Pertanika J. Sci. & Technol. 20 (2): 355 - 369 (2012)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

A New Heuristic Method to Solve Straight Assembly Line Balancing Problem

Mohd Khairol Anuar Mohd Ariffin, Masood Fathi* and Napsiah Ismail

Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ABSTRACT

Assembly line balancing is well-known in mass production system but this problem is non-deterministic polynomial-time(NP)-hard, even for a simple straight line. Although several heuristic methods have been introduced and used by researchers, knowing and using an effective method in solving these types of problems in less computational time have a considerable place in the area of line balancing problem. In this research, a new heuristic approach, known as critical node method (CNM), was introduced and tested by solving several test problems available in the literature so as to solve straight assembly lines. Finally, the obtained results are compared with 9 other heuristic rules in some performance measures. Thus, it is concluded that the proposed CNM is better than the rest in all the measures.

Keywords: Assembly line balancing, heuristic, critical node method, straight line

INTRODUCTION

An assembly line consists of several workstations which are generally arranged along a material-handling system, specifically conveyor belt in which parts sequentially move along the line from station to station. A particular amount of assembly works are done in each workstation and the products are completed as they reach the end of the assembly line. Nowadays, most of the industries that are dealing with mass production system have been using a type of assembly lines due to the high-volume production, and complexity of products. Though, assembly line balancing problem has been under study for 50 years and a number of studies on different types of assembly line balancing problems are done based on the fact that the line balancing problem falls into non-deterministic polynomial-time (NP)-hard category (Gutjahr & Nemhauser, 1964; Ajenblit & Wainwright, 1998), exact methods

Article history:
Received: 12 April 2011
Accepted: 21 April 2011

Email addresses:
khairol@eng.upm.edu.my (Mohd Khairol Anuar Mohd Ariffin),
fathi.masood@gmail.com (Masood Fathi),
napsiah@eng.upm.edu.my (Napsiah Ismail)
*Corresponding Author

such as integer programming, dynamic programming cannot be used effectually to balance the assembly line problems. Thus, finding a new effective method is necessary.

According to the classification proposed by Ghosh and Gagnon (1989), different types of assembly line balancing problems were grouped into different categories, namely, Single and Multi/Mixed Model Deterministic, and Single and Multi/Mixed Model Stochastic. Most research carried out on assembly line is related to the Single Model Deterministic category, where the cycle time is deterministic and the aim is the optimization of efficiency. It consists of two assembly line balancing forms; one of the two is the original and the simplest type of the assembly line balancing problem known as the simple assembly line balancing (SALB) and the other is the added restriction or factors (e.g. parallel stations, zoning restrictions) which become the General Assembly Line Balancing Problem (GALB). In addition, it should be noted that SALBP can be categorized into two main parts (SALBP-1 and SALBP-2) so that their main objective can be considered as minimizing the number of workstations while cycle time is constant and minimizing the cycle time for a given number of workstations, respectively. Moreover, the other types of simple assembly line balancing belong to the GALB problems.

Balancing the assembly line needs some constraints, as follows:

- Precedence constraint should be satisfied.
- The cycle time is greater than or equal to the time of any work element.
- The workstation time should not exceed the cycle time.

Recently, one of the most important SALBP subdivisions, i.e. SALBP-1, has been studied precisely by many researchers (Bautista & Pereira, 2009). Consequently, as in the previous research by Scholl & Becker (2006), the present study focused on SALBP-1. Although several exact and heuristic approaches in the area of SALBP have been introduced by the researchers and a comprehensive survey can be found in Erel and Sarin (1998) and Scholl and Becker (2006), a number of methods have been suggested and developed by the researchers in the recent decades to find the optimum methods to overcome the complexity of assembly line balancing problem (ALBP). According to Rekiek and Delchambre (2005), all the available methods used in solving line balancing problems can be divided into two main categories, namely, the exact and approximated methods. In addition, in the case of SALBP-1, dynamic programming has been used to determine lower bounds on the number of workstations using the exact approaches several methods such as integer programming, branch and bound, and dynamic programming have been vastly applied to date. According to Baybars (1986) and Scholl and Klein (1999), however, most effective techniques are based on the Dynamic Programming (DP) as well as Branch and Bound (B&B) methods.

Furthermore, Rekiek and Delchambre (2005) state that the approximated methods are divided into two main groups, namely, the heuristic and metaheuristic methods. One of the first proposed heuristics used to solve assembly line balancing problems was the Ranked Positional Weight or RPW (Helgeson *et al*., 1961), but the rules may sometimes be mistakenly utilised as Kilbridge and Wester's heuristic (1996), Moodie and Young's (1965) method and so on. Additionally, Metaheuristics includes several methods, such as Ant Colony Optimization, Tabu search, Genetic Algorithms and simulated annealing, which are used to solve different line balancing problems like straight and U-shaped line (Hwang *et al*., 2008; Baykasoglu, 2006), two-sided (Özcan & Toklu, 2008), etc.

Since assembly line balancing problems are categorized as NP-hard problems, all the proposed computational methods face difficulties when solving large size problems. Therefore, the heuristic and metaheuristic methods are applied to overcome the difficulties and to obtain the optimal or near the optimal solution in a reasonable amount of time. Furthermore, Scholl and Becker (2006) asserted that most of effective procedures have been proposed in the area of simple assembly line balancing type-1 (SALBP-1) are based on the priority heuristic rules. Recently, several articles have been published on the metaheuristic methods so they are using priority heuristic rules as a foundation (Fathi *et al*., 2010). For example, Sabuncuoglu *et al*. (2000) and Ponnambalam *et al*. (2000) developed the genetic algorithm-based heuristic for SALBP. Meanwhile, Baykasoglo (2006) introduced a simulated annealing (SA) algorithm using several heuristic rules to solve U-shape and straight line.

METHODOLOGY OF THE PROPOSED HEURISTIC METHOD

Since using the heuristic method has a significant rule to solve assembly line balancing problems, research on this particular method is a hot topic for researchers. More recently, Yeh and Kao (2009) proposed a new heuristic method based on the Critical Path Method (CPM) for solving bidirectional assembly line balancing problem. In this study, a more effective heuristic method called the Critical Node Method (CNM) was introduced based on combining the main concepts of the assembly line balancing problem and project management issues. The main concept of the proposed CNM is based on the well-known rank positional weight (RPW) technique introduced by Helgeson and Birnie (1961) and the proposed method based on CPM by Yeh and Kao (2009).

According to the RPW technique, the task that needs to be assigned is the one that has followers' largest total time. In the RPW, the task with the highest positional weight is selected and assigned to the earlier station. Meanwhile, the weight of each task is computed by summing all the followers' time and each task has its own weight, the tasks with greater weight have more priority to be assigned to the appropriate workstation with respect to all constraints, such as precedence relationship. Moreover, the CPM is a technique used for managing and scheduling the projects during the implementation and it can be defined as the longest path (according to the time duration) from the first (the source) node to the last (the sink) node. In this method, the CPM calculates the longest path of the planned activities to the end of the project, and it computes the earliest and the latest time of every single task that can start and finish without making the project longer. In accordance with the above mentioned explanation about CPM and RPW, the proposed CNM computes the task weight, as follows:

The sequence of the tasks makes the critical path to start with the first task in the project and follows through to the last task in the project. In the first step, CPM is applied to compute the critical path for the assembly network. In this process, the algorithm starts from the first task of the critical path by summing all the critical tasks time to calculate the weight for the considered task. In the next step, the most critical task with the highest weight is removed from the assembly network and a new computation to determine the current critical path is done using the same procedure. This process is continued until all the tasks have gained their own weights. The tasks are assigned in a descending order of the weight in which as it satisfies the precedence relationship and does not exceed the station's remaining cycle time. Using this particular point of view, the critical nodes also play a pivotal role in assembly line balancing whereby any impediment in assigning the critical nodes may conduce to increase the number of the workstations which are directly associated with higher labour cost and inefficient human resource management.

The proposed CNM can be used to solve almost all types of assembly line problems such as straight and U shaped lines. However, the current study focused on solving the straight assembly line balancing problem. In this method, another criterion is introduced and used instead of the cycle time named OCT.

The parameters used in the proposed method as well as the calculation of introduced OTC are as follows:

$T(s_i)$	total time of each station
T(x)	time of each task
CT	cycle time
N	number of workstation
SCT	smallest feasible cycle time
S_t	set of all tasks
S_a	set of assigned tasks
S_u	set of unassigned tasks
MS	minimum number of stations
OCT	optimum cycle time

The new introduced cycle time is calculated as bellow:

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

$$\mathrm{SCT} = \sum_{i=0}^{n} T(x_i)/\mathrm{MS} \tag{2}$$

$$\mathrm{OCT} = [(\mathrm{SCT} + \mathrm{CT})/2] \tag{3}$$

Note that OCT is selected between SCT and CT. Although CT can be replaced by each value between CT and SCT, based on the researchers' experience, OCT will offer better results. To obtain the desired conditions, the following equations should be maintained in the solving process:

$$\mathrm{T}(s_i) = \sum_{X \in S_i} T(x) \leq \mathrm{CT} \qquad i = 1, ..., M \tag{4}$$

If $(x,y) \in P$, $x \in S_i$ and $y \in S_j$ then $i \leq j$ for all x. (5)

Equation (4) expresses that the sum of the times for all the assigned tasks to one station should not exceed the predetermined cycle time. Equation (5) ensures that the precedence constraints during the assigning process are satisfied.

In order to assign tasks to the current workstation, the critical tasks are determined by the proposed CNM. In relation to this fact, i.e. the higher weight of each task represents a higher degree of criticality, a high priority to assign is therefore gained. In other words, the task with a higher weight will be assigned sooner, and some tasks with lower weight (lower priority to assign) may be assigned to the current work station due to two reasons. The two reasons are as follows:

- To satisfy the precedence constraints means that a task with a lower priority can be assigned sooner to preserve the precedence relationship.
- The capacity of workstation is not fully completed means that a task with a higher weight is available to be assigned but the current workstation does not have enough time, but the remaining time is enough to assign some other tasks with lower weight (i.e. a lower priority to assign).

This process is continued until no task is left within the assembly network to be assigned. Here, it should be noted that in the situation with an equal weight for some of the tasks in the candidate list, there is no difference to select the tasks, so a task is selected by random chance.

The heuristic CNM based described above involves determining the criticality of each node within the assembly network and assigning the priority based on the obtained weight using the precedence relationship diagram. The assembly network is used to compute the weight of each task and those tasks with higher weight gain more priority for assigning. Task assignment process is continued until all the tasks have been assigned to any station, as shown by $S_t = Ø$. The proposed method should be done in the following steps:

1. Computing the value of OCT for the corresponding problem and replacing the existent cycle time by this value.
2. The set of all tasks within the assembly network is shown by S_t which represents the set of all the available tasks for assignment. The initial value of S_u is equivalent to S_t.
3. Weighing each single task and computing it using the CNM method; they are assigned to the stations based on the priority of tasks. In the entire task assignment procedure, precedence constraints should be satisfied even though the higher priority tasks are available. This particular process is continued until no task can be assigned to any workstation anymore. The set of the unassigned tasks is $S_u = S_t - S_a$.
4. $S_t \neq Ø$ expresses the task availability in the assembly network and this procedure goes to step 3. $S_t = Ø$ shows that all the tasks have been assigned to any station and the task assignment operation is ended.

SOLVING PROCESS OF THE PROPOSED CNM

In this sub-section, an example available in literature taken from Jackson (1956) is graphically shown to describe the proposed CNM.

Note that the assumption is CT = 21 sec. Then, MS, SCT and finally OCT are calculated using Equations 1, 2 and 3.

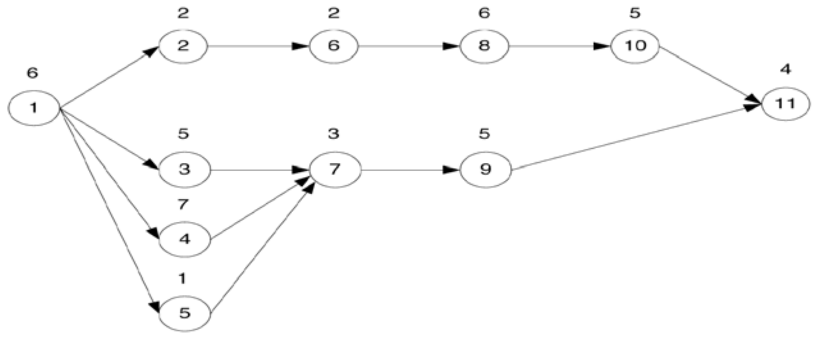

Fig.1: The network of Jackson's problem

Table 1: Weight computation using the CNM method

Task no.	1	2	3	4	5	6	7	8	9	10	11
Weight	25	19	17	19	13	17	12	15	9	9	4

The following is a description of the assigning process to clarify the proposed CNM.

1. Calculating MS = 46/21 = 2.19 and after rounding up, it is 3 and SCT = 46/3 = 15.33; therefore, OCT = [(21+15.33)/2]=18.16, so 15.33<18.16< 21, and since all the task times are integer, it should be rounded and set to 18.
2. Creating workstation 1 and calculating the weight for each task. This phase is shown in Table 1; the first candidate task to assign according to the proposed heuristic is task 1, and because it has a higher weight among the other tasks, so task 1 is assigned according to the CNM rule to the first workstation. In the second step according to the task weight, there are two different choices, namely, tasks 2 and 4. In this situation, a task is selected by random chance, and it is assumed that it is task 4 and then task 2 with higher weight is selected to be assigned. This is continued by tasks 3 and 6 with the same weight which can be assigned to the current station; however, according to the remaining cycle time, task 6 should be assigned to the current station. In accordance with the remaining cycle time for the current station, only task 5 can be assigned to this particular station. Although it does not have a higher weight but it can be assigned accorrding to the predefined assigning procedure. Finally, station time is T = (s_1) = 6+7+2+2+1=18.
3. Creating workstation 2 according to the tasks weight, task 3 should be assigned to the second station. After that, tasks 8 and 7 are assigned to the current station, respectively. According to the remaining cycle time, there is no other task to be assigned. Total station time equals T = (s_2) = 5+6+3=14.
4. Creating workstation 3, tasks 9 and 10 have the same weight, so one of them is selected by random so tasks 10 and 9 are assigned to the current work station, respectively. Then, only task 11 is available to be assigned, so task 11 is assigned to this station. The total process time for this station is T = (s_3) = 5 + 5 +4= 14.

A summary of the whole assigning process described above is shown in Table 2.

Table 2: A summary of the assigning process using the proposed CNM

Iteration	Candidate list	Assigned task	Remaining station time
1	1	1	12
2	2,4	4	5
3	2	2	3
4	6	6	1
5	5	5	0
6	3	3	13
7	8	8	7
8	7	7	4
9	9,10	10	13
10	9	9	8
11	11	11	4

HEURISTIC METHODS AND PERFORMANCE INDEXES

In this section, 9 heuristic rules recently used by the researchers are introduced and several benchmark problems are also solved for all the considered heuristics and the suggested CNM to compare and evaluate the CNM. The heuristic rules and their sign and parameters are given in Table 3. The definitions of the parameters used in the heuristic rules are listed in the following table:

Table 3: List of the heuristic rules

Rule No.	Rule Name	Symbol	Definition
1	Maximum positional weight of follower task	Max $\lvert S_i \rvert$	$\sum_{j \in s_i} t_j$
2	Maximum task time of immediate follower task	Max $\lvert IS_i \rvert$	$\lvert IS_i \rvert$
3	Minimum total number of predecessor tasks	NPS_i	$\lvert P_i \rvert$
4	Minimum total number of successor tasks	MiTNST	$\lvert S_i \rvert$
5	Maximum total time of successor tasks	MaTTST	$T\lvert S_i \rvert$
6	Minimum total time of successor tasks	MiTTST	$T\lvert P_i \rvert$
7	Maximum total number of predecessor tasks	MaTNPT	$\lvert P_i \rvert$
8	Maximum total number of successor tasks	MaTNST	$\lvert S_i \rvert$
9	Maximum total time of predecessor tasks	MaTTPT	$T\lvert P_i \rvert$
t_i	Assembly time required to complete task i		
i, j	Task index		
IS_i	Set of immediate successors of task i		
N	The number of tasks to be balanced into stations		
IP_i	Set of immediate predecessors of task i		
S_i	Set of all successors of task i		
P_i	Set of all predecessors of task i		

PERFORMANCE INDEXES

As stated above, the main objective of the assembly line balancing problem in the area of type-1 is to minimize the number of stations. To the researchers' best knowledge, most of the methods obtained the same results for the number of stations, and thus, evaluating the different heuristic methods using some other performance measures seems to be necessary. Although several indexes are available in the literature, two indexes (SI and LE) were selected and calculated in the current study. A brief definition of the indexes is given below:

1. Number of Work Station (NWS): The minimum index value shows a decrease in the required number of stations for assembly and better task distribution.
2. Smoothness Index (SI): The smoothness index is an index for the relative smoothness of a given assembly line. A smaller SI results in a smoother line, thereby, reducing the in-process inventory (Baykasoglu, 2006).

$$SI = \sqrt{\frac{\sum_{i=1}^{n}(T(s_{\max}) - T(s_i))^2}{N}} \tag{6}$$

3. Line Efficiency (LE): Line efficiency is a ratio between total station time to the product of cycle time and the number of workstations, which is represented as a percentage. The greatest LE results in an efficient line, which is expressed as follows (Ponnambalam *et al*., 2000):

$$LE = \frac{\sum_{i=1}^{n} T(s_i)}{N \times CT} \times 100 \tag{7}$$

It should be noted here that in the above equations $T(S_i)$, is the time of the i-th station, $T(S_{\max})$ is the maximum workstation time, N is the number of workstations and CT is the given cycle time.

RESULTS OF SOLVING TEST PROBLEMS

In this section, several test problems available in the literature, which can be downloaded from Scholl *et al*. (2010), are solved using the proposed CNM and 9 other mentioned heuristic rules. The results obtained for all the performance measures are given in Table 4. Morover, the optimal results in all the indexes obtained from the heuristic methods are bolted and presented in Table 4.

Meanwhile, a summary of the results obtained by the new method and 9 other methods is shown in Table 5. It is important to note that the comparison results of the new method and the other 9 methods in Table 5 are with respect to the assigned activities to the workstations with multi objectives. In other words, each method will get the first place if all their performance indexes are better than those of the others.

As shown in Table 5, all the indexes show the superiority of the proposd heuristic method in balancing the stright assembly line to 9 other methods. Based on the final results, it is clearly found that the proposed method (CNM) has a better situation than the rest of the methods taken into consideration.

Table 4: The results for the SALBP problems using the task assignment rules given in the order defined in Table 3

sample name	CT		CNM	1	2	3	4	5	6	7	8	9
Arcus 83	6842	NS	12.000	**12.000**	12.000	13.000	13.000	12.000	13.000	13.000	13.000	13.000
		SI	1010.00	**706.52**	768.32	1482.6	1448.7	727.70	1448.7	1482.2	1482.6	1482.6
		LE	92.209	**92.209**	92.209	85.116	85.116	92.209	85.116	85.116	85.116	85.116
	4732	NS	**17.000**	18.000	18.000	18.000	18.000	18.000	18.000	18.000	18.000	18.000
		SI	**421.312**	924.23	925.88	954.300	856.22	924.17	856.22	622.27	954.30	954.30
		LE	**94.111**	88.883	88.883	88.883	88.883	88.883	88.883	88.883	88.883	88.883
	4454	NS	**18.000**	19.000	20.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000
		SI	**515.253**	828.383	855.38	644.35	733.53	817.66	733.53	697.01	644.35	836.23
		LE	**94.431**	89.461	84.988	89.461	89.461	89.461	89.461	89.461	89.461	89.461
	4206	NS	**19.000**	20.000	20.000	21.000	22.000	20.000	22.000	20.000	21.000	20.000
		SI	**402.903**	725.53	506.02	929.18	983.41	728.65	983.41	713.16	929.18	755.83
		LE	**94.736**	89.999	89.999	85.713	81.817	89.999	81.817	89.999	85.713	89.999
	3985	NS	21.000	21.000	21.000	22.000	23.000	**21.000**	23.000	22.000	22.000	22.000
		SI	737.081	675.814	590.97	757.28	909.34	**414.63**	909.34	672.13	757.28	786.93
		LE	90.467	90.467	90.467	86.355	82.600	**90.467**	82.600	86.355	86.355	86.355

Table 4 (Cont.)

Mitchell	39	NS	**3.000**	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000
		SI	**2.377**	5.477	3.697	3.464	3.464	6.377	3.464	5.196	3.464	3.464
		LE	**89.744**	89.744	89.744	89.744	89.744	89.744	89.744	89.744	89.744	89.744
	26	NS	**5.000**	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000
		SI	**3.236**	6.957	6.943	6.213	5.020	7.987	5.020	6.213	6.213	6.213
		LE	**80.769**	80.769	80.769	80.769	80.769	80.769	80.769	80.769	80.769	80.769
	21	NS	**6.000**	6.000	6.000	6.000	6.000	6.000	6.000	6.000	6.000	6.000
		SI	**4.767**	6.096	6.096	6.671	5.115	7.382	5.115	6.646	6.671	6.671
		LE	**83.333**	83.333	83.333	83.333	83.333	83.333	83.333	83.333	83.333	83.333
	15	NS	**8.000**	8.000	9.000	9.000	10.000	8.000	10.000	8.000	9.000	9.000
		SI	**1.674**	2.318	4.447	4.738	4.970	2.318	4.970	2.318	4.738	4.738
		LE	**87.500**	87.500	77.778	77.778	70.000	93.750	70.000	87.500	77.778	77.778
	14	NS	**8.000**	8.000	9.000	10.000	10.000	8.000	10.000	9.000	10.000	10.000
		SI	**1.061**	1.173	3.215	4.848	4.970	1.173	4.970	4.069	4.848	4.848
		LE	**93.750**	93.750	83.333	75.000	75.000	93.750	75.000	83.333	75.000	75.000
Mertens	18	NS	**2.000**	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000
		SI	**0.707**	4.950	4.950	4.950	4.950	4.950	4.950	4.950	4.950	4.950
		LE	**80.556**	80.556	80.556	80.556	80.556	80.556	80.556	80.556	80.556	80.556
	15	NS	3.000	**2.000**	3.000	**2.000**	**2.000**	2.000	2.000	2.000	2.000	2.000
		SI	6.028	**0.707**	4.761	**0.707**	**0.707**	0.707	0.707	0.707	0.707	0.707
		LE	64.444	**96.667**	64.444	**96.667**	**96.667**	96.667	96.667	96.667	96.667	96.667
	10	NS	4.000	4.000	4.000	**3.000**	4.000	3.000	4.000	**3.000**	**3.000**	3.000
		SI	2.500	2.500	3.354	**0.577**	2.500	0.577	2.500	**0.577**	0.577	0.577
		LE	72.500	72.500	72.500	**96.667**	72.500	69.667	72.500	**96.667**	96.667	96.667
	8	NS	**5.000**	5.000	5.000	6.000	6.000	5.000	6.000	6.000	6.000	6.000
		SI	**1.414**	2.569	1.414	1.354	1.354	2.569	1.354	1.354	1.354	1.354
		LE	**72.500**	72.500	72.500	60.417	60.417	72.500	60.417	60.417	60.417	60.417
	6	NS	6.000	6.000	6.000	**6.000**	**6.000**	6.000	**6.000**	**6.000**	**6.000**	**6.000**
		SI	1.581	1.581	1.581	**1.354**	**1.354**	1.581	**1.354**	**1.354**	**1.354**	**1.354**
		LE	80.556	80.556	80.556	**80.556**	**80.556**	80.556	**80.556**	**80.556**	**80.556**	**80.556**

Table 4 (Cont.)

Lutz1	2828	NS	**6.000**	6.000	6.000	6.000	6.000	6.000	6.000	6.000	6.000	6.000
		SI	**770.10**	970.10	974.04	974.04	974.04	862.53	862.96	884.13	974.04	974.04
		LE	**83.333**	83.333	83.333	83.333	83.333	83.333	83.333	83.333	83.333	83.333
	2357	NS	7.000	**7.000**	8.000	7.000	7.000	7.000	7.000	7.000	7.000	7.000
		SI	543.55	**367.05**	716.92	408.58	484.67	540.17	484.55	484.80	408.58	408.58
		LE	85.702	**85.702**	74.989	85.702	85.702	85.702	85.702	85.702	85.702	85.702
	2020	NS	8.000	8.000	8.000	8.000	8.000	**8.000**	8.000	8.000	8.000	8.000
		SI	405.35	296.55	366.80	324.49	324.49	**290.31**	303.74	352.83	324.49	324.49
		LE	87.500	87.500	87.500	87.500	87.500	**87.500**	87.500	87.500	87.500	87.500
	1768	NS	9.000	9.000	9.000	9.000	9.000	9.000	**9.000**	9.000	9.000	9.000
		SI	226.09	225.42	230.72	235.03	248.48	260.32	**203.313**	225.88	235.03	235.03
		LE	88.864	88.864	88.864	88.864	88.864	88.864	**88.864**	88.864	88.864	88.864
	1572	NS	**11.000**	11.000	11.000	11.000	11.000	11.000	11.000	11.000	11.000	11.000
		SI	**237.23**	365.79	397.41	396.59	396.59	289.11	398.97	369.29	396.59	396.59
		LE	**81.772**	81.772	81.772	81.772	81.772	81.772	81.772	81.772	81.772	81.772
Roszieg	32	NS	5.000	**4.000**	5.000	5.000	5.000	**4.000**	5.000	5.000	5.000	5.000
		SI	12.657	**1.118**	12.116	12.689	11.713	**1.118**	11.713	12.149	12.689	12.689
		LE	78.125	**97.656**	78.125	78.125	78.125	**97.656**	78.125	78.125	78.125	78.125
	25	NS	6.000	6.000	**6.000**	6.000	6.000	6.000	6.000	6.000	6.000	6.000
		SI	8.727	7.246	**5.148**	6.843	7.200	7.246	7.200	7.583	6.843	6.843
		LE	83.333	83.333	**83.333**	83.333	83.333	83.333	83.333	83.333	83.333	83.333
	18	NS	8.000	8.000	8.000	**8.000**	8.000	8.000	8.000	8.000	**8.000**	8.000
		SI	5.062	5.062	5.062	**1.768**	2.894	4.730	2.894	4.228	**1.768**	2.806
		LE	86.806	86.806	86.806	**86.806**	86.806	86.806	86.806	86.806	**86.806**	86.806
	16	NS	9.000	9.000	9.000	**9.000**	**9.000**	9.000	**9.000**	9.000	**9.000**	**9.000**
		SI	4.203	3.283	4.203	**3.073**	**3.073**	3.283	**3.073**	3.416	**3.073**	**3.073**
		LE	86.806	86.806	86.806	**86.806**	**86.806**	86.806	**86.806**	86.806	**86.806**	**86.806**
	14	NS	11.000	10.000	**10.000**	10.000	10.000	10.000	10.000	10.000	10.000	10.000
		SI	2.860	2.627	**2.470**	2.510	2.550	2.627	2.550	3.826	2.510	2.510
		LE	81.169	89.286	**89.286**	89.286	89.286	89.286	89.286	81.169	89.286	89.286

Table 4 (Cont.)

Gunther		NS	7.000	7.000	7.000	**7.000**	7.000	7.000	7.000	7.000	**7.000**	**7.000**
	81	SI	29.155	29.135	20.410	**16.912**	20.452	29.135	20.452	20.494	**16.912**	**16.912**
		LE	85.185	85.185	85.185	**85.185**	85.185	85.185	85.185	85.185	**85.185**	**85.185**
		NS	**8.000**	8.000	8.000	8.000	9.000	8.000	9.000	9.000	8.000	8.000
	69	SI	**8.870**	13.120	13.262	12.515	18.421	13.120	18.421	17.648	12.515	13.290
		LE	**87.500**	87.500	87.500	87.500	77.778	87.500	77.778	77.778	87.500	87.500
		NS	9.000	9.000	9.000	10.000	10.000	9.000	10.000	10.000	10.000	10.000
	61	SI	11.470	10.520	9.510	16.097	15.852	10.520	15.852	17.009	16.097	16.047
		LE	87.978	87.978	87.978	79.180	79.180	87.978	79.180	79.180	79.180	79.180
		NS	**10.000**	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000
	54	SI	**6.156**	7.880	7.176	7.342	6.834	7.880	6.834	9.160	7.342	7.218
		LE	**89.444**	89.444	89.444	89.444	89.444	89.444	89.444	89.444	89.444	89.444
		NS	12.000	12.000	12.000	12.000	**11.000**	12.000	**11.000**	12.000	12.000	12.000
	49	SI	12.480	11.836	13.817	10.634	**7.224**	11.836	**7.224**	10.712	10.634	10.642
		LE	82.143	82.143	82.143	82.143	**89.610**	82.143	**89.610**	82.143	82.143	82.143
		NS	**15.000**	16.000	17.000	16.000	18.000	16.000	18.000	18.000	16.000	16.000
	41	SI	**12.694**	14.506	17.222	16.491	18.118	14.506	18.118	19.019	16.491	16.342
		LE	**78.537**	73.628	69.297	73.628	65.447	73.628	65.447	65.447	73.628	73.628
Mansoor		NS	3.000	2.000	3.000	3.000	3.000	3.000	3.000	**2.000**	3.000	3.000
	94	SI	33.357	2.121	33.357	33.357	33.357	32.378	33.357	**0.707**	33.357	33.357
		LE	65.603	98.404	65.603	65.603	65.603	65.603	65.603	**98.404**	65.603	65.603
		NS	4.000	4.000	4.000	4.000	4.000	4.000	4.000	**4.000**	4.000	4.000
	62	SI	20.561	18.688	20.106	18.954	18.954	18.688	18.954	**14.603**	18.954	16.741
		LE	74.597	74.597	74.597	74.597	74.597	74.597	74.597	**74.597**	74.597	74.597
		NS	5.000	5.000	5.000	5.000	5.000	5.000	5.000	**4.000**	5.000	5.000
	48	SI	10.918	12.296	12.296	12.296	12.296	12.296	12.296	**2.500**	12.296	12.296
		LE	77.083	77.083	77.083	77.083	77.083	77.083	77.083	**96.354**	77.083	77.083
Bowman		NS	5.000	5.000	5.000	**5.000**	**5.000**	5.000	**5.000**	**5.000**	**5.000**	5.000
	20	SI	5.916	5.916	5.916	**3.873**	**3.873**	5.916	**3.873**	**3.873**	**3.873**	5.916
		LE	75.000	75.000	75.000	**75.000**	**75.000**	75.000	**75.000**	**75.000**	**75.000**	75.000

Table 5: A summary of the results comparing the proposed method and 9 other methods

Rules Number	CNM	1	2	3	4	5	6	7	8	9
Total optimal answer	15	4	3	6	5	4	6	7	5	4
Total problems	35	35	35	35	35	35	35	35	35	35
Percentage	42.85%	11.42%	8.57%	17.14%	14.28%	11.42%	17.14%	20%	14.28%	11.42%

Finally, Fig.2 demonstrates the number of times every method obtained the first place in comparison to the other methods. According to this graph, it can be asserted that the proposed CNM achieved the optimal number of stations in 15 test problems out of 35 and thus took the first place among all the methods. In addition, the second place goes to the heuristic rule 7 (i.e. the maximum total number of the predecessor tasks), whereas rule 6 (the minimum total time of successor tasks) and rule 3 (the minimum total number of predecessor tasks) are both in the third place by achieving the best results for 17.14% of the solved problems.

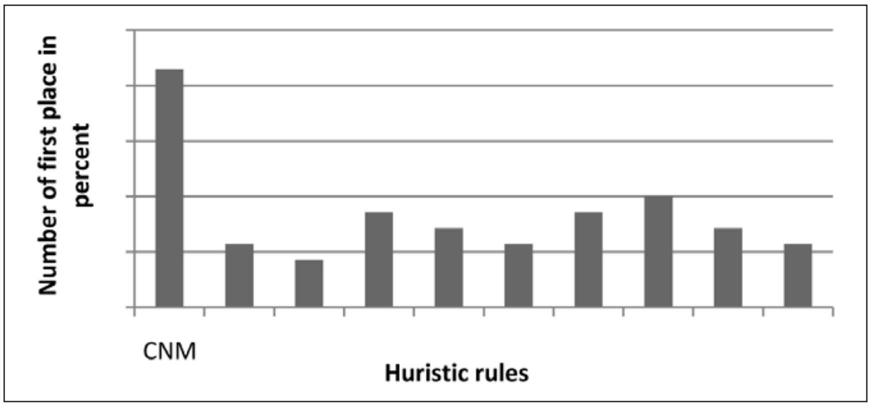

Fig.2: The number of times the best solution is obtained by each method in comparison to the other methods

CONCLUSION

Based on the results obtained for the performance measures, it can easily be concluded that the proposed heuristic method (CNM) has given better results in assigning tasks and in minimizing the number of workstations. Although some other methods have obtained good results for a number of the workstations, they do not give considerable results for other indexes like smoothness index. Another advantage of the proposed CNM can be considered in achieving good results in a reasonable of time. Since the heuristic methods are the foundation of the metaheuristic methods, the proposed method can be used as the main base for most of the metaheuristic methods like simulated annealing, genetic algorithm and ant colony optimization. Furthermore, although the proposed method are introduced and solved for stight line in the area of type-1, it can be efficiently applied for other kind of assembly lines such as parallel, U-shaped, and other types of assembly line balancing problem such as type-2 in future research.

REFERENCES

Ajenblit, D. A., & Wainwright, R. L. (1998) *Applying genetic algorithms to the U-shaped assembly line balancing problem.* Paper presented at the Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, Anchorage, AK, USA.

Bautista, J., & Pereira, J. (2009). A dynamic programming based heuristic for the assembly line balancing problem. *European Journal of Operational Research, 194*(3), 787-794.

Baybars, I. (1986). Efficient heuristic method for the simple assembly line balance problem. *International Journal of Production Research, 24*(1), 149-166.

Baykasoglu, A. (2006). Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. *Journal of Intelligent Manufacturing, 17*(2), 217-232.

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

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

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

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

Gutjahr, A. L., & Nemhauser, G. L. (1964). An algorithm for the line balancing problem. *Management Science, 11*(2), 308-315.

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

Hwang, R. K., Katayama, H., & Gen, M. (2008). U-shaped assembly line balancing problem with genetic algorithm. *International Journal of Production Research, 46*(16), 4637-4649.

Jackson, J. R. (1956). A computing procedure for a line balancing problem. *Management Science, 2*(3), 261-271.

Kim, Y. K., Song, W. S., & Kim, J. H. (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers and Operations Research, 36*(3), 853-865.

Kilbridge, M. D., & Wester, L. (1996). A heuristic method of assembly line balancing. *Journal of Industrial Engineering, 12*, 394-398.

Kao, E. P. C., & Queyranne, M. (1982). *On dynamic programming methods for assembly line balancing*. Paper presented at the Operations Research.

Masood Fathi, M. K. A. Ariffin, & Napsiah Ismail (2010). A note on "A multi-objective genetic algorithm for solving assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology, 50*(5-8), 771-773.

Moodie, C. L., & Young, H. H. (1965). A heuristic method of assembly line balancing for assumptions of constant/variable work element times. *Journal of Industrial Engineering, 16*, 456-467.

Nicosia, G., Pacciarelli, D., & Pacifici, A. (2002). Optimally balancing assembly lines with different workstations. *Discrete Applied Mathematics, 118*(1-2), 99-113.

Özcan, U., & Toklu, B. (2008). A tabu search algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, 1-8.

Ponnambalam, S. G., Aravindan, P., & Mogileeswar Naidu, G. (2000). Multi-objective genetic algorithm for solving assembly line balancing problem. *International Journal of Advanced Manufacturing Technology, 16*(5), 341-352.

Rekiek, B., & Delchambre, A. (2005). Assembly Line Design: *The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms*. Springer Series in Advanced Manufacturing.

Sabuncuoglu, I., Erel, E., & Alp, A. (2000). Ant colony optimization for the single model U-type assembly line balancing problem. *International Journal of Production Economics.*

Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research, 168*(3), 666-693.

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

Scholl, A., Boysen, N., Fliedner, M., & Klein R. (n.d.). Assembly line optimization research. Retrieved on June 20, 2010 from http://www.assembly-line-balancing.de.

Yeh, D. H., & Kao, H. H. (2009). A new bidirectional heuristic for the assembly line balancing problem. *Computers and Industrial Engineering, 57*(4), 1155-1160.