Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 63 (2017) 505 – 510

www.elsevier.com/locate/procedia

The 50th CIRP Conference on Manufacturing Systems

Comparison of Sequential and Integrated Optimisation Approaches for ASP and ALB

M.F.F. Ab Rashid[a,b,*], A. Tiwari[b], W. Hutabarat[b]

[a]*Faculty of Mechanical Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia*
[b]*Manufacturing Department, Cranfield University, Bedford, MK43 0AL United Kingdom*

* Corresponding author. Tel.: +609-424 6321; fax: +609-424 6222. *E-mail address:* ffaisae@ump.edu.my

Abstract

Combining Assembly Sequence Planning (ASP) and Assembly Line Balancing (ALB) is now of increasing interest. The customary approach is the sequential approach, where ASP is optimised before ALB. Recently, interest in the integrated approach has begun to pick up. In an integrated approach, both ASP and ALB are optimised at the same time. Various claims have been made regarding the benefits of integrated optimisation compared with sequential optimisation, such as access to a larger search space that leads to better solution quality, reduced error rate in planning and expedited product time-to-market. These benefits are often cited but no existing work has substantiated the claimed benefits by publishing a quantitative comparison between sequential and integrated approaches. This paper therefore compares the sequential and integrated optimisation approaches for ASP and ALB using 51 test problems. This is done so that the behaviour of each approach in optimising ASP and ALB problems at different difficulty levels can be properly understood. An algorithm named Multi-Objective Discrete Particle Swarm Optimisation (MODPSO) is applied in both approaches. For ASP, the optimisation results indicate that the integrated approach is suitable to be used in small and medium-sized problems, according to the number of non-dominated solution and error ratio indicators. Meanwhile, the sequential approach converges more quickly in large-sized problems. For pure ALB, the integrated approach is preferable in all cases. When both ASP and ALB are considered, the integrated approach is superior to the sequential approach.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems

Keywords: Assembly sequence planning; assembly line balancing; integrated optimisation; sequential optimisation

1. Introduction

Assembly optimisation involves bringing and joining parts and/or sub-assemblies to make the assembly process as efficient as possible [1]. Assembly Sequence Planning (ASP) and Assembly Line Balancing (ALB) are classified to be among major topics in assembly optimisation because both are directly related to assembly efficiency [2]. Traditionally, the ASP and ALB activities are optimised independently since both activities occur in different stages [3]. This approach is known as sequential optimisation, where the ASP is optimised before ALB. Recently, researchers have discovered the benefits of solving and optimising ASP and ALB problems together [4], [5], leading to an increased research focus on testing new or improved algorithms that operate on these combined problems [1], [6]–[8].

Various claims have been made regarding the benefits of integrated optimisation compared with sequential optimisation for ASP and ALB. In one previous work, it was claimed that the integrated ASP and ALB will enhance the quality of the solutions [4]. This is due to avoidance of reduction of the size of search space for ALB. In sequential optimisation, the search space for the second activity (i.e. ALB) will be tremendously reduced because it is formed from the output of the first activity (i.e. ASP). Besides that, integrated optimisation will reduce the error rate in manufacturing planning [5], [9]. Other than that, the integrated ASP and ALB help designers to explore the search space in one shot.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
doi:10.1016/j.procir.2017.03.083

This is important to reduce optimisation time for both activities [5]. Besides that, the integrated optimisation will reduce the lead time and production cost in manufacturing [8], [10].

Although many benefits of integrated ASP and ALB optimisation were discussed by researchers, no existing work has substantiated the claimed benefits by publishing a quantitative comparison between sequential and integrated approaches. This work therefore will compare the quality of solutions of ASP and ALB optimisations that are achieved by sequential and integrated approaches. This work focuses on numerically substantiating the claim of superior solution quality. The rest of the stated benefits, such as reduced error rate and production cost, cannot be compared numerically as yet because they require actual implementation on actual assembly lines.

Substantiating the claim of superior solution quality is important because of its impact on existing practice in both ASP and ALB. It is proof that most manufacturing assembly line, even those that have been optimised using sequential ASP and ALB, are not operating in the best possible way. More importantly, it provides evidence that the integrated ASP and ALB approach is a practical way to increase the assembly line productivity even further than what has been achieved with the standalone ASP and ALB.

2. ASP and ALB Modelling

According to existing ASP and ALB works, there were a few modelling approaches implemented. The first approach is to model the problem based on the assembly components [11], [12]. Besides that, the researchers also model the problem based on the assembly task [13], [14]. Meanwhile, some researchers also model the problem based on the assembly connectors [8], [15].

In this work, we will implement task-based modelling for a simple version of ASP and ALB. The assembly problem based on assembly task is represented using a precedence diagram as shown in Figure 1. In this figure, the numerical nodes represent the assembly task, while the arcs represent precedence constraints among the assembly tasks. As an example, the outgoing arc from node 1 to nodes 2, 3 and 4 means that the assembly task 1 needs to be completed before tasks 2, 3 and 4 can be started. The assembly data for this example is presented in Table 1.

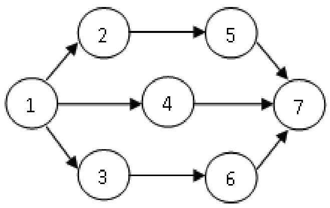

Fig. 1. Example of Precedence Diagram

In Table 1, for each task there are three types of assembly data which are required to calculate the predefined objective functions. To evaluate the ASP objectives (i.e. number of direction change (n_{dc}) and number of tool change (n_{tc}), the assembly direction and tool information for each task are needed. Meanwhile, to evaluate ALB objectives (i.e. cycle time (ct), number of workstation (nws) and workload variation (v)), only the assembly time information is required.

The main constraint in this work is precedence restriction which represents the compulsory sequence that must be followed in assembling a particular product. In handling this constraint, the topological sort approach is applied. Topological sort is an approach to establish feasible sequence by selecting only one available assembly task in each iteration. The topological sort procedure is repeated until all tasks are selected [16].

Table 1. Data table for Fig. 1

Task	Direction	Tool	Time
1	+x	T1	4
2	-x	T2	12
3	+x	T1	7
4	-x	T3	4
5	+x	T1	12
6	+x	T1	5
7	-x	T2	12

2.1. Objective Functions

Various objective functions have been designed and used to optimise ASP and ALB problems. A prior literature survey has collated objective functions that have been used by researchers in both problems [17]. This survey also found that the most frequently used ASP optimisation objectives are to minimise assembly direction change and to minimise the number of tool change. In ALB works, the dominant optimisation objectives are to minimise cycle time, minimise number of workstation and minimise workload variance [17].

Number of assembly direction change (n_{dc}) is counted when the next assembly task requires a different assembly direction compared with the present assembly task. In equations (1) and (2), s refers to the position of a task in a feasible assembly sequence.

$$n_{dc} = \sum_{s=1}^{n-1} d_s \tag{1}$$

$$d_s = \begin{cases} 1 & \text{if direction } s \neq \text{direction } s+1 \\ 0 & \text{if direction } s = \text{direction } s+1 \end{cases}$$

Number of assembly tool change (n_{tc}) is also counted when the next assembly task requires a different assembly tool compared with the present assembly task.

$$n_{tc} = \sum_{s=1}^{n-1} t_s \tag{2}$$

$$t_s = \begin{cases} 1 & \text{if tool } s \neq \text{tool } s+1 \\ 0 & \text{if tool } s = \text{tool } s+1 \end{cases}$$

Cycle time (ct) refers to the duration in between completion of one product unit with the following consecutive unit. The cycle time is important to be complied in order to meet the customers' demand [18]. In this work, the ct is obtained from the maximum processing time (pt) in the workstation. The pt is defined as a summation of assembly time in a particular workstation, which cannot exceed the maximum cycle time (ct_{max}).

Number of workstation (nws) refers to the group of assembly station which consists of assembly task/s. The nws is highly dependent on the predetermined maximum cycle time (ct_{max}) for a particular assembly process. Theoretically, the larger the value of ct_{max}, the smaller the nws will be.

Workload variation (v) measures the mean of working load across all workstations. It calculates the differences between the cycle time and the processing time in all workstations. A smaller v indicated better workload balance in the workstation.

$$v = \frac{\sum_{i=1}^{nws}(ct - pt_i)}{nws} \tag{3}$$

3. Experimental Design

The purpose of the experiment is to compare the solution quality towards Pareto optimum between sequential and integrated optimisation. In a previous work, a tuneable test problem generator for ASP and ALB has been developed to supply sufficient test problems with a different range of difficulties [1]. The results indicate that the ASP and ALB problem difficulties can be increased using a larger number of tasks (n), lower Order Strength (OS), lower Time Variability Ratio (TV) and higher Frequency Ratio (FR).

In this case, n refers to the number of assembly tasks. Meanwhile, the OS measures the relative number of precedence in the graph. The smaller OS means that the problem is more difficult to solve because of greater options. On the other hand, TV indicates the range of task time of all tasks dispersed between the assembly lines. A smaller TV value indicates that task times are distributed in a smaller range, which leads to an increased level of problem complexity. Finally, the FR shows how many times a similar direction or tool appears in the problem. Data with a higher FR value is harder to achieve a minimum number of changes because of the high variability of the data.

Table 2. Level of tuneable input setting

Level	*n*	*OS*	*TV*	*FR*
1	15	0.6	8	0.2
2	20	0.5	6	0.3
3	40	0.4	4	0.4
4	60	0.3	3	0.6
5	80	0.2	2	0.8

Table 3. Experimental design for ASP and ALB problem

Reference Datum	Test Problem Setting				
	Problem No.	*n*	*OS*	*TV*	*FR*
Datum 1	**1**	**15**	**0.6**	**8**	**0.2**
	2	20	0.6	8	0.2
	3	40	0.6	8	0.2
	4	60	0.6	8	0.2
	5	80	0.6	8	0.2
	6	15	0.5	8	0.2
	7	15	0.4	8	0.2
	8	15	0.3	8	0.2
	9	15	0.2	8	0.2
	10	15	0.6	6	0.2
	11	15	0.6	4	0.2
	12	15	0.6	3	0.2
	13	15	0.6	2	0.2
	14	15	0.6	8	0.3
	15	15	0.6	8	0.4
	16	15	0.6	8	0.6
	17	15	0.6	8	0.8
Datum 2	**18**	**40**	**0.4**	**4**	**0.4**
	19	15	0.4	4	0.4
	20	20	0.4	4	0.4
	21	60	0.4	4	0.4
	22	80	0.4	4	0.4
	23	40	0.6	4	0.4
	24	40	0.5	4	0.4
	25	40	0.3	4	0.4
	26	40	0.2	4	0.4
	27	40	0.4	8	0.4
	28	40	0.4	6	0.4
	29	40	0.4	3	0.4
	30	40	0.4	2	0.4
	31	40	0.4	4	0.2
	32	40	0.4	4	0.3
	33	40	0.4	4	0.6
	34	40	0.4	4	0.8
Datum 3	**35**	**80**	**0.2**	**2**	**0.8**
	36	15	0.2	2	0.8
	37	20	0.2	2	0.8
	38	40	0.2	2	0.8
	39	60	0.2	2	0.8
	40	80	0.6	2	0.8
	41	80	0.5	2	0.8
	42	80	0.4	2	0.8
	43	80	0.3	2	0.8
	44	80	0.2	8	0.8
	45	80	0.2	6	0.8
	46	80	0.2	4	0.8
	47	80	0.2	3	0.8
	48	80	0.2	2	0.2
	49	80	0.2	2	0.3
	50	80	0.2	2	0.4
	51	80	0.2	2	0.6

For experimental purposes, each of the input variables is divided into five levels from low to high difficulty values as in Table 2. Then, a reference datum is selected as a baseline, while the rest of the problem variable settings are generated by changing only one variable value at a time. In total, there are 17 test problems (including datum) generated from one datum setting. In order to confirm algorithm performance, three different datum will be used (Level 1, 3 and 5). Therefore, the complete number of test problem in this experiment is 51 problems as shown in Table 3. The bolded problem setting (Problems 1, 18 and 35) represent the datum setting for Levels 1, 3 and 5 respectively.

3.1. Performance Evaluation

To evaluate the performance of each approach when dealing with different complexity problems, the following performance indicators are adopted [19], [20]:

i. Number of non-dominated solution, $\tilde{\eta}$: Shows the quantity of non-dominated solutions obtained by optimisation algorithm in the Pareto solution set. The larger $\tilde{\eta}$ number represents a better performance of algorithm.

ii. Error Ratio, ER: ER counts the number of solutions which are not members of the Pareto optimal set, divided by the number of solutions generated by algorithm q. Smaller ER indicates better algorithm performance.

iii. Generational Distance, GD: GD calculation finds an average distance of solution with the nearest Pareto optimal solution. Smaller GD value indicates better algorithm performance.

iv. Spacing: Indicates the relative distances for a solution with another solution. Smaller Spacing index indicates better solution set having better spacing between each solution.

v. Maximum Spread, $Spread_{max}$: Measures the spread of solutions found by each algorithm. A higher maximum spread shows better algorithm.

3.2. Optimisation Algorithm

Various algorithms have been developed to optimise the combinatorial optimisation problem. In this work, we implement Multi-Objective Discrete Particle Swarm Optimisation (MODPSO) to optimise both sequential and integrated ASP and ALB. This algorithm is selected because of good performance in the ASP and ALB optimization, when compared with other algorithms such as Genetic Algorithm, Ant Colony Optimisation and Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [21]. The general procedure of MODPSO is presented as follows.

Procedure: MODPSO

Initialisation

- Set *npar* (number of particle), $iter_{\max}$ (maximum number of iteration)
- Generate initial swarm, *X* (position) and *V* (velocity)
- Encode *X* into feasible solution (*F*) using Topological Sort procedure

Evaluation

- Evaluate objective functions
- Apply non-dominated sorting procedure

Update *Pbest* and *Gbest*

- Calculate Crowding Distance (*CD*) within solution in swarm
- Update *Pbest* based on maximum *CD*
- Calculate Crowding Distance within non-dominated set
- Update *Gbest* based in maximum *CD*

Update *X* and *V*

- Update *X* and *V* using:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \text{and} \quad (4)$$

$$V_i^{t+1} = c_1 V_i^t + c_2(Pbest_i^t - X_i^t) + c_3(Gbest_t - X_i^t) \quad (5)$$

Subjected to discrete update procedure

End Procedure

The discrete procedures to update position and velocity are applied in this work [22].

Subtraction operator (position – position): $(X_1 - X_2)$.
If the j^{th} element of X_1, $x_{1,j} = x_{2,j}$ then $v_{1,j} = 0$, else $v_{1,j} = x_{1,j}$

Addition operator (position + velocity): $(X_1^t + V_1)$.
If the j^{th} element of V_1, $v_{1,j} = 0$ then $x_{1,j}^{t+1} = x_{1,j}^{t}$, else $x_{1,j}^{t+1} = v_{1,j}$

Multiplication operator (coefficient x *velocity)*: $(V_2 = c.V_1)$.
If $rand<c$, $v_2 = v_1$, else, $v_2 = 0$
$c \in [0,1]$

Addition operator (velocity + velocity): $(V = V_1 + V_2)$
The j^{th} element of V can be derived as follows:

$$v_j = \begin{cases} v_{1,j} \text{ if } v_{1,j} \neq 0, v_{2,j} = 0 \\ v_{1,j} \text{ if } v_{1,j} \neq 0, v_{2,j} \neq 0, r < cp \\ v_{2,j} \text{ otherwise} \end{cases} \quad (6)$$

r is a random number between 0 and 1, while $cp \in [0, 1]$.

The MODPSO algorithm for sequential and integrated ASP and ALB problem has been coded using the MATLAB software. In this work, the population or swarm size is set at 20 with 500 iterations.

4. Results and Discussions

Figure 2 presents the plot of performance indicators for sequential and integrated optimisation approaches. For the number of non-dominated solution in Pareto optimal set indicator ($\tilde{\eta}$) and the maximum spread indicator, larger values show better performance, while for the remaining indicators, smaller values present better performance. Based on Figure 2, the $\tilde{\eta}$ indicator consistently indicates that the integrated optimisation approach produces a larger number of non-dominated solutions in the Pareto optimal set than the sequential approach.

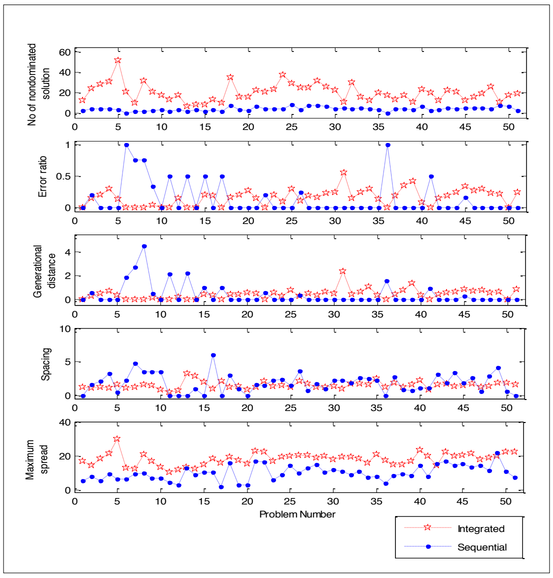

Fig. 2. Plot of performance indicators for sequential and integrated optimisation

On the other hand, the *ER*, *GD* and *Spacing* results indicate the intermixed performance of sequential and integrated optimisation approaches. The sequential approach shows better performance in 69% and 67% of the problem compared with the integrated approach for *ER* and *GD* correspondingly. This percentage is calculated based on the problem with better performance over the total number of problems. More than 80% of these problems are the medium and large size problems. Meanwhile, only 41% of the problems show better results for *Spacing* indicator using the sequential approach. In the meantime, 96% of the test problems come out with better performance using the integrated approach for $Spread_{max}$ indicator.

The results of $\tilde{\eta}$ show that the integrated optimisation approach is able to produce a larger number of solutions in Pareto optimal compared with the sequential approach. This result is related with the size of ASP and ALB search space. In the integrated approach, the ASP and ALB optimisation share the same search space. In contrast, in the sequential approach, the ALB optimisation encounters a tremendously reduced search space since its search space consists of the non-dominated solutions of ASP. This clearly precludes the appearance of solutions which are optimal with respect to ALB, but perhaps less optimal from the ASP perspective. Therefore, the chances to produce better solutions for an integrated optimisation approach is much higher compare with a sequential approach.

On the other hand, the integrated approach displays a mixed performance in *ER* and *GD* because of the number of non-dominated solutions found using this approach. Although the number of non-dominated solutions found using the sequential approach is much smaller than the integrated approach, the solutions have a better chance to belong to the Pareto optimal set. This is because in the first part of sequential approach (i.e. ASP optimisation), only two objective functions are evaluated, while the remaining objective functions are evaluated in the second part (i.e. ALB optimisation). Therefore, the chances to find an ASP non-dominated solution that is in Pareto optimum are better. This can be observed in some cases, where the sequential optimisation approach produces better results for ASP compared with the integrated optimisation approach. However, the integrated approach has consistently performed better or at least equal to the sequential approach in ALB problem.

Although the number of non-dominated solutions found using the sequential approach is less than the integrated approach, most of the solutions found belong to the Pareto optimum because the fewer numbers of objective functions being considered in the first part of optimisation enables quicker convergence. For this reason, 60% of the problem show better *GD* by using the sequential optimisation approach.

The *Spacing* indicator represents the solution uniformity between one solution with the nearest neighbour. This indicator depends on the number of solution found and also the solution spread. For a similar solution spread, the optimisation approach that comes out with a larger number of non-dominated solution will have better *Spacing*. Comparing the performance of sequential and integrated optimisation approaches using this indicator shows mixed results. There is no clear indication of which specific problem category is more suited to which approach.

The $Spread_{max}$ value indicates a capability to explore extreme solutions. For this indicator, 96% of the test problem show better results by using the integrated approach compared with the sequential approach. This result is due to the size of the search space for integrated ASP and ALB compared with sequential optimisation approach. In integrated optimisation, the chance to explore extreme solutions for ALB is higher compared with the sequential optimisation approach.

Finally, for ALB objectives, the integrated optimisation approach has consistently shown better or equal minimum values compared with sequential optimisation. The integrated optimisation produces better minimum values in 88% of the problems, while in the remaining 12% of the problems, the integrated and sequential optimisation approaches share similar minimum objective values. This finding substantiates the claim for the increased ability of the integrated ALB to find better solutions than sequential ALB.

5. Conclusions

This paper investigates and compares the performance of sequential and integrated optimisation approaches for Assembly Sequence Planning (ASP) and Assembly Line Balancing (ALB) in terms of solution quality towards Pareto optimum solution. In the sequential optimisation approach, ASP is optimised first and ALB optimisation follows, where the output from ASP will be the input for ALB. In contrast, the integrated optimisation approach optimises both ASP and ALB together at the same time.

The result from this study concludes that both sequential and integrated optimisation approaches have their own advantages. The sequential optimisation for instance has better Pareto optimum accuracy, but the number of solution is much smaller compared with the integrated approach because of the part by part optimisation approach. On the other hand, the integrated optimisation approach has a better ability to search for more Pareto optimum solutions and better exploration in the search space. This is due to the larger search space, especially for ALB problem. By having this knowledge, a proper optimisation approach can be selected for a particular ASP and ALB problem.

This work has successfully investigated and compared the sequential and integrated optimisation approaches based on solution quality towards Pareto optimum. The integrated optimisation consistently performed better in ALB problem in all problem sizes. In the future, the investigation of optimisation approach in terms of other benefits such as reduced error rate and lead time in production planning is suggested.

Acknowledgements

The author received financial support from Universiti Malaysia Pahang and the Ministry of Higher Education, Malaysia for the research and publication of this article.

References

[1] Ab Rashid, MFF, Hutabarat, W, Tiwari, A. Development of a tuneable test problem generator for assembly sequence planning and assembly line balancing. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf., 2012; 226 (11): 1900-13.

[2] Lu, C, Wong, YS, Fuh, JYH. An enhanced assembly planning approach using a multi-objective genetic algorithm. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf., 2006; 220(2): 255-72.

[3] Marian, RM. Optimisation of assembly sequences using genetic algorithm. PhD thesis. University of South Australia, 2003.

[4] Chen, R, Lu, K, Yu, S. A hybrid genetic algorithm approach on multi-objective of assembly planning problem. Eng. Appl. Artif. Intell., 2002; 15:447-57.

[5] Tseng HE, Tang, CE. A sequential consideration for assembly sequence planning and assembly line balancing using the connector concept. Int. J. Prod. Res., 2006; 44(1): 97-116.

[6] Tseng, HE, Chen, MH, Chang, CC, Wang, WP. Hybrid evolutionary multi-objective algorithms for integrating assembly sequence planning and assembly line balancing. Int. J. Prod. Res., 2008; 46(21):5951-77.

[7] Ab. Rashid, MFF. Integrated multi-objective optimisation of assembly sequence planning and assembly line balancing using particle swarm optimisation. PhD thesis. Cranfield University, 2013.

[8] Lu, C, Yang, Z. Integrated assembly sequence planning and assembly line balancing with ant colony optimization approach. Int. J. Adv. Manuf. Technol., 2016; 83(1): 243-56.

[9] Sakao, T, Lindahl, M. A method to improve integrated product service offerings based on life cycle costing. CIRP Ann. - Manuf. Technol., 2015; 64(1): 33-6.

[10] Fang, HC, Ong, SK, Nee, AYC. An integrated approach for product remanufacturing assessment and planning. Procedia CIRP, 2016; 40: 262 - 7.

[11] Li, X, Qin, K, Zeng, B, Gao, L, Su, J. Assembly sequence planning based on an improved harmony search algorithm. Int. J. Adv. Manuf. Technol., 2015; 84(9): 2367-80.

[12] Li, M, Wu, B, Yi, P, Jin, C, Hu, Y Shi, T. An improved discrete particle swarm optimization algorithm for high-speed trains assembly sequence planning. Assem. Autom., 2013; 33(4):360 - 73.

[13] Zhang, W, Xu, W, Liu, G, Gen, M. An effective hybrid evolutionary algorithm for stochastic multiobjective assembly line balancing problem. J. Intell. Manuf., 2017; 28(3): 783 - 90.

[14] Aydogan, EK, Delice, Y, Özcan, U, Gencer, C, Bali, Ö. Balancing stochastic U-lines using particle swarm optimization. J. Intell. Manuf., 2016; In press.

[15] Wang, HS, Che, ZH, Chiang, CJ. A hybrid genetic algorithm for multi-objective product plan selection problem with ASP and ALB. Expert Syst. Appl., 2012; 39(5): 5440 - 50.

[16] Moon, C, Kim, J, Choi, G, Seo, Y. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. Eur. J. Oper. Res., 2002; 140(3): 606 - 17.

[17] Rashid, MFF, Hutabarat, W, Tiwari, A. A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. Int. J. Adv. Manuf. Technol., 2011; 59(1-4): 335 - 49.

[18] Whitney, DE. Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development, Volume 1. Oxford University Press, 2004.

[19] Deb, K. Multi-Objective Optimization using Evolutionary Algorithms. Sussex, United Kingdom: John Wiley & Sons, 2002.

[20] Yoosefelahi, A, Aminnayeri, M, Mosadegh, H, Ardakani, HD. Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. J. Manuf. Syst., 2012; 31(2): 139 - 51.

[21] Ab Rashid, MFF, Hutabarat, W, Tiwari, A. Multi-objective discrete particle swarm optimisation algorithm for integrated assembly sequence planning and assembly line balancing. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf., 2016; In press.

[22] Rameshkumar, K, Suresh, RK, Mohanasundaram, KM. Discrete particle swarm optimization (DPSO) algorithm for permutation flowshop scheduling to minimize makespan. Adv. Nat. Comput., 2005; 3612: 572 - 81.