Optimization of Automotive Manufacturing Layout for Productivity Improvement

Muhamad Magffierah Bin Razali, Mohd Fadzil Faisae Ab.Rashid,*
Muhammad Razif Abdullah Make
Faculty of Mechanical Engineering,
Universiti Malaysia Pahang,
26600 Pekan, Pahang, Malaysia
*ffaisae@ump.edu.my**

ABSTRACT

This paper deal with an optimization of automotive manufacturing layout by using meta-heuristics approach aided with discrete event simulation (WITNESS Simulation). The objective of this study is to balance the workload, increase line efficiency, and improve productivity by optimizing assembly line balancing (ALB) using Genetic Algorithm. The current assembly line layout operated under the circumstance where idle time is high due to unbalance workload. After the optimization process takes place, the workload distribution in each workstation has shown a significant improvement. Furthermore, productivity improvement was gained after the optimization followed by increment in term of line efficiency by 18%. In addition, the number of workstation needed to assemble the product can be reduced from current layout (17 workstations) to an improved layout (14 workstations). The current study contributes to the implementation of Genetic Algorithm in ALB to improve productivity of related automotive manufacturing industry.

Keywords: *Assembly Line Balancing, Genetic Algorithm, Productivity Improvement*

Introduction

Product quality and the capability to adapt to consumer demands are pivotal perspective that must be thought seriously especially in automotive industry. Administration frameworks need to be design comprehensively to ensure the

ISSN 1823- 5514, eISSN 2550-164X
Received for review: 2017-04-28
Accepted for publication: 2017-05-11
Published: 2017-08-15

end goal which is to take control, arranging, and measuring parameters related to attain a smooth and better company performance. Company or organizations need to understand that the efficiency of production framework is depends upon how well the production line can perform in term of producing the product output by ensuring an efficicent assembly line [1].

Thus, the main objective of assembly line balancing (ALB) is to distribute the task evenly over the workstations on the production assembly line. So that, a better efficent assembly line layout can be achieved to produce a quality product [2]. In that process, idle time of each task also can be minimized. Product, process and fixed-position layout are three basics types of layout design [3]. In general, there are a few variant of line balancing problem. The first variant is single model assembly line balancing. A single model line will be pictured as a line that assembles a single-solitary model [4]. This line produces various units of one product with no extra characteristics as the additional selection. The tasks performed at all station are similar for all units. Product with high demand is expected to be assemble in this kind of line [5].

Next is mixed-model assembly line. This type of assembly line is equipped with the aims at delivering more than one model. The products that being made can be assemble on the same single assembly line [6]. At the point when one model is worked at one stations, the other thing or item are made at substitute stations. Subsequently, every station is arranged to perform distinctive assignments anticipated that would make any model that goes through it. Various customers item is amassed on this type of model.

Lastly, batch model assembly line this line makes every one model in bunches. Ordinarily workstations are arranged up to make obliged measure of the first model then the stations are reproduced to convey other model [7]. Items are frequently accumulated in groups when medium interest. It's more progressive to use one sequential construction system creation framework to make a couple items in packs than develop an alternate line for every one model.

Conventional method in assembly line balancing put more stressed on hands-on balancing and try and error basis. This method however can be time-consuming as well as material costing [8]. Thus, researcher nowadays tend to use a modern approach by using so called ‘artificial intelligence (AI)’ approach. This method can save more time as well as cost-friendly since it mainly involved a computerized usage compared to conventional method. Since then, AI usage in modern manufacturing which called as meta-heuristics method is utilized widely such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), and simulated annealing (SA).

Genetic algorithm (GA) are an optimization and search method which inspired from the principle of natural evolution [9]. It is an established algorithm because it has been used for more than three decades since being introduced in 1970's by John Holland. GA deal with the principal of "survival of the fittest" especially in its search process of quality solution. Nowadays, more researcher in the related fields tend to use this meta-heuristics algorithm approach for addressing a real-world assembly line balancing problem.

The aim of this paper is to optimize assembly line balancing problem using Genetic Algorithm approach. For this purpose, we consider the simple assembly line balancing problem and for optimization purpose, a case study from literature is adopted. To measure the performance of the optimized layout, a discrete event simulation is conducted. The objective for this research included to smoother workload balance, increase productivity, and increase line efficiency. In the progress, we also compared the idle time and busy time for current and improved layout after applying the ALB.

Genetic Algorithm in Assembly Line Balancing

As mentioned earlier, conventional search technique usually incapable of solving an optimization problem that deal with complex problem and non-linear multi-modal functions [10]. In this cases, a meta-heuristic search method is required. The usage of genetic algorithm (GA) in assembly line balancing is crucial since it is a guided semi-random search technique.

The GA techniques consist of five steps. The first step is initialization of population. Generally, initial population is generated randomly in this first step which govern the whole range of possible solution. At times, the solution may be seeded in a region where optimum solutions are possible to be found.

Next is objective function evaluation which is in this study, minimizing idle time and obtaining balanced workstation are the aims. The designated objective functions served as the evaluation criteria to determine the quality of the solutions later on.

Entering the selection phase, several selection outlines have been proposed in the previous literature such as Roulette Wheel, Tournament selection, Rank selection, and Sigma scaling. In this study, a Roulette Wheel selection is used where it selects chromosomes proportionate to their fitness scores.

Genetic Algorithm Operator

Since certain selection method mentioned earlier has their own framework in choosing the fitness of each solution, GA operators are used since it is a process to maintain the genetic diversity. The example of GA operator such as:

- *Crossover*. In GA, crossover is one of the popular tools for recombination of the individual's chromosomes. This GA operator is analogue to those which occur in biological evolutionary processes where the genes of two parents are combines to produce child (offspring).

- *Mutation*. Unlike the crossover, mutation is an operator in GA which its mechanism designed to recombining chromosomes randomly. By doing so, the diversity of given solutions can be ensured.

Problem Definition and Formulation

In this study, an assembly line data is adopted from literature by [11]. It is originated from factory which deals with automotive and defence industries in Sakarya, Turkey. Although many different products are produced in the factory, small bus manufacturing, which covers 70% of all products demand in a year was chosen for the assembly line balancing problem. This problem consists of 17 workstations with 53 number of tasks. The assembly data is presented in Table 1.

Listed below are the parameter and indices used in describing the problem formulation:

Notation	Definition
nws	number of workstations $s = 1,2\ldots, S$
P_T	processing time
N_e	number of task $e = 1,2\ldots, Ne$
I_T	idle time in each workstation
A_{nws}	actual number of workstation
B_T	busy time
C_{lk}	completion time
pre_i	precedence for task *i*
C_T	cycle time
t_e	task time
LE_n	new layout line efficiency
C_{ie}	completion time on station *i* of task *e*
p_{ie}	assembly time on station *i* for task *e*

Objective function and constraints

This study aims to come out with a new assembly line layout that improved in term of workload distribution, increased line efficiency, and improved in productivity. Hence, Eq.1, Eq.2, and Eq.3 show the formulation that related to achieve the designated target [12].

$$\text{Workload balance} = \frac{\sum_{i=1}^{n} C_T \ - P_T}{nws} \tag{1}$$

Line efficiency plays an important role in manufacturing industry to ensure the demand from customer can be met. This can be done by increasing the assembly line efficiency. Line efficiency *(LE)* for each layout can be calculated as follows:

$$LE(\%) = \frac{\text{Total Task Times}}{(A_{nws})\ \text{X}\ (C_T)} \tag{2}$$

$$PI = \frac{|current\ output - new\ output|}{current\ output} \times 100\% \tag{3}$$

Constraints in assembly line balancing include inequality (4) which is a cycle time restriction and inequality (5) represents the precedence restrictions among the assembly tasks.

$$\sum_{s=1}^{S} X_{as} - \sum_{s=1}^{S} X_{bs} \leq 0, for\ \forall(a,b) \in pre_i \tag{4}$$

$$\sum_{s=1}^{S} C_{ik} - \sum_{l\in I_j} C_{lk} \geq \sum_{i\in I_r} p_{ik}x_{ir} \tag{5}$$

Referring to Table 1, the data that being used consist of total 53 number of tasks with the total cycle time as much as 3966.4 minutes to assemble a complete one product in that particular assembly line. Current layout contain 17 number of workstations and each workstation has been assigned with its own task. For example, workstation number 1 only consist of task 1 to be completed. Meanwhile workstation number 3 need to finish task 1, 2, and 4 assigned to it before can proceed to the next workstations.

Table 1: Current system data information

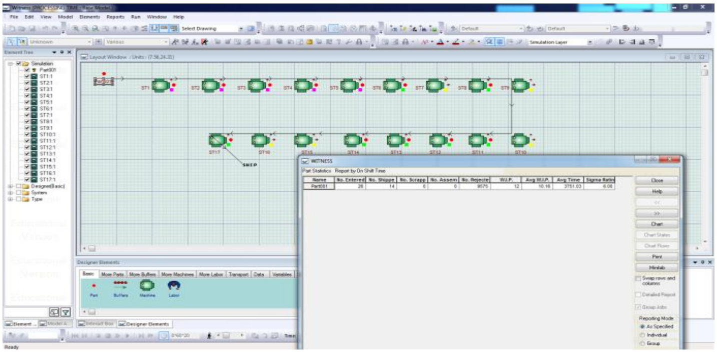
nws	N_e	t_e	*pre*$_i$
1	1	144.3	–
2	2	71.7	1
	3	71.9	–
	4	45.7	–
3	5	128.1	2,1
	6	101.8	–
	7	44.7	1,2,4

4	8	47	5,2,1
	9	128	2,5,1
	10	83.2	4
	11	28.7	–
5	12	125	9,5,2,1
	13	48.2	12,5,2,1
6	14	314.2	13, 12, 11, 10, 8, 5, 4, 2, 1
	15	64	10, 8, 5, 4, 2, 1, 13
	16	30.7	13
7	17	92.8	16, 14, 11, 8
	18	91.7	1, 2, 5, 14
	19	61.3	16, 14
8	20	115.3	14
	21	95.2	17, 18, 14
9	22	85.9	21,20
	23	85.8	7
10	24	59	17, 23
	25	70.5	24
11	26	257.3	25, 23, 20
	27	31.7	19, 17, 14
	28	26.6	25,23
12	29	104.7	26, 23, 20
	30	126.5	26, 17, 16, 15, 14, 10
	31	33.9	26, 17, 16, 15, 14, 10
	32	67.7	22, 18, 17
	33	59.1	32
13	34	77.2	25, 22, 14
	35	26.2	29, 28, 26
	36	16.7	32
	37	21.6	30,26
	38	27.7	26
	39	42.4	31, 29, 26, 30
	40	111.4	19, 18, 32, 22
	41	9.7	30
14	42	63.3	39, 33, 32, 31, 29, 27, 22
	43	105.8	40, 39, 33, 32, 22
	44	30.1	30, 31, 24, 22, 14
15	45	127.3	44, 43, 34, 25
	46	37.8	31, 26
16	47	69.6	40, 36, 33, 27, 22, 3
	48	78.8	46, 39, 37, 35, 31, 29, 26
	49	53.3	43,40
17	50	11	1, 2, 5, 12, 14

	51	94.1	39, 1, 2, 5, 14
	52	15.5	43, 42
	53	4.7	52, 51, 50, 38, 48, 45, 49, 47

WITNESS Modelling and Simulation

As shown in Figure 1, the simulation layout of current assembly line was created based on assembly data gained from Table 1. Here, a simulation of an actual assembly line layout is executed. The purpose of this simulation is to obtain a deep knowledge of the current production system and its level of performances. Besides, information such as idle time, work in progress, busy time and product output can be extract from this activity.



Part Statistics Report by On Shift Time

Name	No. Entered	No. Shippe	No. Scrapp	No. Assem	No. Rejecte	W.I.P.	Avg W.I.P.	Avg Time	Sigma Ratin
Part001	26	14	6	0	9575	12	10.18	3751.03	6.08

Figure 1: Existing layout with 17 number of workstations

Firstly, the data from Table 1 is used and simulate into the WITNESS simulation. The data consists of 53 number of task, a total number of 17 workstations, and also its task time in minute for each of related task. An assemble machine is put into a specific arrangement, and then each workstation is set with the specific cycle time such in Figure 1.

Then, the simulation is conducted by measuring the production in one-month period which gives us a total of 9600 minutes. From the table, the output from the original layout is approximately 14 units per month. Here, we can see that workstation number 6 is the possible busiest workstation and the bottleneck is highly will occur in that particular area.

Defining the Highest Cycle Time and Area of Bottleneck

In order to find and investigate both cycle time and area of bottleneck, we need to summarize the data from WITNESS Simulation into a form of table to make it clearer. Table 2 below explains the detail:

Based on the tabulation, it is clearly shown that workstation number 6 is the region of bottleneck since it has a highest cycle time as much as 408.9

minutes. The optimization by using line balancing can reduce this unbalance workload. From workstation 1 till workstation 17, total cycle time being recorded is 3966.4 minutes. Thus, it brings to a one condition that is the task can only be completed if the total cycle time is still the same after optimization.

Table 2: Data from WITNESS Simulation (Workstation grouping)

Station	Cycle time (min)	Idle time (min)
ST1	144.3	264.6
ST2	189.3	219.6
ST3	274.6	134.3
ST4	286.9	122.0
ST5	173.2	235.7
ST6	408.9	0.0
ST7	245.8	163.1
ST8	210.5	198.4
ST9	171.7	237.2
ST10	129.5	279.4
ST11	315.6	93.3
ST12	391.9	17.0
ST13	332.9	76.0
ST14	199.2	209.7
ST15	165.1	243.8
ST16	201.7	207.2
ST17	125.3	283.6
Total	**3966.4**	**2985.1**

Optimization of ALB using Genetic Algorithm

The following pseudocode presents the GA procedure to optimize the ALB problem.

```
Begin;
  Generate random population of $P$ solutions (chromosomes);
    For each individual $i \in P$: calculate fitness ($i$); For $i$=1 to number of
    generations;
    Randomly select an operation (crossover or mutation);
      If crossover;
       Select two parents at random $i_a$ and $i_b$;
       Generate on offspring $i_c$=crossover ($i_a$ and $i_b$);
      Else If mutation;
       Select one chromosome $i$ at random;
       Generate an offspring $i_c$=mutate ($i$);
      End if;
    Calculate the fitness of the offspring $i_c$; If better than the worst chromosome
    then replace the worst chromosome by $i_c$;
   Next i;
  Check if termination=true;
 End;
```

Results of Optimized Layout

Each of the proposed layouts has a difference output in term of its idle time and number of workstation itself varies for each of layout. Hence, to produce a clearer result and analysis, an overall performance is evaluated. Table 3 below explains the optimization results of various parameters in each layout with different number of workstation such as the product output, total idle time, workload balance and etc.

Table 3 Optimization result summary

	Original layout		**Improved layout**		
Parameter	*nws* = 17	*nws* = 17	*nws* = 16	*nws* = 15	*nws* = 14
Output	14 unit	18 unit	18 unit	18 unit	18 unit
Total Idle (min)	684.1	652.0	446.5	418.8	354.7
Workload Balance	40.2	38.3	27.9	27.9	25.3
Avg.Busy (%)	47.2	60.5	64.5	68.4	73.2
Avg.Idle (%)	40.2	38.4	27.9	27.9	25.3

In this work, the main parameter that being measured closely are the production (unit) in one-month production, the average busy (%), and average idle (%). Referring to Table 3 the average idle (%) shows a significant decrement as the number of workstation reduced. Meanwhile the average busy (%) give an increment as the number of workstation reduced.

The graph in Figure 2 shows the comparison of current layout and optimized layout with 17 workstations. As mentioned earlier, the bottleneck region clearly occurs at workstation 6 by possessing the highest processing time at 408.9 minutes. After applying the optimization to the current layout, the bottleneck in that particular workstation was reduced to 314.2 minutes. Thus, the new improved layout provides better assembly line layout in term of time management as well as enhanced work distribution.

Then, the optimization continues the effort to achieve the others objective function which is to minimize number of workstation. Originally, current layout contains 17 workstations and now been reduced to 16, 15, and 14 workstations.

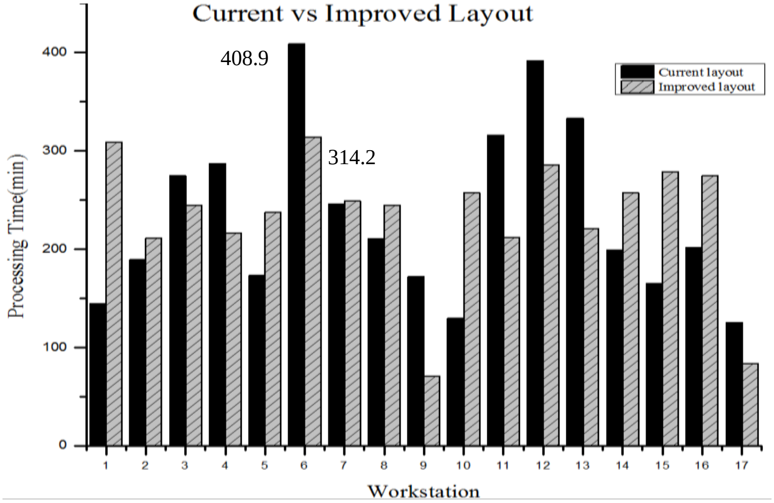


Figure 2: Improvement layout for 17 workstations

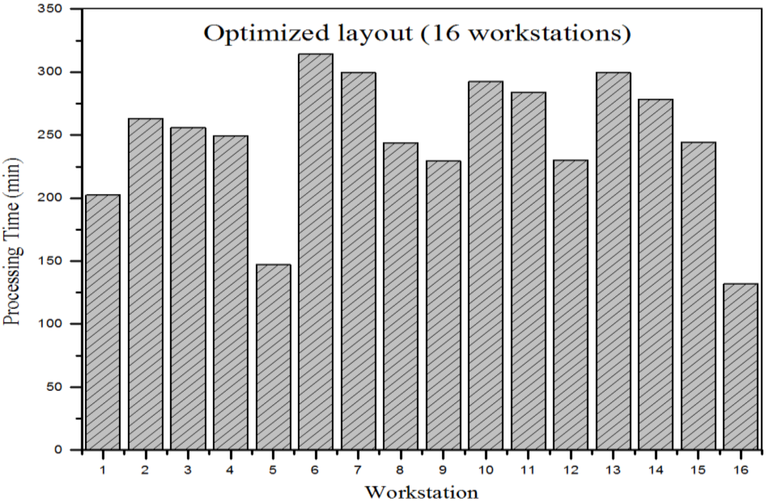


Figure 3 Improved layout (16 workstations)

Based on Figure 3, Figure 4 and Figure 5, it can be observed that the processing time for each workstation from 16 to 14 workstations becomes more balanced. This is consistent with the result simulation from WITNESS software tabulated in Table 3. The workload distribution throughout the whole assembly line layout has shown a significant improvement after optimization. This is proved by the decrement of workload balance from 40.2 to 25.3.

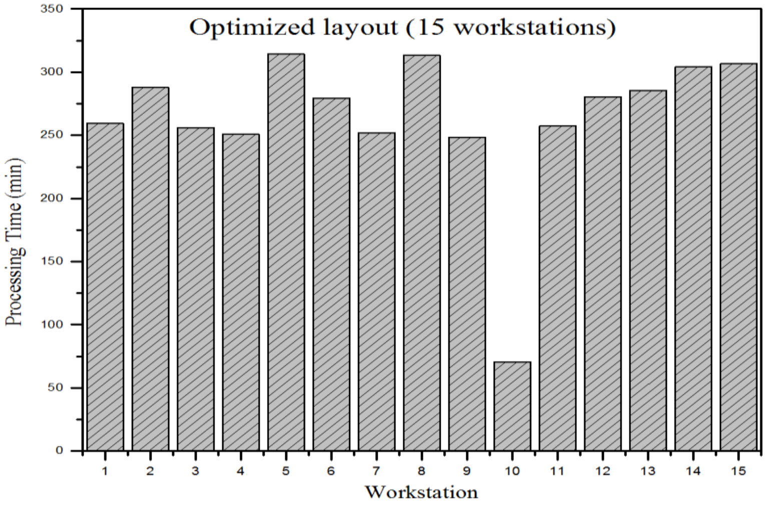


Figure 4 Improved layout (15 workstations)

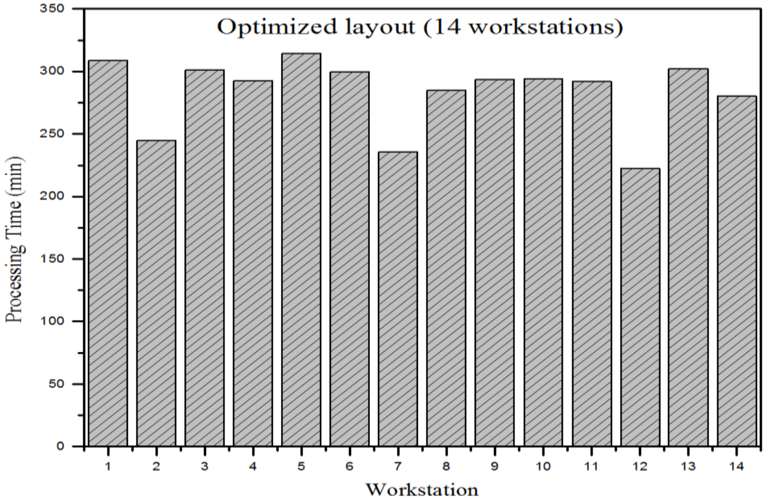


Figure 5 Improved layout (14 workstations)

Lastly, the minimum number of workstation that can be achieve is 14 workstations. In fact, when optimized using GA, not only number of workstation has been reduced, but also produced a better workload distribution. Thus, a smoother work flow can be attained. Clearly, current layout with 17 workstations operated under high total idle time. After assembly line balancing is applied, idle time throughout the assembly line is minimized.

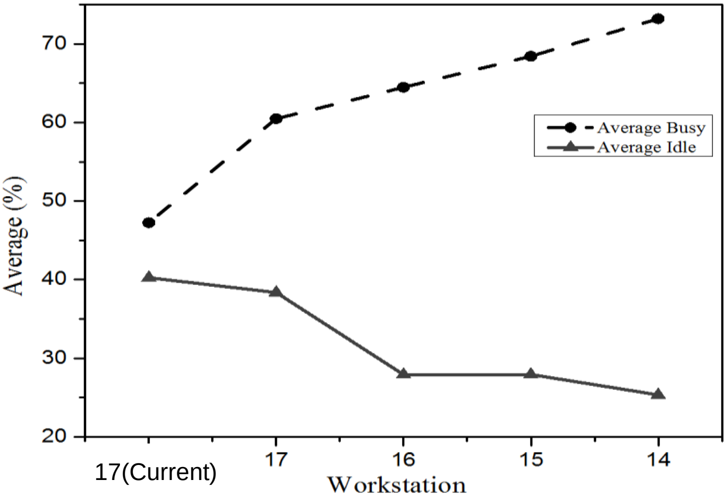


Figure 4 Average busy and idle percentage for each layout

From Figure 4, the time needed to complete all the assembly tasks become better distributed after optimization using assembly line balancing. This is based on the increasing of average busy (%) and decreasing of average idle (%) for each layout compared to original system layout. Then, we can measure the production after optimization. Referred to Table 3**,** the original layout have the capability to produce 14 units of buses in one month production whereas after the optimization, the volume going up from 14 to 18 units. Then, the percentage of productivity improvement (*PI*) based on Eq.3 formulation have been made and calculated as follows:

$$PI = \frac{|current\ output - new\ output|}{current\ output} \times 100\% \qquad (3)$$

$$= \frac{|14-18|}{14} \times 100\%$$

$$= \quad 28\% > 10\%$$

Calculation above clearly shows the productivity improvement as much as 28% is achievable by applying an optimization method of assembly line balancing by using this meta-heuristic approach. Then, the line efficiency (LE_n) for the new improved layout also can be computed by using Eq.2. Noted that, current assembly line layout has a line efficiency of 72%, refer [11].

$$LE_n = \frac{\text{Total Task Times}}{(A_{nws})\ \text{X}\ (C_T)}$$

$$= \frac{3966.4}{(14)\times(314.2)}$$

$$= 0.9017\ @\ 90\%$$

It is apparent from calculation of LE above, the current assembly line efficiency can be enhanced by 18% which is from originally 72% to 90% after applied with the proposed optimization approach.

Conclusion and Discussions

In this research, simulation and optimization techniques using GA have been proposed to improve productivity in aforementioned automotive production layout. From the result shown, the new improved layout can give 28% improvement compared to original layout. After optimization take places, the production output has been increased from 14 to 18 units in one-month production plus the number of workstation can be decrease from 17 workstations to only 14 workstations. Likewise, line efficiency of current layout has been significantly increased from 72% to 90%. This value gives an overview that all machines, tools as well as the workers can be fully utilized due to usage of an efficient assembly line layout.

The result of this study however, has some limitation included the assumption of actual number of workers is always enough for each workstations. In the literature, the authors has some restriction in term of maximum allowable workers. Next, a proper parameter settings for Genetic Algorithm could possibly return a better solution. Likewise, it would be interesting to assess the performance of other meta-heuristics approach besides the one proposed in this paper. For future direction, further investigation also can be implement by consider an actual assembly line balancing problem such as in automotive manufacturing industry and other related assembly company.

Acknowledgement

This research has been supported by research fund from Ministry of Higher Education under RDU140103 and internal grant UMP under PGRS170318.

References

[1] F. De, M. Antonietta, O. Borgia, and M. Tucci, "Layout Design for a Low Capacity Manufacturing Line: A Case Study", *Int. J. Eng. Bus. Manag.*, p. 1, (2013).

[2] C. Yang and J. Gao, "Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment", *Comput. Oper. Res.*, vol. 65, pp. 139–148, (Jan. 2016).

[3] M. M. Davis and J. Heineke, "Operations management," (2005).

[4] M. Fathi and M. Ghobakhloo, "A technical comment on 'a review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches", *Int. J. Adv. Manuf. Technol.*, vol. 71, no. 9–12, pp. 2033–2042, (2014).

[5] J. Sternatz, "Enhanced multi-Hoffmann heuristic for efficiently solving real-world assembly line balancing problems in automotivey industr", *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 740–754, (Jun. 2014).

[6] A. Hamzadayi and G. Yildiz, "A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines", *Comput. Ind. Eng.*, vol. 66, no. 4, pp. 1070–1084, (Dec. 2013).

[7] N. Boysen, M. Fliedner, and A. Scholl, "Level scheduling for batched JIT supply", *Flex. Serv. Manuf. J.*, vol. 21, no. 1–2, pp. 31–50, (2009).

[8] A. Dolgui and J.-M. Proth, *Assembly line balancing: Conventional methods and extensions*, vol. 46, no. 9. IFAC, (2013).

[9] M. M. Razali, M. Fadzil, F. Ab, and M. Razif, "Mathematical modelling of mixed-model assembly line balancing problem with resources constraints", vol. 1, (2002).

[10] M. R. Abdullah Make, M. F. F. Ab. Rashid, and M. M. Razali, "A review of two-sided assembly line balancing problem", *Int. J. Adv. Manuf. Technol.*, (2016).

[11] H. R. Yazgan, I. Beypinar, S. Boran, and C. Ocak, "A new algorithm and multi-response Taguchi method to solve line balancing problem in an automotive industry", *Int. J. Adv. Manuf. Technol.*, vol. 57, no. 1–4, pp. 379–392, (Apr. 2011).

[12] R. Esmaeilbeigi, B. Naderi, and P. Charkhgard, "The type E simple assembly line balancing problem: A mixed integer linear programming formulation", *Comput. Oper. Res.*, vol. 64, pp. 168–177, (2015).