

Time and Space Multi-Manned Assembly Line Balancing Problem Using Genetic Algorithm

Nessren Zamzam , Ahmed Elakkad 

Ain shams university (Egypt)

nessren.zamzam@eng.asu.edu.eg, ahmed.elakkad@eng.asu.edu.eg

Received: March 2021

Accepted: September 2021

Abstract:

Purpose: Time and Space assembly line balancing problem (TSALBP) is the problem of balancing the line taking the area required by the task and to store the tools into consideration. This area is important to be considered to minimize unplanned traveling distance by the workers and consequently unplanned time waste. Although TSALBP is a realistic problem that express the real-life situation, and it became more practical to consider multi-manned assembly line to get better space utilization, few literatures addressed the problem of time and space in simple assembly line and only one in multi-manned assembly line. In this paper the problem of balancing bi-objective time and space multi-manned assembly line is proposed

Design/methodology/approach: Hybrid genetic algorithm under time and space constraints besides assembly line conventional constraints is used to model this problem. The initial population is generated based on conventional assembly line heuristic added to random generations. The objective of this model is to minimize number of workers and number of stations.

Findings: The results showed the effectiveness of the proposed model in solving multi-manned time and space assembly line problem. The proposed method gets better results in solving real-life Nissan problem compared to the literature. It is also found that there is a relationship between the variability of task time, maximum task time and cycle time on the solution of the problem. In some problem features it is more appropriate to solve the problem as simple assembly line than multi-manned assembly line.

Originality/value: It is the first article to solve the problem of balancing multi-manned assembly line under time and area constraint using genetic algorithm. A relationship between the problem features and the solution is found according to it, the solution method (one sided or multi-manned) is defined.

Keywords: assembly line, multi-manned, time and space assembly line, genetic algorithm

To cite this article:

Zamzam, N., & Elakkad, A. (2021). Time and space multi-manned assembly line balancing problem using genetic algorithm. *Journal of Industrial Engineering and Management*, 14(4), 733-749. <https://doi.org/10.3926/jiem.3542>

1. Introduction

Assembly Line Balancing (ALB) has a vital role in current manufacturing system. It is essential to design and balance the line efficiently as its installation requires large capital investments and last for long time (Make, Rashid

& Razali, 2017). Assembly line balancing problem (ALBP) is the problem of finding the best sequence for assigning tasks to station without violating the precedence constraint and the cycle time while optimizing certain objective/s. These lines were presented by Henry Ford and in 1955, Salveson was the first one who formulated this problem in a research (Micieta & Stollmann, 2011) since then it became the matter of concern for researchers for more than 65 years (Eghtesadifard, Khalifeh & Khorram, 2020) due to the increase in the complexity of the products and the advancement in technology that arises new constraints and requires new solution approach. Many researchers have categorized the ALBP into different categories according to the line type, number of products to be assembled, layout of the line and the objective function as (Saif, Guan, Wang, Mirza & Huang, 2014; Hazır, Delorme & Dolgui, 2015).

Literature also classified assembly lines into three types of lines: single side, two-sided and multi-manned assembly line Figure 1. Single sided assembly line is the line that there is only one worker in each station in one side. Two-sided assembly line, there is at most one worker in each side of the line makes different task simultaneously on the product, one on the left and the other on the right side. While multi-manned assembly line (MAL) more than one worker can be assigned to each station according to the product structures (Fattahi, Roshani & Roshani, 2011) The maximum number of operators is decided by the decision maker according to the product size, tools availability, workstation design... According to Dimitriadis (2006) the major advantage of MAL over SAL: it can reduce line length, decrease work in process, and workers in same station can share tools or fixtures which in turn save time and cost.

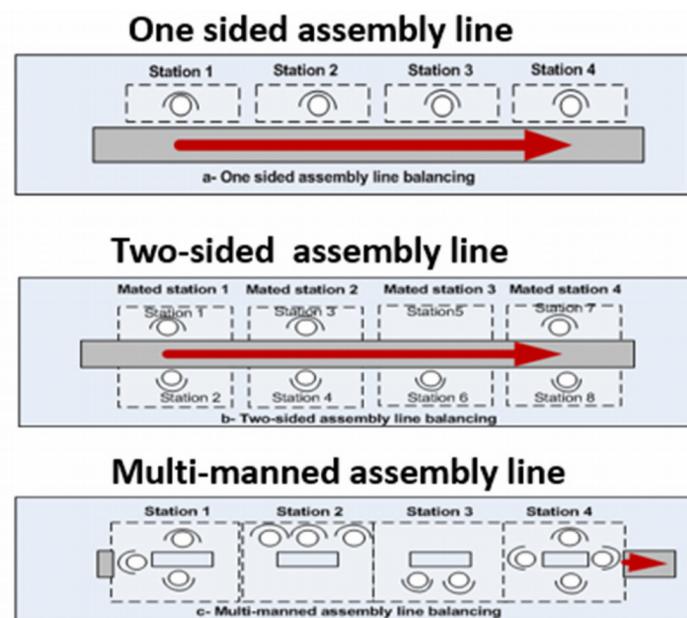


Figure 1. Types of assembly line

Although multi manned assembly line is popular in industry, only few numbers of literatures concentrate on it. Dimitriadis (2006) was the first one to present multi-manned assembly line problem, with the objective of achieving a better space utilization with the same number of workers and the same total idle time of the line. Kellegoz and Toklu (2012) addressed the problem of assembly line that has multi-manned stations parallel to each other using branch and bound algorithm called Jumper to solve the problem optimally. The algorithm succeeded in finding the optimal solution in small sized problem but fail to find them in the big sized problem. Cevikan, Durmusoglua and Unal (2009) proposed a mathematical programming model for mixed model assembly lines to create multi manned station. The objective was to balance the time load between workers for the different model. A scheduling-based heuristic was proposed for this objective. Kazemi and Sedighi (2013) proposed a genetic algorithm to model the MAL balancing problem. The proposed model aimed to maximize utilization of the resources and minimize the cost of production. Two alternatives were considered in the initial population:

1) randomly generated initial population; 2) random population with results of seven proposed heuristics. The result showed that using seven heuristics with the random population performs better than seven heuristic alone and the random initial population. priority rule-based genetic. Chen, Cheng and Li (2018) aimed to minimize the number of workers and stations in assembly line by using simulated annealing algorithm. Li, Wang and Yang (2018) used simulated annealing to solve Type 2 MALBP with the objective of minimizing cycle time. Ferrari, Faccio, Gamberi, Margelli and Pilati (2019) proposed a new mixed integer programming model for “multi-manned assembly line balancing problem” with the objectives of optimizing the line length, the line efficiency and the workload smoothness. A simulated annealing algorithm was then developed to efficiently solve the problem. Yadav and Agrawal (2019) proposed a new mathematical model to solve the “multi- parallel two-sided assembly line problem” with the objective of minimizing the cost of tools in the station and minimizing the total idle time.

Researchers has tackled the ALBP from different perspective as constraints and optimization techniques. There was a variety of **optimization techniques** that concerned with ALBP such as exact and heuristic, and meta-heuristic methods (Eghtesadifard et al., 2020). Exact methods include branch and bound, mathematical model and dynamic programming (Bautista, & Pereira, 2009; Li, Kucukkoc & Zhang, 2018). Exact methods can solve the problem optimally and give an exact solution. However, the exact methods could find the optimal solution for small, and medium sized problems only. Researchers start to seek for heuristic and meta heuristic techniques for solving large sized problem. From the Meta-heuristic optimization techniques are simulated annealing, genetic algorithm, ant colony, tabu search... Taha, El-Kharbotly, Sadek and Afia (2011), and Alavidoost, Zarandi, Tarimoradi and Nemati (2017) used genetic algorithm. Li, Janardhanan, Nielsen and Tang (2018) and Roshani, Roshani, Roshani, Salehi and Esfandyari (2013) used simulated annealing, Roshani and Giglio (2020) used Tabu search.

From the constraint perspective, the most serious constraint that affect the ALBPs are assignment constraints which are occurrence, cycle time and precedence constraint (Battaia & Dolgui, 2013). Occurrence constraints ensure that each task is assigned to only one station, cycle time constraint ensure that the cycle time of each station does not violating the cycle time of the line and the precedence constraint make sure that the precedence relationship of each task is respected. By the time more realistic constraint that express real life situation has been taken in consideration. Constraints as positive and negative zoning constraint which ensures that some sets of tasks are assigned or forbid to assign to the same workstation respectively (Purnomo et al., 2013). Resource constraint is the constraint that takes the space of the assembly line into consideration and time and space constraint is the one that takes the space required by the tasks and place to store the tools into consideration (Chica, Cordon, Damas & Bautista, 2010).

Bautista and Pereira (2007) were the first one tackled time and space constraint in the assembly line balancing problem. They used ant colony optimization technique to solve the problem. They categorized the TSALBP into eight categories according to the variable depending on the variant number of stations, cycle time and available area. Moreover in 2011 Bautista and Pereira solved “time and space assembly line balancing problem” by proposing an adjustment to the Bounded Dynamic Programming (BDP) method (Bautista & Pereira, 2011). They developed several bounding and solution procedures to tackle the problem. Then Chica et al. (2010) approached the 1/3 TSALBP objectives variant. They used ant colony optimization and random greedy research technique to optimize the area and number of stations. In 2011, they studied the presence of preferences based on domain knowledge to tackle the TSALBP-1/3, both in the objective and decision spaces. Next, meta-heuristics were used to tackle the balancing problem of time and space in assembly line, such as ant colony algorithm (Rada-Vilela , Chica, Cordon & Damas, 2013), memetic algorithm (Chica, Cordon, Damas & Bautista, 2012) and genetic algorithm (Chica, Cordon & Damas, 2011).

Zhang, Tang and Chica (2020) was the first one to tackle the time and space assembly line balancing problem with multi-manned workstations. At first, the problem of balancing “multi-manned assembly line with time and space constraint” is formulated. Then a new mixed-integer linear mathematical model (MILP) is presented, and finally a “memetic ant colony optimization” is created to solve the problem.

From the previous literature review it can be concluded that multi-manned assembly line is very common in real life while few researchers seek to solve this problem. The few researchers use different optimization technique, however limited one of them who consider a real-life constraint. It also can be concluded that despite time and

space constraint is considered an important constraint that express the real situation, few researchers try to solve this problem. According to Zhang, Tang and Chica (2020), there is a lack of research on the multi-manned assembly line balancing problem with time and space constraints. He also mentioned that the new problem of multi- manned assembly line with time and space is necessary and realistic in modern assembly enterprises and it work in line with the production of real enterprise. Therefore, the purpose of this paper is to address “multi-manned assembly line balancing problem with time and space constraint” using genetic algorithm with the objective of minimizing number of workers and number of stations. In this work, the initial population is generated using four conventional assembly line heuristics added to the randomly generated initially population.

The organization of the paper is as follows; section two states the problem definition, the methodology and proposed genetic algorithm is in section three, the results and discussion are in section four and finally the conclusion.

2. Problem Definition and Assumptions

In multi-manned assembly line balancing problem, there are more than one worker perform different tasks on the same product simultaneously in the same workstation Figure 2. Since these lines are applied to relatively from medium to large products, the components, and tools range in size from small to large parts. The required components and tools should be available along the sides of the line. It is essential to let them as near as possible to the worker and the workstation. If two tasks are assigned to the same station and each one of them require a large component and big tool to be assembled, the tools and the part may be set far away from the workstation and the worker must exert more effort to get the component or the tool. Thus, it more time economical to take the area of the tools and parts into consideration. In MAL with area and time constraint problem, it is required to assign task (t) to a certain worker in a certain station (S) such that the total time of tasks assigned to each worker (W) does not violate cycle time (CT) and precedence relationship ($P_{i,j}$). It is also required that the total area required by the tasks (a_i) in each station does not exceed the available global area for each station (A). The proposed model aimed to minimize number of workers and number of stations with given cycle time, the maximum allowed number of workers in each workstation, the global available area, a set of tasks with their processing time and required area and the precedence relationship. The tasks are required to be assigned into the workstations satisfying the precedence relation constraint, cycle time and space constraint.

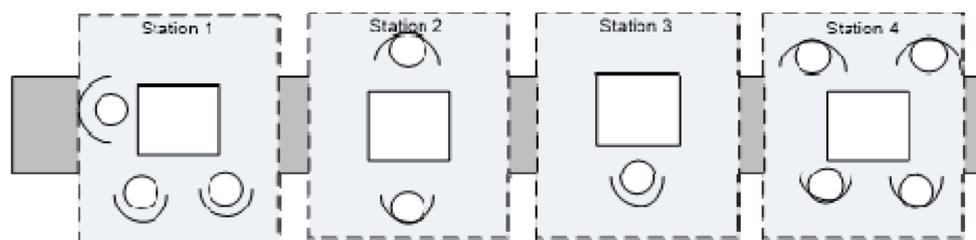


Figure 2. Multi-manned assembly line

We assume the following parameters for the proposed multi-manned assembly line model:

- The maximum allowed number of operators per station is two.
- The minimum allowed number of operators per station is one.
- Each task is allocated to one worker only.
- The precedence relationships between tasks should not be violated.
- The travel time of the workers within the station is ignored.
- Each task time is deterministic and is predetermined.
- The cycle time is predetermined (Type-I assembly line).
- The sum of task times performed by any worker in any station should not exceed the cycle time.

2.1. Notations and Abbreviations

| | | |
|------------|---|--|
| A | : | Global available area of each station |
| S | : | Station as shown in Figure 2. $S = \{1, \dots, NS\}$ |
| t | : | Task number, $t = \{1, \dots, \square\}$ |
| W | : | Worker, $W = \{1, \dots, NW\}$ where NW is the max acceptable number of workers in a station |
| CT | : | Cycle time |
| FF | : | Fitness Function |
| LB | : | Lower bound for the number of stations / workers |
| NS | : | Total number of stations in the line |
| MAL | : | Multi-manned assembly line |
| $TSMAL$ | : | Multi-manned assembly line with time and space |
| a_t | : | Area required by task t |
| EF_t | : | Early finish of task t |
| ES_t | : | Early start of task t |
| F_t | : | Number of followers of task t |
| LBT_t | : | Lower Bound of Task t ; Earliest possible station of task t |
| P_{tj} | : | Where $P_{tj} = \{1 \text{ if task } t \text{ precedes task } j, 0 \text{ otherwise}\}$ |
| W_{LB} | : | Lower bound of number of workers |
| W_{max} | : | Maximum allowed number of workers per station |
| W_{min} | : | Minimum allowed number of workers per station |
| D_{sw} | : | Delay time of Worker W at station S |
| W_{WS} | : | Worker W in station S |
| S_{ATSW} | : | Set of tasks T assigned to worker W in station S |
| IT_s | : | Station idle time |
| S_c_g | : | Sequence of tasks according to genetic algorithm |
| RPW_t | : | Ranked positional weight of task t |
| T_t | : | Time of task t |
| TA_s | : | Sum of all tasks' area in station S |
| TWL_{WS} | : | Total workload of worker W in station S (in terms of time) |
| UBT_t | : | Upper Bound of Task t ; Latest possible station of task t |

Table 1. Parameter of multi-manned assembly line model with time and space constraint

A Flow chart for the proposed “multi-manned assembly line with time and space” presented in section 3 is shown in Figure 3.

3. “Multi-Manned Assembly Line With Time and Space” Algorithm

The problem of balancing the “multi-manned assembly line with time and space” (*TSMAL*) is tackled using a bi-objective genetic algorithm to optimize two objectives simultaneously; minimize the number of stations (NS) and minimize the number of workers (NW). The algorithm involves two main stages: encoding and decoding. In the encoding stage, the feasible sequences of the tasks (chromosomes) taking precedence constraint into consideration are generated. In the decoding stage, the set of sequenced tasks are consecutively assigned to stations / workers according to a proposed heuristic. The fitness function is evaluated for each chromosome to pick the parents for crossover and mutation, and for the selection of the next generation.

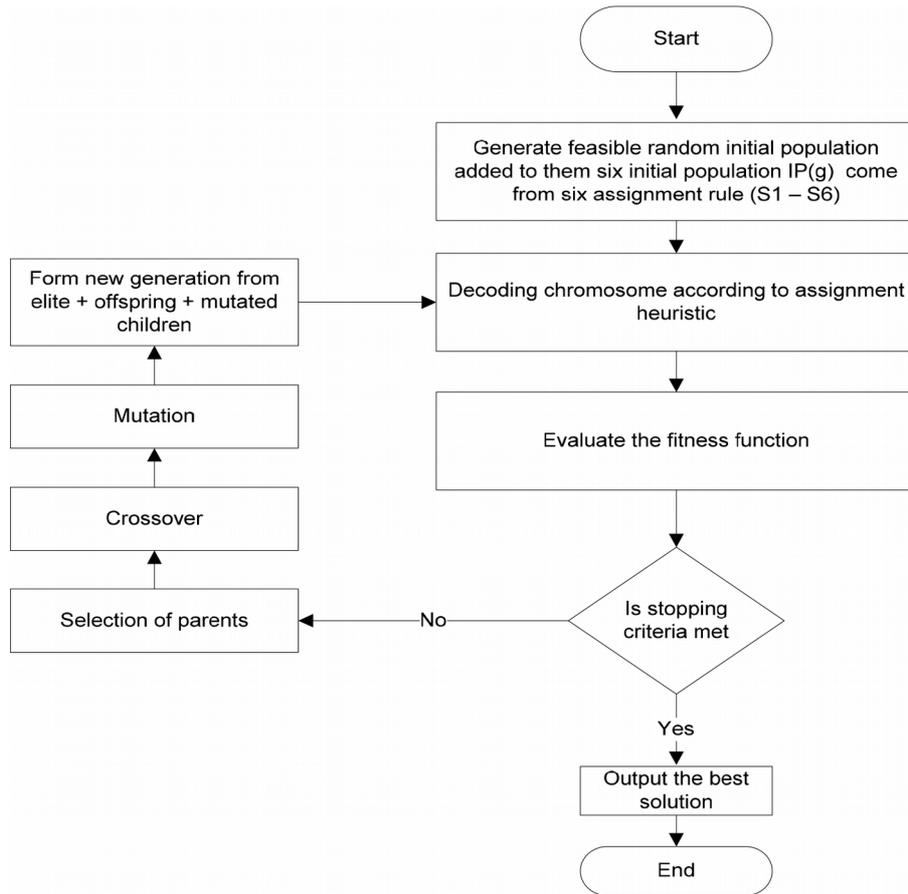


Figure 3. Flow chart of the proposed multi-manned assembly line with time and space

3.1. Encoding Stage

As the assembly line balancing problem is np hard problem, most of the researchers used meta heuristics for sequencing tasks to seek for optimal and near optimal solutions. The genetic algorithm is one of the most popular heuristics used in solving assembly line balancing problems (Chutima, 2020). From the literature, it is found that research who adds some known heuristics to the initial population gets better results than run it randomly (Gonçalves & De Almeida, 2002). In this article, initial population is generated randomly, and some well-known heuristic known in solving assembly line balancing problem is added to it.

The following four well known priority rules heuristics (Baykasoglu & Dereli, 2008) are added to the random initial population:

- Maximum Ranked Positional Weight (*RPW*);

$$RPW_t = t_t + \sum_{j=1}^T P_{tj} * t_j \quad (1)$$

- Largest Candidate Rule (*LCR*); the tasks are arranged according to the maximum task time.

$$LCR_t = t_t \quad (2)$$

- Number of Followers (*F*), the tasks are arranged according to the maximum number of followers.

$$F_t = \sum_{j=1}^T P_{tj} \tag{3}$$

- Early start (*ES*) of task. The tasks are arranged according to the minimum value of their *ES*. It is the maximum early finish (*EF*) of the predecessor tasks.

$$ES_t = \max(EF_j \times P_{jt}) \quad \forall_j = 1, \dots, T \tag{4}$$

The chromosome uses a task-based representation scheme. A task number is represented in each gene. Hence, sequence of tasks is represented in the chromosome.

3.2. Decoding Stage

In this stage the tasks are assigned to station and worker according to the following heuristic, this heuristic is an extension to the algorithm presented by Zamzam, Sadek, Afia, & El-Kharbotly (2015):

Step 1: Initializing.

Set $S = 1, W_{WS} = 0, TWL_{WS} = 0, TA_S = 0$

- Calculate the worker lower bound (W_{LB}) and the upper bound of the idle time (UB).

$$W_{LB} = Round \left(\frac{\sum_{t=1}^N t_t}{ct} \right) \tag{5}$$

$$UB = \beta * \frac{ct + W_{LB} - \sum_{t=1}^N t_t}{W_{LB}} \tag{6}$$

Where β is controlling parameter and is set by 2.

Step 2: Selecting task.

Start with the initial element from S_{c_g} until $S_{c_g} = \{\emptyset\}$; if $S_{c_g} = \{\emptyset\}$ go to Step 6.

Step 3: Assigning tasks.

- Assign task t to the worker who satisfy:

$$TWL_{WS} + t_t \leq CT \ \& \ FTP_t + t_t \leq CT \ \& \ TA_S \leq A$$

- If task t can't be assigned, remove T_i from S_{c_g} and add it to $S_{c_{UA}}$.

$$TWL_{WS} = TWL_{WS} + t_t \tag{7}$$

$$TA_S = TA_S + a_t \tag{8}$$

$$FTP_t = M \quad t = 1, \dots, n, \sum_{j=1}^N P_{tj} = 1 \tag{9}$$

Where M is a big value to prevent assign tasks before its predecessors.

- Modify the value of P_{tj} and remove the task T_i then go to Step 2.

- Assign task t to the worker that will start it earlier, if more than worker fulfills this condition.
- Choose the worker randomly if more than worker fulfill this condition, then go to Step 2.
- Modify value of P_{tj} and remove T_t from S_{c_g} then go to Step 2.

Step 4: From $S_{c_{UA}}$ start with the first element until $S_{c_{UA}} = \{\emptyset\}$; if $S_{c_{UA}} = \{\emptyset\}$ go to Step 7 if $S_{c_{UA}} \neq \{\emptyset\}$ go to Step 5.

Step 5: Calculating the idle time of workers.

- Calculate the average idle time per station (IT_s)

$$IT_s = \frac{ct * NW_s - \sum_{i \in S} t_i}{NW_s} \tag{10}$$

- If $IT_s > UB$ and $NW_s > NW_{min}$ cancel one worker at a time and restart assigning the tasks as in Step 2, until $IT_s < UB$ or $NW_s = NW_{min}$.

Step 6: Open new Station S , $NS = S + 1$, then go to Step 2.

Step 7: End

3.3. Fitness Function

It is the function that assess the solutions and according to its value, the chromosome which will be survived for the next generation is decided. In this paper two objective functions are considered. These functions are minimizing line length (NS) and minimize number of workers (NW). Pareto optimization technique is used for evaluating multi-objective functions.

$$F.F (1) = minimize NS \tag{11}$$

$$F.F (2) = minimize NW \tag{12}$$

3.4. Genetic Operators

The genetic operators that used are selection, crossover, and mutation. For the Selection, remainder selection of Matlab® are used. For Crossover, hybrid crossover technique was used, half of the crossover children used two-point crossover techniques, while the others half used Precedence preservative crossover. For Mutation, the scramble mutation was used. The values are shown in Table 2.

| Parameter | Value small sized problem | Value large sized problem |
|---------------------------|---------------------------|---------------------------|
| Population size | 25 | 60 |
| Crossover rate | 0.75 | 0.75 |
| Mutation rate | 0.25 | 0.25 |
| Elite | 3 | 6 |
| Number of generation (Ng) | 100 | 1000 |

Table 2. Parameter used in developed GA

4. Results and Discussion

To examine the performance of the proposed model with the area constraint, two computational experiments has been carried. **The first set of experiments** aimed to check the validity of the proposed algorithm. To do so the proposed algorithm was applied to solve 29 instances got from Talbot and Patterson (1984) known as Talbot data set. Two mediums sized problems, presented by Heskiouff (1986) and Kilbridge (1961), and three large-sized problems, presented by Tonge (1961) and Arcus (1963) are solved. The model is run as multi-manned model without area constraint. Table 3 shows the results obtained from the proposed algorithm compared with the results obtained by SA algorithm presented by Roshani et al. (2013). The number of workers (NW) and the number of station (NS) are considered the evaluation criteria to assess the performance of the model. Only medium and large sized problems are compared as the MAL addresses mainly large-sized products.

According to the results obtained from the conducted experiments on the medium- and large-sized problems, it can be concluded that the proposed genetic algorithm outperforms the SA algorithm in accordance with the number of stations as in nine problems the GA get better results than SA and in two instances it got worse. In two instances only GA got better number of station/workers at the expense of amount of workers/station. These results prove the effectiveness of the proposed genetic model in solving the MALBP.

The second set of experiments aimed to assess the advantage of MAL balancing problem with area constraint (TSMALBP) over single model assembly line balancing problem with area constraint (TSALBP). Eight problem instances with different order strength and number of tasks have been chosen for the experimentation Table 4: Arc111 with cycle time limits of $c = 5755$ and $c = 7520$, Barthold1, Barthold2, Lutz2, Lutz3, Mukherje and Wee-Mag. The eight instances are publicly available at: <http://www.nissanchair.com/TSALBP>. In all instances the CT is kept constant while the area is varied. The area information has been created by following the procedures of Bautista and Pereira (2007) by providing the processing time value of the last task (t_n) to the area needed by the first task (a_1).

$$a_j = t_{n-j+1} \quad (13)$$

Each instance is solved ten times and the most repeated results are reordered. Each instance is solved once for TSALBP and another time for TSMALBP. Table 5 shows the obtained results. Two evaluation criteria are considered. The deviation in number of workers and in number of stations

Deviation in number of workers/stations is calculated as follows:

$$\text{Dev} = \frac{\text{Objective value of TSMALBP} - \text{objective value of TSALBP}}{\text{Objective value of TSMALBP}} \cdot 100 \quad (14)$$

From the previous table, it can be found that in only four problems out of 51 the proposed TSMALB model get higher number of workers than TSALBP model. However, in these four problems it gets better number of stations, so it is according to the decision maker to make tradeoff between number of workers and number of stations. It also can be found that the proposed model gets better number of stations in 17 instances and in 13 of them with the same number of workers. Which proves the advantage of TSMALB model over TSALB model in better space utilization.

It is also concluded that in not all the problem features, the TSMAL model is the best choice. The proposed TSMALB model is more efficient in large sized problem that has big time variability within tasks and the largest processing time of the tasks is relatively close to the CT as shown in Figure 4 to Figure 8. In case of small processing time relative to CT and small tasks time variability, the model tends to solve the problem as TSALBP to minimize the idle time per station. As the algorithm is designed to cancel one worker in case of the idle time per station exceeds the permissible idle time Equation 4. Thus, in large sized problem, with small processing time relative to the CT it is more appropriate to solve the problem as TSALBP or the decision maker accepts a greater number of workers to get fewer number of stations.

| Author | CT | LB (NW) | SA | | Prop MAL using GA | |
|---------------|-------|---------|----|----|-------------------|----|
| | | | NW | NS | NW | NS |
| Heskia(28) | 138 | 8 | 8 | 5 | 8 | 5 |
| | 205 | 5 | 5 | 4 | 5 | 4 |
| | 216 | 5 | 5 | 3 | 5 | 3 |
| | 256 | 4 | 5 | 3 | 4 | 3 |
| | 324 | 4 | 4 | 2 | 4 | 2 |
| | 342 | 3 | 3 | 2 | 3 | 2 |
| Kilbridge(45) | 57 | 10 | 10 | 6 | 10 | 6 |
| | 79 | 7 | 8 | 4 | 7 | 4 |
| | 92 | 6 | 7 | 4 | 6 | 4 |
| | 110 | 6 | 6 | 3 | 6 | 3 |
| | 138 | 4 | 4 | 3 | 4 | 3 |
| | 184 | 3 | 3 | 2 | 3 | 2 |
| Tonge(70) | 176 | 21 | 22 | 12 | 21 | 17 |
| | 364 | 10 | 10 | 6 | 10 | 6 |
| | 410 | 9 | 9 | 5 | 9 | 5 |
| | 468 | 8 | 8 | 5 | 8 | 4 |
| | 527 | 7 | 7 | 4 | 7 | 4 |
| Arcus(83) | 5048 | 16 | 16 | 12 | 16 | 11 |
| | 5853 | 14 | 14 | 10 | 14 | 10 |
| | 6842 | 12 | 12 | 8 | 13 | 8 |
| | 7571 | 11 | 11 | 10 | 11 | 9 |
| | 8412 | 10 | 10 | 8 | 10 | 8 |
| | 8998 | 9 | 9 | 7 | 10 | 5 |
| | 10816 | 8 | 8 | 5 | 8 | 6 |
| Arcus (111) | 8847 | 18 | 18 | 14 | 18 | 12 |
| | 10027 | 16 | 16 | 12 | 16 | 10 |
| | 10743 | 15 | 15 | 14 | 15 | 14 |
| | 11378 | 14 | 14 | 9 | 14 | 8 |
| | 17067 | 9 | 9 | 7 | 9 | 5 |

Yellow color: means better solution saving in number of workers or number of stations for the proposed MALB model compared to SA.

Pink color: means worse solution increasing in number of workers or number of stations for the proposed MALB model compared to SA.

Table 3. Results of proposed MAL model compared to SA model

| Instance | Order strength | Time variability | Number of tasks | T_{max}/ct |
|------------------------|----------------|------------------|-----------------|--------------|
| Arc111 $CT = 5755$ | 40.4 | 568.9 | 111 | 0.988531712 |
| Arcus111 2 $CT = 7520$ | 40.4 | 568.9 | 111 | 0.756515957 |
| Barthold 1 | 25.8 | 127.7 | 148 | 0.475776398 |
| Barthold 2 | 25.8 | 83 | 148 | 0.976470588 |
| Hesika | 22.5 | 108 | 28 | 0.315789474 |
| Lutz2 | 77.6 | 10 | 89 | 0.986666667 |
| Lutz3 | 77.6 | 74 | 89 | 0.625 |
| Mukherje | 44.8 | 21.4 | 94 | 0.487179487 |
| Wee-Mag | 22.7 | 13.5 | 75 | 0.946428571 |

Table 4. Used instances features.

| Author | CT | T_{max}/ct | A | $TSALBP$ | | $TSMALBP$ | | Dev in worker | Dev in station |
|---------------|------|--------------|-------|----------|----|-----------|----|---------------|----------------|
| | | | | NW | NS | NW | NS | | |
| Heskia(28) | 342 | 0.315 | 138 | 8 | 8 | 8 | 8 | 0 | 0 |
| | | | 205 | 6 | 6 | 6 | 6 | 0 | 0 |
| | | | 216 | 5 | 5 | 5 | 5 | 0 | 0 |
| | | | 256 | 5 | 5 | 5 | 5 | 0 | 0 |
| | | | 324 | 4 | 4 | 4 | 4 | 0 | 0 |
| | | | 342 | 4 | 4 | 4 | 4 | 0 | 0 |
| Weemag (75) | 56 | 0.48 | 27.73 | 64 | 64 | 64 | 64 | 0 | 0 |
| | | | 34 | 61 | 61 | 61 | 61 | 0 | 0 |
| | | | 40 | 60 | 60 | 60 | 60 | 0 | 0 |
| | | | 46 | 37 | 37 | 37 | 37 | 0 | 0 |
| | | | 50 | 33 | 33 | 33 | 33 | 0 | 0 |
| | | | 56 | 31 | 31 | 31 | 31 | 0 | 0 |
| Lutz 3 (89) | 75 | 0.98 | 74 | 28 | 28 | 28 | 27 | 0 | -3.70370 |
| | | | 75 | 28 | 28 | 28 | 27 | 0 | -3.70370 |
| | | | 77 | 28 | 28 | 28 | 27 | 0 | -3.70370 |
| | | | 79 | 27 | 27 | 27 | 27 | 0 | 0 |
| | | | 81 | 26 | 26 | 26 | 25 | 0 | -4 |
| Lutz 2 (89) | 16 | 0.625 | 10 | 58 | 58 | 58 | 58 | 0 | 0 |
| | | | 15 | 40 | 40 | 40 | 40 | 0 | 0 |
| | | | 20 | 36 | 36 | 36 | 30 | 0 | -20 |
| | | | 25 | 33 | 33 | 34 | 28 | 2.9411765 | -17.8 |
| | | | 30 | 33 | 33 | 33 | 24 | 0 | -37.5 |
| Mukherje (94) | 351 | 0.487 | 171 | 27 | 27 | 27 | 27 | 0 | 0 |
| | | | 220 | 20 | 20 | 20 | 20 | 0 | 0 |
| | | | 270 | 17 | 17 | 17 | 17 | 0 | 0 |
| | | | 320 | 14 | 14 | 14 | 14 | 0 | 0 |
| | | | 351 | 14 | 14 | 14 | 14 | 0 | 0 |
| | | | 370 | 13 | 13 | 13 | 13 | 0 | 0 |
| | | | 400 | 13 | 13 | 13 | 12 | 0 | -8.33333 |

| Author | CT | T_{max}/ct | A | TSALBP | | TSMALBP | | Dev in worker | Dev in station |
|-----------------|------|--------------|-------|--------|----|---------|----|---------------|----------------|
| | | | | NW | NS | NW | NS | | |
| Arc 1 | 7520 | 0.75 | 5689 | 29 | 29 | 29 | 29 | 0 | 0 |
| | | | 6689 | 25 | 25 | 25 | 25 | 0 | 0 |
| | | | 7689 | 22 | 22 | 22 | 22 | 0 | 0 |
| | | | 8689 | 21 | 21 | 21 | 21 | 0 | 0 |
| | | | 9689 | 21 | 21 | 21 | 21 | 0 | 0 |
| | | | 10689 | 21 | 21 | 21 | 21 | 0 | 0 |
| Arc111 2 | 5755 | 0.98 | 5689 | 33 | 33 | 33 | 30 | 0 | -10 |
| | | | 6689 | 28 | 28 | 30 | 25 | 6.66666 | -12 |
| | | | 7689 | 28 | 28 | 29 | 22 | 3.44827 | -27.2727 |
| | | | 8689 | 28 | 28 | 28 | 20 | 0 | -40 |
| | | | 10689 | 27 | 27 | 28 | 18 | 3.57142 | -50 |
| Barthold1 (148) | 805 | 0.475 | 605 | 10 | 10 | 10 | 10 | 0 | 0 |
| | | | 705 | 8 | 8 | 8 | 8 | 0 | 0 |
| | | | 805 | 8 | 8 | 8 | 8 | 0 | 0 |
| | | | 905 | 8 | 8 | 8 | 8 | 0 | 0 |
| | | | 1005 | 8 | 8 | 8 | 8 | 0 | 0 |
| | | | 1105 | 8 | 8 | 8 | 7 | 0 | -14.285 |
| Barthold2 (148) | 85 | 0.976 | 85 | 60 | 60 | 60 | 60 | 0 | 0 |
| | | | 90 | 56 | 56 | 56 | 56 | 0 | 0 |
| | | | 95 | 55 | 55 | 55 | 54 | 0 | -1.85185 |
| | | | 100 | 54 | 54 | 54 | 53 | 0 | -1.88679 |
| | | | 105 | 54 | 54 | 54 | 53 | 0 | -1.88679 |

Negative value: means better solution saving in number of workers or number of stations for the proposed *MALBPTS*.

Positive value: means worse solution increasing in number of workers or number of stations for the proposed *TSSALBP*.

Table 5. Comparison results of *TSALBP* and *TSMALBP*

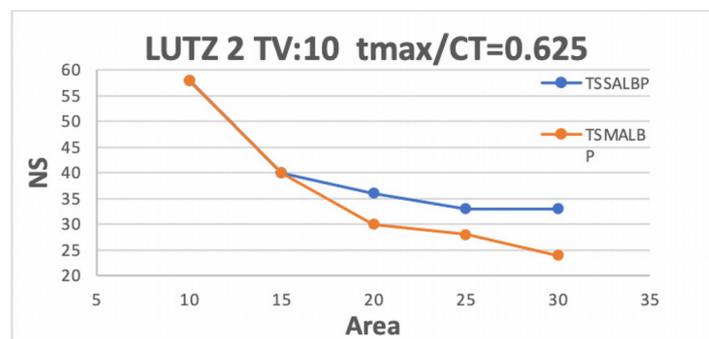


Figure 4. The number of stations plotted against the area for Lutz 2 instance with moderate *TV* and T_{max}/CT

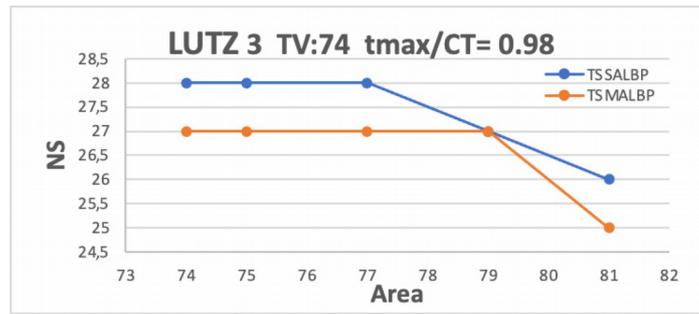


Figure 5. The number of stations drawn against the area for Lutz 3 instance with high TV and T_{max}/CT

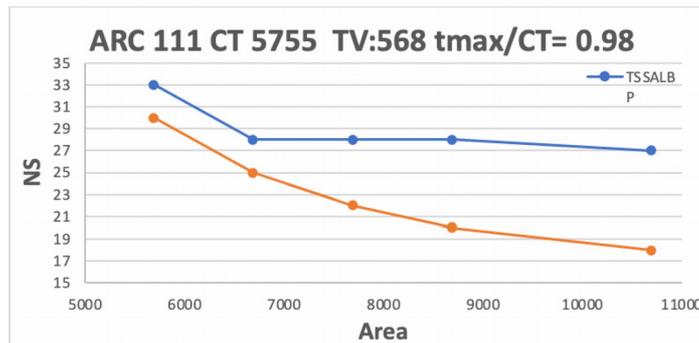


Figure 6. The number of stations drawn against the area for Arc111 $CT = 5755$ instance with high TV and T_{max}/CT

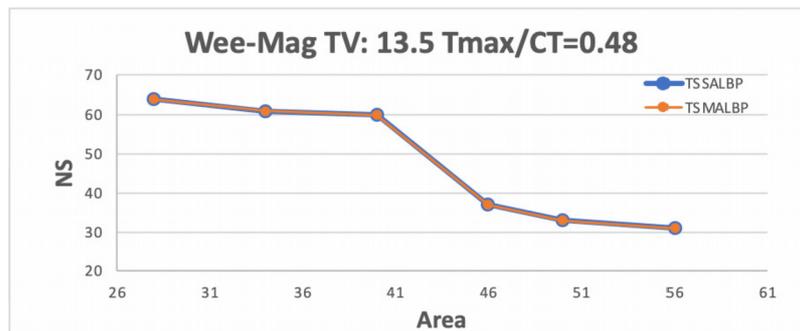


Figure 7. The number of stations drawn against the area for Wee-Mag instance with low TV and T_{max}/CT

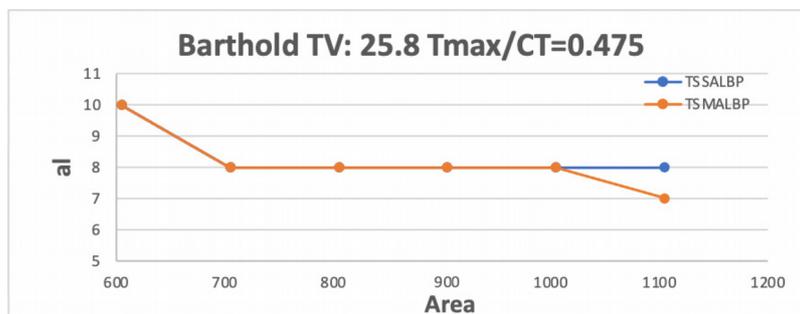


Figure 8. The number of stations drawn against the area for Barthold instance with low TV and moderate T_{max}/CT

Finally, the algorithm was assessed by solving real life problem. The data was provided by the “Nissan Barcelona Plant (see <http://www.nissanchair.com/TSALBP>)”. This problem instance has detailed area information for each task. The results of solving the problem as single model without area constraint, TSALBP and TSMALBP compared to results got from ant algorithms proposed by Bautista and Pereira (2007) are shown in Table 6.

| CT = 180, Number of tasks = 140 TV = 115 $T_{max}/CT = 0.66$ | | | | | | |
|--|---------------------------|----|--------|----|---------|----|
| | Single model without area | | TSALBP | | TSMALBP | |
| | NS | NW | NS | NW | NS | NW |
| Proposed model | 17 | 17 | 20 | 20 | 20 | 20 |
| Bautista et al model | 17 | 17 | 21 | 21 | - | - |

Table 6. The results of real-life problem of Nissan plant in Spain

From the previous table it can be found that the proposed model gets better solution than the one proposed by Bautista and Pereira. However, since the time variability and T_{max}/CT are of middle value the model tends to solve the problem as TSALBP. The problem is solved again but by changing the CT and giving it the values; 150, and 125 with $T_{max}/CT = 0.8$ and 0.96 respectively. The results are shown in Table 7 and Figure 9.

From the previous figure it can concluded that as T_{max}/CT increases the model tends to solve the problem as TSMAL ($NS < NW$) and when it decreases it tends to solve it as TSALBP ($NW = NS$).

| Area LB | WLB | TSMALBP | | | |
|---------|-----|---------|--------------|----|----|
| | | CT | T_{max}/CT | NS | NW |
| 19 | 17 | 180 | 0.666667 | 20 | 20 |
| | 20 | 150 | 0.8 | 23 | 24 |
| | 24 | 125 | 0.96 | 22 | 24 |

Table 7. The results of real-life problem of Nissan plant in Spain by varying the CT

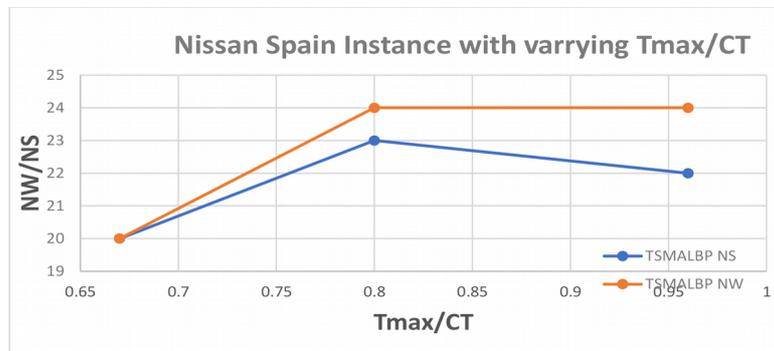


Figure 9. The number of stations and number of workers drawn against the T_{max}/CT for Nissan Spain instance

5. Conclusion

A multi-manned assembly line is widely used in industry for assembly medium-large sized product. The size and area of tools and components required to assemble the parts are essential to be considered while tackling the problem of balancing the line, as the space required by them affect the solution (task assignment), the exerted effort by operators to get the part and tools and the length of the line. Despite its importance to simulate real life case only few literatures that focus on time and space assembly line problem and only one literature focus on multi-manned assembly line with time and space constraint.

In this paper the problem of multi-manned assembly line with time and space constraint is tackled. A genetic algorithm with the objective of minimizing number of workers and number of stations is proposed. The initial population of the model is generated randomly with four conventional heuristic for assembly line were added to them to enhance the solution.

The results showed the effectiveness of the proposed model as multi-manned assembly line model compared to SA algorithm presented in literature. The results prove the advantage of MAL balancing problem with area constraint over single model assembly line balancing problem with area constraint as it saves stations in more than 33% of the tested problems. It is also found that the MAL with time and area constraint is not the best choice in all medium-large sized problem, in large sized problem, with small processing time relative to the CT and small task time variability it is more appropriate to solve the problem as TSALBP or the decision maker accepts a greater number of workers to get fewer number of stations. Finally, the proposed model gets better solution in solving real life problem (Nissan Barcelona Plant) which emphasis the effectiveness of the proposed model.

Future work could extend the proposed model to take the side of tasks into consideration and adding more realistic constraints as positive and negative zoning and synchronous tasks. further investigation is required to determine the effect of the features of the problem on the suitability of solving the problem with multi-manned model taking into consideration time and space constraint.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

References

- Alavidoost, M.H., Zarandi, M.F., Tarimoradi, M., & Nemati, Y. (2017). Modified genetic algorithm for simple straight and U-shaped assembly line balancing with fuzzy processing times. *Journal of intelligent manufacturing*, 28(2), 313-336. <https://doi.org/10.1007/s10845-014-0978-4>
- Arcus A.L. (1963). *An analysis of a computer method of sequencing assembly line operations*. PhD dissertation. University of California.
- Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142(2), 259-277. <https://doi.org/10.1016/j.ijpe.2012.10.020>
- Bautista, J., & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, 177(3), 2016-2032. <https://doi.org/10.1016/j.ejor.2005.12.017>
- Bautista, J., & Pereira, J. (2009). A dynamic programming based heuristic for the assembly line balancing problem. *European Journal of Operational Research*, 194(3), 787-794. <https://doi.org/10.1016/j.ejor.2008.01.016>
- Bautista, J., & Pereira, J. (2011). Procedures for the Time and Space constrained Assembly Line Balancing Problem, *European Journal of Operational Research*, 212(3), 473-481. <https://doi.org/10.1016/j.ejor.2011.01.052>
- Baykasoglu, A., & Dereli, T. (2008). Two-sided assembly line balancing using ant colony- based heuristic. *International Journal of Advanced Manufacturing Technology*, 36, 582-588. <https://doi.org/10.1007/s00170-006-0861-3>
- Cevikcan, E., Durmusoglua, M.B., & Unal, M.E. (2009). A team-oriented design methodology for mixed model assembly systems. *Computers & Industrial Engineering*, 56, 576-599. <https://doi.org/10.1016/j.cie.2007.11.002>
- Chen, Y.Y., Cheng, C.Y., & Li, J.Y. (2018). Resource-constrained assembly line balancing problems with multi-manned workstations. *Journal of Manufacturing Systems*, 48, 107-119. <https://doi.org/10.1016/j.jmsy.2018.07.001>
- Chica, M., Cordon, O., & Damas, S. (2011). An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem. *Computers & Industrial Engineering*, 61(1), 103-117. <https://doi.org/10.1016/j.cie.2011.03.001>
- Chica, M., Cordon, O., Damas, S., & Bautista, J. (2010). Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Information Sciences*, 180(18), 3465-3487. <https://doi.org/10.1016/j.ins.2010.05.033>

- Chica, M., Cordon, O., Damas, S., & Bautista, J. (2012). Multiobjective memetic algorithms for time and space assembly line balancing. *Engineering Applications of Artificial Intelligence*, 25(2), 254-273. <https://doi.org/10.1016/j.engappai.2011.05.001>
- Chutima, P. (2020). Research Trends and Outlooks in Assembly Line Balancing Problems. *Engineering Journal*, 24(5), 93-134. <https://doi.org/10.4186/ej.2020.24.5.93>
- Dimitriadis, S.G. (2006). Assembly line balancing and group working: a heuristic procedure for workers' groups operating on the same product and work- station. *Computers & Operations Research*, 33, 2757-2774. <https://doi.org/10.1016/j.cor.2005.02.027>
- Eghtesadifard, M., Khalifeh, M., & Khorram, M. (2020). A systematic review of research themes and hot topics in assembly line balancing through the web of science within 1990-2017. *Computers & Industrial Engineering*, 139, 106182. <https://doi.org/10.1016/j.cie.2019.106182>
- Fattahi, P., Roshani, A., & Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *International Journal of Advanced Manufacturing Technology*, 53, 363-378. <https://doi.org/10.1007/s00170-010-2832-y>
- Ferrari, E., Faccio, M., Gamberi, M., Margelli, S., & Pilati, F. (2019). Multi-manned assembly line synchronization with compatible mounting positions, equipment sharing and workers cooperation. *IFAC-PapersOnLine*, 52(13), 1502-1507. <https://doi.org/10.1016/j.ifacol.2019.11.412>
- Gonçalves, J.F., & De Almeida, J.R. (2002). A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics*, 8(6), 629-642. <https://doi.org/10.1023/A:1020377910258>
- Hazır, Ö., Delorme, X., & Dolgui, A. (2015). A review of cost and profit-oriented line design and balancing problems and solution approaches. *Annual Reviews in Control*, 40, 14-24. <https://doi.org/10.1016/j.arcontrol.2015.09.001>
- Heskiaoff, H. (1968). A heuristic method for balancing assembly lines. *Western Electric Engineering*, 12(3), 9-16.
- Kazemi, A., & Sedighi, A. (2013). A Cost-oriented model for multi-manned assembly line balancing problem, *Journal of Optimization in Industrial Engineering*, 13, 13-25
- Kellegoz, T., & Toklu, B. (2012). An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations. *Computers & Operations Research*, 39, 3344-3360. <https://doi.org/10.1016/j.cor.2012.04.019>
- Kilbridge, M.D., & Wester, L. (1961). A heuristic method of assembly line balancing. *Journal of Industrial Engineering*, 12(4), 292-298.
- Li, Y., Wang, H., & Yang, Z. (2018). Type II assembly line balancing problem with multioperators. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-018-3834-1>
- Li, Z., Janardhanan, M.N., Nielsen, P., & Tang, Q. (2018). Mathematical models and simulated annealing algorithms for the robotic assembly line balancing problem. *Assembly Automation*. <https://doi.org/10.1108/AA-09-2017-115>
- Li, Z., Kucukkoc, I., & Zhang, Z. (2018). Branch, bound and remember algorithm for U-shaped assembly line balancing problem. *Computers & Industrial Engineering*, 124, 24-35. <https://doi.org/10.1016/j.cie.2018.06.037>
- Make, M.R.A., Rashid, M.F.F.A., & Razali, M.M. (2017). A review of two-sided assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 89(5-8), 1743-1763. <https://doi.org/10.1007/s00170-016-9158-3>
- Micieta, B., & Stollmann, V. (2011). Assembly line balancing. *DAAAM International* (257-264). <https://doi.org/10.2507/daaam.scibook.2011.21>
- Purnomo, H., Wee, H., Rau, H. (2013). Two-sided assembly lines balancing with assignment restrictions. *Mathematical and Computer Modeling*, 57, 189-199. <https://doi.org/10.1016/j.mcm.2011.06.010>

- Rada-Vilela, J., Chica, M., Cerdón, Ó., & Damas, S. (2013). A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem. *Applied Soft Computing*, 13(11), 4370-4382. <https://doi.org/10.1016/j.asoc.2013.06.014>
- Roshani, A., Roshani, A., Roshani, A.; Salehi, M., & Esfandyari, A. (2013). A simulated annealing algorithm for multi-manned assembly line balancing problem. *Journal of Manufacturing Systems*, 32; 238-247. <https://doi.org/10.1016/j.jmsy.2012.11.003>
- Roshani, A., & Giglio, D. (2020). A tabu search algorithm for the cost-oriented multi-manned assembly line balancing problem. *International Journal of Industrial Engineering & Production Research*, 31(2), 189-202.
- Saif, U., Guan, Z., Wang, B., Mirza, J., & Huang, S. (2014). A survey on assembly lines and its types. *Frontiers of Mechanical Engineering*, 9(2), 95-105. <https://doi.org/10.1007/s11465-014-0302-1>
- Taha, B., El-Kharbotly, A., Sadek, Y., & Afia, N. (2011). A Genetic Algorithm for solving two-sided assembly line balancing problems. *Ain Shams Engineering Journal*, 227-240. <https://doi.org/10.1016/j.asej.2011.10.003>
- Talbot, F.B., & Patterson JH. (1984). An integer programming algorithm with network-cuts solving the assembly line balancing problem. *Management Science*, 30, 85-89. <https://doi.org/10.1287/mnsc.30.1.85>
- Tonge, F.M. (1961). *A heuristic program of assembly line balancing*. Englewood Cliffs: Prentice-Hall.
- Yadav, A., & Agrawal, S.A. (2019). A multi-manned parallel two-sided assembly line balancing with tool sharing approach-a company case study solved by exact solution approach. *International Journal of Mechanical and Production Engineering Research and Development*, 9(2), 51-60. <https://doi.org/10.24247/ijmperdapr201905>
- Zamzam, N., Sadek, Y., Afia, N., & El-Kharbotly, A. (2015). Multi-manned Assembly Line Balancing using Genetic Algorithm. *International Journal of Engineering Research & Technology*, 73(9-12), 1665-1694.
- Zhang, Z., Tang, Q., & Chica, M. (2020). Multi-manned assembly line balancing with time and space constraints: A MILP model and memetic ant colony system. *Computers & Industrial Engineering*, 150, 106862. <https://doi.org/10.1016/j.cie.2020.106862>

