step 5.

Step 5: The final solution S^* is the best solution between S_1 and S_2^u based on their E_b values.

4. Problem Instances

As mentioned, this research aims to increase the entire productivity of the famous Japanese manufacturer producing digital rice cookers and digital hot pots, located in Thailand. To do so, this research focuses on the high-demand product models. In digital rice cookers, this research focuses on the RC-18NMF model and the RC-10NMF model which are the highest-demand product model and the second highest-demand product model in 2015, respectively. The percentages of the sums of these two models' demands over the total demands of digital rice cookers in the second quarter through the fourth quarter of 2015 are given as follows: 58% in the second quarter, 55% in the third quarter, and 55% in the fourth quarter. In digital hot pots, this research focuses on the PLK-45SF model, the highest-demand product models in 2015. The percentages of the demands of PLK-45SF over the total demands of digital hot pots in the second quarter through the fourth quarter of 2015 are given as follows: 66% in the second quarter, 55% in the third quarter, and 72% in the fourth quarter.

This section then develops six instances based on the actual details of the RC-18NMF, RC-10NMF and PLK-45SF models as follows:

- RC-10NMF-36 is the instance which uses the details of the work elements of RC-10NMF, see Table 1, with the most preferred cycle time (T_c) of 36 seconds. For this instance, the repositioning time of every workstation is 1.67 seconds.

- RC-10NMF-72 is the instance which uses the details of the work elements of RC-10NMF, see Table 1, with the most preferred cycle time (T_c) of 72 seconds. For this instance, the repositioning time of every workstation is 1.67 seconds.
- RC-18NMF-36 is the instance which uses the details of the work elements of RC-18NMF, see Table 2, with the most preferred cycle time (T_c) of 36 seconds. For this instance, the repositioning time of every workstation is 1.67 seconds.
- RC-18NMF-72 is the instance which uses the details of the work elements of RC-18NMF, see Table 2, with the most preferred cycle time (T_c) of 72 seconds. For this instance, the repositioning time of every workstation is 1.67 seconds.
- PLK-45SF-42.35 is the instance which uses the details of the work elements of PLK-45SF, see Table 3, with the most preferred cycle time (T_c) of 42.35 seconds. For this instance, the repositioning time of every workstation is 1.66 seconds.
- PLK-45SF-84.70 is the instance which uses the details of the work elements of PLK-45SF, see Table 3, with the most preferred cycle time (T_c) of 84.70 seconds. For this instance, the repositioning time of every workstation is 1.66 seconds.

The instances RC-10NMF-36, RC-18NMF-36 and PLK-45SF-42.35 use the most preferred cycle times which are currently used in the factory. The instances RC-10NMF-72, RC-18NMF-72 and PLK-45SF-84.70 use the most preferred cycle times which are doubles of the most preferred cycle times currently used in the factory; these instances are generated for the production during their obsolete periods in the future.

The details of all work elements of RC-10NMF, RC-18NMF and PLK-45SF are provided in Tables 1, 2 and 3, respectively. Each table presents the descriptions of all work elements k (k = 1, 2,…, n_e), the work element time of the work element k (T_{ek}) in seconds, and the immediate predecessors of the work element k. The work element k can be started only when all its immediate predecessors have already been finished.

Table 1 Details of Work Elements of RC-10NMF

k	Description of Work Element *k*	T_{ek}	Immediate Predecessors
1	Assemble the inner frame to the base part.	4.87	—
2	Insert the wires into the base part.	9.86	1
3	Assemble the unit case to the base part.	14.00	2
4	Assemble the shield plate to the base part.	12.71	3
5	Insert the wires into the inside of the shield plate.	11.24	4
6	Arrange positions of the wires and check them.	8.62	5
7	Insert the wires into the upper of the shield plate.	10.78	5
8	Pick up the bottle plate and insert the ground line.	5.88	5
9	Assemble the bottle plate to the base part by hand.	7.48	6, 7, 8
10	Fix the bottle plate and the base part by screws.	10.23	9
11	Put the outer lid down on the base part.	5.27	10
12	Insert the wires and assemble the hinge spring and the hinge shaft into the outer lid.	9.95	11
13	Fix the hinge spring and the outer lid by screws.	8.67	12
14	Stick the stickers onto the sides of the base part.	20.14	10
15	Stick the stickers onto the backside of the base part.	7.53	10
16	Assemble the clamp button set to the base part.	4.08	10
17	Insert the plug cable into the base part.	5.70	10
18	Assemble the outer lid to the base part.	6.79	13
19	Assemble the steam case set to the base part.	4.45	18
20	Machine test.	18.87	14, 15, 16, 17, 19
21	Assemble the timer and the handle into the base part.	18.57	20
22	Assemble the shield for the control set.	10.20	21
23	Power-consumption test.	10.14	22
24	Put the paper down inside the base part.	1.59	23
25	Put the inner pot inside the base part.	2.03	24
26	Assemble the inner lid to the outer lid of the base part.	2.12	23
27	Stick the sticker on the outer lid of the base part.	6.38	25, 26
28	Inspect the inner lid.	2.67	27
29	Clean the inside of the base part.	4.37	27
30	Inspect the inside of the whole base part.	11.45	29
31	Put the measuring cup and the rice paddle into the base part.	6.90	28, 30
32	Stick the stickers on the backside of the base part. (The base part will be called the finished product unit from now on.)	5.77	31
33	Put the product warranty into the finished product unit.	2.94	31
34	Assemble the finished product unit with the shockproof foams and put the finished product unit into the paper box.	6.11	32, 33
35	Stick the barcode onto the paper box.	6.47	34
36	Cover the PE plastic sheet onto the finished product unit.	4.63	34
37	Put the instruction manual into the paper box.	3.90	36
38	Put the additional guide book into the paper box.	4.39	36
39	Put the shockproof foam on the top of the finished product unit.	4.24	35, 37, 38
40	Packing.	4.09	39

Table 2 Details of Work Elements of RC-18NMF

k	Description of Work Element *k*	T_{ek}	Immediate Predecessors
1	Assemble the inner frame to the base part.	3.83	—
2	Insert the wires into the base part.	9.90	1
3	Assemble the unit case to the base part.	13.40	2
4	Assemble the shield plate to the base part.	13.63	3
5	Insert the wires into the inside of the shield plate.	12.65	4
6	Arrange positions of the wires and check them.	14.61	5
7	Insert the wires into the upper of the shield plate.	9.11	5
8	Assemble the bottle plate to the base part by hand.	9.57	6, 7
9	Fix the bottle plate and the base part by screws.	11.33	8
10	Put the outer lid down on the base part.	2.97	9
11	Insert the wires and assemble the hinge spring and the hinge shaft into the outer lid.	10.70	10
12	Fix the hinge spring and the outer lid by screws.	10.16	11
13	Stick the stickers onto the backside of the base part.	7.36	9
14	Assemble the clamp button set to the base part.	4.67	9
15	Insert the plug cable into the base part.	4.97	9
16	Assemble the outer lid to the base part.	4.85	12
17	Assemble the steam case set to the base part.	8.57	16
18	Machine test.	19.48	13, 14, 15, 17
19	Assemble the timer and the handle into the base part.	19.66	18
20	Assemble the shield for the control set.	10.30	19
21	Power-consumption test.	9.90	20
22	Put the paper down inside the base part.	1.56	21
23	Put the inner pot inside the base part.	2.75	22
24	Assemble the inner lid to the outer lid of the base part.	2.56	21
25	Stick the sticker on the outer lid of the base part.	9.12	23, 24
26	Inspect the inner lid.	4.93	25
27	Clean the inside of the base part.	3.62	25
28	Inspect the inside of the whole base part.	9.48	27
29	Put the measuring cup and the rice paddle into the base part.	5.66	26, 27, 28
30	Stick the stickers on the backside of the base part. (The base part will be called the finished product unit from now on.)	6.98	29
31	Put the product warranty into the finished product unit.	4.65	29
32	Assemble the finished product unit with the shockproof foams and put the finished product unit into the paper box.	6.98	30, 31
33	Stick the barcode onto the paper box.	6.06	32
34	Cover the PE plastic sheet onto the finished product unit.	4.43	32
35	Put the instruction manual into the paper box.	1.93	34
36	Put the additional guide book into the paper box.	6.24	34
37	Put the shockproof foam on the top of the finished product unit.	6.97	34, 35, 36
38	Packing.	4.63	37

Table 3 Details of Work Elements of PLK-45SF

k	**Description of Work Element *k***	T_{ek}	**Immediate Predecessors**
1	Check the quality of the base part, named the tank assy; bind the top of the the tank assy by a rubber band, named the tank pk; and put the tank assy onto a jig.	9.85	—
2	Cover the top of the tank assy by a power lid.	6.57	1
3	Drop the silicone liquid, named T-16, into the hold of the bottom of the tank assy.	3.08	2
4	Insert a wire, named the warm lead set, into the power lid of the tank assy.	12.01	2
5	Assemble a part, named the base plate assy, onto the bottom of the tank assy.	7.69	4
6	Press a rubber onto the base plate assy of the tank assy, and insert a wire of the base plate assy into an electronic circuit board, named the until assy.	5.38	5
7	Insert three wires of a heater into the base plate assy of the tank assy.	5.22	6
8	Insert a wire of the part, named the TEMP.FUSE set into the base plate assy of the tank assy, and then insert another wire of the TEMP.FUSE set into the until assy of the tank assy.	6.49	6
9	Put two 'Fool Metal' parts onto the base plate assy of the tank assy.	8.06	6
10	Fix the two 'Fool Metal' parts with the base plate assy of the tank assy by two screws, and then fix the base plate assy tightly with the tank assy by a screw.	10.58	9
11	Fix a boiling sensor with the base plate assy of the tank assy by a screw and insert a wire of the boiling sensor into the until assy of the tank assy.	10.21	7, 8, 10
12	Twist the end of a warm lead set, and then hook the end of the warm lead set with the terminal of the base plate assy of the tank assy.	11.63	11
13	Assemble a spring washer, named SPRING WASHER M3, onto the terminal of the base plate assy of the tank assy, and then assemble a nut, named NUT M3, onto the terminal.	14.84	12
14	Tight the NUT M3 by a jig.	9.26	13
15	Check the until assy of the tank assy, and assemble the until assy with a case, named the until case.	11.56	14
16	Assemble a water level pipe set with the power pump of the tank assy.	8.09	15
17	Fix the until case tightly with the until assy of the tank assy by a screw, and then fix the water level pipe tightly with the tank assy.	12.51	16
18	Cover the tank assy by a part, named the body assy, and then assemble the body assy with another part, named the lower lid; the completed assembly of the tank assy, the body assy and the lower lid is called the tank.	23.32	17
19	Put two slip pipes into the bottom of the tank.	3.58	18
20	Connect two pairs of two wires located at the bottom of the tank.	17.46	19
21	Put a part, named the bottom sheet BA, onto the body assy of the tank.	4.18	20
22	Fix the bottom sheet BA tightly with the body assy of the tank by two screws.	7.58	21
23	Brush the silicone liquid onto the hold of the body assy of the tank.	3.33	20
24	Put a rubber band, named the rotary base, into the hold of the body assy of the tank.	4.62	23
25	Air leak test.	20.77	22, 24
26	Leakage current test.	7.60	25
27	Power consumption test.	11.97	26
28	Stick a sticker at a side of the tank.	7.91	27
29	Inspect the tank, and assemble the upper lid with the tank. (The tank will be called the digital hot pot from now on.)	12.23	28
30	Clean the digital hot pot.	5.09	29
31	Stick a barcode at a side of the digital hot pot.	6.54	30
32	Lay down the manual, caution, and warrantee documents and a plug beside the digital hot pot.	10.40	31
33	Unpack 2 shockproof foams from their packages.	5.74	32
34	Put the digital hot pot into its package.	5.43	33
35	Put a PE sheet and 2 shockproof foams for the digital hot pot inside the package.	7.13	34
36	Put the manual, caution, and warrantee documents and a plug inside the package.	4.13	35
37	Packing.	5.01	36
38	Stick a barcode at the package.	4.08	37

5. Evaluation of Algorithm Performances

This paper uses the balance efficiency of the assembly line, or E_b, as the indicator to evaluate the performances of these four proposed algorithms:

- Double largest candidate rule method (D-LCR)
- Double ranked positional weight method (D-RPW)
- Double most following work element method using longest work element time rule as the secondary criterion (D-MFWEL)
- Double most following work element method using shortest work element time rule as the secondary criterion (D-MFWES).

Each algorithm is manually computed on the six instances, namely, RC-10NMF-36, RC-10NMF-72, RC-18NMF-36, RC-18NMF-72, PLK-45SF-42.35 and PLK-45SF-84.70. Note that, the factory currently uses the ad hoc procedure which returns these following balance efficiencies: 66.6% for RC-10NMF-36, 73.8% for RC-18NMF-36 and 74.3% for PLK-45SF-42.35. The results from the ad hoc procedure are based on the actual observations; thus there are no balance efficiencies for RC-10NMF-72, RC-18NMF-72 and PLK-45SF-84.70.

Table 4 shows % balance efficiencies of LCR, RPW, MFWEL, MFWES, R-LCR, R-RPW, R-MFWEL, R-MFWES, D-LCR, D-RPW, D-MFWEL, and D-MFWES on the six instances. The highest value on each instance shows in the italic font. Based on the results shown in Table 4, D-MFWES performs best, MFWES is the runner-up, R-MFWES is in the third position, D-MFWEL is in the fourth position, and MFWEL is in the fifth position. The best solution found in this paper for each instance will show in Tables 5 through 10. Each solution shows all work elements arranged and assigned for the workstations as well as the service times in seconds.

Table 4 % Balance Efficiencies of Algorithms on six Instances

Algorithm	Instance						
	RC-10NMF-36	RC-10NMF-72	RC-18NMF-36	RC-18NMF-72	PLK-45SF-42.35	PLK-45SF-84.70	Average
Ad hoc	66.6	n.a.	73.8	n.a.	74.3	n.a.	n.a.
LCR	*82.3*	88.1	79.6	88.3	*82.1*	80.3	83.5
RPW	81.6	88.1	*82.0*	88.1	*82.1*	80.3	83.7
MFWEL	81.6	*91.0*	*82.0*	88.1	*82.1*	80.3	84.2
MFWES	81.6	88.1	*82.0*	*92.8*	*82.1*	*84.8*	85.2
R-LCR	81.6	87.6	81.0	87.1	81.8	81.7	83.5
R-RPW	81.6	87.6	80.9	87.0	81.8	81.7	83.4
R-MFWEL	81.6	87.6	80.9	87.0	81.8	81.7	83.4
R-MFWES	81.6	90.1	81.5	90.5	81.8	82.7	84.7
D-LCR	*82.3*	88.1	81.0	88.3	*82.1*	81.7	83.9
D-RPW	81.6	88.1	*82.0*	88.1	*82.1*	81.7	83.9
D-MFWEL	81.6	*91.0*	*82.0*	88.1	*82.1*	81.7	84.4
D-MFWES	81.6	90.1	*82.0*	*92.8*	*82.1*	*84.8*	*85.6*

On RC-10NMF-36 instance, D-LCR performs best with the E_b = 82.3%. The solution given by D-LCR on this instance is shown in Table 5. The solution shows that, by using D-LCR, the number of all workstations required is 11. In this solution, the workstation 11 is the bottleneck workstation with the service time of 33.83 seconds and the repositioning time of 1.67 seconds. The actual cycle time of the entire assembly line is thus 35.50 seconds, less than the most preferred cycle time (T_c = 36 seconds). It means that the assembly line operates slightly faster than the pace really needed by the factory.

On RC-10NMF-72 instance, D-MFWEL performs best with the E_b = 91.0%. The solution given by D-MFWEL is shown in Table 6. The solution of D-MFWEL requires 5 workstations. The bottleneck workstation in this solution is the workstation 2 with the service time of 67.34 seconds and the repositioning time of 1.67 seconds. The actual cycle time of the entire assembly line is thus 69.01 seconds, faster than the most preferred cycle time (T_c = 72 seconds).

Table 5 Best Found Solution on RC-10NMF-36 with E_b = 82.3%

Workstation	Work Elements	Service Time
1	1, 2, 3	28.73
2	4, 5, 6	32.57
3	7, 8, 9	24.14
4	10, 14	30.37
5	15,17,11,12	28.45
6	13,18,19,16	23.99
7	20	18.87
8	21, 22	28.77
9	23, 24, 25, 26, 27, 29	26.63
10	30, 28, 31, 32, 33	29.73
11	34, 36, 37, 38, 35, 39, 40	33.83

Table 6 Best Found Solution on RC-10NMF-72 with E_b = 91.0%

Workstation	Work Elements	Service Time
1	1, 2, 3, 4, 5, 7	63.46
2	6, 8, 9, 10, 11, 12, 13, 18, 19	67.34
3	14, 15, 17, 16, 20	56.32
4	21, 22, 23, 24, 25, 26, 27, 29, 30	66.85
5	28, 31, 32, 33, 34, 36, 38, 37, 35, 39, 40	52.11

On RC-18NMF-36 instance, D-RPW, D-MFWEL and D-MFWES returns the different solutions with the same E_b = 82.0%. Table 7 thus shows only the solution taken from D-RPW. In this solution, the number of all required workstations is 11, and the bottleneck workstation of this assembly line is the workstation 3 with the service time of 33.29 seconds and the repositioning time of 1.67 seconds. The actual cycle time of the entire assembly line is thus 34.96 seconds, faster than the most preferred cycle time (T_c = 36 seconds).

On RC-18NMF-72 instance, D-MFWES performs best with the E_b = 92.8%. The solution given by D-MFWES is shown in Table 8. By using D-MFWES, the number of all required workstations is 5. The bottleneck workstation in this solution is the workstation 3 with the service time of 64.71 seconds and the repositioning time of 1.67 seconds. The actual cycle time of the entire assembly line is hence 66.38 seconds. This solution makes the assembly line operate slightly faster than the most preferred cycle time (T_c = 72 seconds).

Table 7 Best Found Solution on RC-18NMF-36 with E_b = 82.0%

Workstation	Work Elements	Service Time
1	1, 2, 3	27.13
2	4, 5	26.28
3	7, 6, 8	33.29
4	9,10,11	25.00
5	12,16,17,13	30.94
6	15,14,18	29.12
7	19, 20	29.96
8	21, 22, 23, 24, 25	25.89
9	28, 26, 27, 29, 30	30.67
10	31, 32, 33, 34, 35, 36	30.29
11	37, 38	11.60

Table 8 Best Found Solution on RC-18NMF-72 with E_b = 92.8%

Workstation	Work Elements	Service Time
1	1, 2, 3, 4, 5, 7	62.52
2	6, 8, 9, 10, 11, 12, 16	64.19
3	17, 14, 15, 13, 18, 19	64.71
4	20, 21, 22, 23, 24, 25, 27, 26, 28, 29, 31	64.53
5	30, 32, 34, 35, 36, 33, 37, 38	44.22

On PLK-45SF-42.35 instance, D-LCR, D-RPW, D-MFWEL, and D-MFWES returns an identical solution with the E_b = 82.1%, shown in Table 9. In this solution, the number of all required workstations is 10. The bottleneck workstation in this solution is the workstation 7 with the service time of 40.34 seconds and the repositioning time of 1.66 seconds. The actual cycle time of the entire assembly line is hence 42.00 seconds which is slightly faster than the most preferred cycle time (T_c = 42.35 seconds).

On PLK-45SF-84.70 instance, D-MFWES performs best with a solution of E_b = 84.8%. Table 10 shows the solution of D-MFWES which requires 5 workstations. The bottleneck workstation in this solution is the workstation 2 with the service time of 78.10 seconds and the repositioning time of 1.66 seconds. The actual cycle time of the entire assembly line is hence 79.76 seconds which is slightly faster than the most preferred cycle time (T_c = 84.70 seconds).

Table 9 Best Found Solution on PLK-45SF-42.35 with E_b = 82.1%

Workstation	Work Elements	Service Time
1	1, 2, 4, 3, 5	39.20
2	6, 9, 10, 8, 7	35.73
3	11, 12, 13	36.68
4	14, 15, 16	28.91
5	17, 18, 19	39.41
6	20, 21, 22, 23, 24	37.17
7	25, 26, 27	40.34
8	28, 29, 30, 31	31.77
9	32, 33, 34, 35, 36, 37	37.84
10	38	4.08

Table 10 Best Found Solution on PLK-45SF-84.70 with E_b = 84.8%

Workstation	Work Elements	Service Time
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	74.93
2	11, 12, 13, 14, 15, 16, 17	78.10
3	18, 19, 20, 23, 24, 21, 22	64.07
4	25, 26, 27, 28, 29, 30, 31	72.11
5	32, 33, 34, 35, 36, 37, 38	41.92

Based on the results taken from Table 4, D-MFWES performs best, as mentioned. This paper hence uses the paired *t* test to test the population mean of all paired differences between the balance efficiencies of D-MFWES and any specific algorithm (e.g., D-LCR, D-RPW, D-MFWEL, MFWES and R-MFWES). This paper uses the significance level of 0.15 for all tests.

This paper first conducts the hypothesis test for H0: the population mean of all paired differences between the balance efficiencies of D-MFWES and D-LCR is zero versus H1: the population mean of all paired differences between the balance efficiencies of D-MFWES and D-LCR is greater than zero with the significance level of 0.15. The test returns the *p*-value of 0.046 which is less than 0.15; thus, H0 is rejected. It concludes that, on average, D-MFWES performs better than D-LCR.

For testing the hypothesis that H0: the population mean of all paired differences between the balance efficiencies of D-MFWES and D-RPW is zero versus H1: the population mean of all paired differences between the balance efficiencies of D-MFWES and D-RPW is greater than zero with the significance level of 0.15. As a result, the *p*-value is 0.050, less than the significance level; H0 is thus rejected. It concludes that, on average, D-MFWES performs better than D-RPW.

For testing the hypothesis that H0: the population mean of all paired differences between the balance efficiencies of D-MFWES and D-MFWEL is zero versus H1: the population mean of all paired differences between the balance efficiencies of D-MFWES and D-MFWEL is greater than zero with the significance level of 0.15. This test returns the *p*-value of 0.130 which is less than the significance level; H0 is thus rejected. It hence concludes that, on average, D-MFWES performs better than D-MFWEL.

This paper then tests the hypothesis that H0: the population mean of all paired differences between the balance efficiencies of D-MFWES and MFWES is zero versus H1: the population mean of all paired differences between the balance efficiencies of D-MFWES and MFWES is greater than zero with the significance level of 0.15. The test returns the *p*-value of 0.182 which is higher than the significance level. This experiment thus fails to reject H0. In other words, it fails to conclude that D-MFWES outperforms MFWES on average with the significance level of 0.15.

In testing of the hypothesis that H0: the population mean of all paired differences between the balance efficiencies of D-MFWES and R-MFWES is zero versus H1: the population mean of all paired differences between the balance efficiencies of D-MFWES and R-MFWES is greater than zero with the significance level of 0.15. As a result, the *p*-value is 0.050 which is less than the significance level; H0 is thus rejected. The conclusion is that, on average, D-MFWES outperforms R-MFWES with the significance level of 0.15.

As a summary, the results of the hypothesis tests concludes that, on average, D-MFWES performs better than D-LCR (*p*-value = 0.046), D-RPW (*p*-value = 0.050), D-MFWEL (*p*-value = 0.130), and R-MFWES (*p*-value = 0.050) with the significance level of 0.15. However, the test fails to reject that the mean of all paired differences between the balance efficiencies of D-MFWES and MFWES is zero with the significance level of 0.15 (*p*-value = 0.182). In other words, it fails to conclude that, on average, D-MFWES outperforms MFWES.

6. Conclusions

This paper proposed the four algorithms for ALBP, i.e., D-LCR, D-RPW, D-MFWEL and D-MFWES. D-LCR is composed of LCR and the reversed LCR (i.e., R-LCR), while D-RPW is composed of RPW and the reversed RPW (i.e., R-RPW). Similarly, D-MFWEL consists of MFWEL and the reversed MFWEL (i.e., R-MFWEL). D-MFWES consists of MFWES and the reversed MFWES (i.e., R-MFWES). All four proposed algorithms performed much better than the ad hoc procedure currently used in the factory. The results from the hypothesis tests concluded that, on average, D-MFWES outperformed D-LCR, D-RPW, D-MFWEL, and R-MFWES with the significance level of 0.15; however, they failed to conclude that, on average, D-MFWES outperformed MFWES, its original version. This is possibly because the original precedence diagram is more proper to use than the reversed precedence diagram for the set of the instances given here. D-MFWES may significantly outperform MFWES on the set of instances where the reversed precedence diagram is more proper to use than the original precedence diagram.

A further work of this research should be a study on the performances of the proposed algorithms on the real-world ALBP instances from other industries. Another interesting further work should be an introduction of some more simple-but-powerful algorithms for ALBP.

7. Acknowledgment

This article is a revised and expanded version of the paper entitled “Double assembly line balancing methods for a manufacturer of digital rice cookers” presented in the *Proceedings of the International Conference on Science and Technology*, Pathum Thani, Thailand, 2015.

8. References

[1] P. Phiphatsomporn and P. Pongchairerks, “*Double assembly line balancing methods for a manufacturer of digital rice cookers*”. in the Proceedings of the International Conference on Science and Technology. Pathum Thani, Thailand. 2015. 424-429.

[2] P. M. Groover. ***Work Systems and the Methods, Measurement, and Management of Work***. New Jersey : Pearson Prentice Hall; 2007.

[3] B. Rekiek, A. Delchambre. ***Assembly Line Design : The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms***. Germany : Springer; 2006.

[4] I. Baybars. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*. **32** (1986): 909-932.

[5] N. Kriengkorakot, N. Pianthong. The assembly line balancing problem : review articles. *KKU Engineering Journal*. **34** (2007):133-140.

[6] N. Kumar, D. Mahto. Assembly line balancing : a review of developments and trends in approach to industrial application. *Global Journal of Researches in Engineering : Industrial Engineering*. **13** (2013): 19-50.

[7] V. Pachghare, R. S. Dalu. Assembly line balancing – a review. *International Journal of Science and Research*. **3** (2014): 807-811.

[8] P. Sivasankaran, P. Shahabudeen. Literature review of assembly line balancing problems. *International Journal of Advanced Manufacturing Technology*. **73** (2014): 1665-1694.

[9] A. L. Gutjahr, G. L. Nemhauser. An algorithm for the balancing problem. *Management Science*. **11** (1964): 23-35.

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

[11] W. B. Helgeson, D. P. Birnie. Assembly line balancing using ranked positional weight technique. *Journal of Industrial Engineering*. **12** (1961): 394-398.

[12] B. Ibrahim. Production flow in gas cooker assembly line. *Journal of Babylon University*. **21** (2013): 1752-1761.

[13] C. Ünal, A new line balancing algorithm for manufacturing cell transformation in apparel industry. *Industria Textilǎ,* **64** (2013): 155-162.

[14] J. K. Shim, J. G. Siegel. ***Operations Management***. New York : Barron's Educational Series; 1999.

[15] J. Konnully. ***Assembly Line Balancing – Example***. Internet: http://www.slideshare.net/ JosephKonnully/asslylinbalexample. 2013 [Nov. 5, 2014].

[16] S. Bhattacharya. ***Operations Management***. Delhi: PHI Learning Provate; 2014.

[17] J. Heizer, B. Render. ***Principle of Operations Management***. 10th ed. New Jersey : Pearson Prentice Hall; 2011.

[18] G. Nirmala, D. Ramprasad. Genetic algorithm with topology to solve minimizing waiting time of ALBP. *Aryabhatta Journal of Mathematics and Informatics*. **6** (2014): 169-174.

[19] J. C. Chen, C.-W. Wu, T. D. Thao, L.-H. Su, W.-H. Hsieh, T. Chen. Hybrid genetic algorithm for solving assembly line balancing problem in footwear industry. *Advanced Material Research*. **939** (2014): 623-629.

[20] U. Ozcan, H. Cercioglu, H. Gokcen, B. Toklu. A tabu search algorithm for the parallel assembly line balancing problem. *G.U. Journal of Science*. **22** (2009): 313-323.

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