System Improvement Through the Application of Assembly Line Balancing

Dyan C. Gonzales-Rodriguez
Assistant Professor of Industrial Engineering Department
College of Engineering
Bulacan State University
City of Malolos, Bulacan, Philippines
dyan.rodriguez@bulsu.edu.ph

Abstract

Assembly lines are characterized by the movement of the workpiece from one workstation to the next. An assembly line balancing problem consists of assigning the individual tasks to the workstations to optimize some appropriate measure of the line performance. If a line is balanced perfectly, all stations have an equal amount of work to perform, and a smooth product flow with no delay will be achieved. In this paper, two types of assembly line problems are analyzed. The first one is when the cycle time is given, and it needs to determine the minimum number of workstations. The second one is when the number of workstations is given, and it needs to determine the minimum cycle time. To determine the feasible and optimal solutions, the proponent uses software called ProPlanner. ProPlanner determines the possible solutions and assignment of tasks in each workstation that optimize the objective of the problem.

Keywords
Assembly lines, assembly line balancing, workstations, cycle time, ProPlanner

1. Introduction

Assembly lines are characterized by the movement of the workpiece from one workstation to the next. The individual tasks required to complete the product are divided and assigned to the workstation s so that each station performs the same operation on every unit of the product. The workpiece remains at each station for a duration of time called the cycle time. An assembly line balancing problem consists of assigning the individual tasks to the workstations to optimize some appropriate measure of the line performance. If a line is balanced perfectly, all stations have an equal amount of work to perform, and a smooth product flow with no delay should be achieved. We note that the overall design of a production line includes planning the proper sequence of operations, setting the production rate for the line, and balancing the load on the individual workstation. (Montgomery & Johnson)

1.1 Objectives

The proponent aims to balance the line for a cycle time of 15 minutes. It is assumed that there are equal probabilities assigned to available tasks. The assembly line balancing of a particular Company ABC is shown in Figure 1. The performance times, in minutes, for the tasks are shown above the nodes.

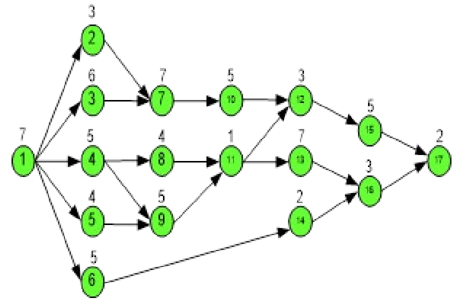

Figure 1. Assembly line of Company ABC

2. Literature Review

Assembly line balance (ALB) is a method of assigning work in an assembly line to achieve the desired production rate and achieve the desired goals. Tasks are distributed to workstations following precedence in an assembly line, ensuring that constraints such as task priority and performance indicators are met (Eghtesadifard, Halifeh, & Khorram, 2020). Assembly lines are specialized flow-line production systems that play a critical role in producing high-volume, standardized goods. Assembly lines have recently acquired popularity in the low-volume production of customized products (mass-customization). Practitioners should consider its configuration planning because of the significant capital costs of building or redesigning a line. As a result, numerous researchers have attempted to assist real-world configuration planning using appropriate optimization models (assembly line balancing problems) (Boysen, Fliedner, & Scholl, 2007).

Assembly lines are still a traditional method of mass and large-scale production. Since Henry Ford's days, several developments have shifted assembly lines from strictly paced and straight single-model lines to more flexible systems such as lines with parallel workstations or tasks, customer-oriented mixed-model and multi-model lines, U-shaped lines, and unpaced lines with intermediate buffers, among others. In any event, while (re-) setting an assembly line, a significant decision problem known as the assembly line balancing problem develops and must be resolved. The whole burden for manufacturing any unit of the product to be assembled is distributed across the workstations along the line. The simple assembly line balancing problem (SALBP) has historically been the focus of assembly line balancing research, which includes several constraining assumptions. To explain and solve more realistic generalized assembly line balancing problems (GALBP), many research works have been done recently. (GALBP) (Becker & Scholl, 2006). The paper of Scholl & Vob (1997) described Type 1 and Type 2 of the Simple Assembly Line Balancing Problem (SALBP). Type 1 of SALBP (SALBP-1) consists of assigning tasks to workstations such that the number of stations is minimized for a given production rate. In contrast, Type 2 (SALBP-2) maximizes the production rate or minimizes the sum of idle times for a given number of stations. In both problem types, precedence constraints between the tasks must be considered.

According to Sivasankaran & Shahabudeen (2014), the use of heuristics is unavoidable because this problem falls under the combinatorial category. It is also possible to create a mathematical model that will allow researchers to compare the heuristics' solutions to the models. Battaïa & Dolgui (2013) discussed that the fundamental problem, which was first proposed for manual assembly, has been expanded to robotics, machining, and disassembly contexts for decades. Despite the differences in industrial contexts and line configurations, mathematical models that are typically highly comparable, if not identical, have been devised.

In several papers, assembly line balancing problems were solved using different models. According to Peeters and Degraeve (2006), the simple assembly line balancing problem is a classical integer programming problem in operations research. A set of tasks, each one being an indivisible amount of work requiring a number of time units, must be assigned to workstations without exceeding the cycle time. They provide a new lower bound, namely the LP relaxation of a Dantzig–Wolfe decomposition-based integer programming formulation. To solve the issue, they suggest a column-generating algorithm. As a result, they devise a branch-and-bound algorithm to solve the price problem precisely. They evaluate the lower bound's quality by comparing it to other lower bounds and the best-known solution in the literature for specific circumstances. For most cases, computational studies indicate that the lower bound is equivalent to the best-known objective function value. Furthermore, the new lower bound based on LP can be used to verify optimality for an open problem.

This paper of Esmaeilbeigi, et. al. (2015) presents a mixed integer linear programming formulation for the type E simple assembly line balancing problem. In type E cycle time and number of stations are both decision variables, and the objective is to maximize the line efficiency. Furthermore, two augmentation strategies in the form of valid inequalities and auxiliary variables are proposed to strengthen the presented formulation further. Minimization of the number of stations, cycle time, and smoothness index are also investigated as supplementary aims of the challenge. Three distinct linearization approaches are used and compared for minimizing the smoothness index in the case of workload smoothing. The findings of a computational investigation on the benchmark data set show that the modified formulation is effective.

The work of Bautista and Pereira (2009) suggests a new method for solving the problem known as Bounded Dynamic Programming. This use of the term Bounded refers to both the use of bounds to reduce the state space and heuristics to reduce the state space. This approach can attain the best-known performance for the problem by obtaining an optimal solution rate of 267 out of 269 examples previously used in earlier publications. Even when employing shorter computing durations, these results outperform any previous approach discovered in the literature.

This research of Hazır and Dolgui (2013) deals with line balancing under uncertainty and presents two robust optimization models. Interval uncertainty for operation times was assumed. The proposed methods develop line patterns that are resistant to such interruptions. A decomposition-based method was devised and paired with augmentation tactics to solve large-size instances optimally. The effectiveness of this algorithm was investigated, and the results were provided. The innovative models provided in this research and the decomposition-based exact algorithm created are the paper's theoretical contributions. Furthermore, it is of practical importance since the assembly lines developed with our approach will have a higher output rate if uncertainty is factored in. Furthermore, this is a ground-breaking study of robust assembly line balance that should serve as the foundation for a decision-support system on the topic.

Balancing assembly lines is a crucial task for manufacturing companies in order to improve productivity and minimize production costs. Despite some progress in exact methods to solve large scale problems, software implementing simple heuristics are still the most used tools in industry. Some metaheuristics have also been presented and proved to improve conventional heuristics. Still, to their knowledge, no computer studies on simple industrial applications have been undertaken to assess their performance and adaptability fully. They offer a new tabu search algorithm and explore how it differs from others previously published. The system's performance is then assessed using the Type I assembly line balance problem. Finally, they put their method to the test on a real-world industrial data set with 162 tasks, 264 precedence constraints, and assembly on a sequential line with workstations on both sides of the conveyor, two possible conveyor heights, and no product repositioning. They discuss the flexibility of the metaheuristic and its ability to solve the real industrial case.

3. Methods

The following notations are used for the formulation of the programming model.

J_j = task for i= 1, 2, 3, …, 17
p_j = performance time of task i
X_{ij} = binary variable
k_0 = smallest number of workstation s theoretically available

The objective function of the model is to minimize the number of workstations. The objective function is given by:

$$min\, Z = \sum_{i=1}^{k}\sum_{j=1}^{17} C_{ij}X_{ij}$$

Where

$$C_{ij} = \begin{cases} p_j\left[\sum_{h=F} p_h + 1\right]^{(i-k_0-1)} & , i = k_0+1, \dots, k; j \in F \\ 0\, , otherwise & \end{cases}$$

This objective function uses more than k_o stations very costly and forces tasks to be assigned to the earliest possible station on the line. One unit of a later assignment is more expensive than the sum of all preceding assignments. Since at least the first k_o stations must be used, they need not be assigned a cost. Also, only tasks that have no followers need positive costs, as they may be placed last on the line.

This objective function is subject to the following set of constraints:

(1) All tasks must be performed:

$$\sum_{j=1}^{17} X_{ij}p_i \leq c, i = 1,2,.., k$$

(2)

$$1-\sum_{i=1}^{k} X_{ij} \geq 0, j = 1,2,\ldots,17$$

(3)

$$-17+\sum_{i=1}^{k}\sum_{j=1}^{17} X_{ij} \geq 0$$

(4) Precedence relations:

$$\sum_{i=1}^{k}(k-i+1)(X_{iu}-X_{iv}) \leq 0, (u,v) \in R$$

(5) Binary constraint:

$$X_{ij} = 0,1$$

$$X_{ij} = \begin{cases} 1\ if\ task\ J_j is\ assigned\ to\ station\ i, (i = 1,2,\ldots,k) \\ 0\ otherwise \end{cases}$$

(6) Nonnegativity assumption

$$all\ variables \geq 0$$

4. Data Collection

Table 1 shows the summary of precedence and performance time of tasks, as shown in Figure 1. To solve this model, ProPlanner is used instead of COMSOAL. The trial version of COMSOAL (Computer Method of Sequencing Operation for Assembly Line) requires a maximum of 15 tasks, but the problem involves 17 tasks. So to determine the solution to the problem, ProPlanner is used.

Table 1. Precedence and performance time of tasks

Tasks	Performance Time	Predecessors
1	7	None
2	3	1
3	6	1
4	5	1
5	4	1
6	5	1
7	7	2, 3
8	4	4
9	5	4, 5
10	5	7
11	1	8, 9
12	3	10, 11
13	7	11
14	2	6
15	5	12
16	3	13, 14
17	2	15, 16

5. Results and Discussion

5.1 Numerical and Graphical Results

In minimizing the number of workstations with a given cycle time of 15 minutes, the following tables (Tables 2 to 6) are the five possible tasks assignments for each workstation.

Figure 2, Table 2, and Figure 3 show the first line balancing with six workstations.

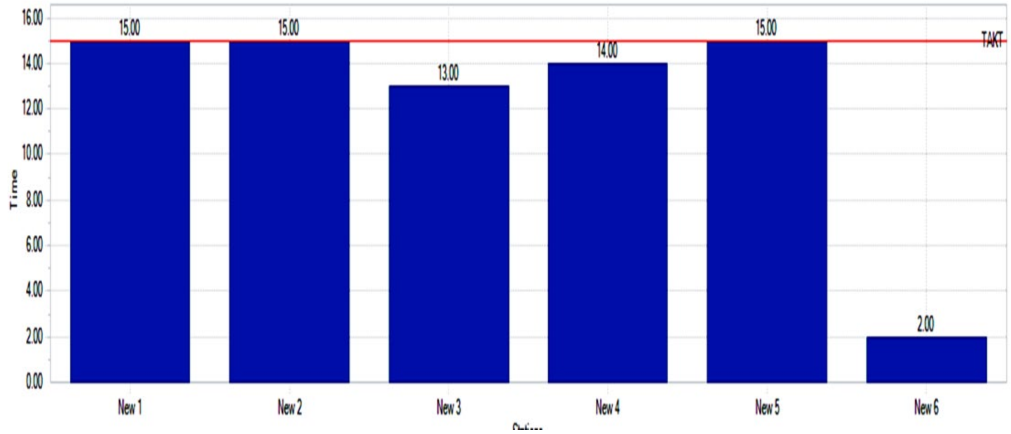

Figure 2. Line Balancing 1 Workstations

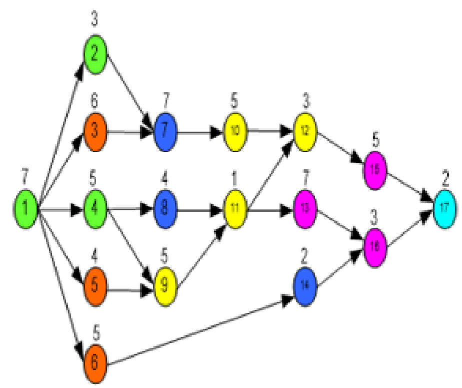

Figure 3. Precedence diagram for LB 1

Table 2. Line balancing 1

Work stations	Tasks	Performance Time	Cycle Time	Utilization
1	1	7	15	100%
	2	3		
	4	5		
2	3	6	15	100%
	6	5		
	5	4		
3	7	7	13	86.67%
	14	2		
	8	4		
4	9	5	14	93.33%
	10	5		
	11	1		
	12	3		
5	13	7	15	100%
	16	3		
	15	5		
6	17	2	2	13.33%
Average				82.22%

Figure 4, Table 3, and Figure 5 show the second line balancing with six workstations.

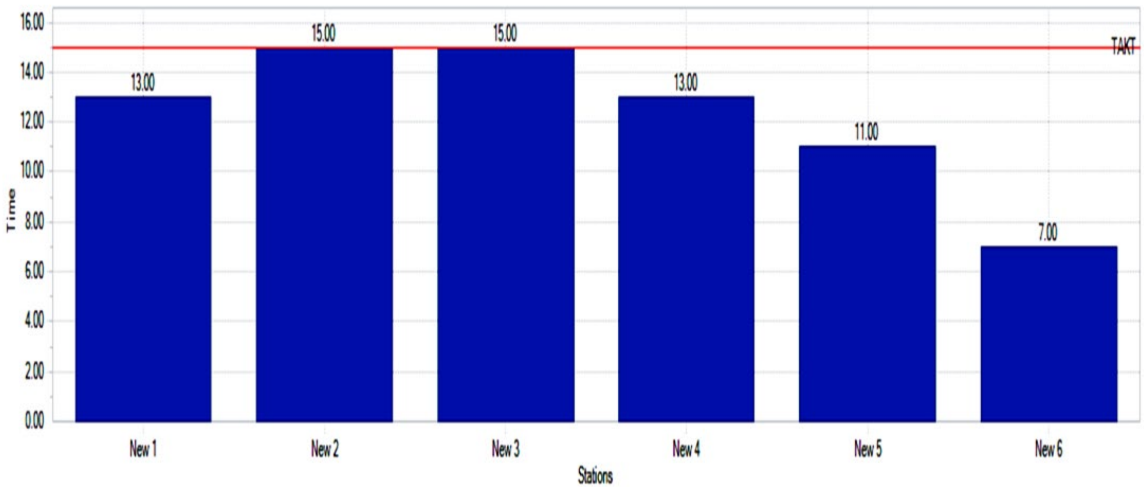

Figure 4. Line balancing 2 workstations

Table 3. Line balancing 2

Work stations	Tasks	Performance Time	Cycle Time	Utilization
1	1	7	13	86.67%
	3	6		
2	6	5	15	100%
	2	3		
	4	5		
	14	2		
3	5	4	15	100%
	7	7		
	8	4		
4	9	5	13	86.67%
	11	1		
	13	7		
5	16	3	11	73.33%
	10	5		
	12	3		
6	15	5	7	46.67%
	17	2		
Average				82.22%

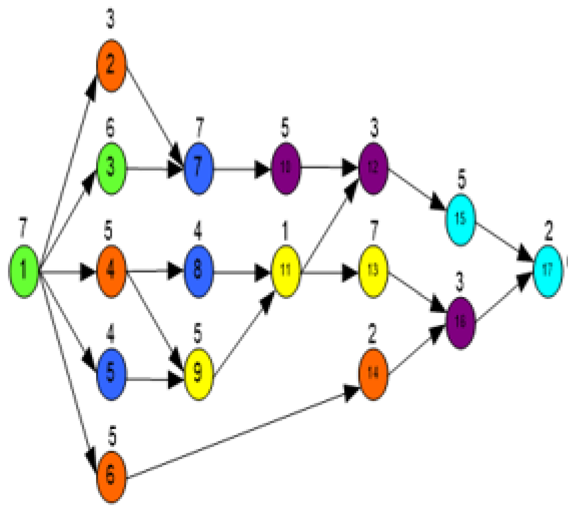

Figure 5. Precedence diagram for LB 2

Figure 6, Table 4, and Figure 7 show the third line balancing with six workstations.

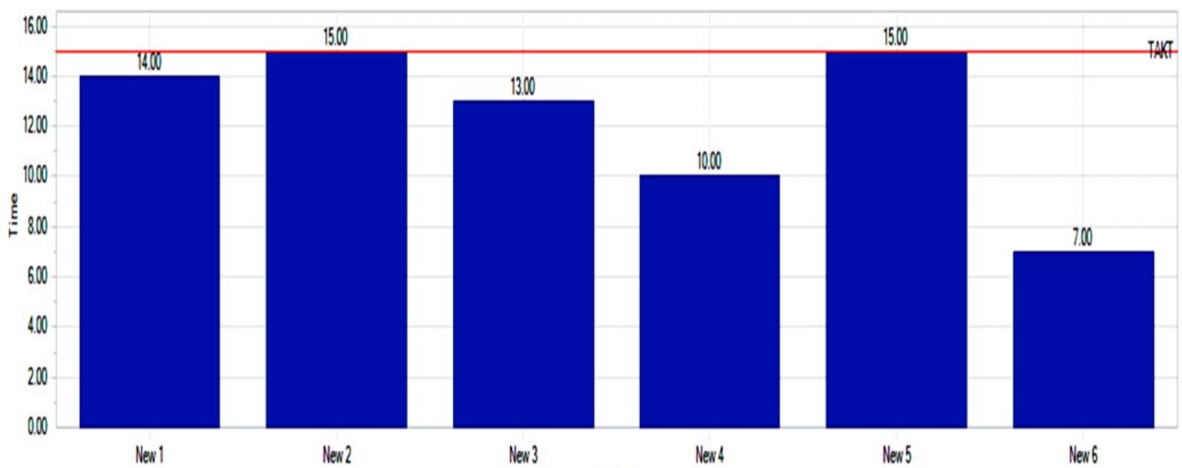

Figure 6. Line balancing 3 workstations

Table 4. Line balancing 3

Work stations	Tasks	Performance Time	Cycle Time	Utilization
1	1	7	14	93.33%
	2	3		
	5	4		
2	4	5	15	100%
	9	5		
	8	4		
	11	1		
3	3	6	13	86.67%
	6	5		
	14	2		
4	13	7	10	66.67%
	16	3		
5	7	7	15	100%
	10	5		
	12	3		
6	15	5	7	46.67%
	17	2		
Average				82.22%

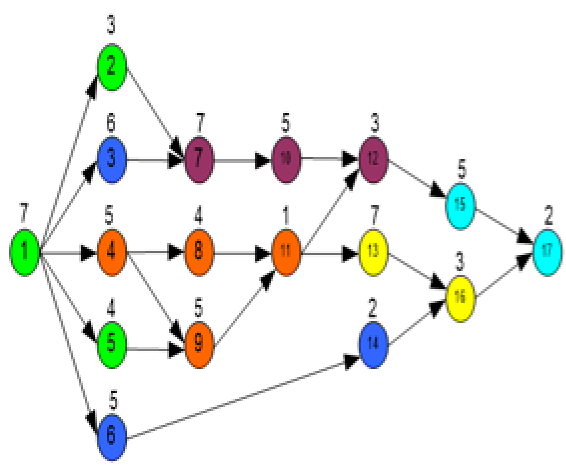

Figure 7. Precedence diagram for LB 3

Figure 8, Table 5, and Figure 9 show the fourth line balancing with six workstations.

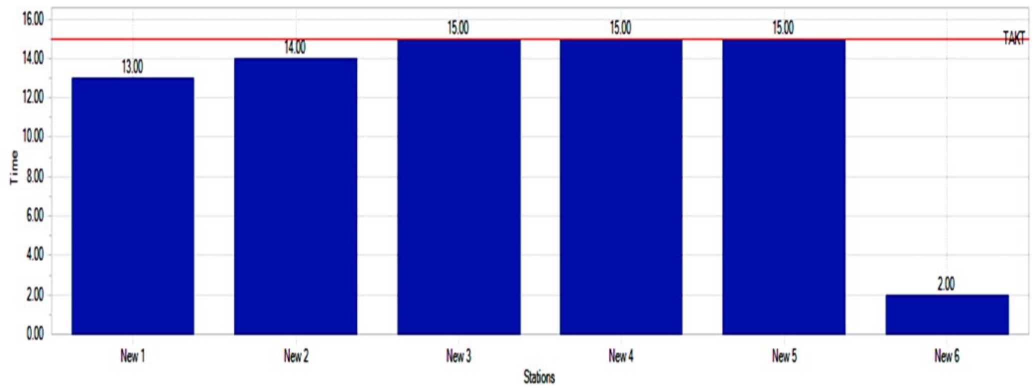

Figure 8. Line balancing 4 workstations

Table 5. Line balancing 3

Work stations	Tasks	Performance Time	Cycle Time	Utilization
1	1	7	13	86.67%
	3	6		
2	4	5	14	93.33%
	5	4		
	6	5		
3	2	3	15	100%
	9	5		
	7	7		
4	8	4	15	100%
	14	2		
	11	1		
	10	5		
	12	3		
5	13	7	15	100%
	16	3		
	15	5		
6	17	2	2	13.33%
Average				82.22%

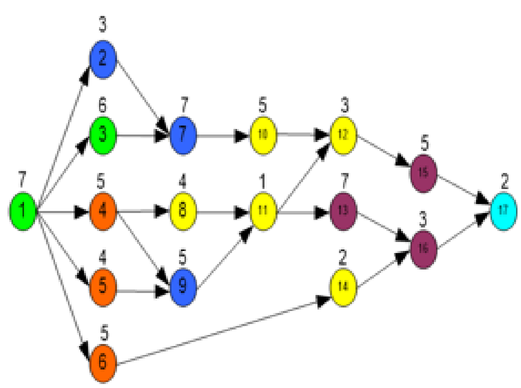

Figure 9. Precedence diagram for LB 4

Figure 10, Table 6, and Figure 11 show the fifth line balancing with six workstations.

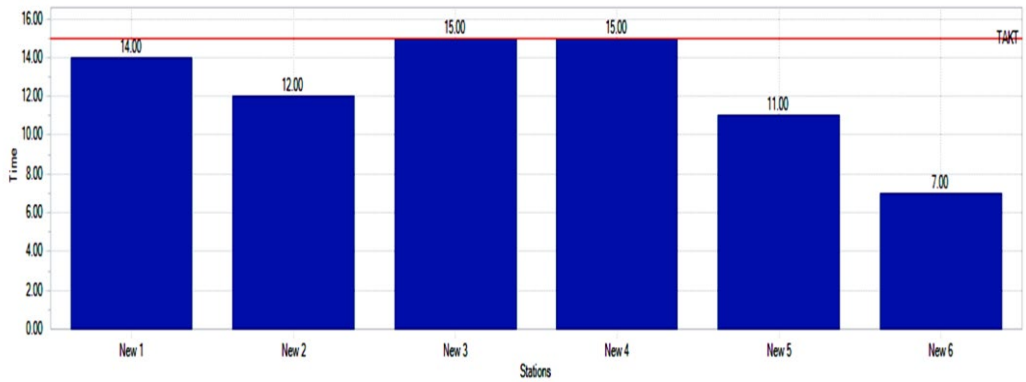

Figure 10. Line balancing 5 workstations

Table 6. Line balancing 5

Work stations	Tasks	Performance Time	Cycle Time	Utilization
1	1	7	14	93.33%
	6	5		
	14	2		
2	2	3	12	80%
	4	5		
	8	4		
3	3	6	15	100%
	5	4		
	9	5		
4	11	1	15	100%
	7	7		
	13	7		
5	16	3	11	73.33%
	10	5		
	12	3		
6	17	2	7	46.67%
	15	5		
Average				82.22%

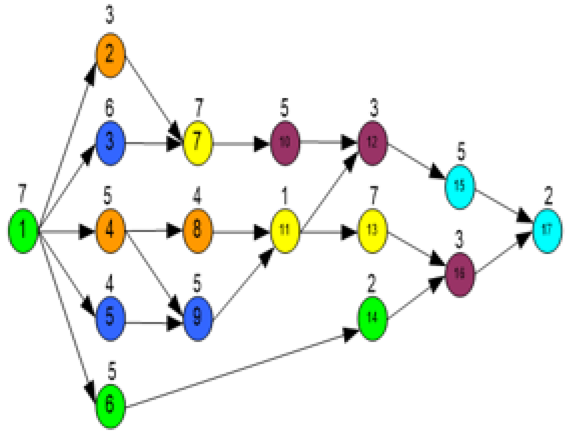

Figure 11. Precedence diagram for LB 5

Sensitivity Analysis

From the previous discussion, the cycle time is given. It wants to determine the minimum number of workstations and the assignment of tasks in each workstation that balances the assembly line. This time, the scenario will be reversed. The number of workstations is given and the minimum cycle time and the assignment of tasks in each workstation are determined. There are five working stations. Using the ProPlanner again, the tasks are assigned to five workstation s. Figure 12 shows the cycle time at each workstation, and Table 7 and Figure 13 show the assignment of tasks to each station. The cycle time is 16 minutes.

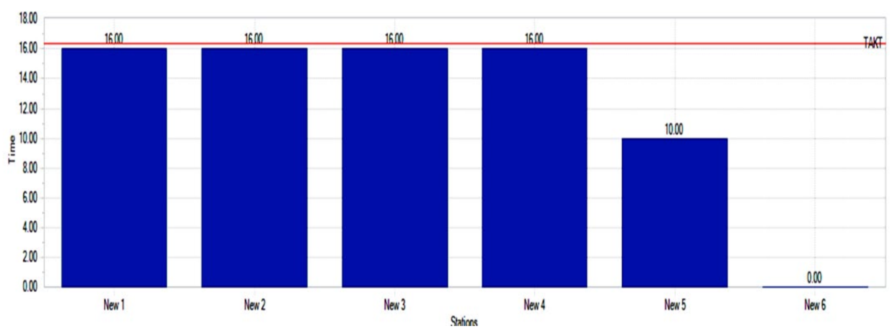

Figure 12. Line balancing of 5 workstations

Table 7. Line balancing for 5 workstations

Work stations	Tasks	Performance Time	Cycle Time	Utilization
1	1	7	16	100%
	3	6		
	2	3		
2	4	3	16	100%
	7	7		
	8	4		
3	10	5	16	100%
	6	5		
	14	2		
	5	4		
4	9	1	16	100%
	11	1		
	12	3		
	13	7		
5	16	3	10	62.5%
	15	5		
	17	2		
Average				92.5%

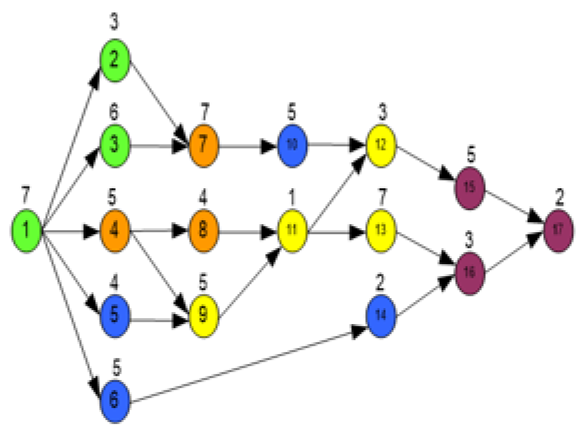

Figure 13. Precedence diagram for 5 workstations

5.2 Proposed Improvements

As the results from the ProPlanner given in the previous of this paper, it was found out that assigning tasks with a cycle time of 15 minutes gives five different sets of solutions. But all the solutions have the same average utilization of 82.22%, and all have the same number of workstations, which is six. The software, ProPlanner, makes it easier to determine different sets of task assignments that minimize the number of workstations.

Also, the ProPlanner determines the minimum cycle time if the number of workstations is given and gives a high utilization of each workstation. The minimum cycle time was 16 minutes, and the tasks were assigned to each workstation.

To distinguish easily what tasks are assigned to a specific workstation, precedence diagrams for each solution are presented with color codes.

5.3 Validation
The summary of the line balancing is given in Table 8.

Table 8. Summary of the line balancing

Line Balancing	No. of Workstations	Cycle Time (min)	Average Utilization
1	6	15	82.22%
2	6	15	82.22%
3	6	15	82.22%
4	6	15	82.22%
5	6	15	82.22%
Sensitivity Analysis	5	16	92.5%

6. Conclusion
Assembly lines are characterized by the movement of the workpiece from one workstation to the next. An assembly line balancing problem consists of assigning the individual tasks to the workstation s so that some appropriate measure of the line performance is optimized. If a line is balanced perfectly, all stations have an equal amount of work to perform, and a smooth product flow with no delay should be achieved.

In this paper, two types of assembly line problems are analyzed. The first one is when the cycle time is given, and it needs to determine the minimum number of workstation s. The second one is when the number of workstations is given, and it needs to determine the minimum cycle time.

Johnson and Montgomery's book entitled Operations Research in Production Planning, Scheduling and Control suggested using COMSOAL (Computer Method of Sequencing Operation for Assembly Line). But due to the unavailability of the original software, the trial version is insufficient to use because it solves only a maximum of 15 tasks. In the problem stated in this study, there are 17 tasks to be considered. To determine the feasible and optimal solutions, the proponent uses software called ProPlanner. ProPlanner determines the possible solutions to the problem as presented in this study.

Other software and method of balancing an assembly line can be used in this study. As a recommendation, future researchers may consider using different line balancing methods.

References
Battaïa, O., & Dolgui, A., A taxonomy of line balancing problems and their solution approaches, *International Journal of Production Economics,* vol. 142, no. 2, pp. 259-277, 2013.
Bautista, J., & Pereira, J., A dynamic programming based heuristic for the assembly line balancing problem, *European Journal of Operational Research,* vol. 194, no. 3, pp. 787-794, 2009.
Becker, C., & Scholl, A., A survey on problems and methods in generalized assembly line balancing, *European Journal of Operational Research,* vol. 168, no. 3, pp. 694-715, 2006.
Boysen, N., Fliedner, M., & Scholl, A., A classification of assembly line balancing problems, *European Journal of Operational Research,* vol. 183, no. 2, pp. 674-693, 2007.
Eghtesadifard, N., Halifeh, M., & Khorram, M., A systematic review of research themes and hot topics in assembly line balancing through the web of science within 1990–2017, *Computers and Industrial Engineering,* vol. 139, 2020.
Esmaeilbeigi, R., Nade, B., & Charkhgard, P., The type E simple assembly line balancing problem: A mixed integer linear programming formulation, *Computers & Operations Research,* vol. *64*, pp. 168-177, 2015.
Hazır, Ö., & Dolgui, A., Assembly line balancing under uncertainty: Robust optimization models and exact solution method, *Computers & Industrial Engineering,* vol. 65, no. 2, pp. 261-267, 2013.
Lapierre, S. D., Ruiz, A., & Soriano, P., Balancing assembly lines with tabu search, *European Journal of Operational Research,* vol. 168, no. 3, pp. 826-837, 2006.
Montgomery, D., & Johnson, L., *Operations Research in Production Planning, Scheduling and Control,* 1st Edition, John Wiley and Sons, Inc., New York, 1974.

Peeters, M., & Degraeve, Z., An linear programming based lower bound for the simple assembly line balancing problem, *European Journal of Operational Research,* vol. 168, no. 3, pp. 716-731, 2006.

Scholl, A., & Vob, S., Simple assembly line balancing—Heuristic approaches, *Journal of Heuristics*, pp. 217-244, 1997.

Sivasankaran, P., & Shahabudeen, P., Literature review of assembly line balancing problems, *The International Journal of Advanced Manufacturing Technology*, pp. 1665-1694, 2014.

Biography

Dyan C. Gonzales-Rodriguez is an Assistant Professor IV and former College Research and Development Unit Head of the College of Engineering at Bulacan State University, Philippines. She has been in the academe since 2011. She finished her Bachelor of Science degree in Industrial Engineering at Bulacan State University in 2009 and her Master of Science degree in Industrial Engineering at the University of the Philippines, Diliman, Quezon City, in 2017. She is conferred as a Professional Industrial Engineer by the Philippine Institute of Industrial Engineers and an ASEAN (Association of Southeast Asian Nations) Engineer by the ASEAN Federation of Engineering Organizations. She has presented several research outputs in different conferences sponsored by Asia Pacific Industrial Engineering and Management Society, Korean Convergence Society, Daehan Society of Industrial Management, and Industrial Engineering and Operations Research Society since 2014. She is an active member of the Philippine Institute of Industrial Engineers, Institute of Industrial and Systems Engineering, and International Association of Engineers. Aside from being a professor, she is also a consultant in various projects of government offices and local government units in the Philippines.