IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Evaluation of rank-based crossovers to optimize real-life assembly line balancing with resource constraint

To cite this article: K H Khalib *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **469** 012014

IOP Conf. Series: Materials Science and Engineering 469 (2019) 012014 doi:10.1088/1757-899X/469/1/012014

Evaluation of rank-based crossovers to optimize real-life assembly line balancing with resource constraint

K H Khalib[1], **N H Kamarudin**[1], **M F F Ab. Rashid**[1,*]

[1]Manufacturing Focus Group, Faculty of Mechanical Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang Malaysia.

*Corresponding author: ffaisae@ump.edu.my

Abstract. Assembly line balancing (ALB) problem has evolved in lined with the manufacturing advancement. Previous research in ALB mostly assumed that all workstations are having similar capabilities including the machines, tools and worker skills. Recently, researchers started to consider the ALB with resource constraint (ALB-RC) such as machine and worker. This paper aim to evaluate new rank-based crossovers to optimize real-life ALB-RC problem. Prior to this work, the authors had proposed rank-based crossover type I and II (RBC-I and II) to enhance the performance of Genetic Algorithm (GA) in optimizing ALB-RC problem. An industrial case study has been conducted in electronics industry. The results of industrial case study confirmed that the proposed ALB-RC model is capable to be used for the real industrial problem. At the same time, the result indicated that the GA with rank-based crossover is capable to optimize real-life problem. As a comparison, the number of workstation, resources and workers had reduced between 10 – 15% for the optimised layout using GA with RBC, compared with the original layout in the case study problem

1. Introduction

Assembly Line Balancing (ALB) is the process of assigning tasks to workstations without violating the precedence relation between tasks. However, in assigning assembly task to workstation, there are many possibilities of the sequence to be evaluated [1]. The possible task of sequence for *n* task and *r* constraints will be $n!/2^r$ [2]. Every assembly sequence will be analysed by considering other assembly constraints in achieving the most optimum and best solution for ALBP.

In the majority of the previous works, researchers make assumptions where any of assembly tasks can be processed or assembled in any workstations [3], [4]. This is certainly true for the product that only requires a common or simple tool to be assembled. However, when the complexity of a product increased, it requires a special tool, machine or highly skilled labour to assemble that particular component. Therefore, the limitation of resources will be another constraint for the industry. In fact, the issue of line balancing with the minimum number of resources has always been a serious problem in the industry [5]. This problem is known as assembly line balancing with resource constraints (ALB-RC).

Previously, researchers had studied the line balancing with resource constraints. Ağpak and Gökçen [5] started the ALB-RC by considering two resources and solve the problem using integer programming. Next, Corominas, Ferrer, and Pastor [6] proposed a model to support generalised constraints problem. Özdemir and Ayağ [7] consider equipment constraint when assigning task to workstation for SALBP. In this paper, the researcher use branch and bound algorithm together with the analytic hierarchy process (AHP) method to determine the optimum solution in minimising the equipment cost for production line. Koltai and Tatay [8] later proposed a model and optimise the ALB with worker skill constraint. The purpose is to match the assembly task with the level of the worker skill. Jayaswal and Agarwal [9] conducted research on assign tasks to workstations, and resources (equipment and assistants) to tasks with the objectives function is to minimised total cost of workstation and resource utilisation. This research is modelled to a U-shaped assembly line balancing using Simulated Annealing. Besides that, [10] optimise the multi-objective ALB with general resources using domination concept.

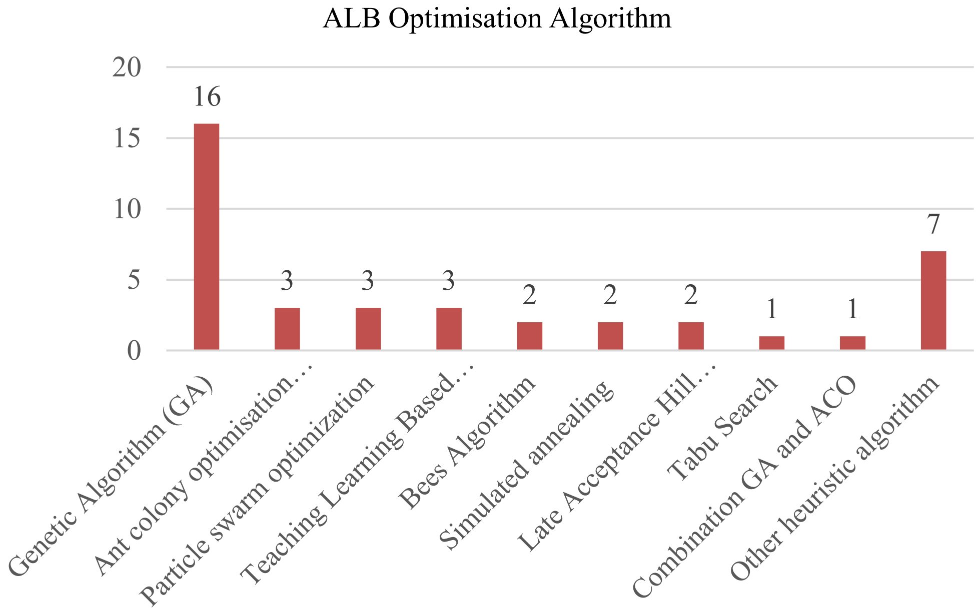

Figure 1. Previous research on ALB using heuristic/metaheuristic algorithm

Previous research on ALB optimisation shows that various metaheuristics methods were used by researchers. Figure 1 shows the number of papers that used different metaheuristic methods to optimise ALB problems from the year 2005. According to the diagram, the three most dominant optimisation methods, used in 55% of the cited research, are Genetic Algorithm (GA), Ant Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO).

This paper presents an optimization of ALB-RC for an industrial case study problem. Previously, an improved Genetic Algorithm has been proposed to optimize ALB-RC [11]. However, in that paper, researchers only tested the algorithm against a set of generic problems. The purpose of this study is to evaluate the performance of the improved GA with other GA based algorithms in optimizing real-life industrial problem. The result had been shared with the company and the implementation is depending on the company decision.

2. Case Study

An industrial case study has been conducted in an electronics manufacturing company. The industrial case study started with understanding the assembly process for the studied product. Then, the assembly task for the studied product is identified. In this case, the assembly tasks are directly defined based on the work elements used by the company.

Next, the precedence constraints are defined according to the engineer's input and the assembly process observation. Then, the assembly data collection is made. For the assembly task time, five repetitions are made and the average time is calculated. Besides the assembly time, the main machine or equipment used to conduct the assembly task is also recorded. In addition, the details of worker skills and requirements are gathered. The precedence graph of the studied product is shown in Figure 2.

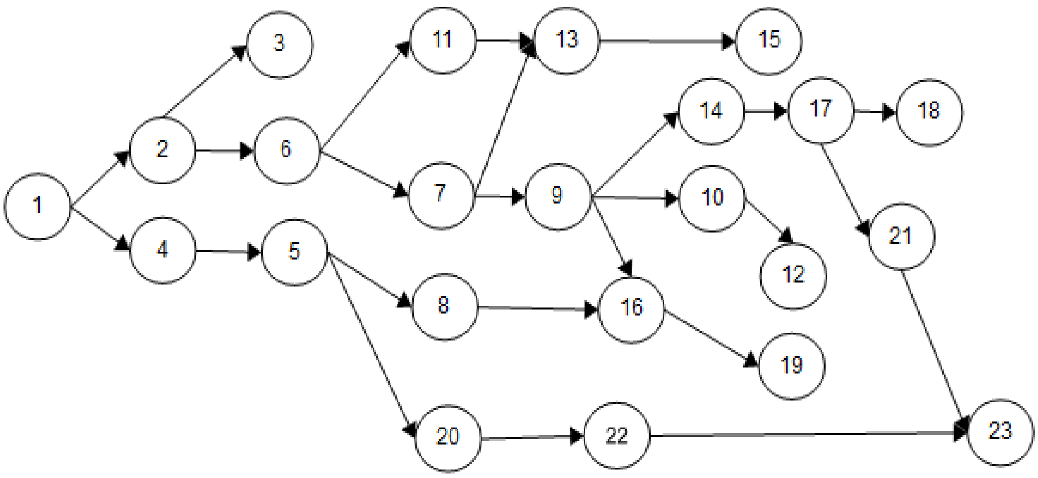

Figure 2. Precedence graph for the case study

The assembly data for this problem is presented in Table 1. For the optimisation purpose, the average time will be used. Table 1 also presents the required machine or tool for the equipment. Since the proposed model allows one machine per assembly task, only the main machine or equipment is considered. It should be noted that some of the assembly tasks requires more than one tool to be completed. The last column in Table 1 shows the worker that needs to conduct a particular assembly task. Currently, 10 workers are needed to perform the assembly process. According to the line supervisor, the workers can conduct any assembly task with minimum training required. Therefore, it is assumed that any assembly task can be assigned to any worker.

Table 1. Assembly data for case study problem

Task	Task time (s)	Machine	Worker	Task	Task time (s)	Machine	Worker
1	38.82	M1	W1	13	57.92	M6	W5
2	31.14	M1	W1	14	59.88	M5	W6
3	47.96	M2	W1	15	44.94	M5	W6
4	88.86	M2	W2	16	31.8	M7	W7
5	27.32	M3	W2	17	27.26	M6	W7
6	47.74	M2	W3	18	69.82	M6	W8
7	73.14	M3	W3	19	22.28	M7	W8
8	20.9	M2	W4	20	43.9	M4	W9
9	64.3	M3	W4	21	19.36	M8	W9
10	11.88	M4	W4	22	20.08	M7	W9
11	15.14	M4	W4	23	33.82	M8	W10
12	64.78	M5	W5				

The efficiency of the assembly line configuration can be measured using the objective function as follows:

(i) Minimise number of workstation
(ii) Minimise number of tool/machine
(iii)Minimise number of worker

The first objective function which is to minimise the number of workstations for a given cycle time. The second objective function (2) is to minimize the number of machines used which subject to resource availability constraints, which ensure that the total number of resources in the workstation is not more than the number of available machines. The third objective function (3) is to minimize the total number of workers used in an assembly line with the restriction that only one worker to be assigned to exactly one workstation depending upon his/her skills.

$$f_1 = \min \sum_{s=1}^{S} A_s \tag{1}$$

$$f_2 = \sum_{m=1}^{r} y_{ms} \tag{2}$$

$$f_3 = \sum_{w=1}^{h} z_{ws} \tag{3}$$

For optimization purpose, a weighted sum approach is used to combine the objective functions. However, due to different unit and ranges for objective functions, its need to be normalized into a similar range. For this purpose, the following formula is used:

$$\bar{f_i} = \frac{(f_i - f_{i\,min})(\bar{f}_{i\,max} - \bar{f}_{i\,min})}{(f_{i\,max} - f_{i\,min})} + \bar{f}_{i\,min} \tag{4}$$

Then, a fitness function is employed to minimize the summation of normalized objective functions.

$$F = w_1\bar{f_1} + w_2\bar{f_2} + w_3\bar{f_3} \tag{5}$$

Where;

$$w_1 = w_2 = w_3 = 0.33$$

w_i represent the weights of objectives and $\bar{f_1}$, $\bar{f_2}$ and $\bar{f_3}$ represent respectively the normalised values derived from the equations.

In addition to the optimisation objective function, the following indicators are used to measure the efficiency of the assembly line.

Smoothness index (*SI*)

$$SI = \sqrt{\sum_{k=1}^{K} (ct - pt_k)^2} \tag{6}$$

Line efficiency (*LE*)

$$LE = \frac{\sum_{i=1}^{n} t_i}{nws \times ct} \tag{7}$$

SI measure how balance the workload assignment between workstation. The smaller SI represent better workload balance. This will smoothen the flow of the assembled product in the assembly line.

Meanwhile, the LE shows the level of value added time utilization in assembly line. The higher LE indicated the lower time wasting in the assembly line.

2.1. Genetic Algorithm

Genetic Algorithm (GA) is one of the popular metaheuristic algorithms and received a huge number of attention from researchers compared to other type of metaheuristic optimisation approach [12]. GA manipulating a population of solutions by randomly searching the best feasible solution in the solution space, based on the mechanism of natural selection and natural genetics [13]. Figure 3 shows the flowchart of Genetic Algorithm.

Some highlight from previous research regarding ALB with GA including one of the earliest study from Falkenauer and Delchambre [14] where they introduce an efficient crossover and mutation operators for bin packing. Haq, Rengarajan, and Jayaprakash on the other hand incorporated a hybrid genetic algorithm approach that used the solution from the modified ranked positional method to solve mixed-model assembly line balancing problem [15] and recently Müller, Grunewald, and Spengler [16] utilize GA to maximise the production rate of robotic assembly line.

In previous work, new crossover operators has been proposed [11]. The crossovers named rank-based crossover type I and II (RBC-I and RBC-II) were proposed based on the ranking approach. The detail mechanism of RBC-I and II are presented in the earlier research [11]. For optimization in this paper, the RBC-I and II are compared with popular crossover operators for the combinatorial problem. The comparison crossovers are the ordered crossover (OX), partially matched crossover (PMX) and Moon crossover.

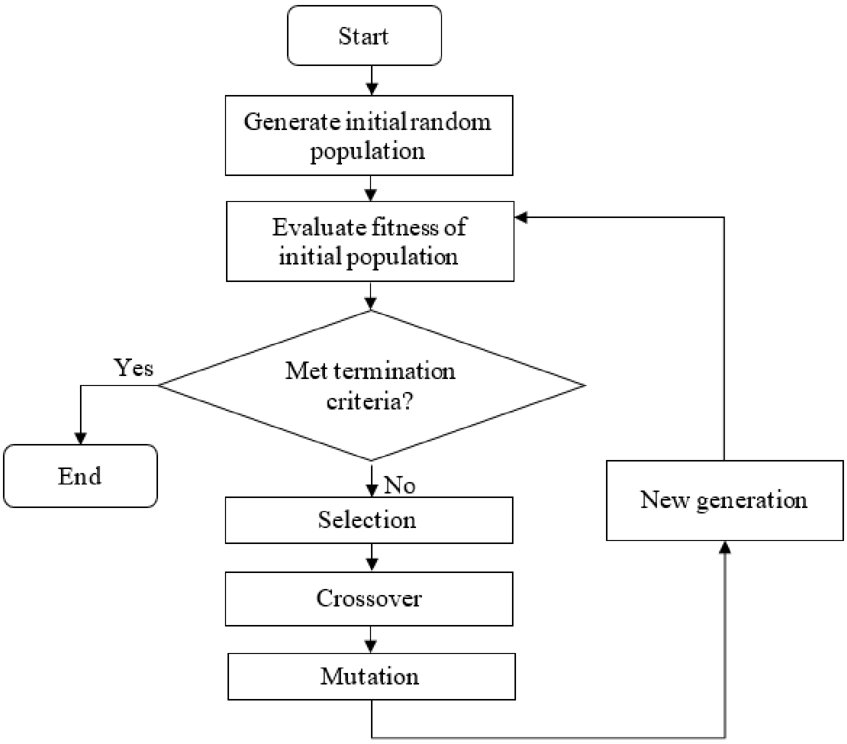

Figure 3. Flowchart of Genetic Algorithm

3. Results and Discussions

Optimisation for the case study problem has been conducted using Genetic Algorithm with different crossovers. For the case study optimisation, ten optimisation runs with different pseudo-random seeds have been done. The number of maximum generation is set to 300, while the probability of crossover (p_c) and mutation (p_m) are 0.8 and 0.2 respectively. Table 2 presents the fitness for the case study problem from ten different runs.

Table 2. Fitness value for the case study problem

No.	**Crossover type**				
	PMX	**Moon**	**OX**	**RBC-I**	**RBC-II**
Min	2.40	2.40	2.00	2.00	2.00
Max	2.61	2.60	2.80	2.60	2.41
Average	2.4614	2.4800	2.4414	2.3229	2.2014

Based on the optimization result for the case study problem in Table 2, the minimum fitness is 2.00, while the maximum fitness is 2.80. For the minimum fitness, three crossovers able to search for this solution. These crossovers were OX, RBC-I and RBC-II. Based on the average fitness value, the best crossover is the RBC-II. This is followed by RBC-I, OX, PMX and finally the Moon crossover. The optimization result for the case study problem indicated that the PMX and Moon crossovers have almost similar performance. Both crossovers are incapable to converge to minimum fitness. The OX crossover on the other hand able to search for optimum solution. However, the obtained fitness range was too diverged since the largest fitness value was also obtained by OX crossover. Meanwhile, the RBC-II has the best performance with 2.20 average fitness. In comparison with other crossover types, the RBC-II was also obtained the smallest maximum fitness value.

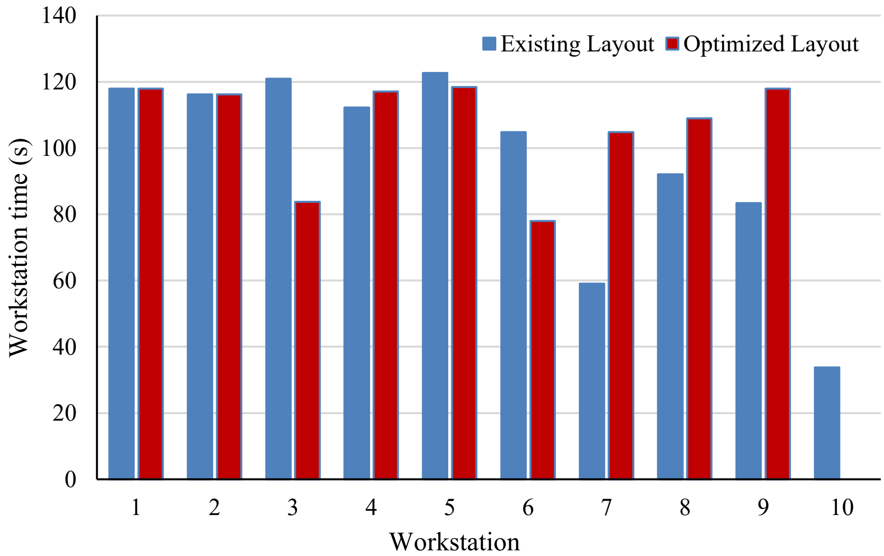

Figure 4. Comparison of workstation time

Figure 4 shows a comparison of workstation time for existing and optimized layout. Based on the bar graph, the optimized layout is more balanced within a smaller gap between minimum and maximum time compared with the existing layout. It indicated that the optimized layout has better (smaller) idle time compared with the existing layout. Based on this observation, the percentage of value added time utilization in the optimize layout will increase.

Besides that, the maximum workstation time for the optimised layout is slightly smaller than the existing layout. In the assembly line, the maximum workstation time is also known as cycle time for the assembly. The cycle time will control the production pace for the whole assembly line. With a smaller cycle time, the optimised layout is predicted to produce more output compared with the existing layout.

Besides the objective function that measure the number of workstation, machine and worker, a few other indicators to measure the line balance and efficiency can be used to compare the assembly layout before and after optimization. The comparison of existing and optimized assembly layout indicators is shown in Table 3.

Table 3. Comparison of existing and optimised assembly layout indicators

Indicator	Existing Layout	Optimised Layout
Number of workstation	10	9
Number of machine	20	17
Number of worker	10	9
Smoothness index	122.20	55.74
Line efficiency (%)	78.48	90.39

Based on the comparison of the existing and optimized layout, the number of workstation and worker were reduced about 10% from the existing configuration. Meanwhile, the number of machine also was reduced from 20 to 17 units. This is about 15% reduction from the existing layout. Besides that, the smoothness index and line efficiency for the optimized layout were also better than the existing. This is because the number of workstation and also the cycle time for the optimized layout is lower than the current layout. The cycle time refers to the maximum workstation time among all workstations. In the existing layout, the cycle time is 122.7 seconds, while in the optimized layout, the cycle time was reduced to 118.38 seconds. The cycle time will control the production pace in assembly line.

4. Conclusion

This paper presents an evaluation on rank-based crossovers (RBC) in GA to optimize real-life industrial problem. In previous work, RBC has been proposed and tested using generic benchmark problems. An industrial problem from electronics industry has been modelled as ALB-RC problem. Then the problem is optimized using GA with RBC, and being compared with the existing layout. The optimisation result indicated that the best solution provided by RBC-II able to reduce the number of workstation, machine and worker. In comparison with the existing layout configuration, the proposed solution by RBC-II was reduced the resources between 10 – 15%. At the same time, the assembly line efficiency was also increased. Therefore, it can be concluded that the ALB-RC model and GA with RBC able to optimize real-life industrial problem.

Acknowledgements
This research is funded by Universiti Malaysia Pahang under grant number RDU180331.

References

[1] M. Azizoğlu and S. İmat, “Workload smoothing in simple assembly line balancing,” *Comput. Oper. Res.*, vol. 89, pp. 51–57, 2018.
[2] I. Baybars, “An efficient heuristic method for the simple assembly line balancing problem,” *Int. J. Prod. Res.*, vol. 24, no. 1, pp. 149–166, 1986.
[3] J. Dong, L. Zhang, and T. Xiao, “A hybrid PSO/SA algorithm for bi-criteria stochastic line balancing with flexible task times and zoning constraints,” *J. Intell. Manuf.*, vol. 29, no. 4, pp. 737–751, 2018.

[4] M. Gansterer and R. F. Hartl, "One- and two-sided assembly line balancing problems with real-world constraints," *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 3025–3042, 2018.
[5] K. Ağpak, H. Gökçen, K. Ağpak, and H. Gökçen, "Assembly line balancing: Two resource constrained cases," *Int. J. Prod. Econ.*, vol. 96, no. 1, pp. 129–140, Apr. 2005.
[6] A. Corominas, L. Ferrer, and R. Pastor, "Assembly line balancing: General resource-constrained case," *Int. J. Prod. Res.*, vol. 49, no. 12, pp. 3527–3542, Jun. 2011.
[7] R. G. Özdemir and Z. Ayağ, "An integrated approach to evaluating assembly-line design alternatives with equipment selection," *Prod. Plan. Control*, vol. 22, no. 2, pp. 194–206, 2011.
[8] T. Koltai and V. Tatay, "Formulation of workforce skill constraints in assembly line balancing models," *Optim. Eng.*, vol. 14, no. 4, pp. 529–545, Nov. 2013.
[9] S. Jayaswal and P. Agarwal, "Balancing U-shaped assembly lines with resource dependent task times: A Simulated Annealing approach," *J. Manuf. Syst.*, vol. 33, no. 4, pp. 522–534, 2014.
[10] M. Jusop and M. Ab. Rashid, "Optimization of Assembly Line Balancing with Resource Constraint using NSGA-II : A Case Study," *Int. J. Appl. Eng. Res.*, vol. 12, no. 7, pp. 1421–1426, 2017.
[11] N. H. Kamarudin and M. F. F. Ab Rashid, "Assembly line balancing with resource constraints using new rank-based crossovers," in *Journal of Physics: Conference Series*, 2017, vol. 908, no. 1.
[12] M. R. Abdullah Make, M. F. F. Mohd Fadzil, and M. M. Razali, "A review of two-sided assembly line balancing problem," *International Journal of Advanced Manufacturing Technology*. 2017.
[13] H. Zhang, "An immune genetic algorithm for simple assembly line balancing problem of type 1," *Assem. Autom.*, vol. InPress, 2018.
[14] E. Falkenauer and A. Delchambre, "A genetic algorithm for bin packing and line balancing," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 1992.
[15] A. N. Haq, K. Rengarajan, and J. Jayaprakash, "A hybrid genetic algorithm approach to mixed-model assembly line balancing," *Int. J. Adv. Manuf. Technol.*, 2006.
[16] C. Müller, M. Grunewald, and T. S. Spengler, "Redundant configuration of robotic assembly lines with stochastic failures," *Int. J. Prod. Res.*, 2018.