G.U. Journal of Science
22(4): 313-323 (2009)

www.gujs.org

A Tabu Search Algorithm for the Parallel Assembly Line Balancing Problem

Uğur ÖZCAN [1♠], Hakan ÇERÇİOĞLU [2], Hadi GÖKÇEN [2], Bilal TOKLU [2]

[1] *Selçuk University, Faculty of Engineering, Department of Industrial Engineering, 42075 Konya, Turkey*

[2] *Gazi University, Faculty of Engineering and Architecture, Department of Industrial Engineering, 06570 Ankara, Turkey*

Received: 18.02.2009 Revised: 17.04.2009 Accepted: 08.09.2009

ABSTRACT

In a production facility there may be more than one straight assembly line located in parallel. Balancing of parallel assembly lines will provide the flexibility to minimize the total number of workstations due to common resource. This type of problem is called as parallel assembly line balancing (PALB) problem. In this paper, a tabu search based approach is proposed for PALB problem with aim of maximizing line efficiency (*LE*) (or minimizing number of stations) and minimizing variation of workloads (*V*). This study is based on the study of Gökçen et al. [1]. The proposed approach is illustrated on a numerical example and its performance is tested on a set of well-known problems in the literature. This study is the first multi objective parallel assembly line balancing study in the literature.

Key Words: Assembly line balancing; Parallel assembly lines; Tabu search.

1. INTRODUCTION

Assembly line balancing (ALB) is an attempt to allocate equal amounts of work to the various workstations along the line. The fundamental line balancing problem is how to assign a set of tasks to an ordered set of workstations, such that the precedence relations and some performance measures (minimizing the number of workstation, cycle time, idle time, *etc*.) are satisfied [2]. The first analytical study on ALB problem was done by Salveson [3], who presented a mathematical formulation of ALB problem and suggested a solution procedure. After that time, many heuristics and optimal procedures have been proposed for the solution of ALB problem.

Assembly lines can be classified into two general groups: traditional assembly lines (with single and multi/mixed products) and U-type assembly lines (with single and multi/mixed products). In traditional assembly lines, the line configuration is straight and the entrance and the exit of the assembly line are in different position. For the studies on traditional assembly line balancing problem, the review papers of Baybars [4], Ghosh and Gagnon [2], Erel and Sarin [5], and Scholl and Becker [6] can be investigated. U-type assembly line configuration which has improved the line balance efficiency has been widely used by manufacturers. Design of these lines is differ from the others, that is, entrance and exit of the assembly line is in same position. U-type assembly line balancing problem was first studied by Miltenburg and Wijngaard [7]. Then, Urban [8] has developed an integer programming formulation of the U-type assembly line balancing problem. Several solution techniques have been developed for the U-type assembly line balancing problems to date. A detailed review of exact and heuristic procedures for solving this problem can be seen in Becker and Scholl [9].

Although the literature on traditional and U-type ALB problems is extensive, the studies on parallel lines are quite little. In designing the parallel lines, Süer and Dagli [10] have suggested heuristic procedures and

♠Corresponding author, e-mail: uozcan@selcuk.edu.tr

algorithms to dynamically determine the number of lines and the line configuration. Also, Süer [11] has studied alternative line design strategies for a single product. Other researches involving parallel workstation have focused on the simple assembly line balancing problem [12] and mixed-model production line balancing problem [13-15]. These studies on parallel lines are logically different from the approach of Gökçen et al. [1]. The new problem presented by Gökçen et al. [1] which more than one assembly line is balanced simultaneously with common resources has been derived from the traditional and U-type ALB problem. Aim of the PALB problem is to minimize the number of workstations by balancing of two or more assembly lines together. In the literature, only two studies have been published to solve the PALB problem. First study belongs to Gökçen et al. [1]. They have developed a mathematical model and heuristic procedure for PALB problem. Second one is the study of Benzer et al. [16]. They have proposed a network model for PALB problem. The presented study is directly related to the PALB problem of Gökçen et al. [1].

In this study a new approach for PALB problem is developed with aim of maximizing *LE* (or minimizing number of stations) and minimizing *V*. This study is the first multi objective parallel assembly line balancing study in the literature. The reminder of this paper is organized as follows: In the following section; PALB problem is explained and Gökçen et al. [1]'s mathematical formulation is presented. In Section 3, the proposed approach is described and clarified on an example problem. The performance of the proposed approach is tested and discussed in Section 4. Finally, the conclusions are given in Section 5.

2. PALB PROBLEM

The idea of common balancing of more than one line is defined first by Gökçen et al. [1]. PALB problem aims to minimize the number of workstations by balancing of two or more assembly lines together. Task assignment to common workstations of the parallel assembly lines is realized by using the precedence diagrams of each product manufactured on each assembly line separately. The common assumptions of PALB problem are presented as follows:

- Only one product is produced on each assembly line.
- Precedence diagrams for each product are known.
- Task performance times of each product are known.
- Operators working in each workstation of the each line are multi-skilled (flexible workers).
- It can also be worked each side of any line.

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

Gökçen et al. [1]'s mathematical formulation and notations used for PALB problem are given below;

C : cycle time

h : line number, $h = 1, \ldots, H$.

k : station number, $k = 1, \ldots, K$.

$\|M_{hk}\|$: total number of tasks (that can be) assigned to station k on line h.

n_h : number of tasks on line h.

t_{hi}: performance time of task i on line h.

K_{max}: maximum number of stations.

P_h: set of precedence relationships in precedence diagram of line h.

$x_{hik} = 1$ if task i in line h is assigned to station k; 0 otherwise.

$U_{hk} = 1$ if station k is utilized in line h; 0 otherwise.

$z_k = 1$ if station k is utilized; 0 otherwise.

Objective Function:

$$Min \sum_{k=\lceil k_{min} \rceil}^{K_{max}} z_k \quad (1)$$

Constraints:

$$\sum_{k=1}^{K_{max}} x_{hik} = 1, \quad i = 1,\ldots,n_h, \quad h = 1,..,H. \quad (2)$$

$$\sum_{i=1}^{n_h} t_{hi} x_{hik} + \sum_{i=1}^{n_{h+1}} t_{(h+1)i} x_{(h+1)ik} \le C z_k, \quad k = 1,\ldots,K_{max}, \quad h = 1,\ldots,H\text{-}1. \quad (3)$$

$$\sum_{i=1}^{n_h} x_{hik} - \|M_{hk}\| \; U_{hk} \le 0, \quad h = 1,\ldots,H \;\; k = 1,\ldots,K_{max}. \quad (4)$$

$$U_{hk} + U_{(h+a)k} = 1, \quad h = 1,\ldots,H\text{-}2 \quad a = 2\ldots\ldots H\text{-}h \;\; k = 1,\ldots\ldots,K_{max}. \quad (5)$$

$$\sum_{k=1}^{K_{max}} (K_{max} - k + 1)(x_{hrk} - x_{hsk}) \ge 0, \quad \forall\, (r,s) \in P_h. \quad (6)$$

$x_{hik}, z_k, U_{hk} \in \{0,1\}$ for all h, i, k.

The objective of the above formulation is to minimize the total number of workstations utilized in the production facility. Constraint (2) ensures the assignment of all tasks to a station and also each task is assigned only once. Constraint (3) ensures that the work content of any station does not exceed the cycle time. Constraints (4) and (5) ensure that an operator working at station k and line h can perform task(s) from only one adjacent line (i.e. operator in line h can perform tasks in line h+1 or h-1). Constraint (6) ensures that the precedence constraints are not violated on the line h precedence diagrams.

In order to explain the PALB concepts, we have used two classical problems (Merten's 7-task problem and Jaeschke's 9-task problem) which precedence diagrams for two different products (two assembly lines) are given In Figure 1. The numbers within the nodes represent tasks and the arrow (or arcs) connecting the nodes specifies the precedence relations. The numbers next to the nodes represent task performance times. When each product in the problem is balanced with a cycle time of 11, it can be seen that all tasks are performed at 7 workstations in the traditional assembly line (Figure 2a, 2b), whereas all tasks are performed at 6 workstations (one less than the traditional line solutions) in parallel assembly line (Figure 2c).

The operator of Station I in Figure 2c, first completes tasks 1 and 2 on the line I and then completes task 1 on the line II. As mentioned before, operators in a parallel assembly line may work on two different items within the same cycle time. The workstations which are opened both Assembly Line I and Assembly Line II are called as *common workstations*, and the other workstations are called as *separate workstations*. In this case, Workstations I, II, V and VI are common workstations, and Workstations III and IV are separate workstations.

3. THE PROPOSED TABU SEARCH ALGORITHM

3.1. Introduction to the proposed tabu search algorithm

In the mathematical complexity, ALB problem is NP-hard class of combinatorial optimization problems [17]. The combinatorial structure of this problem makes it difficult to obtain an optimal solution when the problem size increases. PALB problem is also NP-hard class. In addition to the mathematical complexity of ALB problem, PALB problem has also an additional level of complexity, since two or more assembly lines are balanced simultaneously. A binary mathematical formulation is presented in Section 2 is NP-hard. Since it is impossible for the mathematical formulation presented in Section 2 to reach the optimal or feasible solutions in a reasonable computation time for large-sized problems, the feasible solutions which may be close optimal solutions can be obtained by heuristic or approximate solution methods. Tabu search (TS), defined and developed primarily by Glover [18, 19], is one of the most effective heuristic optimization methods using local search techniques to find possible optimal or near-optimal solutions of many combinatorial optimization problems. There are several applications of tabu search algorithm for ALB problems in the literature [20-22]. Tabu search algorithm consists of several elements called as move, neighborhood, initial solution, search strategy, memory structure, aspiration criterion and stopping rules. In this section, the specific characteristics of the proposed TS algorithm to TALB problem are presented.

The proposed TS algorithm starts with an *initial solution* (x_0) and stores it as the *current solution* (x_k) and the *best solution* (x^*). The cost of initial solution ($f(x_0)$) becomes the current value of the objective function ($f(x_k)$) and the best value of the objective function ($f(x^*)$). The *neighborhood solutions* of x_k are then generated by a *move* (m). These are candidate solutions. They are evaluated by the objective function and a *candidate solution* (x'_k) which is the best not tabu or satisfies the *aspiration criterion* is selected as the new x_k. This selection is called a move and added to *tabu list* (TL), the oldest move is removed from tabu list if it is overloaded. If the new x_k is better than x^*, it is stored as the new x^*. Otherwise, x^* remains unchanged. This searching process is repeated until the termination criterion is met. In the remainder of this section, a detailed description of the proposed TS algorithm is given.

<u>Initial solution:</u> Initial solution can be obtained by using a constructive heuristic or by getting a feasible solution generated randomly. The proposed approach uses an assignment order of tasks which is generated randomly.

<u>Move:</u> In this study, in generating a new neighborhood solution from current solution, *swap* has been used. This neighborhood contains all those permutations $M(x_k)$ obtained from x_k by swapping the assignment order of tasks placed at the bth position and the randomly selected ath position, *i.e.*,

$$x_k = (AO_1, \ldots, AO_a, \ldots, AO_b, \ldots, AO_n)$$

$$m(x_k) = (AO_1, \ldots, AO_b, \ldots, AO_a, \ldots, AO_n)$$

$$M(x_k) = (AO_1, \ldots, AO_b, \ldots, AO_a, \ldots, AO_n)\ ;$$

$$\forall b \in \{1,\ldots,n\}, a \neq b.$$

where,

$m(x_k)$: a neighborhood of x_k which is obtained by swapping the assignment order of tasks in positions a and b.

n: total number of tasks on parallel assembly lines ($n = \sum_{h=1}^{H} n_h$)

AO_j : assignment order of task j

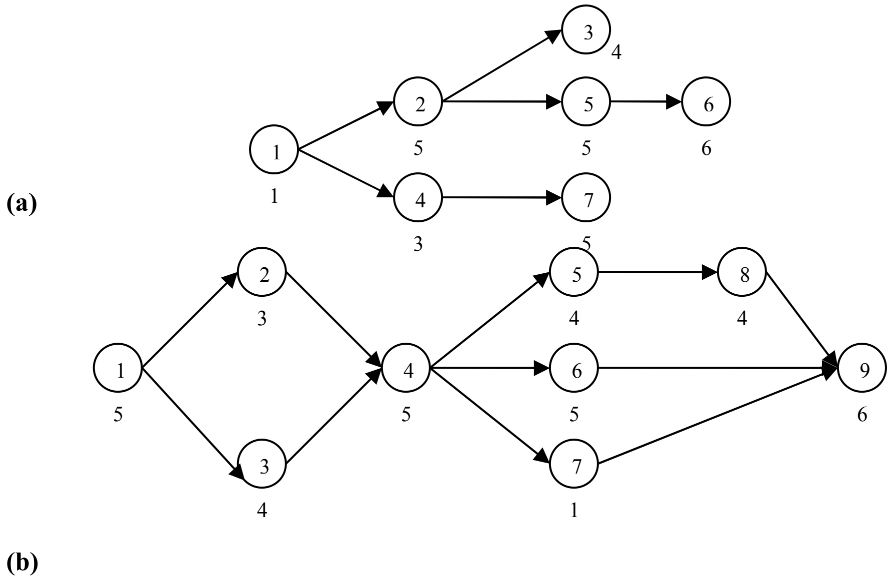


Figure 1. The precedence diagram and the task times of (a) the 7-task problem and (b) the 9-task problem.

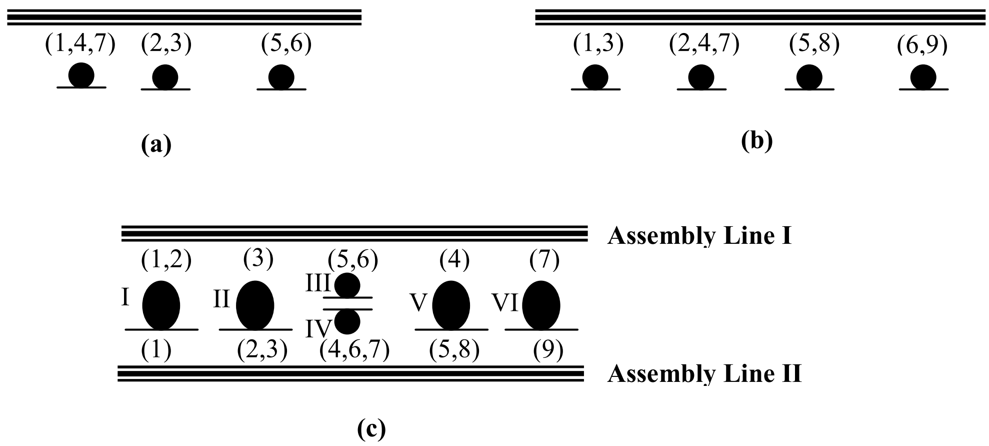


Figure 2. The task assignments of (a) traditional assembly line for the 7-task problem, (b) traditional assembly line for the 9-task problem and (c) parallel assembly lines.

Tabu list: In this study, tabu list which consists of two-dimensional array has been used to check if a move from a solution to its neighborhood is forbidden or allowed. When a pair of tasks is declared as tabu, TL[a][b] is determined as *current iteration number* + *tabu size*. If TL[a][b] is empty, then the assignment orders of task a and task b are not forbidden for swap move. Otherwise, if TL[a][b] is T, then the assignment orders of task a and task b cannot be moved until iteration T. Initially, TL is empty and *tabu size* is $\sqrt{n}$.

Size of neighborhood: Size of neighborhood determines how many neighborhood solutions will be searched before choosing a move which leads to next step. In this study, neighborhood structure of *swap is used and t*he size of neighborhood solutions is also (n-1).

Aspiration criterion: In this study, if a move is in the tabu list and it gives a better solution than the best objective function value obtained so far, then this move is applicable.

Termination rule: In this study, iteration number is used as termination rule and considered as n.

3.2. Building a feasible solution

Cycle times of parallel assembly lines can be same or different. For both cases, the following procedure can be used.

a) Normalize operation times of tasks.

$$t_{ih}^{norm} = t_{ih} / c_h \qquad i = 1,...,n_h \qquad h = 1,...,H.$$

b) Define the cycle time of system.

$$c = 1.0$$

In obtaining a feasible solution, normalized operation times should be used and cycle time condition (C=1.0) should be controlled for each assembly line. Feasible solution procedure is given below:

a) Start with $m(x_{iter})$ $k = 1$, $IY_k = 0$.

b) Establish an assignable task set from the tasks that have not assigned yet, by treating the precedence relations for all independent assembly lines. In this set, there may be several tasks from both assembly lines. If all tasks are assigned, calculate the objective function.

c) Select the first task i *(*on assembly line h*)* with highest priority assignment order from the assignable task set that satisfies the $IY_k + t_{ih}^{norm} \leq 1.0$ constraint. Assign the related task to station k, calculate the station workload, $IY_k = IY_k + t_{ih}^{norm}$ and go to **b**. If the cycle time constraint is not satisfied, then $k = k + 1$, $IY_k = 0$ and go to **b**.

3.3. Performance criteria and objective function

Two performance criteria (LE and V) have been taken into consideration. These performance criteria have also been used by Hwang et al. [23]. Number of stations can be decreased by maximizing the LE. Also workload difference between stations can also be reduced by the minimizing the V (that is, it is possible to distribute these workloads to stations as equal as possible). Calculations of $f_1(LE)$ and $f_2(V)$ for a given solution with K stations are as follows;

$$f_1(LE) = \frac{\sum_{h=1}^{H}\sum_{i=1}^{n_h} t_{ih}^{norm}}{K} \times 100 \qquad (7)$$

$$f_2(V) = \sqrt{\frac{\sum_{s=1}^{K}\left(\frac{\sum_{i\in s} t_{ih}^{norm}}{\max_{y=1}^{K}\left\{\sum_{i\in y} t_{ih}^{norm}\right\}} - \frac{\sum_{h=1}^{H}\sum_{i=1}^{n_h} t_{ih}^{norm}}{K}\right)^2}{K}} \qquad (8)$$

The objective functions which are given in Equations (7) and (8) should be combined into a single objective function. In order to combine these objective functions into a single objective function, we have used the minimum deviation method (MDM) which is applicable when the analyst has partial information of the objectives. Aim is to find the best compromise solution which minimizes the sum of individual objective's fractional deviations [24]. Let $f_1^0(LE)$ and $f_2^0(V)$ be the least desirable objective value of $f_1(LE)$ and f_2(V), respectively, which are obtained from initial solution. The objective function used in this study is formulated as follows:

$$\text{Minimize:} \quad f = \frac{f_1^{\max}(LE) - f_1(LE)}{f_1^{\max}(LE) - f_1^0(LE)} + \frac{f_2^{\min}(V) - f_2(V)}{f_2^{\min}(V) - f_2^0(V)} \qquad (9)$$

where, $f_1^{\max}(LE)$ and $f_2^{\min}(V)$ are the target values of LE and V, respectively. Since minimum level of V and maximum level of LE represent the perfect balance, the target values of V and LE are 0 and 1, respectively. Equation (9) guarantees maximizing the $f_1(LE)$ and minimizing the f_2(V) simultaneously. Geometrical interpretation of minimization of f is given in Figure 3.

Points A and B give the target values of objectives $f_1(LE)$ and $f_2(V)$, respectively. The denominator $f_1^{\max}(LE) - f_1^0(LE)$ and $f_2^{\min}(V) - f_2^0(V)$

can be represented by the length of |BD| and |AD|, respectively. If E is any point in the solution space which is candidate for the best solution, then the length of |EF| and |EG| represent the magnitude of $f_1^{\max}(LE) - f_1(LE)$ and $f_2^{\min}(V) - f_2(V)$, respectively. The objective function given in Equation (9) is the minimization of (|EF|/|BD|+|EG|/|AD|).

4. NUMERICAL EXAMPLE

In this section, an explanatory example given in Figure 1 is used to describe the proposed approach clearly. Cycle times of the lines are assumed as 8 (for line 1) and 10 (for line 2), respectively.

Initial solution

A random task assignment order list with size of (n_1+n_2) is established from the tasks of the each parallel line. Initial assignment order and the parallel assembly line balance with 9 stations are given in Figure 4a and 4b, respectively.

New solution generation

By using swap operator, all neighborhood solutions ($M(x_1)$) are generated and objective function values ($f(m(x_1))$) are calculated. For this purpose, a random task of any line is selected and whole examination is applied between assignment orders. In the Table 1, all candidate moves (m) and objective function values for the first iteration are given. Randomly selected task for first assembly line is task of 2 and the assignment order of this task is 9.

As it can be seen from Table 1, 15 neighborhood solutions are generated and the best objective function value among them is belong to move of 2[1]-1[2]. So, move of 2[1]-1[2] is accepted as the new solution and this move is added to tabu list. New assignment order, new line balance and the tabu list after the first iteration are shown in Figure 5(a), 5(b) and 5(c), respectively.

Final solution

Total number of tasks on parallel assembly lines has been used as termination criterion. The best solution with 8 stations achieved after 16 iterations is given in Figure 6.

5. COMPUTATIONAL RESULTS

Performance of the proposed approach is tested on 82 well known test problems (55 test problems with same cycle time and 27 test problems with different cycle time) in the literature. The number of parallel assembly line is considered as two for all test problems. The proposed approach is coded by using the Visual Basic 6.0 programming language, and the set of test problems are solved on a Pentium IV 3.0 GHz PC with 512 MB RAM. All parameters of the algorithm are obtained experimentally (*iteration number,* (n_1+n_2) and *tabu size,* $\sqrt{n_1 + n_2}$). Each test problem is solved five times using by these parameters and only best solutions are reported here.

Table 2 represents the computational results for same cycle time case. Aim of the proposed approach is to minimize the number of stations while the station workload is being smoothed, that is, two performance criteria are optimized simultaneously. Here, results of the proposed approach is compared with results of Gökçen et al. [1]'s study, to form an idea about the quality of obtained results. From the comparison results, it is seen that proposed approach have optimized the two performance criteria.

In Table 2, in 6 of the 55 test problems, the proposed approach has obtained better solution than Gökçen et al. [1]'s. Only one of them is worse than that solution and other solutions are the same with them. Clearly, the proposed approach results are better than the Gökçen et al. [1]'s results for this problem set.

Table 3 represents the computational results for different cycle time case. 27 test problems are combined from the literature problems. These problems are different from the Gökçen et al. [1]'s. Columns of the Table 3 represent problem types, number of tasks, cycle times and optimal number of stations for each line (independently), theoretical minimum number of workstations and the results obtained by the proposed approach, respectively. The theoretical minimum number of workstation (K_{min}) is a lower bound for the solution and calculated by Equation 10.

$$K_{\min} = \left[\sum_h \left(\sum_i {t_{hi}}/{C_h} \right) \right]^+ \qquad (10)$$

where C_h is the cycle time of line h, and $[X]^+$ denotes the smallest integer greater than or equal to X.

We know that the optimal number of stations is not less than the K_{min}. For the values of the number of stations, which is different from the K_{min} in Table 3, it is not possible to say anything about whether the values of the number of stations are optimal or not. This comparison may only give an idea about the performance of the procedure. As seen from Table 3, in 19 of 27 test problems, number of stations obtained from the proposed approach is equal to the K_{min} (*i.e.* optimal value). In 7 of the test problems, the proposed approach obtained 1 station more than the K_{min}. In 1 of the results, the proposed approach produced 2 stations more than the K_{min}. Ideal LE value is 100% and the V value is 0. But, it is so difficult to reach these values except for perfect balance. As seen from Table 3, obtained LE and V values are quite near to 100% and 0. Moreover, main objective of proposed approach is not only optimizing the single objective, but only optimize two objectives simultaneously. For this reason, as a result, it can be seen that the performance of the proposed approach is successful and it has sufficient performance.

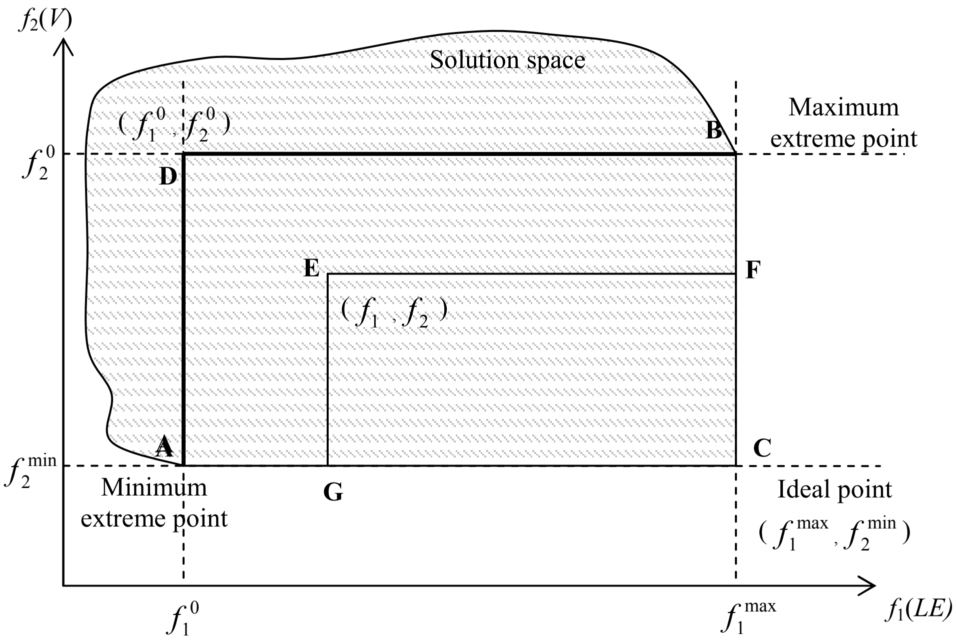
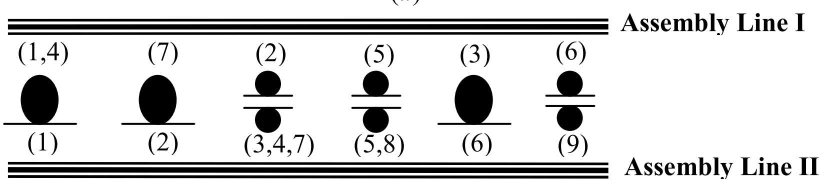


Figure 3. Geometric interpretation of the MDM.

Task [*Line*]	1[1]	2[1]	3[1]	4[1]	5[1]	6[1]	7[1]	1[2]	2[2]	3[2]	4[2]	5[2]	6[2]	7[2]	8[2]	9[2]
Assignment order	7	9	16	6	10	15	8	5	12	13	1	3	11	4	2	14

(*a*)



$LE_0 = 81.39$
$V_0 = 0.163$

(*b*)

k=1, TL = {Ø}, $x^*=x_0$, $f^*=f(x_0)$, $x_1=x_0$.

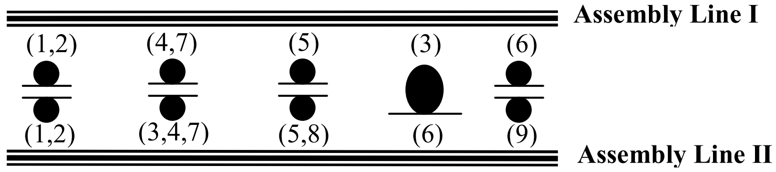
Figure 4. (*a*) Initial assignment order and (*b*) the initial line balance.

Table 1. Candidate moves and objective function values for the first iteration.

m (Swap)		*LE*	*V*	$f(m(x_1))$	*m* (Swap)		*LE*	*V*	$f(m(x_1))$
2[1]	1[1]	81.39	0.163	2	2[1]	3[2]	73.25	0.183	2.56
2[1]	3[1]	73.25	0.183	2.56	2[1]	4[2]	81.39	0.163	2
2[1]	4[1]	81.39	0.163	2	2[1]	5[2]	81.39	0.163	2
2[1]	5[1]	81.39	0.163	2	2[1]	6[2]	81.39	0.163	2
2[1]	6[1]	73.25	0.183	2.56	2[1]	7[2]	81.39	0.163	2
2[1]	7[1]	81.39	0.163	2	2[1]	8[2]	81.39	0.163	2
2[1]	1[2]	81.39	0.147	1.90*	2[1]	9[2]	73.25	0.183	2.56
2[1]	2[2]	81.39	0.163	2					

Task [*Line*]	1[1]	2[1]	3[1]	4[1]	5[1]	6[1]	7[1]	1[2]	2[2]	3[2]	4[2]	5[2]	6[2]	7[2]	8[2]	9[2]
Assignment Order	7	5	16	6	10	15	8	9	12	13	1	3	11	4	2	14

(*a*)



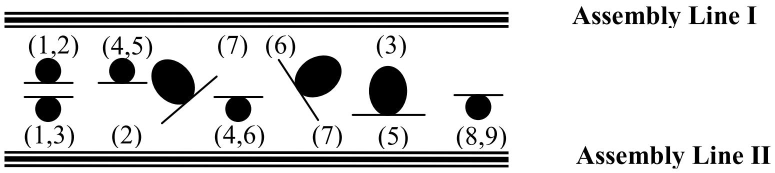
$LE_1 = 81.39$
$V_1 = 0.147$

(*b*)

	2[1]	3[1]	4[1]	5[1]	6[1]	7[1]	1[2]	2[2]	3[2]	4[2]	5[2]	6[2]	7[2]	8[2]	9[2]
1[1]															
2[1]							**4**								
3[1]															
4[1]															
5[1]															
6[1]															
7[1]															
1[2]															
2[2]															
3[2]															
4[2]															
5[2]															
6[2]															
7[2]															
8[2]															

(*c*)

Figure 5. (*a*) New assignment order, (*b*) new line balance and (*c*) tabu list after the first iteration.



$LE^* = 91.56$
$V^* = 0.081$

Figure 6. Best solution of the example problem.

Table 2. Computational results for same cycle time case.

Test problems	Number of task (Line 1-Line 2)	Cycle time	Gökçen et al. [1] Number of Station (*K*)	TS		
				K	*LE*%	*V*
Kilbridge	45-43	57	20	20	95,96	0,026
		79	14	14	98,91	0,010
		92	12	12	99,09	0,007
		110	10	10	99,45	0,006
		138	8	8	99,09	0,004
		184	6	6	99,09	0,004
Hahn	53-51	2004	14	14	93,47	0,073
		2338	12	12	93,47	0,085
		2806	10	10	93,46	0,068
		3507	8	8	93,47	0,060
		4676	6	6	93,47	0,051
Wee-Mag	75-71	28	123	123	85,27	0,081
		29	123	123	82,33	0,080
		31	121	121	78,29	0,083
		33	119	119	74,79	0,096
		34	119	119	72,59	0,093
		41	116	116	61,75	0,112
		42	107	107	65,35	0,161
		43	98	98	69,69	0,183
		49	62	63	95,14	0,036
		54	60	60	90,64	0,065
Arcus1	83-79	3786	40	40	95,00	0,043
		3985	38	38	95,01	0,056
		4206	36	36	95,01	0,036
		4454	34	34	95,00	0,038
		4732	32	32	95,01	0,057
		5853	26	26	94,54	0,034
		6842	22	22	95,58	0,028
		7571	20	20	95,01	0,041
		8412	18	18	95,01	0,043
		10816	14	14	95,01	0,088
Lutz3	89-85	75	45	44*	96,90	0,029
		79	43	42*	96,38	0,028
		83	40	40	96,32	0,056
		87	38	38	96,73	0,042
		92	36	36	96,55	0,036
Mukherje	94-90	176	48	48	96,10	0,041
		183	46	46	96,44	0,055
		192	44	44	96,10	0,072
		201	42	42	96,17	0,040
		211	40	40	96,19	0,099
		222	38	38	96,24	0,075
		234	36	36	96,37	0,046
		248	34	34	96,28	0,053
		263	32	32	96,47	0,041
		281	30	30	96,31	0,068
		301	28	28	96,33	0,040
		324	26	26	96,37	0,039
		351	24	24	96,37	0,059
Arcus2	111-107	5785	55	54*	96,11	0,059
		6016	53	51*	97,86	0,031
		6267	50	50	95,82	0,041
		6540	48	48	95,64	0,049
		6837	46	45*	97,59	0,025
		7162	44	43*	97,49	0,024

Table 3. Computational results for different cycle time case.

Problem (Line1-Line2)	No. of tasks (Line1-Line2)	Cycle time (Line1-Line2)	Optimal no. of station (Line1-Line2)	Theoretical min. no. of station (K_{min})	TS		
					K	*LE*%	*V*
Mitchell-Heskiaoff	21-28	14-138	8-8	15	16	93,25	0,032
Mitchell-Heskiaoff	21-28	15-205	8-5	12	13	92,27	0,048
Mitchell-Heskiaoff	21-28	21-324	5-4	9	9	90,67	0,042
Heskiaoff-Sawyer	28-30	216-41	5-8	13	13	97,25	0,015
Heskiaoff-Sawyer	28-30	324-54	4-6	10	10	91,60	0,049
Heskiaoff-Sawyer	28-30	342-75	3-5	8	8	91,42	0,044
Sawyer-Kilbridge	30-45	25-57	14-10	23	24	95,35	0,039
Sawyer-Kilbridge	30-45	27-92	13-6	18	19	94,73	0,031
Sawyer-Kilbridge	30-45	36-110	10-6	15	15	93,45	0,043
Sawyer-Kilbridge	30-45	41-138	8-4	12	12	99,18	0,007
Kilbridge-Tonge	45-70	79-364	7-10	17	17	97,82	0,012
Kilbridge-Tonge	45-70	92-410	6-9	15	15	97,07	0,016
Kilbridge-Tonge	45-70	138-468	4-8	12	12	95,83	0,030
Kilbridge-Tonge	45-70	184-527	3-7	10	10	96,60	0,019
Tonge-Arcus1	70-83	176-5048	21-16	35	37	94,43	0,030
Tonge-Arcus1	70-83	364-6842	10-12	21	22	94,12	0,044
Tonge-Arcus1	70-83	364-7571	10-11	20	20	98,21	0,015
Tonge-Arcus1	70-83	410-8412	9-10	18	18	97,56	0,013
Tonge-Arcus1	70-83	468-8898	8-9	17	17	94,16	0,051
Tonge-Arcus1	70-83	527-10816	7-8	14	14	94,57	0,014
Arcus1-Arcus2	83-111	5048-8847	16-18	32	33	96,96	0,026
Arcus1-Arcus2	83-111	5853-10027	14-16	28	29	96,32	0,039
Arcus1-Arcus2	83-111	6842-10743	12-15	26	26	96,40	0,030
Arcus1-Arcus2	83-111	7571-11378	11-14	24	24	96,74	0,025
Arcus1-Arcus2	83-111	8412-11378	10-14	23	23	96,60	0,024
Arcus1-Arcus2	83-111	8898-17067	9-9	18	18	96,22	0,029
Arcus1-Arcus2	83-111	10816-17067	8-9	16	16	98,82	0,008

6. CONCLUSION

In this paper, a tabu search based approach is proposed for PALB problem with the aim of maximizing *LE* (or minimizing number of stations) and minimizing *V*. Performance of the proposed approach is tested on 82 well known test problems in the literature. Obtained results from the test problems with same cycle time are compared with results of Gökçen et al. [1]'s. Moreover, main objective of proposed approach is not only optimizing the single objective, but also optimize two objectives simultaneously. According to the comparison results, the results of the proposed approach that optimize the two performance criteria, is better than Gökçen et al. [1]'s results. For problems with different cycle times, new problems are generated and efficiency of the approach is evaluated. The results of the computational study on various test problems indicate that the proposed approach is successful and it has sufficient performance. To the best knowledge of the authors, this study is first multi objective parallel line balancing study in the literature.

REFERENCES

[1] Gökçen, H., Ağpak, K., Benzer, R., "Balancing of parallel assembly lines", ***Int. J. Prod. Econ.***, 103: 600-609 (2006).

[2] Ghosh, S., Gagnon, J., "A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems", ***Int. J. Prod. Res.***, 27: 637-670 (1989).

[3] Salveson, M.E., "The assembly line balancing problem", ***J. Ind. Eng.***, 6: 18-25 (1955).

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

[5] Erel, E., Sarin, S.C., "A survey of the assembly line balancing procedures", ***Prod. Plan. Control.***, 9: 414-434 (1998).

[6] Scholl, A., Becker, C., "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing", ***Eur. J. Oper. Res.***, 168(3): 666-693 (2006).

[7] Miltenburg, J., Wijngaard, J., "The U-line line balancing problem", ***Manage. Sci.***, 40: 1378-1388 (1994).

[8] Urban, T.L., "Optimal balancing of U-shaped assembly lines", ***Manage. Sci.***, 44: 738-741 (1998).

[9] Becker, C., Scholl, A., "A survey on problems and methods in generalized assembly line balancing", ***Eur. J. Oper. Res.***, 168: 694-715 (2006).

[10] Süer, G.A., Dagli, C., "A knowledge-based system for selection of resource allocation rules and algorithms", In: A. Mital and S. Anand, eds. Handbook of expert system applications in manufacturing; Structures and rules, ***Chapman and hall***, 108-147 (1994).

[11] Süer, G.A., "Designing parallel assembly lines", ***Comput. Ind. Eng.***, 35: 467-470 (1998).

[12] Simaria, A.S., Vilarinho, P.M., "The simple assembly line balancing problem with parallel workstations-a simulated annealing approach", ***Int. J. Ind. Eng.***, 8: 230-240 (2001).

[13] Askin, R.G., Zhou, M., "A parallel station heuristic for the mixed-model production line balancing problem", ***Int. J. Prod. Res.***, 35: 3095-3105 (1997).

[14] McMullen, P.R., Frazier, G.V., "A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations", ***Int. J. Prod. Econ.***, 51: 177-190 (1997).

[15] Vilarinho, P.M., Simaria, A.S., "A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations", ***Int. J. Prod. Res.***, 40: 1405-1420 (2002).

[16] Benzer, R., Gökçen, H., Çetinyokuş, T., Çerçioğlu, H., "A network model for parallel line balancing problem", ***Math. Probl. Eng.***, Art. No: 10106 (2007).

[17] Gutjahr, A.L., Nemhauser, G.L., "An algorithm for the line balancing problem", ***Manage. Sci***., 11(2): 308-315 (1964).

[18] Glover, F., "Tabu search, part I", ***ORSA Journal of Computing***, 1: 190-206 (1989).

[19] Glover, F., "Tabu search, part II", ***ORSA Journal of Computing***, 2: 4-32 (1990).

[20] Scholl, A., Voß, S., "Simple assembly line balancing-Heuristic approaches", ***J. Heuristics***, 2: 217-244 (1996).

[21] Chiang, W.C., "The application of a tabu search metaheuristic to the assembly line balancing problem", ***Ann. Oper. Res.***, 209-227 (1998).

[22] Lapierre, S.D., Ruiz, A., Soriano, P., "Balancing assembly lines with tabu search", ***Eur. J. Oper. Res.***, 168 (3): 826-837 (2006).

[23] Hwang, R.K., Katayama, H., Gen, M., "U-shaped assembly line balancing problem with genetic algorithm", ***Int. J. Prod. Res.***, 46 (16): 4637-4649 (2008).

[24] Tabucanon, M.T., "Multiple criteria decision making in industry (Studies in production and engineering economics; 8)", ***Elsevier***, Amsterdam-Oxford-New York-Tokyo (1988).