PORT SAID ENGINEERING RESEARCH JOURNAL
Faculty of Engineering - Port Said University
Volume 26 No. 3 pp: 134-145

A genetic algorithm for multi-manned multi-position assembly line under technological constraint

Raghda B. Taha[*1], *Nessren Mohamed Zamzam*[2]

[1] *Department of Supply Chain Management College of International Transport and Logistics, Arab Academy for Science, Technology and Maritime Transport, Egypt, email: email: raghda.bahaa@aast.edu*

[2] *Design and Production Engineering, Faculty of Engineering, Ain Shams University, Cairo, Egypt, email: Nessren.zamzam@eng.asu.edu.eg*

**Corresponding author, Email address, raghda.bahaa@aast.edu* *DOI: 10.21608/PSERJ.2022.74000.1107*

Received 25-4-2021,
Revised 16-12-2021
Accepted 10-3-2022

ABSTRACT

A multi-manned assembly line is a generic type of assembly line which is common industrial procedures in industry of large sized products. In such line, the position of tasks in the product is critical to be considered to eliminate worker interference and, as a result, decrease unplanned waiting time. However, limited literature considers the position of tasks from the mounting position perspective. This paper introduces a Genetic Algorithm with the objective of minimizing line length to address the multi-manned multi-position assembly line balancing problem. A new constraint is defined, technological constraint where the tasks are constrained with their installation and position in the product and equipment required for them in the station. The performance of the model is compared to the existing approaches, and it proves its effectiveness. The practical applicability was also examined through balancing a real-life assembly line problem and the results show its validity in practical application.

Keywords: Assembly line, multi-manned, genetic algorithm, multi-position, technological constraint

1.INTRODUCTION

Assembly is the final phase in the product manufacturing process, in which all pieces are placed together for product realization. In assembly line there is a conveyor around it, worker/workers execute various activities to assemble the products. Assembly lines are classified into three types: one-sided, two-sided, and multi-manned assembly lines.

The assembly line in which each station has just one worker on one side of the line is known as one-sided assembly line. In two-sided assembly line, there are two workers executing different tasks simultaneously in parallel stations, one on the left side L of the line and the other on the right-side R. The right and left stations are alled mated station. "Multi-manned Assembly Line" (MAL), there is more than worker perform different tasks simultaneously around the product in each station as shown in Figure 1. The "multi-manned assembly line" has an advantage over two-sided and one-sided assembly line according to Dimitriadis [1] that it can: shorten the line length, minimize the cost of the tools and equipment as workers can share it in the same station and it can decrease the line idle time and increase its efficiency. However multi-manned assembly line is appropriate for certain industry sector, where the product should be medium or large sized products as automobile and refrigerator, Etc. The MAL depicts the reality of today large-scaled product manufacturing as the complexity and number of parts in products grows in tandem with customer expectations and demands which are in continuous increase.

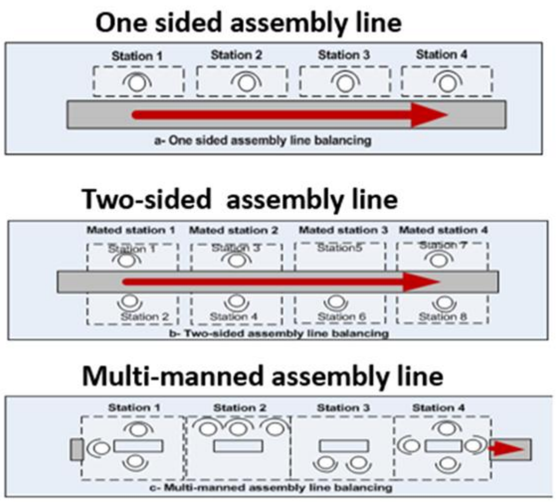

Figure 1 Assembly line types (zamzam et al [2])

Dimitriadis [1] was the first researcher to address the multi-manned assembly line problem. Two level heuristic procedures were developed to solve the problem. The first procedure was to find all feasible tasks assigned to L workers working together simultaneously in same workstation and the second procedure was to allocate each task to each worker. The results showed shorten in the line length and an improvement in the production space utilization. Kellegoz et al [3] developed a branch and bound algorithm called Jumper, to solve the MAL problem. The interpretation of the results revealed that the Jumper outperformed the one from the literature. However, the algorithm's performance in solving large scale ALBP is debatable. Giglio et al [4] proposed a mixed integer programming formulation to solve the MAL balancing issue with qualified workers. The model's goal was to reduce the overall operating cost of the line as well as the total cost of the workers' salaries. Their findings revealed that reducing the number of stations and workers reduced the system's operating costs. However, their model could be applied to small-scale problems, which is not the MMAL's typical application. Şahina et al [5] presented particle swarm meta heuristic to solve the problem of balancing MAL under resource investment by considering renewable resources required by each task. The goal of the paper was to determine the best sequence of tasks for lowering the total cost of the line, which included the cost of the required renewable resources as well as the opened stations. The results demonstrated the ability of the new mixed integer programing formulation to solve optimally the small sized problem with fewer than 35 tasks. The results also showed that the developed constructive heuristic with particle swarm optimization could solve the medium and large sized problems outperforming the tabu search algorithm taken from literature. Zhang et al [6] addressed the multi-manned assembly line balancing under space constraint. The problem of balancing a "multi-manned assembly line with time and space constraints" is first formulated. Then, to solve the problem, a new mixed-integer linear mathematical model (MILP) is presented, followed by a "memetic ant colony optimization to solve the large sized problem. The results proved that the MILP was successful in determining the optimal assignment plan for small sized problem, and the memetic ant colony optimization demonstrated high performance. Zamzam et al [7] addressed the time and space multi-manned assembly line balancing problem. A genetic algorithm was used to tackle the problem. The results demonstrated the ability of the model to get comparative results. A relationship between the solution and problem features was also discovered, according to which the line type (one-sided or multi-manned) is defined.

Different meta heuristics were used to solve the MAL balancing problem as ant colony, simulated annealing, practical swarm, tabu search and genetic algorithm. Fattahi et al [8] solved medium and large sized problem using an ant colony (ACO) approach. The model objective functions were to minimize the number of workers and multi-manned stations. The results showed that the proposed ACO outperformed other algorithms in terms of solution quality. Roshani et al [9] presented simulated annealing algorithm (SA) for problem of MALB with the objective of increasing the line efficiency and decreasing the smoothness index and the line length. The results showed that the proposed algorithm with the same number of workers could outperform ACO in terms of reducing the number of stations however the computational time of the proposed algorithm, was longer than that of the ACOs. Roshani [10] proposed simulated annealing for solving the mixed model MAL balancing problem. It was the first model to solve the problem in mixed model domain. The results showed that the model has a satisfactory performance from the solution accuracy and computational time efficiency. Practical swarm, tabu search and genetic algorithm were also used in Kazmi et al [11], Yilmazaz et al [12], Chen et al [13] respectively. There was no study can prove that one Meta-heuristic is superior to the others. Each's performance is determined by the parameters chosen and the amount of time allotted to them. The genetic algorithm (GA) was, however, the most used Meta-heuristic for balancing assembly lines.

Different objectives were as well used to solve MAL as minimize number of workers, number of stations, total cost of the line and total idle time of the line [1,4,9,14]. Literature also tackled the problem from considering different constraint as sequence of workers constraint, number of worker constraint and resource constraint besides the assembly line traditional constraints (precedence constraint and cycle time constraint) [3, 5,8]. However, only one literature who took into consideration the mounting position of the task [15]. Ferrari et al [15] presented a new mixed integer programming model for MAL synchronization problem to optimize simultaneously the line length and workload smoothness. Many problem features are considered as compatibility/incompatibility between mounting positions, cooperation between workers and sharing tools and equipment. The studied problem was concerned with the automobile industry, with fixed positions. The cars were divided into four different heights with 13 different assembly positions for each height to reduce worker interference. A simulated annealing algorithm was created with tailored procedures to solve the large sized problem. A case study was tackled to evaluate the performance of the proposed model. The results showed that the line efficiency was 89.85%. According to Ferrari et al [15] there can be interference between tasks of the same position, moreover, completing a task in a certain mounting position may prevent other tasks in the opposing positions. Hence, considering the tasks position may decrease the delay time occurred by the mutual interference.

The technological constraint (task position) is important to be considered in real life case, as when the optimum solution from different algorithms is implemented, it is found that it does not perform as planned. This is because those algorithms ignored the interference between tasks which can result in unplanned delay. For example, if there are two tasks i, and h where there is no precedence relation between them and both of them is in the same position (position 1 as shown in Figure 2) and assigned to different workers w1 and w2 respectively, w2 will not be able to start in working in task h until worker 1 finishes first.

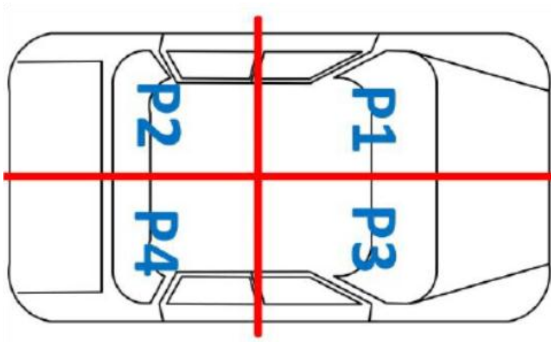

Figure 2 product layout (4 positions)

From the previous survey to the author knowledge, there is no published work in multi manned literature consider the technological constraint. Almost all of them assigned tasks to worker without taking into consideration the position of these tasks, except Ferrari et al [15] who consider the position of the tasks from the mounting level perspective. Hence, the goal of this paper is to study the MAL balancing problem taking into consideration the technological constraint (technological positions of tasks). A genetic algorithm is used for tackling the problem. A new constraint is defined in which the product is divided into four blocks/positions and the tasks assignment are constrained based on their position on the product (P1, P2, P3 or P4). The station is divided also into four blocks that correspond to the product. The workers are then, assigned to one of these blocks. The objective of the present work is to minimize the line length.

The structure of this paper is as follows: section 2 shows problem definition, section 3 introduces the proposed algorithm and methodology, section 4 displays results and discussion and finally section 5 shows conclusion and future work.

2.PROBLEM DEFINITION

In a multi-position MAL under technological constraint, the product is awaiting during the cycle time to be assigned to the station where there are number of workers do different tasks on the product simultaneously. The station is divided into 4 positions/blocks as show in Figure 3. Each task will be constrained with its installation and position in the product and will take one of the four indices: p1, p2, p3, or p4. Each worker is assigned to one of the 4 block/position of the station and the task can only be assigned to a worker if it has the same position index as the worker and the sum of the task time and total task times of the tasks performed before that task in the same position of this station is less than or equal to the cycle time.

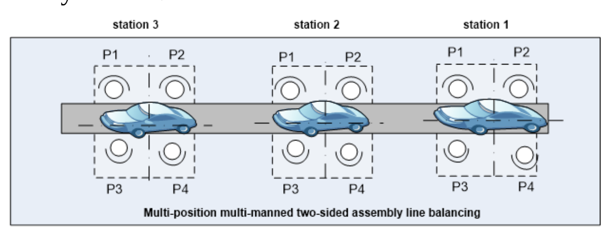

Figure 3 Multi-position multi-manned assembly line under technological constraint

3.THE DEVELOPED MULTI-MANNED MULTI-POSITION ASSEMBLY LINE BALANCING ALGORITHM

In this article, Genetic algorithms meta heuristic (GA) is used for solving the problem. GA are a general technique for solving complex np hard optimization problems that based on operating a population of solutions using genetic operators such as selection, recombination, and mutation. The proposed GA model is an extension version of GA model of two-sided assembly line balancing problem presented by Taha et al. [16]. In this model, a hybrid crossover technique is used to obtain the crossover children, which involves both forward and backward methods. A modified mutation operator is also used. This modification helps in getting better solution by searching the solution space more efficiently than the traditional methods.

Notation

Table 1 Notation and abbreviation

Notation and abbreviation	
i,h	Task number 1,2,3,……n
n	Number of task
P1,P2,P3,P4	Front left position, rear left position, front right position, rear right position respectively
j	index of workstation = {1,2,3,……j_{max} } where j_{max} max valid number of workstation (j_{max} = n)
P	Precedence matrix
P_i	immediate predecessors of task i
P_i^*	total predecessors of task i
t_i	Processing time of task i
ts_i	start time of task i
FT_i	Finishing time for task i
CT	Cycle time
k	index of worker = {1,2,3,….,M_{max}}
$WL_{p1,j}$	Workload of the position 1 of the j station
$WL_{p2,j}$	Workload of the position 2 of the j station
$WL_{p3,j}$	Workload of the position 3 of the j station
$WL_{p4,j}$	Workload of the position 4 of the j station
$SAT_{i,j}$	Set of assignable tasks(i) to the station j
SP_1T_j	Set of tasks(i) requiring position 1 in station j
SP_2T_j	Set of tasks(i) requiring position 2 in station j
SP_3T_j	Set of tasks(i) requiring position 3 in station j
SP_4T_j	Set of tasks(i) requiring position 4 in station j
TP_1	Set of tasks(i) assigned to the position 1 in station j
TP_2	Set of tasks(i) assigned to the position 2 in station j
TP_3	Set of tasks(i) assigned to the position 3 in station j
TP_4	Set of tasks(i) assigned to the position 4 in station j

3.1 Mathematical formula

Decision variables:

$$X_{ikj} = \begin{cases} 1 & \text{if task } i \text{ is assigned to worker } k \text{ in station } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ih} = \begin{cases} 1 & \text{if task } h \text{ is assigned to the same position after task } i \\ 0 & \text{otherwise} \end{cases}$$

$$l_{jkp} = \begin{cases} 1 & \text{if worker } k \text{ is assigned to position } p \text{ in station } j \\ 0 & \text{otherwise} \end{cases}$$

Objective function:

Minimize number of stations for a given cycle time.

$$\text{Min } Z = \sum_{k \in K} \sum_{j=1}^{j_{max}} j X_{ikj}$$

Constraints

This constraint to ensure that each task is assigned to only one worker at one workstation.

$$\sum_{k=1}^{M_{max}} \sum_{j=1}^{j_{max}} X_{ikj} = 1, \forall\, i = 1,2,3, \dots \dots n \qquad (1)$$

This constraint to ensure the precedence relationship is respected.

$$\sum_{k \in K} \sum_{j=1}^{j_{max}} j X_{ikj} \geq \sum_{k \in K} \sum_{j=1}^{j_{max}} j X_{hkj}, \forall\, i = 1,2 \dots n, h \in P_i \qquad (2)$$

This constraint to ensure that the total workload of the worker doesn't violate the cycle time

$$\sum_{k \in K} \sum_{i=1}^{n} (t_{si} + t_i) X_{ikj} \leq CT \ \forall\ j = 1,2,3, j_{max} \qquad (3)$$

This constraint to ensure that if task h is an immediate predecessor of task i and both is assigned to the same station and different workers (position) the start time of task i is greater than or equal the finish time of task h.

$$\sum_{k \in K} t_{si} X_{ikj} \geq \sum_{k \in K} (t_{sh} + t_h) X_{hkj}, \ \forall\ j = 1, 2 \dots, j_{max}, , i = 1,2 \dots n, h \in P_i \qquad (4)$$

This constraint to control the sequencing of the tasks: if task i, and h are assigned to same worker (position) and there is no precedence relation between them then,

If task i is assigned before h:

$$\sum_{k \in K} t_{sh} \geq \sum_{k \in K} (t_{si} + t_i) y_{ih} \text{, and vice versa} \qquad (5)$$

This constraint to ensure that number of workers in station doesn't exceed the max permitted number of workers (M_{max})

$$\sum_{k \in K} k\, l_{jk} \leq M_{max}, \quad \forall \; j = 1, 2,..j_{max} \qquad (6)$$

Definition of decision variables

X_{ijk}, $Y_{i,h}$, $l_{jk} \in \{0\,;1\}$ $\forall$ j = 1,2,..j_{max}, $i = 1,2 \ldots n$, k $\in$K (7)

Assumptions:

- Task times are deterministic.
- The case of single model is concerned.
- The precedence relationships among tasks are known.
- Workers complete their tasks in the workstation within a predefined cycle time.
- Parallel stations and parallel tasks are not permitted.

3.2 Task assignment rule

Each chromosome Figure 4 is considered a sequence through which the tasks is chosen to be assigned to station. The assignment of this task to certain worker in the station is done according to proposed assignment rule shown in the following algorithm.

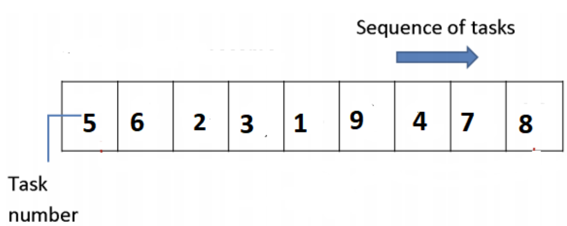

Figure 4 Chromosome illustration for problem of 9 tasks

Algorithm

1. Set j =1, $WL_{p1,j}$= 0, $WL_{p2,j}$= 0, $WL_{p3,j}$= 0, and $WL_{p4,j}$= 0, k=0

2. Determine SAT_j, SP_1T_j, SP_2T_j, SP_3T_j, SP_4T_j, if $SAT_j = \{\emptyset\}$ then go to step 6 else go to step 3

3. For each task i in SAT_j

   If $i \in SP_1T_j$, or $i \in SP_2T_j$, or $i \in SP_3T_j$, or $i \in SP_4T_j$ then calculate the FT_i, where FT_i = Max $\left\{t_i + WL_{p_1\,(p_2,p_3,p_4),j}\,, st_i + t_i\right\}$

   If $FT_i \leq CT$, Assign task i to its position and worker k to this position, otherwise open a new station and go to step 5
4. If task i couldn't be assigned to the current station, then open a new station, and If $TP_1 \neq \{\emptyset\}$, or $TP_2 \neq \{\emptyset\}$, or $TP_3 \neq \{\emptyset\}$, or $TP_4 \neq \{\emptyset\}$, then , j = j+1, and go to step 2
5. Stop.

3.3 Fitness function

The objective function is used to define the GA's fitness function. It provides a measure of chromosome's performance. The fitness function used by the proposed GA is given in Eq. (8).

Minimize number of opened stations =

$$\sum_{j=1}^{jmax} j \qquad (8)$$

3.4 Stopping criteria.

The stopping condition in the developed GA is approaching a certain number of generations, as shown in Table 2,

Table 2 The developed GA parameters

Parameter	Small sized problems	Large sized problems
Population size (Ps)	20	100
Crossover rate (Rc)	0.8	0.8
Mutation rate (Rm)	0.2	0.2
Elite (e)	2	5
Generations	1000	10000

4.COMPUTATIONAL RESULTS

To the author knowledge there are no benchmark problems for the developed model. For this reason, the verification of the model was analyzed in two-fold:

1- The developed GA will be solved as two-sided assembly line with two positions only and will be compared to the two-sided assembly line balancing problem (TSALBP) found in the literature. This model will be called two-sided assembly line heuristic (TSALH).

2- Then, the developed GA will be solved as multi-manned assembly line with maximum four positions and constrained with the task position. The positions of the tasks will be assumed and will be compared to the results found in point one This model will be called multi-manned multi-position assembly line with technological constraint heuristic (MALTCH).

Finally, the validation of the model is examined through solving real-life automobile factory problem. The data of real-life case study are taken from one the auto manufacturing factories in Egypt.

4.1 Comparison with benchmark problems

4.1.1 Two-sided assembly line heuristic (TSALH)

The proposed algorithm was applied to solve seven test problems taken from the literature with different sizes shown in Table 2. In this comparison the max number of workers (K) in the proposed model will be two as in the TSALBP and there will be only two positions are available which are left side and right side only: P1P2, P3P4 respectively. Error! Reference source not found. summarizes the results of TSALH (white columns), as well as TSALPB obtained by using the heuristics presented by Kim et al. [17] using a Genetic Algorithm (GA), Lee et al. [18] using group assignment procedure (GAPR Ozcan and Toklu [19] using Tabu Search Algorithm (TSA). Figure 1 shows the comparison between the results of the proposed GA solving the TSALBP of 205 tasks. The number of stations (NS) which is equal to the number of workers (K) is the solution assessment metric that is considered.

Table 2 Benchmark problems

Problem name	Problem size type	Reference
P9	small	Kim et al. (2000)
P12	small	Kim et al. (2000)
P16	small	Kim et al. (2000)
P24	medium	Kim et al. (2000)
P65	medium	Lee et al. (2001)
P148	large	Lee et al. (2001)
P205	large	Lee et al. (2001)

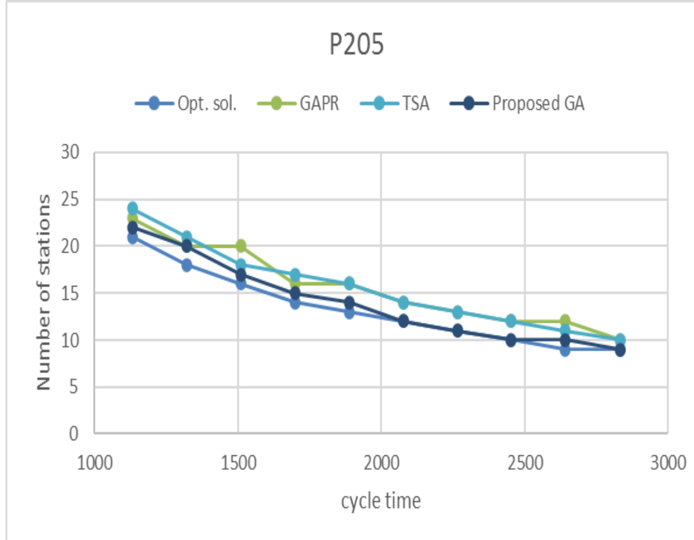

Figure 1 comparison between the proposed GA and the results of benchmark problems

From Table 3 and Figure 1 the proposed algorithm could find optimum and near optimum solution in most cases and it gives better results than other algorithms in some instances especially in large sized problem which verify the effectiveness of the proposed method

4.1.2 Multi-manned multi-position assembly line with technological constraint heuristic (MALTCH)

The results of MAL balancing problem when number of workers equal four and the tasks are constrained with their position in the product compared to TSALH is added to Table 3 (grey columns). In this comparison the position of the tasks is assumed. The numbers of stations (NS) which resemble to number of mated station (NMS) in TSALH, the number of workers (K), the percentage of increase in number of workers, and the percentage of decrease in the total line length are considered the three solution evaluation criteria for assessing the performance of the model. The percentage of increase in number of workers and the decrease in line length of the proposed MALTCH compared to TSALH for large and medium sized problem (P148and P65) are shown in Figure 2 and Figure 3

Table 3 MALTCH compared to SALBP.

Problem	CT	Opt. sol.	k=2 and position =2 GA K/NS	GAPR K/NS	TSA K/NS	Proposed GA two-sided K/NS	Proposed GA two-sided NMS	k=4 and position =4 Proposed GA -multi manned K	NS	% Increase of K	% Decrease in line length
P9	3	6	6	-	6	6	3	8	3	25%	0%
	5	4	4	-	4	4	2	6	2	33%	0%
	6	3	3	-	3	3	2	5	2	40%	0%
p12	4	7	-	-	-	7	4	11	4	36%	0%
	5	5	6	-	6	6	3	9	3	33%	0%
	6	5	-	-	5	5	3	7	3	29%	0%
	7	4	5	-	4	4	2	7	3	43%	-33%
	8	4	4	-	4	4	2	7	2	43%	0%
P16	16	6	-	-	-	6	4	9	4	33%	0%
	19	5	-	-	-	5	3	8	3	38%	0%
	21	4	-	-	4	**5**	3	8	3	38%	0%
	22	4	-	4	-	4	2	7	3	43%	-33%
P24	18	8	8	-	8	8	4	13	4	38%	0%
	20	7	-	-	8	**8**	4	10	4	20%	0%
	24	6	6	-	6	6	3	9	3	33%	0%
	25	6	5	-	6	6	3	9	3	33%	0%
	30	5	5	-	5	5	3	8	3	38%	0%
	35	4	4	-	4	4	2	7	2	43%	0%
	40	4	-	-	4	4	2	7	2	43%	0%
P65	326	16	-	17	17	**17**	9	21	7	19%	29%
	435	12	-	13	13	**13**	7	16	4	19%	75%
	490	11	-	12	11	11	6	15	4	27%	50%
	544	10	-	10	10	10	5	13	4	23%	25%
P148	204	26	-	27	26	**27**	14	32	10	16%	40%
	255	21	-	21	21	21	11	26	8	19%	38%
	306	17	-	18	18	**18**	9	23	7	22%	29%
	357	15	-	15	15	15	8	19	6	21%	33%
	408	13	-	14	13	13	7	17	5	24%	40%
	459	12	-	13	12	12	6	15	4	20%	50%

Negative value in percentage of decrease in line length means there is an increase in the line length

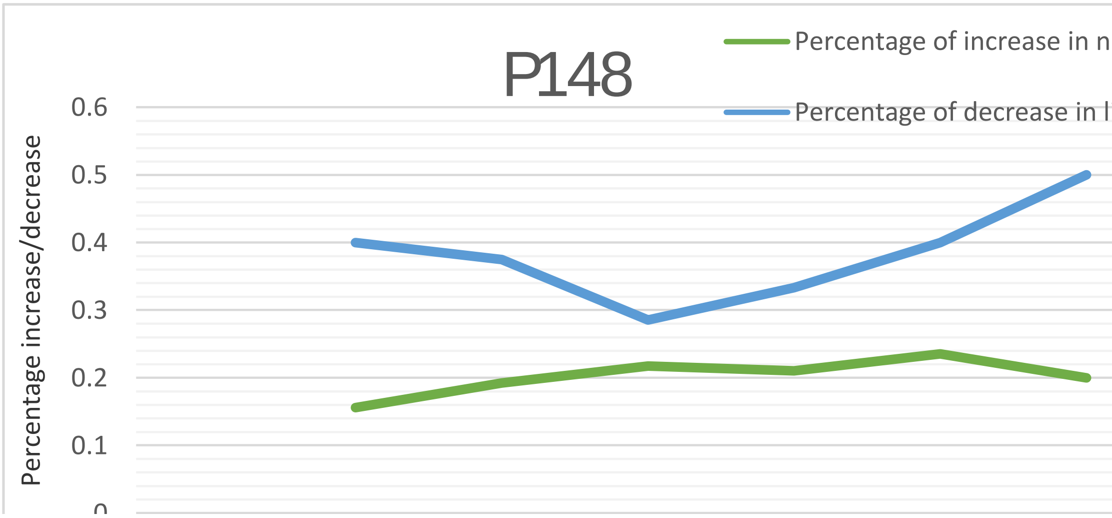

Figure 2 The percentage of increase in number of workers and the decrease in line length of the proposed MALTCH compared to TSALH for P148

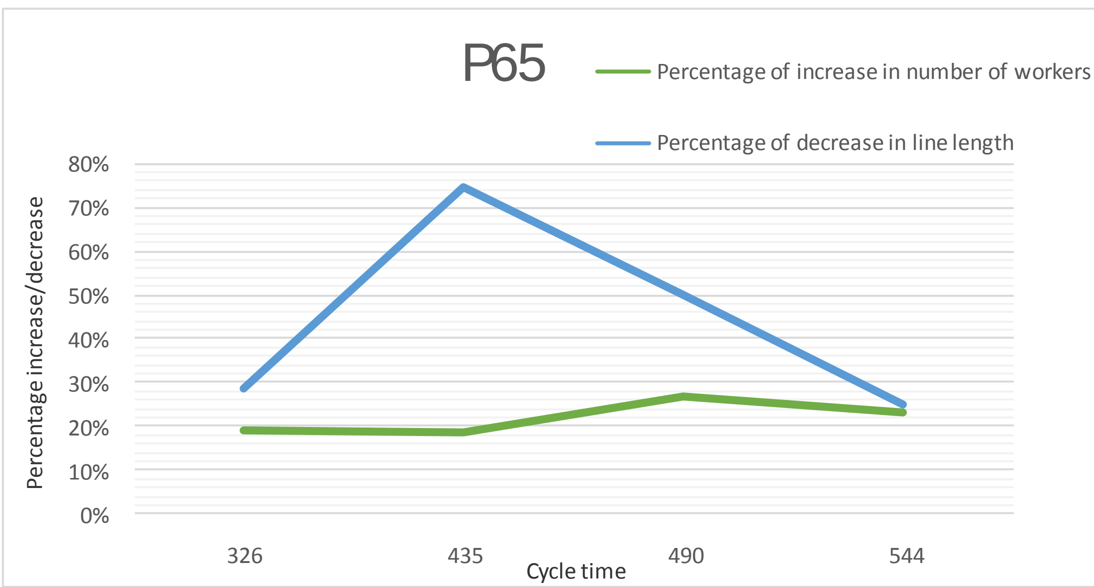

Figure 3 The percentage of increase in number of workers and the decrease in line length of the proposed MALTCH compared to TSALH for P65

The benefit of the MAL balancing line under the defined constraint over the Two-sided ALB can be seen in Table 3 as it saves from 25% to 75% of the line length, which means better use of the available space, lower cost of tools and fixtures, and less worker movement. However, by comparing the number of workers in the developed algorithm by the TSALBP, there is an increase with range from 15% to 43%. This is because that the position of the tasks was taken into consideration to prevent interference between workers as for being more realistic.

As the constraints increase, the problem becomes more complex, and the chances of finding the optimum solution decrease. From Figure 2 and Figure 3, it can be also concluded that for medium and large sized problems, the percentage of decrease in line length is greater than the percentage of increase in number of workers.

4.2 Real life case study

In this section, a real-life problem that arises in an auto manufacturing company in Egypt is presented. The company assembly line, which consists of three workshops, produces 24 cars per day in one shift of 8 hours. The first workshop is for welding, the second for painting, and the third for assembly line. As the company needs to meet customer demand while keeping labor costs to a minimum, it decided to rebalance its line with a new cycle time. In this article, the problem in the assembly line workshop is tackled. The current assembly line workshop contains 170 work elements (tasks) with cycle time equal to 17 min. The objective of the model is to balance the assembly line and increase the production volume from 24 cars per day to 40 cars per day with new cycle time 12 min.

The assembly line workshop is divided into three sections, each of which has 14 stations, nine of them is in trim (T) assembly line, two stations is in underbody (UB) and three stations is in finish (F) as shown in Figure 4. The first section, the Trim line (T), is made up of nine stations on a movable conveyer, with inventory for trim stations on both sides of the conveyer and two monorails at the start and end of the conveyer. There are 103 tasks assembled on that line. The second section, the underbody line (UB), is made up of two stations that use monorail to lift the car for installation of all parts. There are 22 tasks are assembled on that line. The third section, the finish line (F), consists of three stations; in this section, the car's tires will be on the floor, and the worker will push it. There are 45 tasks were completed at the finish line.

In the current case the product is constrained by side constraint and station position constraint. **In side constraint** the tasks is constrained by its side left, right side or can be done in right or left position so those tasks will be either tasks. **In station position constraint**, each task is constrained to its installation and position in the car, as well as the location of labor.

Positions (2,2 – 2,3 – 3,2 – 3,3 – 4,2 – 4,3 – 5,2 – 5,3) are represent the location of the tasks in the car. Positions (1,1 – 1,2 - 1,3 - 1,4 - 2,1 - 2,4 - 3,1 - 3,4 - 4,1 - 4,4 - 5,1 - 5,4 - 6,1 - 6,2 - 6,3 - 6,4) are represent the permissible position of labor around the car as shown in Figure 5. The car will be divided into four blocks (P1, P2, P3, P4) each block as seen in Figure 5. Task installation position in the car and labor permissible position around the car and their corresponding position in the station are shown in Table 5 and Table 6.

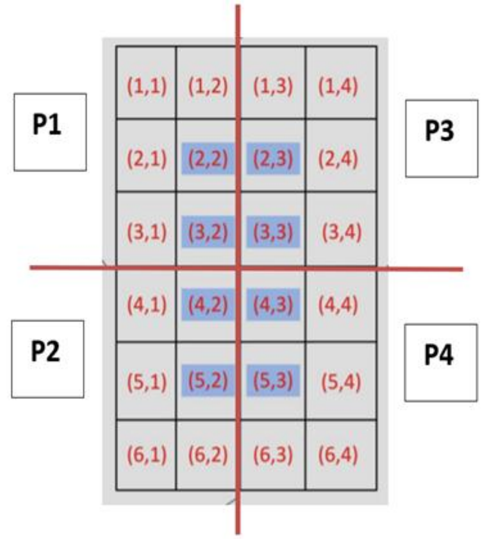

Figure 9 Car layout shows the positions of labors and component on the car.

Table 4 Task installation position in the car and its corresponding position in the station

Task installation position in the car as figure 9	**Corresponding tasks position in the station**
2,2 and 3,2	P1
2,3 and 3,3	P3
4,2 and 5,2	P2
4,3 and 5,3	P4

Table 5 labor permissible position around the car and its corresponding position in the station

Labor permissible position around the car as figure 9	**Corresponding labor position in the station**
1,1 1,2 2,1 and 3,1	P1
1,3 1,4 2,4 and 3,4	P3
4,1 5,1 6,1 and 6,2	P2
4,4 5,4 6,3 and 6,4	P4

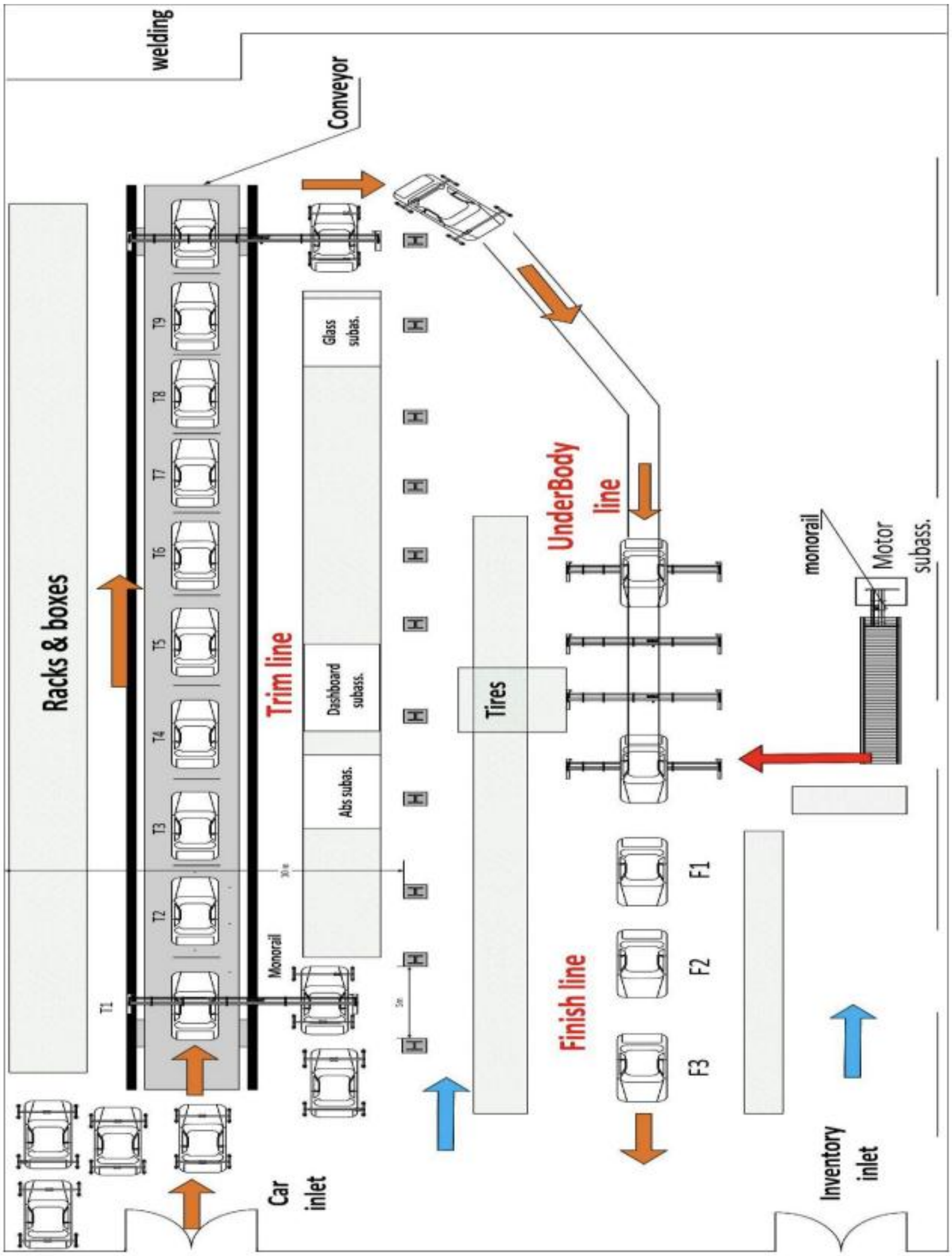

Figure 4 Auto-manufacturing company assembly line workshop layout

Table 6, summarize the current state of the company and the solution obtained by the proposed algorithm. the proposed algorithm is solved under the following constraints:

- Precedence constraints.
- Task side constraint (left, right and either).
- station position constraint (P1, P2, P3 and P4)
- single objective genetic algorithm (minimize number of workers)

The space utilization of the line is calculated by the following equation:

$$\text{Space utilization} = 1 - \frac{\left(NS - \frac{\sum_{i=1}^{n} t_i}{CT}\right)}{\frac{\sum_{i=1}^{n} t_i}{CT}} * 100$$

Table 6 The actual state in the company and the result obtained by proposed GA

	Current state of the auto company	Solution obtained by the proposed algorithm
Number of workers	39	29
Number of stations	15 stations	12 stations
Cycle time	17 min	17 min
Total tasks time	176.86	176.86
Available time (NS*CT)	255	204
Space utilization	63.63%	90.9 %

From Table 6 the proposed algorithm yields improved results regarding the number of workers and number of stations. The reduction in manpower and number of stations is 34.4 % and 25 % respectively which prove the applicability of the proposed algorithm.

5.CONCLUSION

To approach a real-life case, multi-position multi-manned assembly line balancing problem (MPMALPB) with technical constraints is addressed in this paper. A genetic algorithm with a new constraint is defined (technological constraint) where the tasks are classified according to its installation place in the product and technology required for it in the station.

The results reveal the effectiveness of the proposed model as it could find optimum and near-optimal solutions in most cases when compared to the two-sided assembly line benchmark problem, and it also outperforms other algorithms in some cases. The findings also show that multi-manned multi-position assembly line under technological constraint is better than simple assembly line under the same condition in space utilization. The proposed model saves between 25% and 70% of the line length, resulting in better use of available space, lower costs for equipment and fixtures, and less worker movement. However, the algorithm's success in providing the optimal number of workers with the new proposed constraint is doubtful. Finally, a real-world case study was solved to test the applicability of the newly identified problem. The findings demonstrated the applicability of the proposed model, as there was a 34.4 percent reduction in manpower and a 25 percent reduction in the number of stations, respectively.

For future work, in this algorithm the worker is constrained with only one position and cannot move within this position. Moreover, it is required to make further research to allow mobility of worker between two adjacent positions if the idle time of worker is greater than the upper bound which can give better result for number of workers. The model can be extended to take the physical effort of the workers into consideration to approach real life constraints. It can be extended to solve mixed models as well

Credit Authorship Contribution Statement

Raghada Taha: Conceptualization, algorithm, and software

Nessren Zamzam: Methadology, original draft, writing, review and editing.

Declaration of competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6.REFERENCES:

[1]. Dimitriadis, S. G. (2006). Assembly line balancing and group working: A heuristic procedure for workers’ groups operating on the same product and workstation. Computers & Operations Research, 33(9), 2757-2774

[2]. 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.

[3]. Kellegöz, T., & Toklu, B. (2012). An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations. Computers & Operations Research, 39(12), 3344-3360.

[4]. Giglio, D., Paolucci, M., Roshani, A., & Tonelli, F. (2017). Multi-manned assembly line balancing problem with skilled workers: a new mathematical formulation. IFAC-PapersOnLine, 50(1), 1211-1216.

[5]. Şahin, M., & Kellegöz, T. (2019). A new mixed-integer linear programming formulation and particle swarm optimization-based hybrid heuristic for the problem of resource investment

and balancing of the assembly line with multi-manned workstations. Computers & Industrial Engineering, 133, 107-120
[6]. 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
[7]. 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.
[8]. Fattahi, P., Roshani, A., & Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. The International Journal of Advanced Manufacturing Technology, 53(1-4), 363-378.
[9]. 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(1), 238-247.
[10]. Roshani, A., & Nezami, F. G. (2017). Mixed-model multi-manned assembly line balancing problem: a mathematical model and a simulated annealing approach. Assembly Automation.
[11]. Kazemi, A. and Sedighi, A. (2013). A cost-oriented model for balancing mixed-model assembly lines with multi-manned stations. International Journal of Services and Operations Management, 16(3), 289–309
[12]. Yilmazaz, H. and Yilmaz, M. (2019). A mathematical model and tabu search algorithm for multi-manned assembly line balancing problems with assignment restrictions. Engineering Optimization, 1–19.
[13]. Chen, Y.Y., Cheng, C.Y. and Li, J.Y. (2018). Resource-constrained assembly line balancing problems with multi-manned stations. Journal of Manufacturing Systems, 48, 107–119.
[14]. Chen, Y. Y. (2017). A hybrid algorithm for allocating tasks, operators, and workstations in multi-manned assembly lines. Journal of manufacturing systems, 42, 196-209
[15]. 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
[16]. Taha, R. B., El-Kharbotly, A. K., Sadek, Y. M., & Afia, N. H. (2011). A Genetic Algorithm for solving two-sided assembly line balancing problems. Ain Shams Engineering Journal, 2(3-4), 227-240.
[17]. Kim, Y. K., Kim, Y., & Kim, Y. J. (2000). Two-sided assembly line balancing: a genetic algorithm approach. Production Planning & Control, 11(1), 44-53.
[18]. Lee, T. O., Kim, Y., & Kim, Y. K. (2001). Two-sided assembly line balancing to maximize work relatedness and slackness. Computers & Industrial Engineering, 40(3), 273-292.
[19]. Özcan, U., & Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. Computers & Industrial Engineering, 57(1), 217-227.