A Multi-Objective Particle Swarm Optimization for Mixed-Model Assembly Line Balancing with Different Skilled Workers

Parviz Fattahi[a,*], Parvaneh Samouei[b]

[b] *Associate Professor,Department of Industrial Enginerring,* Faculty of Engineering, *Bu-Ali Sina University, Hamedan, Iran*

[b] *Assistant Professor, Department of Industrial Enginerring,* Faculty of Engineering, *Bu-Ali Sina University, Hamedan, Iran*

Received 2 September, 2014; Revised 12 January, 2015; Accepted 27 April, 2015

Abstract

This paper presents a multi-objective Particle Swarm Optimization (PSO) algorithm for worker assignment and mixed-model assembly line balancing problem when task times depend on the worker's skill level. The objectives of this model are minimization of the number of stations (equivalent to the maximization of the weighted line efficiency), minimization of the weighted smoothness index and minimization of the total human cost for a given cycle time.In addition, the performance of proposed algorithm is evaluated against a set of test problems with different sizes. Also, its efficiency is compared with a Simulated Annealing algorithm(SA)in terms of the quality of objective functions. Results show that the proposed algorithm performs well, and it can be used as an efficient algorithm.

*Keywords:*Mixed-model,assembly line balancing problem (ALBP), Multi-objective optimization, Different skilled workers, Particle swarm optimization, Simulated annealing.

1. Introduction

An assembly line is a production line on which the unfinished products move continuously through a sequence of stations and workstations are linked together by a material handler. Balancing assembly line that is defined as the allocation of tasks to an ordered sequence of stations subject to precedence constraints has a very important role in many industries.

Salveson (1955) conducted the first study on ALBP.Then,many Authors considered this problem with different assumptions, constraints, objective(s) and solving methods. There are multiple recent surveys and taxonomies on the ALBP such as Baybars (1986); Ghosh and Gagnon (1989); Amen (2000); Becker and Scholl (2006); Amen (2006); Scholl and Becker (2006); Boysen et al. (2007); Boysen et al. (2008); Hu et al. (2011); Battaïa and Dolgui (2013).

There are several criteria for classification of ALBP. For example, based on the number of product models assembled in a line, this problem can be divided into single, mixed and multi models. Additionally, according to the characteristics of the products and technical or operational requirements, assembly lines can be classified into one and two-sided lines. In the one-sided assembly line only left or right side of the line is used, whereas in two-sided line usually both sides are utilized.

There are two famous objective functions (Type-I and Type-II) for solving ALBP. Minimization of the number of workstations for a known cycle time is called Type-I. Minimization of the cycle time for the given number of workstations is Type-II.

Based on the number of objective function(s), ALBP is divided into one objective(i.e., Erel&Gokcen (1999) and Karabatı&Sayın (2003)) and multi-objective(i.e.,Kara et al., (2011) andYagmahan (2011).Recently, multi-objective optimization has been used more than one objective in assembly line balancing problems (Cakir et al. (2011)).

Another category for ALBP is concenend with solving methods. We can find different accurate , heuristic and metaheuristic algorithms for solving mixed model ALBP. Table1 presents a few studies that used these methods for solving this problem.

* Corresponding author Email address: Fattahi@basu.ac.ir

Table 1
Several Exact, heuristic and metaheuristics for solving mixed-model ALBP

Exact	Branch And Bound (B&B)	Bukchin&Rabinowitch (2006)
Heuristic		Jin & Wu (2002)
Metaheuristic	Simulated Annealing (SA)	Özcan&Toklu(2009)
	Genetic Algorithm (GA)	Simaria&Vilarinho (2004); Akpınar&Bayhan (2011), Rabbani et al. (2012)
	Ant Colony Optimization (ACO)	Simaria&Vilarinho (2009); Yagmahan (2011),
	Tabu Search (TS)	Bock (2008)
	Particle Swarm Optimization (PSO)	Chutima&Chimklai (2012)

Based on the nature of processing times, assembly line balancing problems are classified into deterministic and non-deterministic classes. Most of the authors in the field of ALBPassumed that the processing times are deterministic (Hamta et al. (2012)),and it is not dependent on the skilled workers. However, in many realistic manufacturing environments,the tasks are done manually. So, the processing times of tasks depend on the skill of workers. For example, tasks will be done faster if the worker is a high-skilled worker.Furthermore, distinguishing between worker types enables management to decide which tasks should be done by which level of skills. This brigs about a good saving for human cost because in this situation, the salary of each worker can be dependent on his skill.

There are a few studies(such as Naked and Nishiwaki (2008) and Corominas et al. (2008)) conducted on the skills of workers in assembly line balancing. Nakade and Nishiwaki (2008) developed an algorithm to solve U-shaped production line problem with different workers' skill. In that article, processing, operation and walking times were deterministic.

To the best of our knowledge, no study has considered the skills of workers for processing time in mixed model assembly line balancing problems.On the other hand, this study is the first one that considers mixed-model assembly line balancing and worker assignment, simultaneously. In this paper the task times are dependent on the skill of workers. It means if a high skill worker executes a task, he/she is expected to spend lower time than a low or medium skill worker's time. This leads to doing more tasks in a station and it means we can have lower station in the line. But when we use high skill worker, we should pay more costs and it cannot be suitable for managers and decision makers. So, as well as minimization of number of stations (which is equivalent to maximizing the weighted line efficiency), total human cost minimization is considered. Furthermore, another objective function is used to minimize the variances of workloads in stations. For NP-Hardness of this problem (Gutjahr and Nemhauser (1964)), we use a multi objective PSO algorithm to solve it.The efficiency of the presented algorithm is compared with the obtained results of a simulated annealing algorithm.

The remainder of this paper is structured as follows:The assumptions of the problem and the proposed PSO algorithm are presented in Section 2.Numerical example and experiments are given in Section 3 and 4. Finally, Section 5 is devoted to conclusions and recommendations for future research.

2. Problem Definition

In this section, the problem assumptions, the standard PSO algorithm and the procedure of the proposed algorithm for multi objective mixed model assembly line balancing problem with different skilled workers are introduced.

2.1. Assumptions

Assumptions of this problem are given as follows:

1. Combined precedence diagramis known.
2. Workers with different levels of skills are available, and the operation time depends on the worker's skill.
3. Common tasks exist among different models.
4. Productswith similar characteristics are assembled on the same station.
5. A task completion time can be different from one model to another.
6. Parallel tasks and stations are not allowed.
7. Work-in-process inventory is not allowed.
8. Tasks must be processed only once.
9. It is assumed that no machine breaks down when performing each task assigned to it.
10. Workstations are in a simple straight assembly line.
11. The cycle time is given.
12. Demand rate is deterministic.

2.2. Mathematical model

In this paper, a multi-objective mathematical model for mixed-model ALBP and workers assignment is proposed with using the following indices, parameters and variables:

Indices:

i, h, p, r	Task
j, g	Station
l, q	Skill
m	Product model

Parameters and variables:

I	Set of tasks in the combined precedence diagram
J	Set of stations
L	Number of skills (low, high, …)
$P(i)$	Set of immediate predecessors of task i
P_0	Set of tasks that have no immediate predecessors
C	Cycle time
M	Number of models

t^f_{iml}	Finish time of task *i* for model m with skill *l*
D_m	Market demand for model *m*
x_{ijl}	1, if task *i* is assigned to station *j* with skill *l*; 0, otherwise
t_{iml}	Operation time of task *i* for model *m* with skill *l*
y_{jl}	1, if a worker with skill *l* is assigned to station *j*; 0,otherwise
HC_l	Human cost of a worker with skill *l*
${}_mWL_j$	Load station j including unavoidable idle times for model *m*.
WL_{max}	Maximum of Load stations

$$Min\ \ NWS = \sum_{j\epsilon J}\sum_{l=1}^{L} y_{jl} \tag{1}$$

$$Min\ WSI = \sqrt{\sum_{m=1}^{M}\frac{D_m(\sum_{j\epsilon J}({}_mWL_j - WL_{max})^{\ 2}}{\sum_{m=1}^{M} D_m . NWS}} \tag{2}$$

$$Min\ \ HC = \sum_{j\epsilon J}\sum_{l=1}^{L} HC_l . y_{jl} \tag{3}$$

S.to:

$$\sum_{j\epsilon J}\sum_{l=1}^{L} x_{ijl} = 1 \qquad \forall i\epsilon I \tag{4}$$

$$\sum_{l=1}^{L} y_{jl} = 1 \quad \forall\ j\epsilon J \tag{5}$$

$$x_{ijl} \le y_{jl} \qquad \forall\ \ j\epsilon J\ ;\ \ l = 1, \dots L \tag{6}$$

$$\sum_{g\epsilon J} g . x_{hgz} - \sum_{j\epsilon J} j . x_{ijl}\ \le 0 \qquad \forall\ i\epsilon I - P_0\ \ , h\epsilon P(i),\ z, l = 1, \dots, L \tag{7}$$

$$t^f_{iml} \le C \qquad \forall i\epsilon I\ , m = 1, \dots, M, l = 1, \dots L \tag{8}$$

$$t^f_{iml} \ge t_{iml} . x_{ijl} \quad \forall i\epsilon I\ , m = 1, \dots, M, l = 1, \dots L,\ \ j\epsilon J \tag{9}$$

$${}_mWL_j - \sum_{i\epsilon I}\sum_{l=1}^{L} x_{ijl} . t^f_{iml} = 0 \qquad for\ m = 1, \dots, M, j\epsilon J \tag{10}$$

$$WL_{max} \ge {}_mWL_j \tag{11}$$

$$x_{ijl}\epsilon\{0,1\} \qquad \forall\ i\epsilon I,\ j\epsilon J,\ l = 1, \dots, L \tag{12}$$

$$y_{jl}\epsilon\{0,1\} \qquad \forall\ j\epsilon J,\ l = 1, \dots, L \tag{13}$$

Objective function (1) minimize the number of stations (this objective function is equivalent to maximization of weighted line efficiency that will be described in section 2.4). Equation 2 shows weighted smoothness index and the third objective function deals with minimizing the total human cost. Constraint 4 shows that each task should be assigned to exactly one station. Constraint (5) demonstrates each station has only one operator. Constraint (6) shows that tasks can be assigned to stations which equipped by a worker.Constraint (7) represents the precedence relations between the tasks. Constraint (8) and (9) determine the completion time of each task *i* for model *m* that is done with a worker with skill *l* is less than the cycle time and also it is equal or greater than its operation time. Constraint (10) and (11) show the workload of each station to calculate WSI. Finally, Constraint (12) and (13) express that x_{ijl}*and* y_{jl} are binary variables.

2.3. The Standard and the proposed PSO algorithm

PSO is a population-based stochastic optimization technique that was developed by Kennedy and Eberhart (1995). In this algorithm, an individual potential solution to the problem being optimized is called a particle. Each particle has its own position (current solution), velocity and fitness value. The particle flies through the multidimensional problem space with a velocity regularly adjusted using navigational guidance from its best flying history experience (local best) and the whole population's best flying experience (global best).

In PSO algorithm, positive constants such as personal and social learning factors (C_1 and C_2) and inertia weight(W)and two random real numbers for each iteration (r_1 and r_2)between 0 and 1 are used.

2.3.1. Initial solution generation

Each solution in proposed algorithm is a string of integer values. The initial solution is shown in a list that is named priority list (PL)which is initially created randomly, and its length is equal to the number of tasks. In this list, both position and value of position have important roles. For example, if there are five tasks in an assembly line, an initial and random priority list can be shown with PL= {2, 1, 4, 5, 3}. It means that task 2 has highest priority value, and task 3 has the lowest priority value.Since we want to use a PSO algorithm and it is basicaly for continuous space,applying a method that changes the continuous space to discrete one is necessary. For this purpose, we use the column and sorting method as follows. The details of this change is shown in Table 2.

Table 2
changing continuous space to discrete space

Start	Number of column (task)	1	2	3	4	5
Step 1	Creating random numbers in a continuous space	0.079	0.0034	0.339	0.134	0.275
Step 2	Sorting (ascending)	0.0034	0.079	0.134	0.275	0.339
Step 3 (finish)	Number of column(task) which is relted to this number	2	1	4	5	3

For creating a feasible solution, assignable tasks which satisfy the precedence constraints are initiallay assigned to the station based on their priority values. Then the set of assignable tasks is updated, and this process continues until there is no task for assignment.

In this scheme, a random worker is assigned to a work centre. If the time of this station after adding the new task is greater than the cycle time, another station will be opened, and a random worker will beassigned to this station, and the current task will be executed there.

2.3.2. Building a feasible solution

In addition to the notations in section 2.2, the following notations are used for bulding a feasible solution:
NS: Current station
SAT: Assignable tasks set
${}_mWL_{NS}$: The station load, including unavoidable idle times at the station NS for all m∈M
TL_{NS}: The set of tasks which are assigned to station NS
The Procedure of building a solution is as follows:
1. Set $NS = 1$, ${}_mWL_{NS}=0$ for all $m\in M$ and Skill 1=0, Skill 2=0, Skill 3=0.
2. Determine SAT ($SAT = \{i$ | (all $p\in P(i)$ have already been assigned or $P(i) = \{Ø\}$) and task i has not been assigned}). If $SAT= \{Ø\}$, then go to Step 6.
3. Sort the tasks in SAT in increasing order of priority value of tasks in PL.
4. Assign the first task h in SAT for which;
4.1. If ${}_mWL_{NS}= 0$ assign a random worker to the current station and add one unit to this skill.
4.1.2. If $t_{hml}+{}_mWL_{NS}\leq C$ and $t_{hml}+t^f_{rml}\leq C$ for all $m\in M, (t^f_{rml}=max\{t^f_{pml}|\ p\in P(h)$ have already been assigned to the current station}), then assign task h to station NS; $TL_{NS}=TL_{NS}+\{h\}$, and *set* $t^f_{hml}=max\{(t_{hml}+{}_mWL_{NS}),\ (t_{hml}+t^f_{rml})\}$ for all $m\in M$. Set ${}_mWL_{NS}=t^f_{hml}$ for all $m\in M$ and go to Step 2; otherwise go to Step 5.
5. If none of these tasks in SAT could be assigned to the current station, then open a new one. If $TL_{NS}\neq\{Ø\}$ then $NS=NS+1$, ${}_mWL_{NS} = 0$ for all $m\in M$, and go to Step 2.
6. Stop and calculate WLE, WSI and total human cost.
The flowchart of building a feasible solution is shown in Figure 1.

2.4. Objective function

The objectives of the proposed algorithm for mixed model assembly line balancing with different skilled workers for a given cycle time are as follows:

1. Minimizing the number of stations. It is equivalent to maximization of the Weighted Line Efficiency *(WLE)*, minimization of the line length or the number of operators. *WLE* is described in Eq. (14).

$$WLE=\left(\frac{\sum_{m\in M} q_m(\sum_{i\in I}\sum_{j=1}^{J} t_{iml}\cdot x_{ijl})}{C.NWS}\right).100 \qquad x_{ij}\in\{0,1\} \quad (14)$$

Where q_m is the overall proportion of the number of units for model m, and it can be computed by the following equation where D_m denotes the demand, over the planning horizon, for model m.

$$q_m=\frac{D_m}{\sum_{m\in M} D_m} \qquad (15)$$

2. Minimizing the weighted smoothness index *(WSI)*. By using the smoothness index, we can decrease the workload difference between stations. Eq. (16) shows it:

$$WSI=\sqrt{\frac{\sum_{m\in M} q_m\cdot(\sum_{j\in J}({}_mWL_j-WL_{max})^2}{NWS}} \qquad (16)$$

Where, WL_{max} is the maximum station time.

3. Minimizing the total human cost *(HC)*. It can be calculated as follows:

$$HC=\sum_{j\in J}\sum_{l=1}^{L} HC_l.\,y_{jl} \qquad (17)$$

According to the weighted sum method (Deb (2001)), the objective function of the proposed approach is given as follows:

$$Minimize\ E=W_1\left(\frac{WLE0}{WLE}\right)+W_2\left(\frac{WSI}{WSI0}\right)+W_3\left(\frac{HC}{HC_0}\right) \qquad (18)$$

Where, WLE_0, WSI_0 and HC_0 are the initial objective function and W_1, W_2 and W_3 are the weights of the objective functions. This equation shows *that WLE* will be maximized, and the *WSI* and *HC* will be minimized for a given cycle time. If $W_1=W_2=W_3=\frac{1}{3}$, then the following objective function will be achieved:

$$Minimize\ E=\frac{1}{3}\left(\frac{WLE0}{WLE}\right)+\frac{1}{3}\left(\frac{WSI}{WSI0}\right)+\frac{1}{3}\left(\frac{HC}{HC_0}\right) \qquad (19)$$

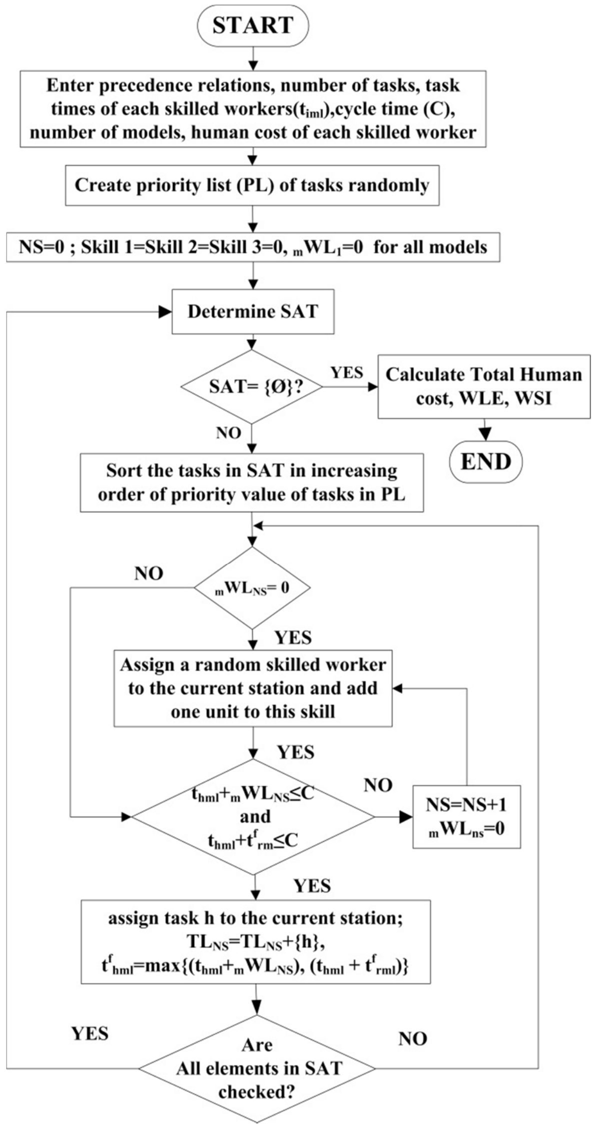

Fig. 1.Flowchart of building a feasible solution

3. Numerical Example

In this section, the proposed algorithm is illustrated by using a nine-task and two-model example problem. The task times for different skilled workers are generated randomly. The required data is given in Table 3. The given cycle time is 6. Furthermore, the human costs of a worker with skill 1, 2 and 3 are 90, 60 and 40 dollars per period, respectively.

Suppose a random solution (priority list) is constructed as: PL = {1, 2, 3, 4, 5, 6, 7, 8, 9}.

The procedure of creating an initial line balancing is shown inTable 4.

The initial tasks and skills assignments to the stations are presented in Table 5. It shows that there are four stations in this system. The objective function values of *WLE*, *WSI*, human cost and *E* of the initial line balance are 71.87%, 1.704, 230 and 1, respectively.

Table 3
Data of the example problem

Task	Immediate predecessor(s)	Model A (q_A=0.5)			Model B(q_B=0.5)		
		Skill 1	Skill 2	Skill 3	Skill 1	Skill 2	Skill 3
1	–	1.5	2	3	0	0	0
2	–	2	3	4	0.5	1.5	2.5
3	–	0	0	0	1	2.5	3.5
4	1	2	3	4	0	0	0
5	2	1	3	4	1.5	3	4
6	2,3	1	2	3	1	2	3
7	4,5	1.5	3	4	2	3	4
8	5	0	0	0	3	3.5	4
9	6	1	3	4	0.5	1	1.5

Table 4
Building an initial line balance

Step1	Step2 (SAT)	Step3 (PL)	Step4	Step5	Step6
NS=1, ${}_AWL_1$=0, ${}_BWL_1$=0	{1,2,3}	{1,2,3}	Select task 1, P(1) ={Ø}, ${}_AWL_1$=0, ${}_BWL_1$=0, random skill is skill 1, 1.5+0≤6, 0+0≤6, TL_1= TL_1+ {1},t^f_{1A}=1.5, t^f_{1B}=0, ${}_AWL_1$=1.5, ${}_BWL_1$=0.		
	{2,3,4}	{2,3,4}	Select task 2, P(2) ={Ø},2+1.5≤6, 0.5+0≤6, TL_1= TL_1+{2},t^f_{2A}=3.5, t^f_{2B}=0.5, ${}_AWL_1$=3.5, ${}_BWL_1$=0.5.		
	{3,4,5}	{3,4,5}	Select task 3, P(3) ={Ø}, 3.5+0≤3.5, 0.5+1≤1.5, TL_1= TL_1+{3}, t^f_{3A}=3.5, t^f_{3B}=1.5, ${}_AWL_1$=3.5, ${}_BWL_1$=1.5.		
	{4,5,6}	{4,5,6}	Select task 4, P(4) ={1},3.5+2≤6; 1.5+0≤6, TL_1= TL_1+{4}, t^f_{4A}=5.5, t^f_{4B}=1.5, ${}_AWL_1$=5.5, ${}_BWL_1$=1.5.		
	{5,6}	{5,6}	Select task 5, P(5) ={2},5.5+1>6, go to step 5.	Task 5 could not be selected.	
NS=2, ${}_AWL_1$=0, ${}_BWL_1$=0	{5,6}	{5,6}	Select task 5, P(5) ={2},${}_AWL_2$=0, ${}_BWL_2$=0, random skill is skill 3,4+0≤6, 4+0≤6, TL_2= TL_2+{5}, t^f_{5A}=4, t^f_{5B}=4, ${}_AWL_2$=4, ${}_BWL_2$=4.		
	{6,7,8}	{6,7,8}	Select task 6, P(6) ={2,3}, 4+3>6.	Task 6 could not be selected	
NS=3, ${}_AWL_1$=0, ${}_BWL_1$=0	{6,7,8}	{6,7,8}	Select task 6, P(6) ={2,3},${}_AWL_3$=0, ${}_BWL_3$=0, random skill is skill 2, 2+0≤6, 2+0≤6, TL_3= TL_3+{6}, t^f_{6A}=2, t^f_{6B}=2, ${}_AWL_3$=2, ${}_BWL_3$=2.		
	{7,8,9}	{7,8,9}	Select task 7, P(6) ={4,5}, 3+2≤6,3+2≤6, TL_4= TL_4+{7},t^f_{7A}=5, t^f_{7B}=5, ${}_AWL_3$=5, ${}_BWL_3$=5		
	{8,9}	{8,9}	Select task 8, P(8) ={5}, 5+3.5>6.	Task 8 could not be selected	
NS=4, ${}_AWL_1$=0, ${}_BWL_1$=0	{8,9}	{8,9}	Select task 8, P(8) ={5},${}_AWL_4$=0, ${}_BWL_4$=0, random skill is skill 3, 0+0≤6, 4+0≤6, TL_4= TL_4+{8}, t^f_{8A}=0, t^f_{8B}=4, ${}_AWL_4$=0, ${}_BWL_4$=4.		
	{9}	{9}	Select task 9, P(9) ={6},0+4≤6; 4+1.5≤6 TL_4= TL_4+{9}, t^f_{9A}=4, t^f_{9B}=5.5, ${}_AWL_4$=4, ${}_BWL_4$=5.5.		
	SAT={Ø}				Stop

Table 5
The assignment of tasks and skills to the stations

	Station 1	Station 2	Station 3	Station 4
Task	1,2,3,4	5	6,7	8,9
Skill	1	3	2	3

4. Experiments and Analysis

4.1. Parameter settings

In metaheuristic algorithms, choosing the best combination of the parameters can intensify the search process and prevent premature convergence. So, setting the parameters can influence upon the performance of these algorithms.

In the proposed PSO algorithm, the Taguchi (1986) method is used for the best parameter selections. Four levels are selected for each parameter (swarm size, C1, C2 and W). They are shown in Table 6.

Each test is run five times, and the average of the objective function is obtained to calculate the (S/N) ratio. In the Taguchi method, the S/N ratio is as follows (Taguchi (1986)):

$$SN \;\; = -10\log(\frac{1}{n}\sum_{i=1}^{n}(objective\ function)^2) \qquad (20)$$

The larger S/N ratio is equivalent to the least objective function. So, each factor's level that shows the maximum S/N ration is the best ones.

Table 7 reports the best level of each factor which is obtained by Taguchi method.

4.2. Numerical experiments

In order to verifyi the efficiency of the proposed algorithma set of test problems ,P9, P12, P14, P20, P25, P30, P39, P47 and P65 are solved.

We ran this algorithm five times for a fixed cycle time by a PC 3.2 GHz CPU and 2 GB of RAM. The best, the average and the worst results of *NS, WLE, WSI*, human cost, the objective function*(E)* and the number of each skill level of workers are presented in Table 8.

Figure 3 shows the relations between human cost and different problem sizes and cycle times. It shows that for a specified problem, by increasing the cycle time; human cast will be decreased.For example, when the cycle time is 40 for P47, total human cost is $690. But by increasing the cycle time to 50 and 60, the total human cost will be decreased to 500 and 420, respectively.

Figure 4 shows the relations between the best number of stations and different problem size and cycle times. It shows that for the same problem, in the most of cases, number of stations will be decreased, if cycle time is increased (For example, see P14, P25, P30, P39, P47 and P65). But in the others, for the structure of the problem, by increasing the cycle time, number of stations cannot decrease (See P9, P12 and P20). In these cases, minimum number of stations for different cycle times are the same.

The proposed PSO algorithm is compared with SA algorithm. The obtained results of both algorithms in five iterations for the best, the worst, the average and standard deviationof objective functions (E) are reported in Table 9.

Table 6
Factors and their levels

Factors	Swarm size				C_1				C_2				W			
level	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
value	5	10	15	20	0.7	1	1.5	2	1	1.5	2	2.5	0.8	1	1.2	1.5

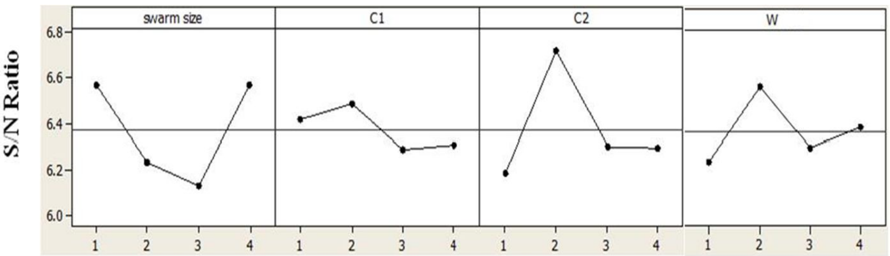

Fig. 2. The mean S/N ratio plot for the selected levels of each factor

Table 7
of the PSO algorithm and their selected levels

Factor	Swarm size	Cognitive coefficient (C_1)	Social coefficient (C_2)	Inertia weight (W)
Level	1	2	2	2
value	5	1	1.5	1

Table 8
The results of the proposed algorithm for different problem tests and cycle times

	C	NS	WLE			WSI			Human Cost			E				S1		S2		S3		T
		(W,M,B)	B	M	W	W	M	B	W	M	B	W	M	B	S	M	B	M	B	M	B	M
P9	4	(6, 5.2,5)	90.0	85.4	81.2	0.9	0.7	0.5	340	320	290	0.46	0.39	0.34	0.04	1.6	1	1.6	0	2	1	1.9
	6	(2,2,2)	81.2	81.2	81.2	0.9	0.9	0.9	180	180	180	0.43	0.40	0.38	0.02	0	2	0	0	0	0	2.0
	8	(2,2,2)	82.8	77.9	58.1	3.8	1.2	0.6	150	150	150	0.51	0.48	0.46	0.02	1	1	1	1	0	0	2.0
P 12	5	(3,2,2)	88	86.8	81.8	1.4	1.0	0.9	240	228	180	0.49	0.43	0.40	0.03	2	2	0.8	0	0	0	2.7
	7	(7,4,2)	81.8	68	46.5	4.3	2.4	1.4	380	258	180	0.51	0.49	0.46	0.02	1.4	0	1.4	0	1.2	0	2.8
	8	(2,2,2)	90.9	90.6	89.4	1.1	1.1	1.1	150	150	150	0.50	0.46	0.41	0.03	1	1	1	1	0	0	2.7
P 14	10	(5,5,5)	88.2	87.2	85.6	1.5	1.4	1.2	370	370	370	0.45	0.41	0.38	0.02	3	3	1	1	1	1	3.3
	12	(5,4.8,4)	86.9	85.8	83.4	2.2	2.2	2.1	320	300	290	0.49	0.47	0.44	0.02	1.4	1	1.4	0	1.8	1	3.3
	15	(6,4.2,3)	90.4	76.1	54.1	6.8	4.1	2.3	380	294	240	0.53	0.51	0.47	0.02	1.8	1	1.8	1	0.6	0	3.4
P 20	18	(6,6,6)	86.7	86.2	85.9	3.4	3.3	3.2	370	370	370	0.47	0.44	0.42	0.02	1	1	4	4	1	1	5.2
	21	(6,5.4,5)	88.4	86.4	82.2	3.9	3.7	3.6	320	310	300	0.54	0.49	0.46	0.03	2.8	0	2.8	2	2	1	5.3
	25	(5,5,5)	84.1	84.1	84.1	3.9	3.9	3.9	260	260	260	0.61	0.56	0.51	0.03	0	0	3	3	2	2	5.1
P 25	30	(10,9.4,9)	76.7	73.9	70.6	9	8.3	7.7	620	582	550	0.61	0.56	0.54	0.02	2.2	1	4.8	3	2.4	1	7.2
	35	(8,8,8)	78.1	77.2	76.6	10.0	9.7	9.2	480	468	450	0.64	0.60	0.57	0.03	4.4	1	4.4	3	2.4	2	7.3
	45	(7, 6.6,6)	79.9	76.6	73.5	12.0	11.5	10.9	340	320	300	0.63	0.62	0.62	0.01	0.8	0	0.8	0	5	4	7.3
P 30	20	(8,8,8)	91.1	90.2	89.0	2.8	2.5	2.2	470	454	440	0.55	0.53	0.52	0.01	0.2	0	6.2	5	1.6	1	9.1
	28	(6,6,6)	91.4	90.8	90.4	3.0	2.7	2.4	310	302	300	0.57	0.56	0.53	0.01	2.6	0	2.6	1	3.2	3	9.1
	35	(5,4.4,4)	93.9	92.5	90.5	2.6	1.6	0.9	270	250	220	0.55	0.52	0.48	0.02	0.6	0	2.2	1	1.6	0	9.1
P 39	20	(10,9.6,9)	88.7	86.0	83.0	3.6	3.2	2.5	560	546	540	0.55	0.54	0.52	0.01	1	0	5.6	4	3	2	14.1
	28	(7,6.4,6)	95.3	93.1	89.3	3.7	2.9	2.5	360	360	360	0.61	0.55	0.52	0.03	5.2	0	5.2	4	1.2	0	14.2
	33	(6,5.8,5)	93.8	91.9	90.2	3.4	3.0	2.8	310	298	280	0.56	0.53	0.50	0.02	0.2	0	2.8	2	2.8	1	14.2
P 47	40	(15,13.2,12)	84	80.6	76.0	10.4	9.4	8.5	760	716	690	0.66	0.62	0.59	0.02	0.8	0	7.4	5	5	2	13.7
	50	(11,10.4,10)	87.3	84.3	80.5	11.2	10.4	10	570	540	500	0.65	0.63	0.61	0.01	4.2	0	4.2	1	5.4	3	13.9
	60	(9,8.2,8)	87.5	86.5	85.1	12.7	12.3	12	450	440	420	0.69	0.65	0.61	0.03	0.4	0	4.6	3	3.2	2	13.9
P 65	360	(8,8,8)	88.1	87.4	86.8	64.1	58.7	53.7	450	442	430	0.59	0.55	0.53	0.02	1.4	1	2.6	1	4	3	26.9
	400	(7,7,7)	91.4	90.2	89.5	57.6	53.1	50.5	390	376	360	0.60	0.57	0.55	0.01	3.8	0	3.8	2	2.8	2	27.2
	430	(7,6.8,6)	92.7	88.9	87.8	67.0	60.9	48.8	390	318	300	0.57	0.55	0.52	0.02	0.2	0	1.8	1	4.8	0	26.9

W*:the worst result, M**: the average result, B***: The best result, T#: Elapsed time, S1: Skill 1, S2:Skill 2, S3: Skill 3.

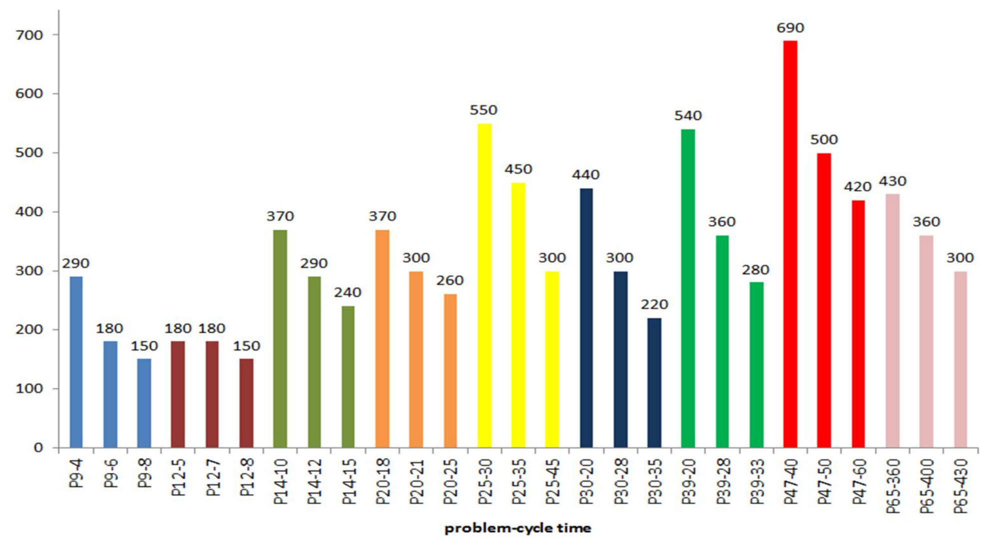

Fig. 3. The relations between different test problems and the best humancost

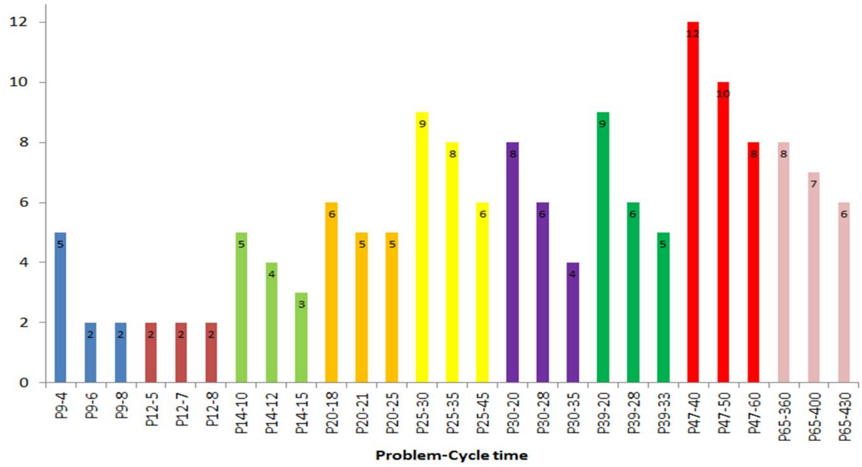

Fig. 4. The relations between different test problems and the best number of stations

Table 9 indicates that the objective functions results obtained by the PSO algorithm are better than the results of SA algorithm. For example, for similar objective function (E) which we would like to minimize, the results of the new algorithm for the worst, the average and also the best cases are lower than the similar objective function in SA algorithm. Also, the standard deviation of the obtained results by the proposed algorithm is better than the standard deviation of SA.Figure 5 and Figure 6 show these results in details for the best and the average results.They show that for each case the results of propsed algorithm are better than the SA.

Table 9
Comparision between PSO and SA algorithm for the objective functions

		E(PSO)				E(SA)			
	C	W	M	B	S	W	M	B	S
P9	4	0.46	0.39	0.34	0.04	0.96	0.77	0.63	0.13
	6	0.43	0.40	0.38	0.02	1.00	0.66	0.44	0.19
	8	0.51	0.48	0.46	0.02	0.64	0.59	0.55	0.04
P12	5	0.49	0.43	0.40	0.03	0.86	0.62	0.47	0.13
	7	0.51	0.49	0.46	0.02	0.87	0.69	0.55	0.12
	8	0.50	0.46	0.41	0.03	0.75	0.57	0.49	0.09
P14	10	0.45	0.41	0.38	0.02	0.75	0.61	0.51	0.09
	12	0.49	0.47	0.44	0.02	0.92	0.67	0.57	0.13
	15	0.53	0.51	0.47	0.02	0.67	0.62	0.59	0.04
P20	18	0.47	0.44	0.42	0.02	0.80	0.69	0.64	0.06
	21	0.54	0.49	0.46	0.03	0.93	0.75	0.65	0.10
	25	0.61	0.56	0.51	0.03	0.81	0.72	0.61	0.07
P25	30	0.61	0.56	0.54	0.02	0.95	0.85	0.71	0.09
	35	0.64	0.60	0.57	0.03	0.91	0.84	0.80	0.04
	45	0.63	0.62	0.62	0.01	0.91	0.87	0.81	0.04
P30	20	0.55	0.53	0.52	0.01	0.87	0.75	0.58	0.11
	28	0.57	0.56	0.53	0.01	0.77	0.73	0.65	0.05
	35	0.55	0.52	0.48	0.02	0.80	0.68	0.57	0.08
P39	20	0.55	0.54	0.52	0.01	0.76	0.70	0.62	0.05
	28	0.61	0.55	0.52	0.03	0.85	0.76	0.68	0.06
	33	0.56	0.53	0.50	0.02	0.87	0.76	0.63	0.09
P47	40	0.66	0.62	0.59	0.02	0.88	0.79	0.73	0.05
	50	0.65	0.63	0.61	0.01	0.94	0.8	0.70	0.08
	60	0.69	0.65	0.61	0.03	0.85	0.75	0.68	0.05
P65	360	0.59	0.55	0.53	0.02	0.79	0.72	0.58	0.07
	400	0.60	0.57	0.55	0.01	0.83	0.74	0.65	0.06
	430	0.57	0.55	0.52	0.02	0.91	0.79	0.70	0.08

W:The worst result, M: The average result, B: The best result, S:Standard deviation

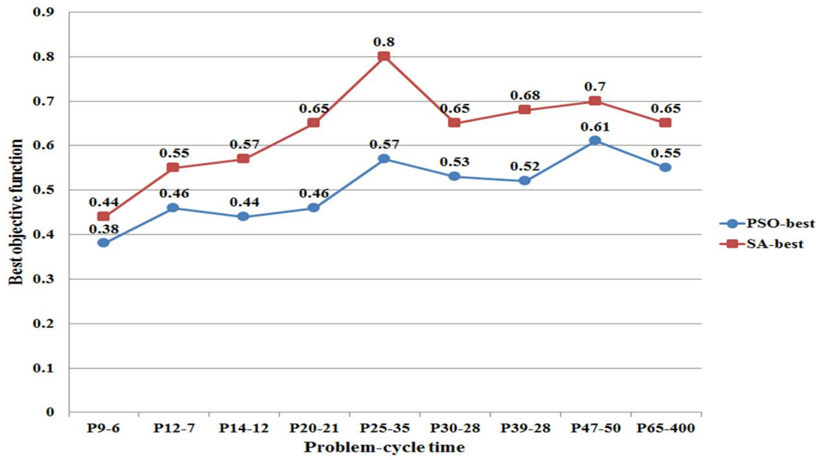

Fig. 5. A comparison between the proposed algorithm and SA obtainedresults for the best objective functions

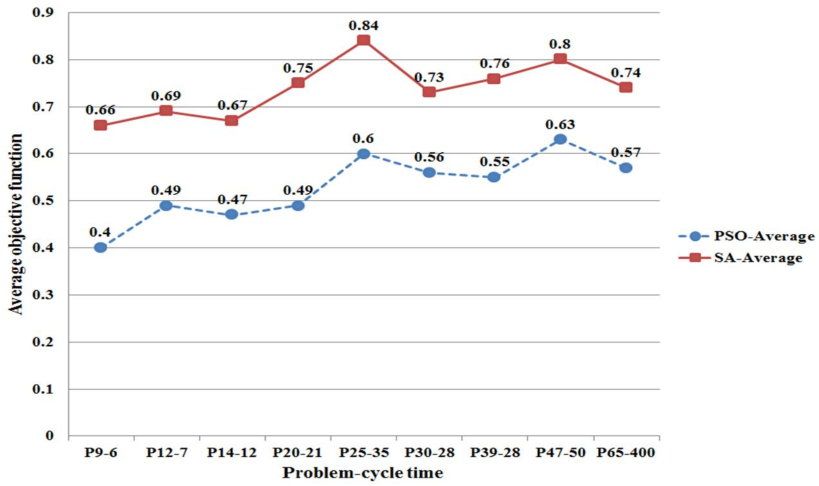

Fig. 6. A comparison between the proposed algorithm and SA obtained results for the average objective functions

5. Conclusion

In this paper, a multi-objective particle swarm algorithm for mixed-model ALBP withdifferent skilled workers for minimizing the number of stations (equivalent to maximizing the weighted line efficiency), minimizing the weighted smoothness index and total human cost was considered. In this problem, the cycle time is given. An illustrative example problem is solved by using the proposed algorithm.Furthermore, several numerical experiments are conducted to demonstrate its efficiency. In addition, the results of the proposed algorithm are compared with those of the simulated annealing algorithm. The results show that the proposed algorithm yields good solutions within a reasonable computational time for different test problems. For example, for similar objective function (E) which we would like to minimize, the results of the new algorithm for the worst, average and also the best cases are lower than the similar objective function in SA algorithm. Also, the standard deviation of the obtained results by the proposed algorithm is better than the standard deviation of SA.

Further studies can focus on developing this algorithm for a given number of stations. Also, adding other costraints to this problem to make a better decision for the real world problems can be an interesting idea.

References

Akpınar, S.and Bayhan, G. M. (2011). A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel work stations and zoning constraints. Engineering Applications of Artificial Intelligence, 24, 449–457.

Amen, M. (2000). Heuristic methods for cost-oriented assembly line balancing: A survey. Int. J. Production Economics, 68, 1-14.

Amen, M. (2006). Cost-oriented assembly line balancing: Model formulations, solution difficulty, upper and lower bounds. European Journal of Operational Research, 168, 747–770.

Battaïa, O. Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. Int. J. Production Economics, 142(2), 259–277.

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

Becker, C.and Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research, 168 (3), 694–715.

Bock, S. (2008). Supporting offshoring and nearshoring decisions for mass customization manufacturing processes. European Journal of Operational Research, 184, 490–508.

Boysen, N. Fliedner, M.and Scholl, A. (2007). A classification of assembly line balancing problems. European Journal of Operational Research, 183, 674–693.

Boysen, N. Fliedner, M.and Scholl, A. (2008). Assembly line balancing: Which model to use when?. International Journal of Production Economics, 111, 509–528.

Bukchin, Y.and Rabinowitch, I. (2006). A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. European Journal of Operational Research ,174, 492–508.

Cakir, B. Altiparmak, F.and Dengiz, B. (2011). Multi-objective optimization of a stochastic assembly line balancing: a hybrid simulated annealing algorithm. Computers and Industrial Engineering, 60, 376–384.

Chutima, P.and Chimklai, P. (2012). Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. Computers and Industrial Engineering, 62, 39–55.

Corominas, A. Pastor, R.and Plans, J. (2008). Balancing assembly line with skilledand unskilled workers. OMEGA, The International Journal of Management Science, 36(6), 1126-1132.

Deb, K. (2001). Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Inc, New York, NY, USA.

Erel, E.and Gokcen, H. (1999). Shortest-route formulation of mixed-model assembly line balancing problem. European Journal of Operational Research 116, 194-204.

Ghosh, S.and 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.

Hamta, N. FatemiGhomi, S.M.T. Jolai, F. Akbarpour Shirazi, M. (2013). A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect". International Journal of Production Economics 141(1), 99-111.

Hu, S.J. Ko, J. Weyand, L. El Maraghy, H.A. Lien, T.K. Koren, Y. Bley, H. Chryssolouris, G. Nasr, N.and Shpitalni, M. (2011). Assembly system design and operations for product variety. CIRP Annals-Manufacturing Technology, 60, 715–733.

Jin, M.and Wu, S. D.(2002). A new heuristic method for mixed model assembly line balancing problem. Computers and Industrial Engineering, 44, 159–169.

Kara, Y. Özgüven, C. Seçme, N.Y.and Chang, C. T. (2011). Multi-objective approaches to balance mixed-model assembly lines for model mixes having precedence conflicts and duplicable common tasks. International Journal of Advanced Manufacturing Technology, 52, 725–737.

Karabatı, S.and Sayın, S. (2003). Assembly line balancing in a mixed-model sequencing environment with synchronous transfers. European Journal of Operational Research 149, 417–429.

Kennedy. J.and Eberhart, R. (1995). Particle swarm optimization". In Proceedings of IEEE International Conference on Neural Networks, 1942-1948.

Nakade, K.and Nishiwaki, R. (2008). optimal allocation for heterogeneous workers in a U-shaped production. Computers and Industrial engineering, 54(3), 432-440.

Özcan, U.and Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. Computers and Industrial Engineering, 57, 217–227.

Rabbani, M. Kazemi, S. M.and Manavizadeh, N. (2012). Mixed model U-line balancing type-1 problem: A new approach. Journal of Manufacturing Systems, 31, 131–138.

Salveson, M.E. (1955). The assembly line balancing problem. Journal of Industrial Engineering, 6 (3), 18–25.

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

Simaria, A. S.and Vilarinho, P. M. (2004). A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. Computers and Industrial Engineering, 47, 391–407.

Simaria, A. S.and Vilarinho, P. M. (2009). 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines. Computers and Industrial Engineering, 56, 489–506.

Yagmahan, B. (2011). Mixed-model assembly line balancing using a multi-objective ant colony optimization approach. Expert Systems with Applications,38, 12453–12461.