



HAL
open science

Assembly Line Worker Assignment and Balancing Problem with Positional Constraints

Hyungjoon Yang, Je-Hun Lee, Hyun-Jung Kim

► **To cite this version:**

Hyungjoon Yang, Je-Hun Lee, Hyun-Jung Kim. Assembly Line Worker Assignment and Balancing Problem with Positional Constraints. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France. pp.3-11, 10.1007/978-3-030-85914-5_1. hal-03897890

HAL Id: hal-03897890

<https://inria.hal.science/hal-03897890v1>

Submitted on 14 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Assembly line worker assignment and balancing problem with positional constraints

Hyungjoon Yang^[0000-0002-5166-382X], Je-Hun Lee^[0000-0001-8578-1237],
and Hyun-Jung Kim*^[0000-0001-7190-3264]

Korea Advanced Institute of Science and Technology (KAIST),
Daejeon 34141, Republic of Korea
*hyunjungkim@kaist.ac.kr

Abstract. One of the most important operational issues in the assembly line is to assign tasks and workers to stations while balancing the workload of the workers. We consider the assembly line worker assignment and balancing problem (ALWABP) where the process time of a task depends on worker skill levels. We also consider new positional constraints in the ALWABP to secure the working area of the workers. Two mathematical programming models are proposed to assign workers and tasks when new products are introduced and when a worker is absent or leaves a position temporary, respectively. A heuristic algorithm for the first model is proposed due to its extremely high complexity. The experimental results show the efficiency of the proposed methods. Finally, we explain how the proposed models can be used with a real-time dashboard of a digital twin.

Keywords: Assembly line, Line balancing, Worker skill level, Positional constraints, Digital twin

1 Introduction

An assembly line balancing problem (ALBP) which distributes tasks to workstations was introduced in 1955 [1]. The problem is known to be NP-hard due to its high complexity [2]. The early studies have focused on the ALBP to balance task times on each station while neglecting the heterogeneity of the workers. Later, many studies have considered practical environments, such as mixed-model assembly lines [3], multi-manned stations [4], and various shapes of assembly lines [5]. An assembly line worker assignment and balancing problem (ALWABP) further considers the different task time of each worker to the ALBP and assigns workers and tasks to stations [6]. Many studies have developed exact methods [7-9] and heuristic algorithms [10-12] for the ALWABP. ALWABP also has been

We address the ALWABP with new positional constraints which are required from one of the automotive parts manufacturing companies in Korea. Since multiple workers perform tasks in a workstation simultaneously, it is important to secure the working area of the workers. In addition, it is preferred not to assign tasks located far from each other to a worker [13]. These positional constraints are newly considered as well as task

precedence constraints. We propose two mathematical models, for the assignment, which can be used when new products are introduced and when a worker is absent or leaves a position temporary, respectively. Absenteeism is one of the significant factors that break the balance of assembly lines. Some workers are absent for the whole day while some others may leave the position for a while during work hours. It happens unexpectedly and is hard to predict in advance. Therefore, it is important to quickly and effectively respond to such a situation [14], and the second model we propose addresses this problem. We finally explain how the proposed models can be used in practice with a digital twin.

2 Problem definition

The problem we consider has been motivated from one of the automotive parts manufacturing companies in Korea. The company assembles a wiring harness which consists of wires, terminals, and connectors and runs throughout the entire vehicle. The wiring harness is assembled on a panel which moves on a conveyor belt. We assume the assembly line produces a single type of wiring harness. Fig. 1 shows assembly tasks on a panel and the precedence graph of the tasks, and Fig. 2 shows the assembly line we consider.

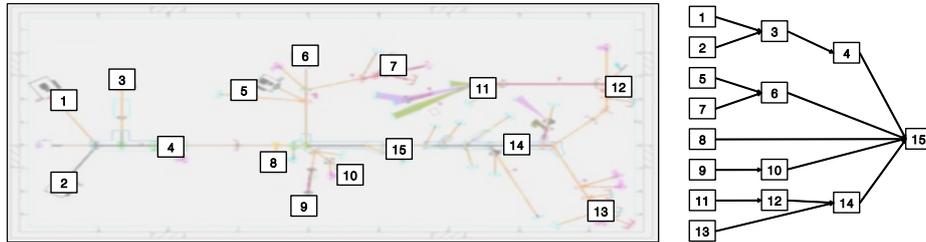


Fig. 1. An example of tasks and their precedence graph



Fig. 2. Photo of an assembly line

We consider two positional constraints required from the company. The first one is a task positional constraint which restricts the assignment of tasks located far from each other to a worker. The other one is a worker positional constraint where the working

areas of adjacent workers are not overlapped. The objective is to minimize the cycle time.

3 Solution Approach

3.1 Mathematical programming model for regular assignment

Table 1 below shows the indices, parameters and decision variables we use for the mathematical model.

Table 1. Notations of a mathematical model for regular assignment

Indices & Sets	
$h, i (\in I)$	Indices of tasks (Set of tasks)
$j (\in J)$	Index of stations (Set of stations)
$k (\in K)$	Index of workers (Set of workers)
P	Set of task pairs with precedence
P_t	Set of task pairs that are too far to be assigned to the same worker
P_w	Set of task pairs that are too close to be assigned to the adjacent worker
Parameters	
t_{ik}	Process time of task i when performed by worker k
M	Total number of tasks (used as a sufficiently big number)
Decision Variables	
x_{ijk}	1 if task i is performed by worker k at station j , otherwise 0
y_{jk}	1 if worker k is assigned to station j , otherwise 0
C	Cycle time of the line

$$\min C \quad (1)$$

Subject to:

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \quad \forall i \in I \quad (2)$$

$$\sum_{j \in J} y_{jk} = 1, \quad \forall k \in K \quad (3)$$

$$\sum_{k \in K} y_{jk} = 1, \quad \forall j \in J \quad (4)$$

$$\sum_{j \in J} \sum_{k \in K} j x_{hjk} \leq \sum_{j \in J} \sum_{k \in K} j x_{ijk}, \quad \forall (h, i) \in P \quad (5)$$

$$\sum_{i \in I} \sum_{j \in J} t_{ik} x_{ijk} \leq C, \quad \forall k \in K \quad (6)$$

$$\sum_{i \in I} x_{ijk} \leq M y_{jk}, \quad \forall j \in J, k \in K \quad (7)$$

$$\sum_{j \in J} (x_{hjk} + x_{ijk}) \leq 1, \quad \forall k \in K, (h, i) \in P_t \quad (8)$$

$$\sum_{k \in K} (x_{h(j-1)k} + x_{ijk}) \leq 1, \quad \forall j \in J \setminus \{1\}, (h, i) \in P_w \quad (9)$$

This formulation derives the optimal assignment of tasks and workers to reduce the cycle time. Objective function (1) and Constraints (2)-(7) are the basic formulation of the ALWABP. Constraint (2) indicates that each task is assigned to exactly one station. Constraints (3) and (4) mean that each worker should be assigned to one station and each station should have only one worker, respectively. Constraint (5) is used for the precedence relation of the tasks. Constraint (6) computes the cycle time which takes the maximum value among the completion times of all workers, and Constraint (7) makes sure that tasks are assigned to a worker and a station when the worker is assigned to the station. Constraints (8) and (9) are the task positional constraint and worker positional constraint, respectively. After obtaining a solution, the workstations are then assigned to panels evenly.

3.2 Mathematical programming model for task reassignment

We now propose another mathematical model that adjusts the initial assignment obtained in Section 3.1. When a worker is absent at the beginning of the day or leaves a position during work hours, the tasks assigned to the worker should be reassigned. The following notations and formulation are used for this case. We note that x_{ijk} and y_{jk} used in this section are obtained from the model in Section 3.1.

Table 2. Notations of a mathematical model for task reassignment

Set	
I_U	Set of unassigned tasks (tasks that should be reassigned)
Parameter	
k_a	Absent worker
Decision Variable	
a_{ik}	1 if reassign task i to worker k , otherwise 0

$$\min C \quad (10)$$

Subject to:

$$\sum_{k \in K \setminus \{k_a\}} a_{ik} = 1, \quad \forall i \in I_U \quad (11)$$

$$\sum_{j \in J} \sum_{k \in K} j a_{hk} y_{jk} \leq \sum_{j \in J} \sum_{k \in K} j x_{ijk}, \quad \forall (h, i) \in P, h \in I_U, i \notin I_U \quad (12)$$

$$\sum_{j \in J} \sum_{k \in K} j x_{hjk} \leq \sum_{j \in J} \sum_{k \in K} j a_{ik} y_{jk}, \quad \forall (h, i) \in P, h \notin I_U, i \in I_U \quad (13)$$

$$\sum_{j \in J} \sum_{k \in K} j a_{hk} y_{jk} \leq \sum_{j \in J} \sum_{k \in K} j a_{ik} y_{jk}, \quad \forall (h, i) \in P, h, i \in I_U \quad (14)$$

$$a_{ik} + \sum_{j \in J} x_{hjk} \leq 1, \quad \forall (h, i) \in P_t, k \in K, i \in I_U \quad (15)$$

$$\sum_{k \in K} (a_{hk} y_{(j-1)k} + a_{ik} y_{jk}) \leq 1, \quad \forall (h, i) \in P_w, j \in J \setminus \{1\}, h, i \in I_U \quad (16)$$

$$\sum_{i \in I} \sum_{j \in J} t_{ik} (a_{ik} y_{jk} + x_{ijk}) \leq C, \quad \forall k \in K \setminus \{k_a\} \quad (17)$$

Constraint (11) reassigns the tasks of the absent worker. Constraints (12), (13) and (14) are used for the precedence relation of tasks after the reassignment. Constraints (15) and (16), respectively, are for the task positional constraint and worker positional constraint after the reassignment. Constraint (17) calculates the cycle time.

3.3 Heuristic algorithm for regular assignment

Since it takes a long time to obtain an optimal solution of the model in Section 3.1, a heuristic algorithm is proposed. We note that an optimal solution of the model in Section 3.2 is easily obtained within a short period of time. The proposed algorithm consists of the initial solution generation step and improvement step with local search.

First, the tasks with no predecessor are randomly selected and assigned to the first station. The number of tasks assigned to each station is either $\lfloor |I|/|k| \rfloor$ or $\lceil |I|/|k| \rceil$. Similarly, the remaining tasks are also randomly assigned to the stations by considering the precedence constraints and positional constraints. After that, workers who can finish the tasks in a station at the earliest time are assigned. Then an initial solution is obtained.

A local search process is applied to the initial solution with task shift, task swap, and worker swap operators to reduce the cycle time. If there is any improvement after applying the three operators, the current best solution is updated. Otherwise, a new neighbor solution is obtained from the current solution by randomly changing the task assignment regardless of the solution quality. Then the local search process is again applied to the neighbor solution. This procedure is repeated until a give number of iterations or the given maximum computation time.

4 Computational results

The two optimization models in Sections 3.1 and 3.2 were programmed and solved by using OR-Tools, an open source software for optimization. The maximum number of iterations of the heuristic algorithm is set to 10,000, and the time limit is 10 minutes. Experiments are performed by comparing the proposed heuristic algorithm to an optimal solution for initial assignment and by comparing the reassignment result of the proposed mathematical model to the current practice. Table 3 shows the comparison result of the model solved with OR-Tools and the proposed heuristic algorithm for regular assignment. Table 4 shows the effectiveness of the model for task reassignment by

comparing the result with the current practice. The average cycle time (CT) and computation time with 30 instances are presented in the tables.

In Table 3, the model for regular assignment found optimal solutions with up to 10 tasks with OR-Tools. When the number of tasks is more than 20, it is not able to provide optimal solutions within one hour. The best cycle time found so far is written in the CT column in the table. We can observe that the proposed heuristic algorithm provides optimal solutions with 10 tasks and 5 workers. In addition, it gives a better solution than OR-Tools with more than 40 tasks and 10 workers when there is a time limit of one hour. The computation time of the heuristic algorithm is less than 2 minutes. This is enough for practical use because the heuristic algorithm is required to run in the early morning of the day when a new product is assembled, which does not occur frequently.

In Table 4, the current practice indicates the results from the simple rule used in the company. The operations manager reassigns tasks based on his/her experience in the company. We can see that the proposed model can reduce the cycle time at most 17.5% with 100 tasks and 20 workers. The computation time of the optimization model with OR-Tools is within 10 seconds even for the large-sized instances.

Table 3. Computation results for OR-Tools and heuristic algorithm

Number of Tasks	Number of Workers	Optimization model		Heuristic algorithm	
		CT(s)	Computation time(s)	CT(s)	Computation time(s)
7	3	378*	< 1	378*	< 1
10	5	341*	35	341*	< 1
20	7	524	> 3,600	524	8
40	10	714	> 3,600	703 (-1.5%)	28
100	20	910	> 3,600	874 (-3.9%)	110

* : optimal solution

Table 4. Comparison of the proposed model and current practice for task reassignment

Number of Tasks	Number of Workers	Current practice	Optimization model
		CT(s)	CT(s)
10	5	555	543 (-2.1%)
20	7	739	679 (-8.1%)
40	10	951	833 (-12.4%)
100	20	1,249	1,031 (-17.5%)

5 Applications to practice

With the advancement of IoT, it is possible to identify the real-time position of workers and to detect whether a worker is away from the assembly line and returns to the line.

In the company we consider, the workers wear a bracelet with a sensor, and the real-time position of each worker is collected by a beacon system.

Our proposed method is implemented in the digital twin which collects real-time data and monitors the current operations state. Fig. 3 shows a scenario of our method applied with a digital twin. All the information about the current position of workers and their task assignment are described in the dashboard of a digital twin. Step 1 in Fig. 3 shows the optimal worker-task assignment when all workers are present in the assembly line. When a worker leaves his/her position for a certain period, absenteeism is automatically recognized (Step 2 in Fig. 3) and the proposed model in Section 3.2 is run with the real-time information of the task assignment, remaining workers, and tasks requiring the reassignment. Then an optimal solution to minimize the cycle time is obtained with a commercial solver and displayed in the dashboard with the modified conveyor belt speed and production rate (Step 3 in Fig. 3). The conveyor belt then decreases its speed according to the solution obtained. When the worker returns to the assembly line, the initial assignment is again applied (Step 4 in Fig. 3).

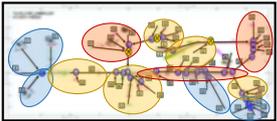
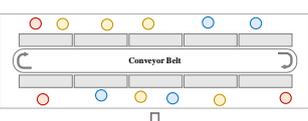
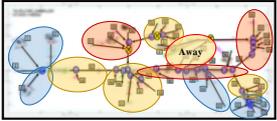
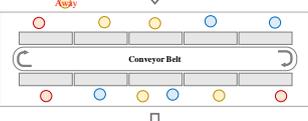
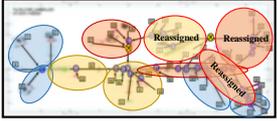
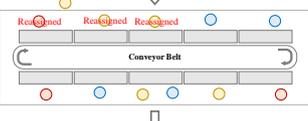
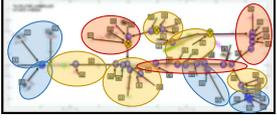
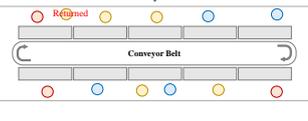
	Current worker-task assignment	Current worker position	Details	
Step 1			Belt Speed	2.4 m/min
			Production Rate	40 units/hour
Step 2			Belt Speed	2.4 m/min
			Production Rate	40 units/hour
Step 3			Belt Speed	2.2 m/min
			Production Rate	37 units/hour
Step 4			Belt Speed	2.4 m/min
			Production Rate	40 units/hour

Fig. 3. Use of proposed methods in practice

6 Conclusion

We have addressed the ALWABP with new positional constraints to minimize the cycle time. Two mathematical programming model and one heuristic algorithm have been proposed, and the experimental results have shown their effectiveness. We have also provided a scenario of using the proposed model with a digital twin by detecting the real-time position of workers. We further need to compare the digital twin result with the actual result, to develop exact algorithms for the regular assignment, to consider the

work efficiency of an individual operator, and to develop AI models for solving the problem.

References

1. Salvesson, M. E.: The assembly line balancing problem. *Journal of Industrial Engineering* 6(3), 18–25 (1955).
2. Gutjahr, A. L., Nemhauser, G. L.: An algorithm for the line balancing problem. *Management Science* 11(2), 308–315 (1964).
3. Thomopoulos, N. T.: Line Balancing-Sequencing for Mixed-Model Assembly. *Management Science* 14(2), B59-B75 (1967).
4. Fattahi, P., Roshani, A., Roshani, A.: A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *International Journal of Advanced Manufacturing Technology* 53(1-4), 363-378 (2011).
5. Miltenburg, G. J., Wijngaard, J.: The U-line Line Balancing Problem. *Management Science* 40(10), 1378-1388 (1994).
6. Miralles, C., García-Sabater, J. P., Andrés, C., Cardos, M.: Advantages of assembly lines in sheltered work centres for disabled. A case study. *International Journal of Production Economics* 110(1-2), 187–197 (2007).
7. Vilà, M., Pereira, J.: A branch-and-bound algorithm for assembly line worker assignment and balancing problems. *Computers & Operations Research* 44, 105-114 (2014).
8. Borba, L., Ritt, M.: A heuristic and a branch-and-bound algorithm for the Assembly Line Worker Assignment and Balancing Problem. *Computers & Operations Research* 45, 87-96 (2014).
9. Li, Z., Kucukkoc, I., Tang, Q.: Enhanced branch-bound-remember and iterative beam search algorithms for type II assembly line balancing problem. *Computers & Operations Research* 131, 105235 (2021). doi: 10.1016/j.cor.2021.105235
10. Blum, C., Miralles, C.: On solving the assembly line worker assignment and balancing problem via beam search. *Computers & Operations Research* 38, 328-339 (2011).
11. Mutlu, Ö., Polat, O., Supciller, A. A.: An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II. *Computers & Operations Research* 40, 418–426 (2013).
12. Moreira, M. C. O., Ritt, M., Costa, A., Chaves, A. A.: Simple heuristics for the assembly line worker assignment and balancing problem. *Journal of Heuristics* 18(3), 505-524 (2012).
13. Gansterer, M., Hartl, R. F.: One- and two-sided assembly line balancing problems with real-world constraints. *International Journal of Production Research* 56(8), 3025-3042 (2018).
14. Ritt, M., Costa, A. M., Miralles, C.: The assembly line worker assignment and balancing problem with stochastic worker availability. *International Journal of Production Research* 54(3), 907-922 (2016).