A New Heuristic Approach to Solving U-shape Assembly Line Balancing Problems Type-1

M. Fathi, M. J. Alvarez, V. Rodríguez

Abstract**—**Assembly line balancing is a very important issue in mass production systems due to production cost. Although many studies have been done on this topic, but because assembly line balancing problems are so complex they are categorized as NP-hard problems and researchers strongly recommend using heuristic methods. This paper presents a new heuristic approach called the critical task method (CTM) for solving U-shape assembly line balancing problems. The performance of the proposed heuristic method is tested by solving a number of test problems and comparing them with 12 other heuristics available in the literature to confirm the superior performance of the proposed heuristic. Furthermore, to prove the efficiency of the proposed CTM, the objectives are increased to minimize the number of workstation (or equivalently maximize line efficiency), and minimizing the smoothness index. Finally, it is proven that the proposed heuristic is more efficient than the others to solve the U-shape assembly line balancing problem.

Keywords**—**Critical task method, Heuristic, Line balancing problem, U-shape

I. Introduction

ASSEMBLY line is an order of production equipment called workstations, which are connected together by a kind of material-handling system. The aim of assembly line balancing problems (ALBPs) is to assign activities to stations with respect to the precedence relationships and other constraints while some measurements of performance are optimized [1]. In accordance with Ghosh and Gagnon [2], only two main types of measurements have been used in the ALBPs. The first one is technical measurements such as cycle time, balance delay or total idle time, and minimizing the number of workstations. The second one is economic measurements like profit maximization and cost minimization.

In general, assembly line balancing problem occur when an assembly line has to be designed or redesigned. The assembly line problem was first introduced by Henry Ford in 1915, the father of modern assembly lines used in mass production. The first researcher that published a paper on the assembly line was Salveson [3], who recommended a linear programming solution. Since then, the topic of line balancing has been a hot one for researchers, and a lot of theoretical and practical studies have been done on different types of assembly lines, especially straight and U-shape.

However, Gutjahr and Nemhauser [4] and Ajenblit and Wainwright [5] showed that traditional (linear) and U-shape ALBPs fall into the NP-hard class of combinatorial optimization problems.

According to the literature [6]-[8] assembly line balancing can be categorized in two main groups; one is the original and simplest type of assembly line balancing problem, which is called SALBP, and the second group with added restrictions or factors (e.g. parallel stations, zoning restrictions) becomes the general assembly line balancing problem (GALBP). In addition, the most important types of assembly line balancing problems are considered as type-1 and type-2. The main objective of type-1 is to assign tasks to workstations in such a way that the number of stations, and consequently the total idle time, be minimized for a given production rate while the cycle time is deterministic [9]. Type-2 can be defined as a given set of tasks and their precedence relationships (restrictions) so that minimizing the cycle time, and equivalently maximizing the production rate for the presumed number of workstations, are the main objectives of this type of problem.

Nowadays, most industries are using some type of assembly line (e.g., straight and U-shape) due to high-volume production, the complexity of products, technological growth, and the ability to respond quickly to changes in demand and challenges in a contemporary competitive environment. The line balancing problem is aimed at arranging the assembly tasks for workstations so that all workers have an approximately equal amount of work, as well as total work time in each workstation [10]. In such circumstances, the number of workers or workstations required on an assembly line would be minimized while the precedence constraints are satisfied and some measurements of productivity are optimized. Consequently, the main objective in line balancing is to minimize unit assembly cost. Since in the classic assembly line balancing problems assembly costs are postulated to include the cost of task performance and the cost of idle time, the assembly cost can only be minimized through the reduction of idle time, which sometimes might result in the removal of some stations. In addition, some advantages of assembly line balancing in industries and in mass production in particular include:

- Decline in inventories.
- Increasing worker satisfaction.
- Decreasing human resources.

M. Fathi and M. J. Alvarez are with the Department of Industrial Organization, School of Engineering (Tecnun), University of Navarra, P Manuel Lardizabal 13, 20018 San Sebastian, Spain (e-mail: mfathi@tecnun.es and mjalvarez@tecnun.es)

V. Rodríguez is with the Economics and Management School, University of Navarra, Campus Universitario, 31080 Pamplona, Spain (e-mail: vrodriguez@unav.es)

- Decreasing the total production costs including labor, inventories and machines.
- Increasing the productivity and efficiency of the assembly line.
- Decreasing the number of workstations and consequently the amount idle time.

The line balancing problem falls in the category of problems that provide a greater degree of efficiency by improving the manufacturing throughput and reducing the required number of human resources by optimizing task assignment and resource allocation. Therefore, it has generated an interest in many researchers in this area, and so many methods have been developed by researchers to solve assembly line balancing problems up to now. Most western producers arrange assembly lines in a straight-line for high volume production, while Japanese manufactures using the just-in-time philosophy prefer the U-shaped layout. However, as stated above both straight and U-shape line balancing problems are categorized into NP-hard problems; this means that exact methods have difficulty dealing with a large size problem. Therefore, it is necessary to use heuristic methods to solve this problem. Furthermore, heuristic methods are considered as the foundation of metaheuristics, and thus the efficiency of these methods will determine the success of the optimization. A number of heuristics and metaheuristic approaches have been introduced and used so far by researchers to solve different types of assembly line balancing problems, especially straight and U-shape. Nevertheless, there is relatively less literature on the U-shape assembly line balancing problem (UALBP) than on the straight assembly line balancing problem and finding an optimum or near-optimum solution in less computational time with a more efficient method has yet to be discovered.

In this research, a new heuristic method named CTM is introduced to solve U-shape assembly line balancing problems in the area of type-1. According to the definition of type-1 the main objective is to minimize the number of stations; however, in this study, the main objective is to show the efficiency and effectiveness of the new method when one more performance measure known as the smoothness index is considered as well. Finally, the performance of the proposed CTM is tested by solving a number of test problems and comparing them with 12 other heuristics available in the literature to confirm the superior performance of the proposed heuristic method. In addition, it is proven that the proposed heuristic is more efficient than the others in solving U-shape assembly line problems.

The rest of paper is organized as follows. In the first section the proposed heuristic is explained, in the second section the algorithm of the proposed CTM is described in steps, a brief description of the priority heuristic rules is included in third section, performance criteria are explained in section four, the computational results are shown in tables in the fifth section, and last section is about the conclusions and further work.

II. Description of Proposed Heuristic Method

In this section a new heuristic named the critical task method (CTM) is introduced and all the steps of the algorithm are explained in detail. Finally, an illustrative example is solved to show and clarify the proposed CTM.

As stated above, heuristic methods can be considered as the best methods for solving assembly line balancing problems. In the last decades a lot of research has been done on this issue and a number of heuristic methods have been presented to date. However, using a heuristic method to solve this kind of problems in less computational time still has a very important place in real world.

Recently, Yeh and Kao [11] proposed a new approach based on critical path methods (CPM) in order to solve bidirectional assembly lines, and the time complexity of this method is only O(mn^2), meaning that this method can be solved within a polynomial-time. Therefore, regarding the above mentioned reasons to use heuristic methods and taking the advantage of less time complexity, in this study a new effective heuristic method is presented which is based on combining the proposed approach by Yeh and Kao [11] and the well-known rank positional weight technique (RPW) introduced by Helgeson and Birnie [12] to solve U-shape assembly line problems in the area of type-1.

Due to the fact that the proposed heuristic (CTM) is based on CPM and RPW, reviewing some definitions of both is necessary.

The critical path method (CPM) is very popular and it is used widely in project management problems. CPM is a mathematically-based algorithm for scheduling a set of project activities and is widely used in computing project scheduling. In other words, CPM is a technique for managing and scheduling projects during implementation, and it can be defined as the longest path (according to the time duration) from the first node to the last node. In this method, CPM calculates the longest path of planned activities to the end of the project, and for every single task it computes the earliest and latest time a task that can start and finish without making the project longer.

The rank positional weight (RPW) is a well-known heuristic method that is widely used in solving assembly line balancing problems and it is calculated by adding a task processing time to the summation of the processing times of all its successors. According to the RPW technique, each task has its own weight and the weight is computed by summing all the following tasks' times, and those tasks with greater weight gain more priority in assignment. In other words, the task with the highest positional weight is selected and assigned to the earlier station.

In accordance with above mentioned explanations of CPM and RPW, the proposed CTM is explained as follows.

The CPM is used in the proposed method to solve assembly line balancing problems for two reasons:

1. Assembly line balancing problems and project management problems have similar network structures.

2. In the project management all of tasks that are on a critical path have a high performance priority, and each delay in their performance ends in postponing the whole project, which can have a similar meaning for the assembly line in that if a suitable task is somehow not assigned to each workstation there might need to be more workstations and consequently expenses and the human resources needed increase.

Additionally, the main concept from RPW of calculating a weight for each task and assigning a priority to it based on its weight is used in the proposed CTM.

The proposed heuristic is a procedure for assigning the most suitable tasks to each workstation in order to minimize the number of workstations as well as the difference in workloads. In the first step of the algorithm, the tasks' weight is calculated so that these weights are equal to the time of the longest path from the beginning of the operation through the remainder of the network, and all tasks are sorted in descending order. In the next step, the most critical task with the highest weight will be assigned to the proper workstation and will be removed from the assembly network, and a new computation for determining the current tasks' weight will be performed by the same procedure. This process continues until all the tasks gain their own particular weights. The tasks are assigned in descending order of the weight in such a way which satisfies the precedence relationship and does not exceed the station remaining cycle time. To clarify the proposed CTM, all steps of the algorithm are described below.

Step 1. Find the critical path (CP).

Step 2. Calculate the weight of the first task on the CP by summing the time of all the tasks on the CP plus the time of the first task.

Step 3. Check the calculated weight for each task. If the weight of all the tasks has been calculated, the next step of the algorithm should be applied; otherwise the algorithm should be started from the first step to find the new critical path.

Step 4. Sort the tasks in descending order and the weight of each task shows the criticality of the task. In other words, the task with the higher weight is more critical than the others and consequently has a higher assignment priority.

Step 5. Assign the tasks to workstations according to their criticality with respect to all constraints.

The proposed CTM can be used to solve almost all types of assembly line problems. However, this study is focused on solving the U-shape assembly line balancing problem. Here it should be noted that for U-shape lines the mentioned weight should be calculated in two directions (forward and backward).

The parameters used in this method are:

$T(s_i)$	Total time of each station.
$T(x)$	Time of each task.
CT	Given cycle time.
N	Number of tasks.
M	Number of workstations.
S	Minimum feasible number of workstation.
MCT	Minimum feasible cycle time.
CT^*	Modified cycle time.

In this method, at first a precedence network is used to find the tasks' weight using the above mentioned definition (time of the longest path from the beginning of the operation through the remainder of the network), and then those tasks that have more weight have a higher selection priority. Those tasks that have less weight will be assigned to a workstation according to two reasons; first, to preserve succession and precedence priorities, and second, if workstation capacities have not yet been completely used. If there is more than one task in the candidate assignment list, the task with the greatest time will be chosen. Assigning tasks to the workstations is completed when all tasks are assigned. In this method, another criterion named CT^*, whose calculation is shown just below is used instead of cycle time:

$S = \sum_{i=0}^{n} T(x)/CT$ If S is not an integer then it will be rounded up. (1)

$$\mathrm{MCT} = \sum_{i=0}^{n} T(x)/S \quad (2)$$

$$\mathrm{CT}^* = [(\mathrm{MCT} + \mathrm{CT})/2] \quad (3)$$

CT^* is between MCT and CT ($\mathrm{MCT} < \mathrm{CT}^* < \mathrm{CT}$). Although, CT can be substituted by each value between CT and MCT, but it has been shown through experience that CT^* offers better results. To obtain the desired conditions, the following relations should be maintained:

$$\mathrm{T}(s_i) = \sum_{x \in s_i} T(x) \leq CT \quad i = 1 \ldots, \mathrm{M} \quad (\mathrm{I})$$

$$\text{If } (x,y) \in p,\ x \in S_i \text{ and } y \in S_j \text{ then } i \leq j \text{ for all } x. \quad (\mathrm{II})$$

$$\text{If } (y,z) \in p,\ y \in S_j \text{ and } z \in S_k \text{ then } k \leq j \text{ for all } z. \quad (\mathrm{III})$$

Relation (I) above means that the sum of the times assigned to each station should not be more than a predetermined cycle time. Relation (II) says that if activity *x* precedes *y* and activity *x* is done at i-th station and *y* is done at j-th station, then i ≤ j, means that *x* is done before y or at the same station as y. Relation (III) is same as the second one to ensure the precedence constraints are satisfied in the backwards direction.

A. Algorithm of the Proposed CTM to Solve U-Shape Assembly Line Balancing Problem

In this part a comprehensive overview of the CTM rule algorithm is described based on the above mentioned explanation as follows:

(1) Calculating minimum feasible number of workstation *S* and the minimum feasible cycle time MCT and the adjusted value of $\mathrm{CT}^* = [(\mathrm{MCT} + CT)/2]$.

(2) Creating a new workstation, calculating the weight for each task in two stages, one time from the forward direction and another time from the backwards direction and then identifying activities permitted for assigning and creating a candidate list.

(3) Assigning activities with high weight on the candidate list; if there are two or more activities with the same weight one of them can be selected to be assigned at random.

(4) Computing the remaining time for the current station and updating the candidate list based on the new calculated weights and constraints; if the station has enough time for any feasible unassigned task go to step 3, otherwise go to step 5.

(5) The assigning process will be repeated until no tasks are left. If there are unassigned tasks, go to step 2.

Note:

(1) The obtained CT^* is the supposed upper bound of the work capacity of the stations instead of CT.

(2) This order in each stage is continued by finding the new weight for each task using the critical path, because when solving U-shape line the tasks' weight should be updated in the forward direction when the assigned task is from the end of network; otherwise, the tasks' weight in the backward direction should be calculated again, until all the activities are assigned to the workstations.

(3) The candidate list stated in this method includes all tasks that can be assigned to the current workstation according to all that constraints.

B. Numerical Illustration

An example with 12 tasks and a cycle time of 12 sec. is shown for illustration. The precedence network of the presented example is graphically shown in Fig. 1 and all steps of the assigning procedure by proposed CTM are described in following.

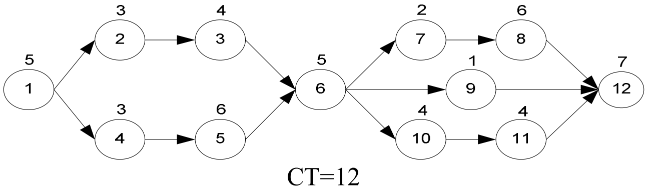

Fig. 1 Precedence diagram of assembly network for illustration

The assumption is that CT = 12 sec., then S, MTC and finally CT^* are calculated by using the above mentioned equations. Moreover, the initial calculated weights of the tasks are given in Table I.

Here a brief overview of the results without the modified network in each step is provided:

TABLE I
WEIGHT COMPUTATION BY CTM IN FORWARD AND BACKWARDS DIRECTION

Task number	1	2	3	4	5	6	7	8	9	10	11	12
Backward weight	34	27	24	29	26	20	15	13	8	15	11	7
Forward weight	5	8	12	8	14	19	21	27	20	23	27	34

(1) Calculate S = 50/12 = 4.16 and after rounding up it is 5 and MCT = 50/5 = 10, therefore CT^* = [(12+10)/2] =11, so 10<11< 12.

(2) Create workstation 1, calculate the activity weights in the forward and backwards directions for the first time; the candidate tasks to assign according to the proposed heuristic are 1 and 12, so task 12 is assigned according to the CTM procedure and random selection. In the second step, after calculating the new task weights, there is one choice, which is task 11, so it will be assigned. Finally, the station time is $T(_{s1})$ = 7+4=11.

(3) Create workstation 2, calculate the task weights and according to the candidate list tasks 1 and 8 can be assigned to the current work station, but through random selection, task 8 will be assigned. As a result, the task weight in the backward direction changes and the next candidate tasks are 1 and 10. Both of the candidate tasks have the same weight, but task 1 is assigned to the current workstation randomly. According to the remaining station time and activities, more activities cannot be assigned to this station. Total station time equals $T(s_2)$ = 6+5=11.

(4) Create workstation 3, and according to the CTM procedure tasks 4 and 10 can be assigned to the current work station. Task 10, having a higher weight, is assigned first. Subsequently, between tasks 7 and 4, task 4 is assigned. Then with the same procedure task 2 is assigned. Finally, in accordance with the remaining station time task 9 will be assigned. The total process time for this station is $T(s_3)$ = 4 + 3 +3+1= 11.

(5) Create workstation 4, and the candidate tasks are 5 and 7. Task 5 will be assigned using random selection. In the next step, tasks 3 and 7 are candidates and task 3 is selected for assignment. The total process time for this station is $T(s_4)$ = 6 + 4 = 10.

(6) Create workstation 5; according to the candidate list, from tasks 6 and 7, task 6 is assigned. Finally, the last remaining task is 7, and it will be assigned to the last station. Here the algorithm ends because all the activities have been assigned. The total process time for this station is $T(s_3)$ = 5 + 2= 7.

The summary of results of the assigning process using the proposed method is given in Table II.

TABLE II
THE SUMMARY OF ASSIGNING PROCESS FOR U-SHAPE LINE USING PROPOSED CTM

CT=12 , CT^*=11			
Iteration	Candidate list	Assigned task	Station No.
1	1,12	12	1
2	11	11	1
3	1,8	8	2
4	1,10	1	2
5	4,10	10	3
6	7,4	4	3
7	2,7	2	3
8	9	9	3
9	5,7	5	4
10	3,7	3	4
11	6,7	6	5

12	7	7	5

III. Priority Heuristic Rule

Since assembly line balancing problems are considered as NP-hard problems, using heuristic methods to solve these problems is common as a result of the fact that most of constructive procedures that have been proposed in the area of SALBP-1 are based on priority heuristic rules [13]. These groups of procedures use priority heuristic rules for computing priority values for each task and build an operation ranking based on the given precedence relationships and task times. Some of the most effective priority heuristic rules that have been used recently by researchers are given in Table III.

Here, it should be noted that the notations used in heuristic rules in Table III are as follow:

t_i	Assembly time required to complete task i.
i, j	Task index.
CT	Station cycle time.
IS_i	Set of immediate successors of i.
N	Number of tasks to be balanced into stations.
IP_i	Set of immediate predecessors of i.
S_i	Set of all successors of i.
P_i	Set of all predecessors of i.
UB_i	Upper bound on the station to which i may be assigned.
LB_i	Lower bound on the station to which i may be assigned.
$[X]^+$	Smallest integer greater than or equal to X.

TABLE III
The List of Priority Heuristic Rules

Rule No.	Rule Name	Symbol	Definition	References
1	Shortest Processing Time	SPT	t_i	[7]
2	Maximum Number of Immediate Successors after task i	NIS	$\lvert IS_i \rvert$	[14]
3	Random Priority	RND	random	[15]
4	Smallest Task Number	STN	i	[15]
5	Maximum Ranked Positional Weight	RPW	$t_i + \sum_{j\in Si} tj$	[16]
6	Greatest (Processing Time Divided by the Upper Bound)	G_PT_UB	t_i / UB_i	[7]
7	Smallest Lower Bound	SLB	$[(t_i + \sum_{j\in Pi} tj)/CT]^+$	[7]
8	Minimum Slack	MSLK	$UB_i - LB_i$	[17]
9	Maximum Number of Immediate Predecessors	NIP	$\lvert IP_i \rvert$	[7]
10	Smallest Upper Bound	SUB	$N+1-[(t_i + \sum_{j\in Si} tj)/CT]^+$	[18]
11	Greatest Number of Successors	GNS	$\lvert S_i \rvert$	[7]
12	Greatest Average Rank Positional Weight	GARPW	$(t_i + \sum_{j\in Si} tj)/(\lvert S_i \rvert + 1)$	[18]

IV. Performance Criteria

Perfect balance of the assembly line can be defined as combining the work elements in order to execute them in such a way that at each workstation summation of all the elemental times is equal to cycle time. Whereas in most cases a perfect balance cannot be attained, some other measures are used to assess the performance and effectiveness of the balance. Although there are several performance indexes in the literature, some of the ones that are more efficient and used by most researchers in the area of SALBP-1 can be described by the following:

Number of Work Station (NWS): in this performance criterion, when NWS is less, there is a decrease in the number of stations and a better distribution of tasks so that minimizing the number of station is equivalent to increasing line efficiency. A production line with fewer work stations can result in lower labor costs and decrease the amount of space needed, so it can be considered as a more cost efficient plan [6], [7].

Line Efficiency (LE): line efficiency is the ratio of the summation of all stations' time to the cycle time and the number of stations. It shows the percentage of the line's usage. A greater LE results in an efficient line while the target value is set to 100; in addition, maximization of LE is equivalent to the decreasing the number of stations, and for a given solution it can be expressed as: [6], [19]

$$LE = \frac{\sum_{i=1}^{m} T(s_i)}{M \times CT} \times 100 \tag{4}$$

Smoothness Index (SI): the smoothness index is one of the important performance measures for the relative smoothness of a given production line. The aim of SI is to indicate the amount of idle time caused by the uneven assignment of work elements to stations. The target value for SI is set to zero, and thus a smaller SI results in a perfect balance and a minim amount of SI can be attained by decreasing the workload difference between workstations in the line so that these workloads are distributed at workstations as equally as possible. This measure is calculated as: [7]

$$SI = \sqrt{\frac{\sum_{i=1}^{m} (T(s_{max}) - T(s_i))^2}{M}} \tag{5}$$

Here it should be noted that, in equations (4) and (5) above, S is symbol for station.

V. Computational Results

In this section, the proposed procedure for solving assembly line balancing problems will be applied and tested by solving several assembly line test problems. To evaluate the proposed heuristic (CTM), the solution for each sample is calculated. The efficiency and effectiveness of both proposed procedures will be proved through comparing the results for performance measures for the proposed CTM and 25 other heuristic rules.

In order to demonstrate and evaluate the efficiency and effectiveness of the proposed methods, the computational results are from on three sets of problems available in the literature. The three sets are the Talbot-Set [20], the Hoffmann-Set [21], and the Scholl-Set [22]. Moreover, the sources of these problems along with their detailed descriptions are given on the homepage for assembly line optimization research (these set of benchmark problems can be downloaded from *http://www.assembly-line-balancing.de*). The efficiency of the proposed CTM is measured by six well-known benchmark problems**.** All six benchmark problems are solved with five different cycle times. Thus, on the whole 30 test problems are solved. All 30 test problems are solved for the proposed CTM and 12 other heuristic rules (as previously mentioned in Table III) to compare and show the efficiency and effectiveness of the proposed heuristic among all the mentioned heuristic rules. All mentioned methods are solved in ALBP-1 where the main objective is to minimize the number of workstation. Moreover, to evaluate the performance measures, the proposed approach and all other heuristics are compared by an additional performance index called SI, as stated previously. The computational results of the proposed CTM and the 12 other heuristics for all performance measures are given in Table IV, and for each solved problem the heuristic rule which got the best results for all performance measures appears in bold. The summary of results is depicted in Table V.

TABLE V
SUMMARY OF RESULTS OF COMPARING THE PROPOSED CTM AND 12 OTHER HEURISTIC METHODS

Rules number	CTM	1	2	3	4	5	6	7	8	9	10	11	12
Total optimal answer	14	4	3	1	2	5	2	1	2	1	2	1	3
Total problems	30	30	30	30	30	30	30	30	30	30	30	30	30
Percentage	46.6	13.3	10	3.3	6.6	16.6	6.6	3.3	6.6	3.3	6.6	3.3	10

According to Table V the proposed heuristic method to balance the U-shape assembly line achieved the best results over the 12 other heuristic methods, and according to the final results it has been clearly found that the proposed CTM has a better situation than the other considered methods. Moreover, Fig. 2 indicates the number of first place obtained by each method for the number of stations (line efficiency) and for the smoothness index as simultaneously compared to other methods in all 30 solved test problems in this study. In accordance with this graph, whereas the proposed CTM rule obtained the best result in 46.66% of problems (14 test problems out of 30) compared to the other methods, it is evident that this rule takes first place among the other methods. In addition, heuristic rule 5 (maximum ranked positional weight), which achieved the best results in only 5 problems out of 30 (16.66%), is in second place, though there is a large difference with the proposed CTM rule. Moreover, rules number 1 (shortest processing time) is in third place for having the best results for 13.3 % of the problems solved.

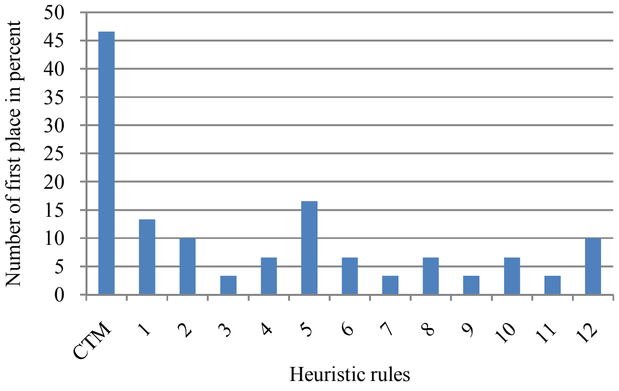

Fig. 2 The number of times of obtaining best solution by each method comparing to other methods

VI. CONCLUSIONS AND FUTURE WORK

In this study, a heuristic method has been presented to solve U-shape assembly line balancing problem in the area of type-l. At the first, the proposed heuristic was tested by solving several benchmark problems available in literature. The results were compared to 12 other heuristic rules in three performance measures, namely number of stations, smoothness index and line efficiency, while the main objective was to minimize the number of stations. According to the results, the proposed CTM rule led to very good results in finding the best results for all indexes simultaneously in 46.66% of solved problems, and it took first place among the 12 other heuristic rules. Accordingly, although it can be asserted that some of the other methods get good results in finding the minimum number of stations, the proposed CTM rule got significantly better results in other indexes under consideration and it is outperformed the other heuristics.

In light of the future implications of this study, further research can take advantage of the proposed methods in two aspects. First, the proposed CTM can be considered as a foundation for metaheuristics like tabu search, genetic algorithm, ant colony and so forth, or it can be used alone as a simple priority heuristic rule to solve the different types of assembly lines to achieve a reasonable solution in much less computational time. In addition, the proposed method can be used to solve different categories of assembly line balancing problems, such as type-2, or different kinds of lines such as straight, two-sided and so on.

TABLE IV
RESULTS OF SOLVING TEST PROBLEMS USING CTM AND HEURISTIC RULES GIVEN IN AN ORDER AS DEFINED IN TABLE III

Sample name	Cycle time	Index	CTM	Heuristic rules number											
				1	2	3	4	5	6	7	8	9	10	11	12
Arcus111	17067	NWS	**9**	10	9	10	9	9	9	9	9	9	10	9	9
		SI	**601.7104**	2460.073	842.2975	4729.029	968.15	1008.152	1053.016	892.1112	972.7018	629.903	4692.889	993.4397	944.9937
		LE	**97.9141**	88.12269	97.9141	88.12269	97.9141	97.9141	97.9141	97.9141	97.9141	97.9141	88.12269	97.9141	97.9141
	11378	NWS	14	15	14	14	14	14	14	14	14	14	**14**	14	14
		SI	1031.239	1929.73	1602.813	936.7861	1844.167	2257.551	1833.158	1844.728	2221.167	1455.276	**834.4851**	1858.735	2109.956
		LE	94.41717	88.12269	94.41717	94.41717	94.41717	94.41717	94.41717	94.41717	94.41717	94.41717	**94.41717**	94.41717	94.41717
	6540	NWS	24	30	25	26	25	24	24	25	24	**24**	26	24	25
		SI	569.2545	1775.663	995.903	1087.817	967.1692	621.5473	968.7222	990.2727	703.3164	**390.5001**	1105.367	498.1858	941.3162
		LE	95.81995	76.65596	91.98716	88.44919	91.98716	95.81995	95.81995	91.98716	95.81995	**95.81995**	88.44919	95.81995	91.98716
	6267	NWS	25	31	25	27	25	**25**	25	26	25	26	27	25	25
		SI	765.4326	1742.826	396.4131	1038.643	409.3497	**380.9961**	1038.203	765.5289	722.0263	765.5791	938.1481	382.7928	507.4585
		LE	95.99426	77.41472	95.99426	88.88357	95.99426	**95.99426**	95.99426	92.30217	95.99426	92.30217	88.88357	95.99426	95.99426
	6016	NWS	26	31	**26**	29	27	26	26	27	26	28	28	27	26
		SI	614.2345	1410.718	**346.3505**	1187.488	1007.214	507.8765	1060.155	1005.019	663.6654	1023.276	1045.571	752.371	447.6891
		LE	96.15321	80.64463	**96.15321**	86.20632	92.59198	96.15321	96.15321	92.59198	96.15321	89.28512	89.28512	92.59198	96.15321
Jackson	21	NWS	**3**	3	3	3	3	3	3	3	3	3	3	3	3
		SI	**5.567802**	8.266398	9.814955	9.255629	9.814955	9.814955	9.255629	9.814955	9.814955	7.023769	8.082904	5.802298	9.814955
		LE	**73.01587**	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587	73.01587
	14	NWS	**4**	4	4	4	4	4	4	4	4	4	4	4	4
		SI	**1.6779**	2.12132	4.527693	4.062019	3.082207	3.082207	5	3.082207	3.082207	4.123106	4.123106	3.082207	3.082207
		LE	**82.14286**	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286	82.14286
	13	NWS	**4**	4	4	4	4	4	4	4	4	5	5	4	4
		SI	**1.6779**	2.12132	2.12132	3	2.236068	2.236068	2.54951	2.236068	2.236068	4.404543	4.404543	1.870829	2.236068
		LE	**88.46154**	88.46154	88.46154	88.46154	88.46154	88.46154	88.46154	88.46154	88.46154	70.76923	70.76923	88.46154	88.46154
	10	NWS	6	6	**5**	6	5	5	5	5	5	6	6	5	5
		SI	2.309401	2.94392	**1.095445**	2.94392	1.414214	1.264911	1.414214	1.414214	1.414214	2.94392	2.94392	1.264911	1.414214
		LE	76.66667	76.66667	**92**	76.66667	92	92	92	92	92	76.66667	76.66667	92	92
	9	NWS	7	6	6	**6**	6	6	6	6	6	6	**6**	6	6
		SI	1.927248	1.914854	1.914854	**1.732051**	2	2.081666	2.081666	2	2.081666	2.081666	**1.732051**	2.081666	2
		LE	73.01587	85.18519	85.18519	**85.18519**	85.18519	85.18519	85.18519	85.18519	85.18519	85.18519	**85.18519**	85.18519	85.18519
Lutz1	2828	NWS	**6**	6	6	6	6	6	6	6	6	6	6	6	6
		SI	**552.4611**	630.1122	955.8612	918.731	974.0493	1046.252	995.9177	974.0493	867.7073	972.5835	824.0631	937.2591	974.0493
		LE	**83.33333**	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333	83.33333
	2357	NWS	7	**7**	7	7	7	7	7	7	7	7	7	7	7
		SI	690.874	**379.2978**	527.8452	587.6515	630.0358	692.8554	642.9446	630.0358	641.1561	540.8343	513.0625	503.8027	630.0358
		LE	85.70216	**85.70216**	85.70216	85.70216	85.70216	85.70216	85.70216	85.70216	85.70216	85.70216	85.70216	85.70216	85.70216
	2020	NWS	9	**8**	8	8	8	8	8	8	8	8	8	8	8
		SI	544.4794	**304.8852**	475.8697	512.5944	409.3531	408.4826	466.2757	409.3531	490.0041	316.1202	387.4364	388.7827	409.3531
		LE	77.77778	**87.5**	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5
	1768	NWS	9	**9**	9	9	9	9	9	9	9	9	9	9	9
		SI	310.4219	**219.6381**	273.8386	355.8514	346.5012	360.6611	344.3474	346.5012	345.8278	292.631	303.2777	270.8956	346.5012
		LE	88.86375	**88.86375**	88.86375	88.86375	88.86375	88.86375	88.86375	88.86375	88.86375	88.86375	88.86375	88.86375	88.86375
	1572	NWS	**10**	11	10	10	10	10	10	10	10	10	10	10	10
		SI	**122.457**	302.9497	135.8381	199.874	192.7465	180.9663	299.5891	181.9341	180.9663	166.7765	174.4615	138.7862	192.7465
		LE	**89.94911**	81.77192	89.94911	89.94911	89.94911	89.94911	89.94911	89.94911	89.94911	89.94911	89.94911	89.94911	89.94911

TABLE IV
RESULTS OF SOLVING TEST PROBLEMS USING CTM AND HEURISTIC RULES GIVEN IN AN ORDER AS DEFINED IN TABLE III

Sample name	Cycle time	Index	CTM	Heuristic rules number											
				1	2	3	4	5	6	7	8	9	10	11	12
Heskiaoff	324	NWS	**4**	4	4	4	4	4	4	4	4	4	4	4	4
		SI	**70.569**	77.7721	126.2953	135.5009	125.1699	126.325	136	125.106	125.704	126.6945	126.6945	126.3962	126.3962
		LE	**79.01235**	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235	79.01235
	256	NWS	5	5	5	5	5	**4**	5	5	5	5	5	5	5
		SI	56.34891	64.58947	82.50455	93.13861	85.89296	**0**	113.1495	67.64318	86.83087	85.5605	86.00116	85.8487	85.89296
		LE	80	80	80	80	80	**100**	80	80	80	80	80	80	80
	216	NWS	**5**	6	5	5	5	5	5	6	5	5	5	5	5
		SI	**12.34928**	56.36784	16.28496	13.82751	18.40652	24.16609	24.60081	67.53024	16.81071	14.44991	14.93988	23.74026	18.40652
		LE	**94.81481**	79.01235	94.81481	94.81481	94.81481	94.81481	94.81481	79.01235	94.81481	94.81481	94.81481	94.81481	94.81481
	205	NWS	6	6	6	6	6	**5**	6	6	6	6	6	6	6
		SI	81.24244	50.06662	57.04969	63.58197	57.29165	**0.447214**	82.87742	57.2771	56.74798	57.42822	58.86142	56.89171	57.29165
		LE	83.25203	83.25203	83.25203	83.25203	83.25203	**99.90244**	83.25203	83.25203	83.25203	83.25203	83.25203	83.25203	83.25203
	138	NWS	**8**	10	8	9	9	8	8	9	8	9	9	8	9
		SI	**11.34646**	35.31855	12.8938	40.11373	30.78961	23.53189	20.31625	32.02777	23.53189	30.01666	25.45803	23.45208	30.78961
		LE	**92.75362**	74.2029	92.75362	82.44767	82.44767	92.75362	92.75362	82.44767	92.75362	82.44767	82.44767	92.75362	82.44767
Roszieg	32	NWS	5	5	4	4	**4**	4	4	**4**	**4**	4	4	4	**4**
		SI	12.55388	7.937254	1.5	1.5	**0.866025**	1.118034	1.118034	**0.866025**	**0.866025**	1.118034	1.118034	1.118034	**0.866025**
		LE	78.125	78.125	97.65625	97.65625	**97.65625**	97.65625	97.65625	**97.65625**	**97.65625**	97.65625	97.65625	97.65625	**97.65625**
	25	NWS	6	6	**5**	6	6	**5**	6	6	6	6	6	**5**	**5**
		SI	2.972092	5.845226	**0**	8.256311	7.17635	**0**	9.407444	7.17635	7.17635	8.631338	6.204837	**0**	**0**
		LE	83.33333	83.33333	**100**	83.33333	83.33333	**100**	83.33333	83.33333	83.33333	83.33333	83.33333	**100**	**100**
	18	NWS	**8**	8	8	8	8	8	8	8	8	8	8	8	8
		SI	**2.573908**	2.893959	4.703722	5.03736	5.03736	5.03736	5.373546	5.062114	5.03736	3.724916	3.691206	4.756574	5.086747
		LE	**86.80556**	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556
	16	NWS	9	**9**	9	9	9	9	9	9	9	9	9	9	9
		SI	2.211083	**1.490712**	3.958114	2.768875	4.176655	4.203173	4.484541	3.511885	4.203173	3	3.036811	3.958114	4.176655
		LE	86.80556	**86.80556**	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556	86.80556
	14	NWS	**10**	11	10	10	10	10	10	10	10	10	10	10	10
		SI	**0.707107**	3.705033	2.983287	1.81659	2.387467	3.535534	2.738613	2.073644	3.24037	2.213594	2.024846	2.428992	2.428992
		LE	**89.28571**	81.16883	89.28571	89.28571	89.28571	89.28571	89.28571	89.28571	89.28571	89.28571	89.28571	89.28571	89.28571
Buxey	54	NWS	**7**	7	7	7	7	7	7	7	7	7	7	7	7
		SI	**9.289837**	10.60997	13.43769	10.30257	13.53303	13.53303	18.92089	10.19804	13.53303	13.76331	14.12192	13.46954	13.53303
		LE	**85.71429**	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429	85.71429
	47	NWS	8	9	8	8	8	8	**7**	8	8	8	8	8	8
		SI	9.67822	7.141428	11.36882	10.35616	10.06231	10.06231	**1.889822**	10.71214	10.06231	11.45644	11.45644	10.42833	10.06231
		LE	86.17021	76.59574	86.17021	86.17021	86.17021	86.17021	**98.48024**	86.17021	86.17021	86.17021	86.17021	86.17021	86.17021
	36	NWS	**10**	11	10	11	10	10	10	11	10	10	10	10	10
		SI	**2.792848**	7.077493	7.348469	8.034019	5.899152	5.899152	8.354639	8.769783	5.899152	4.516636	5.422177	5.91608	5.899152
		LE	**90**	81.81818	90	81.81818	90	90	90	81.81818	90	90	90	90	90
	33	NWS	**11**	13	12	11	11	11	11	13	11	12	11	11	11
		SI	**3.074824**	8.439832	8	3.692745	4.188729	4.188729	5.776126	8.673567	4.188729	8.386497	4.358899	4.431294	4.188729
		LE	**89.2562**	75.52448	81.81818	89.2562	89.2562	89.2562	89.2562	75.52448	89.2562	81.81818	89.2562	89.2562	89.2562
	30	NWS	13	14	13	13	**12**	**12**	**12**	13	**12**	13	13	13	**12**
		SI	6.523045	7.874008	6.409128	6.139406	**4.082483**	**4.082483**	**4.082483**	6.214128	**4.082483**	6.598368	6.917091	6.838803	**4.082483**
		LE	83.07692	77.14286	83.07692	83.07692	**90**	**90**	**90**	83.07692	**90**	83.07692	83.07692	83.07692	**90**

REFERENCES

[1] A. Scholl, and R. Klein, "ULINO: Optimally balancing U-shaped JIT assembly lines," *Int J of Prod Res*. vol. 37, no. 4, pp. 721-736, Mar. 1999.

[2] S. Ghosh, and R. J. Gagnon, "A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems," *Int J of Prod Res*. vol. 27, no. 4, pp. 637-670, Apr. 1989.

[3] M. E. Salveson, "The assembly line balancing problem," *J Ind Eng*. Vol. 6, no. 3, pp. 18-25, May-Jun. 1955.

[4] A. L. Gutjahr, and G. L. Nemhauser, "An algorithm for the line balancing problem," *Manag Sci*, vol. 11, no. 2, pp. 308-315, Nov. 1964.

[5] D. A. Ajenblit, and R. L. Wainwright, "Applying genetic algorithms to the U-shaped assembly line balancing problem," IEEE Int. Conf. Evolutionary Computation, ICEC, Anchorage, AK, USA 1998, pp. 96–101.

[6] S. G. Ponnambalam, P. Aravindan, and G. Mogileeswar Naidu, "Multi-objective genetic algorithm for solving assembly line balancing problem," *Int J Adv Manuf Technol*. vol. 16, no. 5, pp. 341-352, Apr. 2000.

[7] A. Baykasoglu, "Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems," *J Intell Manuf*. vol. 17, no. 2, pp. 217-232, Apr. 2006.

[8] N. Kriengkorakot, and N. Pianthong, "The Assembly Line Balancing Problem," *KKU Enginieering Journal*. vol. 34, no. 2, pp. 133-140, Mar.-Apr. 2007.

[9] A. Pinnoi, A branch and cut approach for certain problems in assembly systems, Texas A&M University, United States, Texas, Unpublished Ph.D. thesis.

[10] K. H. Oh, "Expert Line Balancing System (ELBS)," *Comput Indu Eng*, vol 33, no. 1-2, pp. 303-306, Oct. 1997.

[11] D. H. Yeh, and H. H. Kao, "A new bidirectional heuristic for the assembly line balancing problem," *Comput Indu Eng*, vol. 57, no. 4, pp. 1155-1160, Nov. 2009.

[12] W. B. Helgeson, and D. P. Birnie, "Assembly line balancing using the ranked positional weight technique," *J Ind Eng*. vol. 12, no. 6, pp. 394-398, Nov.-Dec. 1961.

[13] A. Scholl, and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *Eur J Oper Res*. vol. 168, no. 3, pp. 666-693, Feb. 2006.

[14] Tonge, F. M., "Summary of a heuristic line balancing procedure," *Manag Sci*, vol. 7, no. 1, pp. 21-42, Oct. 1960.

[15] C. L. E. Khaw, and S. G. Ponnambalam, "Multi-rule multi-objective ant colony optimization for straight and U-type assembly line balancing problem," 2009 Int. Conf. Automation Science and Engineering, CASE 2009, Bangalore, pp. 177-182.

[16] .Baykasoglu A, Dereli T, "Two-sided assembly line balancing using an ant-colony-based heuristic," *Int J Adv Manuf Technol*, vol 36, no. 5-6, pp. 582-58, Mar. 2008.

[17] R. Kolisch, "Efficient priority rules for the resource-constrained project scheduling problem," *J Oper Manag*, vol 14, no.3, pp. 179-192, Sep. 1996.

[18] J. Bautista, and J. Pereira, "Ant algorithms for assembly line balancing," Lecture Notes in Computer Science. Vol. 2463, pp. 65-75, Sep. 2002.

[19] U. Ozcan, and B. Toklu, "A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems," *J Intell Manuf*. vol. 20, no.1, pp. 123-136, Feb. 2009.

[20] F. B. Talbot, J. H. Patterson, and W. V. Gehrlein, "A comparative evaluation of heuristic line balancing techniques," *Manag Sci*, Vol. 32, no.4, pp. 430-454, Apr. 1986.

[21] T. R. Hoffmann, Eureka, "A hybrid system for assembly line balancing," *Manag Sci*, vol. 38, no.1, pp. 39-47, Jan. 1992.

[22] A. Scholl, "Data of assembly line balancing problems," (Working paper), Schriften zur Quantitativen Betriebswirtschaftslehre, vol 16, no.93, pp.1-32, Nov. 1993.