A Heuristic Approach for Solving Robotic Assembly Line Balancing Problems

Nilufer Pekin Alakoc
Independent Researcher, Bursa, Turkiye
npekinalakoc@gmail.com

Hedi Mhalla
College of Engineering and Technology, American University of the Middle East, Kuwait
hedi.mhalla@aum.edu.kw (corresponding author)

Received: 7 December 2024 | Revised: 8 January 2025 | Accepted: 12 January 2025

ABSTRACT

This study proposes a heuristic algorithm to balance Robotic Assembly Lines (RAL). A flexible line is assumed in which robots can be allocated to any station, perform any task, and have fixed setup costs. To consider both robot allocation costs and limiting the number of stations, the current work aims to minimize the system cost, which includes new station and robot allocation costs. It evaluates the performance of the algorithm with a large set of randomly generated samples and conducts statistical analyses to summarize, compare, and draw conclusions. The experimental results demonstrate the efficacy of the proposed algorithm in addressing large-scale problems in a reasonable timeframe.

Keywords-robotic assembly line balancing; heuristic algorithms; integer programming

I. INTRODUCTION

The field of Simple Assembly Line Balancing Problems (SALBP) has been the focus of scholarly inquiry for over fifty years. In response to the complexity of these problems, numerous variants have been developed, and researchers and manufacturers have proposed a range of resolution methods. This particular interest stems from the necessity to extend the field and consider multiple factors concurrently to adapt to real-life production systems. In the highly competitive global marketplace, industries have been compelled to modify their operational strategies, adopt cutting-edge technological advancements, and align with the principles of Industry 4.0 by incorporating robotics and automated equipment into their production systems. The growing significance of robots has had substantial ramifications for industrial applications and individual lifestyles [1-3]. Consequently, the demand for RAL has witnessed a substantial surge in recent decades. The Robotic Assembly Line Balancing Problem (RALBP) has emerged as a prominent branch of the SALBP. The RALBP can be defined as the optimization of the production process by the simultaneous assignment of tasks and robots to the line to increase production efficiency. The field of robotics and equipment assignment problems has garnered significant interest from researchers and manufacturers alike, leading to the publication of numerous academic papers since the subject's inception [4]. Analogous to the SALBP, the RALBP can be classified according to various characteristics, including task time, layout structure, objective criteria, and solution methodology [5]. This section aims to provide a concise overview of the extant literature on the subject. A substantial proportion of these earlier studies involved simple extensions of SALBP, wherein the term equipment was substituted for robots. Some researchers considered parallel layouts to enhance line flexibility and capacity [6-9]. For instance, authors in [10] examined the Robotic Parallel Assembly Line Balancing Problem (RPALBP). They explored the potential of iterative beam search and cutting algorithms as solutions to this problem. Large-scale and high-volume product manufacturing frequently uses two-sided assembly line layouts. Metaheuristic methods have been proposed to address these balancing problems and have been compared with existing exact or heuristic algorithms [11-13]. In contrast, authors in [12] sought to address the two-sided lines by simultaneously minimizing energy consumption and cycle time. Comparative studies were also conducted for the robotic U-shaped assembly line balancing problems [14-16]. The integration of human flexibility and robot productivity has been identified as a means to enhance the efficient usage of resources in the context of line balancing. Literature has also explored the problem of assembly line balancing through human-robot collaboration [17-21].

Authors in [22] have previously indicated that exact and heuristic methods can be employed to solve RALBP. However, the intricate nature of the problem poses significant challenges in terms of its resolution using exact methods within a reasonable timeframe, even for the most simplified version of the problem. The extant literature indicates that exact methods, such as branch and bound, demonstrate optimal performance primarily on problems of a limited size [23-25]. Consequently, a significant proportion of studies have advocated for the utilization of heuristic methods [26-28], which have been developed with metaheuristic algorithms and those that employ the particle swarm optimization method, a population-based metaheuristic [29, 13, 14]. Consequently, the solutions to RALBP have the potential to enhance system productivity, product quality, and safety in manufacturing environments, thereby ensuring competitiveness in the market. The multifaceted objectives of manufacturers and the varied structures of production systems create a fertile ground for research. The study's objective is twofold: first, to develop a novel approach to address the challenges posed by the intricate nature of the problem and second, to make a meaningful contribution to the extant literature. This study proposes a novel heuristic approach for RALBP, with the objective of minimizing the total cost associated with the number of stations and robots. The proposed algorithm accounts for several realistic constraints. A notable advantage of the proposed algorithm is its capacity for facile modification by users, hence facilitating the incorporation of additional constraints. This feature enables the execution of sophisticated analyses, such as sensitivity analysis, parametrization of the problem, and resolution when problem instances are subject to variability, a common occurrence in industrial processes. As previously mentioned in this section, to the best of the authors' knowledge, the existing algorithms in the literature do not offer the same adaptability and are less designed for extensions.

II. PROBLEM DEFINITION AND MATHEMATICAL MODEL

The objective is to allocate tasks to stations, along with the requisite robots for processing, aiming to minimize the system cost, which encompasses the expenses associated with station opening and robot allocation. Given the intricate nature of RAL, characterized by complex structures, chaotic behaviors [30], and a multitude of variations, the problem is constrained by the assumptions that the cycle time is known and constant, the task times are deterministic and dependent on the robot, each task requires one robot, robot costs are constant for each station and include setup, purchasing, maintenance, and consumption costs, and any robot can be assigned to any station without limitations on the number or placement of robots. To simplify the problem, it is also assumed that all other variables, costs, and times are included or negligible. While it is acknowledged that any robot can execute all tasks, it is postulated that there exists a negative correlation between task times and robot costs. The objective of this study is to minimize the total system cost, which is expressed as:

$$TC = \sum_{k=1}^{S} \sum_{j=1}^{r} RC_j y_{jk} + S \times SC \tag{1}$$

with the constraints:

$$\sum_{k=1}^{S} \sum_{j=1}^{\mathrm{r}} x_{ijk} = 1, for\ all\ i \tag{2}$$

$$\sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{r}} t_{ij}\, x_{ijk} \leq C, for\ all\ k \tag{3}$$

$$\sum_{k=1}^{S} \sum_{j=1}^{\mathrm{r}} k x_{ijk} \leq S, for\ all\ i \tag{4}$$

$$\sum_{j=1}^{\mathrm{r}} \sum_{k=1}^{S} k x_{ajk} \leq \sum_{j=1}^{\mathrm{r}} \sum_{k=1}^{S} k x_{bjk}\ , for\ all\ a, b \tag{5}$$

$$x_{ijk} \leq y_{jk}, for\ all\ i, k, j \tag{6}$$

where C is the cycle time, S is the number of workstations, n is the number of tasks to be assigned, r is the number of robots, t_{ij} is the processing time of task i, when performed by robot j, RC_j is the cost of robot j, SC is the station opening cost,

$$x_{ijk} = \begin{cases} 1 & \text{if task } i \text{ is performed in station } k \text{ by robot } j \\ 0 & \text{otherwise} \end{cases},$$

$$y_{jk} = \begin{cases} 1 & \text{if robot } j \text{ is assigned to workstation } k \\ 0 & \text{otherwise} \end{cases},$$

where $i = 1, \ldots, \mathrm{n}$, $j = 1, \ldots, \mathrm{r}$, $k = 1, \ldots, \mathrm{S}$.

Constraint set (2) guarantees that each workstation is assigned to only one task. The capacity constraint set (3) assures that the capacities of the stations are not exceeded. Constraint (4) imposes a limit on the minimum number of workstations required to complete all tasks, while (5) guarantees that the precedence relations are respected for all tasks and (6) stipulates that if a task is processed by a workstation, then the task's robot is assigned to that workstation.

III. SOLUTION ALGORITHM

This study proposes a heuristic algorithm, a fast constructive algorithm driven by a 2-opt method. The algorithm facilitates the attainment of a solution in a substantially reduced processing time. The algorithm is delineated in three primary phases. Initially, the data pretreatment phase is undertaken, subsequently constructing the initial partial solution. The improvement phase is then applied, enabling the diversification of the search and the eventual completion of the solution.

A. *Pretreatment*

This phase involves a rapid prioritization process to prepare the data for the subsequent construction phase. By definition, tasks can be addressed by at least one robot, though in most cases, they are executed by multiple robots. The initial step entails the establishment of a selection criterion to determine which robot will be assigned to each task. It is possible to define different selection rules during this phase. The implementation of a sequential approach is proposed, wherein the available robots are arranged in an ascending order based on their designated processing times. To this end, the robots are indexed from 1 to r based on the increasing order of their processing times. For any given task i, the robots are then indexed as $j_1\ldots, j_r$, such that $t_{ij1} \leq \cdots \leq t_{ij_r}$. It is imperative to emphasize that r signifies the maximum number of robots.

B. *Construction*

In this phase, tasks are allocated to robots according to their pretreatment order, and a group is formed for each robot, denoted by G_j. In the event that tasks exhibit precedence relationships, the objective is to assign them to the same group, therefore considering them as a single entity. The associated time for this entity is defined as the total time required for the same robot to execute the tasks. The objective of this grouping is to divide the primary problem into smaller components, thereby facilitating a more straightforward approach to the solution. The groups and the tasks assigned to each group are then considered individually. This approach ultimately reduces the complexity of the main problem to r SALBPs. The total time allocated to robot j is $T_{Tot}(j) = \sum_{i=1}^{n} z_{ij} t_{ij}$, where z_{ij} is 1 if task i is assigned to robot j and 0 otherwise. So, the minimum number of stations needed for each group is denoted by $\mathrm{S}(j)$ and defined as $\mathrm{S}(j) = \lfloor T_{Tot}(j)/\mathrm{C} \rfloor$. The objective function is associated with the total time and the allocated robot for each task, as well as the costs of the stations and robots required. It is even more appropriate to say that $\sum_{j=1}^{r} \mathrm{S}(j) \times RC_j + \sum_{j=1}^{r} \mathrm{S}(j) \times SC$ is a lower bound to the optimal solution. Subsequent to the task of grouping, an allocation approach is employed for the purpose of assigning the groups to the stations. Given the fact that the allocation approach affects the efficiency of the main approach, it is considered to be at the core of the algorithm. Two allocation approaches are put forth for consideration. The first approach is a greedy algorithm that considers one station at a time. The second approach seeks a balanced allocation among the stations of each group G_j.

- The initial approach entails the consideration of one station at a time, with the objective of assigning tasks while minimizing idle time. This approach is characterized by its stability and efficiency. For group G_j, initially, there are $S(j)$ stations, with robot j being allocated to each. The algorithm commences with the assignment of tasks from group G_1 and progresses sequentially until all the available groups have been covered (up to a maximum of r groups). Within each group, tasks are prioritized based on their processing times, with the objective of minimizing the idle time of the current station. In the event that no tasks from the current group can be allocated to the current station, the next station in the set $S(j)$ is opened, and the process is repeated until all $S(j)$ stations have been processed. Subsequent to the assignment of all groups, any residual tasks, if applicable, are addressed in the final phase of the algorithm. This approach is founded on a constructive method applied to a single station at a time, which can be assimilated to a greedy algorithm employed for the stations, yet in a sequential manner. Greedy algorithms are recognized for their efficiency and expediency in general. In the current study, this approach leads to a good optimization of the first stations relative to the later ones, which can be relatively corrected and improved globally in the final phase.

- The second approach is designed with a strong emphasis on equally balancing the stations. The algorithm considers groups and the corresponding stations simultaneously and assigns tasks (always considering the precedence rules) to the stations in a descending order of the idle times of the stations. In summary, tasks are allocated with the objective of minimizing the maximum idle time among all stations within each group, denoted by G_j. The objective is to distribute and balance the load of the stations as uniformly as possible, drawing inspiration from well-known combinatorial optimization problems of type knapsack. The steps of certain algorithms draw inspiration from the binary knapsack-sharing problem resolution, where the fundamental principle remains the distribution of a global capacity among different groups [31].

C. *Final Phase*

The current solution is incomplete. Prior to the completion of the construction of the proposed solution, an improvement procedure is applied. This improvement procedure is based on a two-stage swapping procedure of type 2-opt. The first stage involves a 2-opt among the tasks that have already been assigned, while the second stage involves a 2-opt among the tasks that have been assigned and those that have not been assigned yet. It is important to note that both stages of this procedure result in an enhancement of the current solution, thereby reducing the total cost. The swapping of tasks between the stations enables the application of perturbations to the structures of some stations, primarily to change tasks to stations with robots different from the initially attributed robot and following the initially defined order of robots. In the subsequent stage, representing the final improvement step, the assignment of the remaining tasks, if any, is finalized. To accomplish this, the overall time for all remaining tasks is collected, and the minimum number of stations required to accommodate all tasks is determined. At this juncture, the attention is turned to the construction phase groups, which were initially assigned. To optimize the usage of the available capacity, stations that retain sufficient residual time for the remaining tasks are incorporated. The allocation of these stations is determined by the availability of the robot(s) assigned to each station and the robot allocated to the tasks in the construction phase. To ensure precision, an extensive tree is employed, encompassing all potential outcomes for each task. However, as the process progresses, the optimal alternatives are retained, enabling efficient navigation through the construction phase. It is evident that the tree search conducted at the culmination of the algorithm is a beam search algorithm applied exclusively to the reduced instance of the problem, which corresponds to the remaining items following the execution of the algorithm's earlier steps. The problem at hand is characterized by a complex structure and two dimensions, namely the number of stations and robot costs, which must be considered simultaneously. To overcome this structure and avoid eliminating viable solutions, the best l feasible solutions obtained through the combination of the robots are considered. Half of these solutions ($l/2$) correspond to the lowest time, and the remaining half to the lowest cost. The value of l, which is a predefined number of solutions, is examined in the subsequent section.

IV. RESULTS AND DISCUSSION

This section will discuss the algorithm's primary findings, which were coded in C++ and tested on the AUM-Phenix high-performance computing facility. The AUM-Phenix facility is a state-of-the-art system known for its exceptional processing power and speed. The facility consists of ten Dell PowerEdge R730 servers, each of which is equipped with an Intel Xeon E5-2698 v3 2.3GHz processor. To carry out the performance analysis, random problem instances were generated. A comprehensive summary of the outcomes is presented, accompanied by a detailed discussion that explores the efficiency and behavior of the algorithm in addressing problems of varying dimensions. This analysis is supported by rigorous statistical examinations. The distribution of the processing times follows a uniform distribution with parameters [5, 20], while the distribution of robot costs follows a uniform distribution with parameters [20]. The number of robots available for each task varies between six and ten, reflecting real-world scenarios. The station data are also generated randomly but remain indexed on the overall task data. To this end, two random values are employed, designated as x_1 and x_2, which adhere to a uniform distribution with the parameters [5, 20]. The cycle time is defined as the sum of all the minimum times required to complete each task, divided by x_1. The cost of the station is determined randomly as the median cost of all the robots multiplied by x_2. The production lines encompass a wide range of tasks, from 50 to 500, ensuring the direct applicability of this study's findings to diverse industrial settings. To ensure the reliability of the findings, the data are made more realistic by considering precedence constraints for 20% of the tasks, a common occurrence in practical applications. For each problem size and number of robots, the current work generates $m = 4$ random problem instances. The determination of lower bounds on the number of stations and total cost, LB(S_{nr}) and LB(TC_{nr}), for each problem instance is achieved through the calculation of the minimum number of stations required to complete the tasks and the assignment of the robot with the lowest cost to each station. This approach entails the relaxation of additional task-related constraints, including precedence, time, and resource limitations on task execution. While the solution may not be feasible, it serves as a robust metric for assessing the performance of the algorithm. The random problems demonstrate that a significant proportion of the solutions is not feasible but they provide an excellent indicator to measure the algorithm's performance. To calculate the lower bound on the total cost, the minimum number of stations is considered and the robot with the lowest price is allocated to each station. This approach enables the disaggregation of some of the constraints. Through the analysis, the attained solution is compared to the obtained lower bound in terms of the number of stations and cost. Table I offers a synopsis of the experimental outcomes. The first two columns present, respectively, the size and the number of robots considered in the generated instances.

The performance variables are defined by two statistics: the average difference in the number of stations, $\overline{DS}$, and the average ratio of each solution relative to the lower bound, $\bar{P}$. $\overline{DS}$ is the average difference between the algorithm's solution and the one provided by the lower bound for each combination of categories (a couple of values of the number of tasks n, and the number of robots, r) and is calculated as: $\overline{DS} = \frac{1}{m}\sum_1^m(\mathrm{S_{nr}} - \mathrm{LB(S_{nr})})$, where S_{nr} is the number of stations obtained by the approach. As portrayed in Table I, columns 3 and 5 present the mean values for the two approaches delineated in the construction phase. Finally, columns 4 and 6 present the average performance of the algorithms' solutions for the problem instances generated for each category. The latter performance value is calculated as the ratio of the algorithm's solution divided by the corresponding lower bound of the instance $\bar{P} = \frac{1}{m}\sum_1^m \frac{\mathrm{TC_{nr}}}{\mathrm{LB(TC_{nr})}}$. This performance indicator, which must be greater than or equal to one, serves as an objective metric for evaluating the efficiency of the algorithm. Solutions that closely approach one will demonstrate optimal performance. A comparison of the mean ratios of the solutions reveals that approach 1 exhibits superior performance for problem instances of small size (n = 50, r = 6, 8, and n = 100, r = 6), while approach 2 demonstrates superiority over all other pairs. It is evident that the second approach exhibits superiority over the first in terms of the mean discrepancy in the number of stations. Furthermore, the second approach demonstrates enhanced stability in comparison to the initial approach. The initial approach is more sensitive to the nature of the instances tested, and in the improvement phase (2-opt), it exhibits a decline in efficiency when a substantial number of priority rules are present. In contrast, the second approach leverages the structure of the grouping procedure, enhancing its efficiency, while the swapping approach (2-opt) offers additional improvements.

TABLE I. SUMMARY OF THE ALGORITHM PERFORMANCE

		Approach 1		Approach 2	
n	r	$\overline{DS}$	$\bar{P}$	$\overline{DS}$	$\bar{P}$
50	6	0	1.05	0	1.10
50	8	0.25	1.15	0	1.17
50	10	0.25	1.14	0	1.12
100	6	0	1.05	0	1.06
100	8	0.25	1.12	0	1.10
100	10	0.25	1.16	0	1.11
250	6	0.25	1.13	0	1.07
250	8	0.25	1.19	0	1.14
250	10	0.25	1.16	0.25	1.12
500	6	0.25	1.13	0	1.08
500	8	0.25	1.14	0	1.07
500	10	0.25	1.11	0.25	1.07

In addition to calculating the performance averages and providing support for the initial findings, this study conducted a three-way ANOVA to effectively compare the two approaches and investigate the effects of the number of robots and tasks on performance. While the construction principles of the two approaches are comparable, their respective designs differ, impacting the performance of the main approach in terms of the quality of the solutions and execution time. In this study, a set of randomized tests was conducted to compare the two approaches using ANOVA tests. The experimental design encompasses three factors: the Approach (AP), the Number of Robots (NR), and the problem Size (S). The first factor has two levels, the two approaches under consideration, the second factor has three levels (6, 8, and 10), and the third one has four levels (50, 100, 250, and 500). The detailed results of this analysis can be found in Table II.

Table II presents the F statistics and their respective p-values, which were determined through the statistical analysis of the primary effects of the three factors, namely the approach and the number of tasks and robots. The p-values (3.53e-10, <2e-16, and 1.35e-10), which are less than the significance level of 0.05, indicate that the three factors have significant effects on the algorithm's performance. The interaction effects of the three factors are incorporated into the ANOVA model. The presence of small p-values (0.02122, 2.14e-9, 1.74e-10, 0.00272, and 0.00265) further substantiates the significance of these interactions. This finding suggests that the levels of the three factors interact with each other. The validation of the primary assumptions of ANOVA is imperative for the generation of reliable interpretations. To this end, a residual analysis has been conducted to assess the validity of these assumptions. The histogram of residuals displays a symmetrical distribution. The normality assumption is then tested using a q-q plot and the Shapiro-Wilk test, which indicates a normal distribution with a p-value greater than the significance level of 0.05. Subsequently, a Levene's test was implemented to assess the homogeneity of variance. The Levene's test was not significant (p-value=0.464>0.05). Consequently, the homogeneity of variances across the distinct groups can be assumed. Given the significance of the performed analysis, this study proceeds to conduct post hoc tests. Initially, simple two-way interactions are undertaken, incorporating the approach as the third variable. A statistically significant simple two-way interaction between the size and the number of robots for both approaches is observed, as evidenced by p-values smaller than the Bonferroni-adjusted alpha level of 0.025. Specifically, the p-value for Approach 1 is 1.93e-08, and for Approach 2, it is 1.45e-05. Given the significance of these two-way interactions, this work proceeded to compute simple simple main effects of the size of the problem on performance. The data are grouped by the approach and number of robots, and six tests are conducted. The resulting p-values, which are used to determine the statistical significance, are 1.45e-10 (approach 1, r=6), 1.05e-5 (approach 1, r=8), 0.0005 (approach 1, r=10), 0.0143 (approach 2, r=6), 4.19e-11 (approach 2, r=8), and 0.0002 (approach 2, r=10). It is evident that the size of the problem is a substantial factor in the performance, as evidenced by the findings of this study. Subsequently, to ascertain the means of the distinct groups, a comprehensive calculation of all multiple pairwise comparisons for the six pairs is conducted, employing the Tukeys HSD test. The results of these statistical tests are depicted in Table III, which displays the p-values of the respective tests.

TABLE II. THREE-WAY ANOVA: PERFORMANCE VERSUS SIZE AND APPROACH

	DF	Adj SS	Adj MS	*F*-Value	*P*-Value
AP	1	0.01733	0.017334	52.829	3.53e-10
NR	2	0.04973	0.024866	75.781	<2e-16
S	3	0.02279	0.007595	23.148	1.35e-10
AP×NR	2	0.00267	0.001334	4.067	0.02122
AP×S	3	0.01930	0.006434	19.610	2.14e-9
NR×S	6	0.02756	0.004593	13.999	1.74e-10
AP×NR×S	6	0.00736	0.001226	3.737	0.00272
Error	72	0.02362	0.000338		
Total	95	0.17036			

The interactions between the three factors can be also examined graphically. Figures 1 and 2 illustrate the interactions for the two approaches, respectively. These figures plot the group averages against the levels of the size and the number of robots. It is evident from these plots that they do not align in a consistent pattern, thereby substantiating the conclusion regarding the interaction of the three factors. Furthermore, it can be concluded that both approaches demonstrate enhanced performance when the number of robots is set at 8. Specifically, approach 1 exhibits optimal performance for the combination of 250 tasks and 8 robots, while approach 2 achieves its peak performance for the same pair, followed by the best performance pair of 50 tasks and 8 robots. In summary, although the overall performance of approach 2 is superior, approach 1 yields optimal results for a reduced number of robots and tasks.

TABLE III. P-VALUES FOR THE PAIRWISE COMPARISON TESTS

Pairs	Approach 1			Approach 2		
$n_{(1)}$-$n_{(2)}$	r=6	r=8	r=10	r=6	r=8	r=10
50-100	1	0.1068	0.4976	0.0292	0.0002	0.9390
50-250	0.0005	0.0253	0.4976	0.0595	0.1359	1
50-500	0.0011	0.8345	0.1877	0.3013	6.93e-6	0.0084
100-250	0.0005	0.0004	1	0.9758	0.0102	0.939
100-500	0.0011	0.3746	0.0163	0.5036	0.0955	0.0227
250-500	0.9482	0.0058	0.0163	0.739	0.0002	0.0084

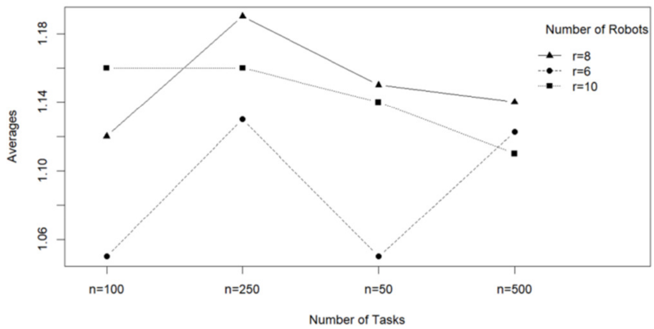

Fig. 1. Interaction plot for approach 1.

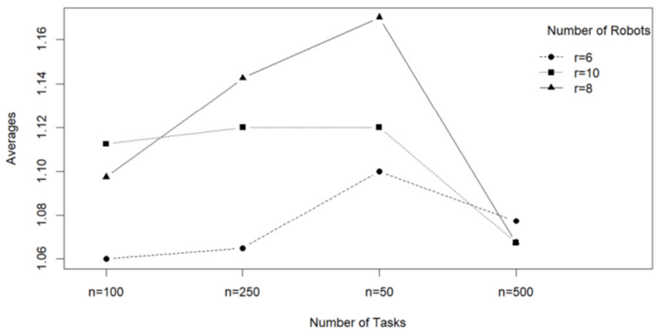

Fig. 2. Interaction plot for approach 2.

V. CONCLUSIONS

This study underscores the significance of the Robotic Assembly Line Balancing Problem (RALBP) and proposes a novel heuristic approach for addressing RALBP. The approach under consideration is characterized by its ability to address task assignment and robot allocation in a simultaneous fashion. It ensures the satisfaction of precedence constraints, balances workload, and minimizes the total cost. The approach comprises three primary phases (arrangement, construction, and improvement) and two sub-algorithms. The validity of the algorithm was ascertained through the implementation of a random sample generation process. The study encompasses not only a comprehensive summary of the computational outcomes, but also the execution of rigorous statistical tests. These evaluations are designed to assess and contrast the efficacy of the algorithm in the context of variations in the task allocation, robotic resources, and sub-strategies. The outcomes of this study demonstrate the efficacy of the proposed approach in balancing lines, thus enhancing production efficiency and reducing the total assembly costs. The algorithm's flexibility is noteworthy, as it can be adapted to accommodate additional constraints and address large problem instances. In the subsequent phase of the research, the proposed approach can be adapted to different assembly line structures, such as parallel or two-sided assembly lines. Furthermore, the approach's framework can be expanded through the integration of novel sub-approaches and the implementation of enhancement procedures, which have the potential to influence performance metrics and execution times.

ACKNOWLEDGEMENT

Calculations have been performed on the AUM-Phenix high-performance facility at the American University of the Middle East.

REFERENCES

[1] V.-T. Nguyen *et al.*, "Robust adaptive nonlinear PID controller using radial basis function neural network for ballbots with external force," *Engineering Science and Technology, an International Journal*, vol. 61, Jan. 2025, Art. no. 101914, https://doi.org/10.1016/j.jestch.2024.101914.

[2] V.-T. Nguyen, C.-D. Do, T.-V. Dang, T.-L. Bui, and P. X. Tan, "A comprehensive RGB-D dataset for 6D pose estimation for industrial robots pick and place: Creation and real-world validation," *Results in Engineering*, vol. 24, Dec. 2024, Art. no. 103459, https://doi.org/10.1016/j.rineng.2024.103459.

[3] V.-T. Nguyen, D.-N. Duong, D.-H. Phan, T.-L. Bui, X. HoangVan, and P. X. Tan, "Adaptive Nonlinear PD Controller of Two-Wheeled Self-Balancing Robot with External Force," *Computers, Materials and Continua*, vol. 81, no. 2, pp. 2337–2356, Nov. 2024, https://doi.org/10.32604/cmc.2024.055412.

[4] S. C. Graves and D. E. Whitney, "A mathematical programming procedure for equipment selection and system evaluation in programmable assembly," in *1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, Fort Lauderdale, FL, USA, Dec. 1979, vol. 2, pp. 531–536, https://doi.org/10.1109/CDC.1979.270236.

[5] P. Chutima, "A comprehensive review of robotic assembly line balancing problem," *Journal of Intelligent Manufacturing*, vol. 33, no. 1, pp. 1–34, Jan. 2022, https://doi.org/10.1007/s10845-020-01641-7.

[6] P. A. Pinto, D. G. Dannenbring, and B. M. Khumawala, "Assembly Line Balancing with Processing Alternatives: An Application," *Management Science*, vol. 29, no. 7, pp. 817–830, 1983.

[7] S. C. Graves and C. H. Redfield, "Equipment selection and task assignment for multiproduct assembly system design," *International Journal of Flexible Manufacturing Systems*, vol. 1, no. 1, pp. 31–50, Sep. 1988, https://doi.org/10.1007/BF00713158.

[8] J. Rubinovitz, J. Bukchin, and E. Lenz, "RALB – A Heuristic Algorithm for Design and Balancing of Robotic Assembly Lines," *CIRP Annals*, vol. 42, no. 1, pp. 497–500, Jan. 1993, https://doi.org/10.1016/S0007-8506(07)62494-9.

[9] J. Bukchin and M. Tzur, "Design of flexible assembly line to minimize equipment cost," *IIE Transactions*, vol. 32, no. 7, pp. 585–598, Jul. 2000, https://doi.org/10.1023/A:1007646714909.

[10] Z. A. Cil, S. Mete, E. Özceylan, and K. Agpak, "A beam search approach for solving type II robotic parallel assembly line balancing problem," *Applied Soft Computing*, vol. 61, pp. 129–138, Dec. 2017, https://doi.org/10.1016/j.asoc.2017.07.062.

[11] M. Aghajani, R. Ghodsi, and B. Javadi, "Balancing of robotic mixed-model two-sided assembly line with robot setup times," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 5, pp. 1005–1016, Sep. 2014, https://doi.org/10.1007/s00170-014-5945-x.

[12] Z. Li, Q. Tang, and L. Zhang, "Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm," *Journal of Cleaner Production*, vol. 135, pp. 508–522, Nov. 2016, https://doi.org/10.1016/j.jclepro.2016.06.131.

[13] Z. Li, M. N. Janardhanan, Q. Tang, and P. Nielsen, "Co-evolutionary particle swarm optimization algorithm for two-sided robotic assembly line balancing problem," *Advances in Mechanical Engineering*, vol. 8, no. 9, Sep. 2016, Art. no. 1687814016667907, https://doi.org/10.1177/1687814016667907.

[14] J. M. Nilakantan and S. G. Ponnambalam, "Robotic U-shaped assembly line balancing using particle swarm optimization," *Engineering Optimization*, vol. 48, no. 2, pp. 231–252, Feb. 2016, https://doi.org/10.1080/0305215X.2014.998664.

[15] Z. Li, M. N. Janardhanan, A. S. Ashour, and N. Dey, "Mathematical models and migrating birds optimization for robotic U-shaped assembly line balancing problem," *Neural Computing and Applications*, vol. 31, no. 12, pp. 9095–9111, Dec. 2019, https://doi.org/10.1007/s00521-018-3957-4.

[16] Z. Zhang, Q. Tang, and L. Zhang, "Mathematical model and grey wolf optimization for low-carbon and low-noise U-shaped robotic assembly line balancing problem," *Journal of Cleaner Production*, vol. 215, pp. 744–756, Apr. 2019, https://doi.org/10.1016/j.jclepro.2019.01.030.

[17] M. Dalle Mura and G. Dini, "Designing assembly lines with humans and collaborative robots: A genetic approach," *CIRP Annals*, vol. 68, no. 1, pp. 1–4, Jan. 2019, https://doi.org/10.1016/j.cirp.2019.04.006.

[18] Z. A. Cil, Z. Li, S. Mete, and E. Özceylan, "Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human–robot collaboration," *Applied Soft Computing*, vol. 93, Aug. 2020, Art. no. 106394, https://doi.org/10.1016/j.asoc.2020.106394.

[19] T. Koltai, I. Dimény, V. Gallina, A. Gaal, and C. Sepe, "An analysis of task assignment and cycle times when robots are added to human-operated assembly lines, using mathematical programming models," *International Journal of Production Economics*, vol. 242, Dec. 2021, Art. no. 108292, https://doi.org/10.1016/j.ijpe.2021.108292.

[20] Z. Li, M. N. Janardhanan, and Q. Tang, "Multi-objective migrating bird optimization algorithm for cost-oriented assembly line balancing problem with collaborative robots," *Neural Computing and Applications*, vol. 33, no. 14, pp. 8575–8596, Jul. 2021, https://doi.org/10.1007/s00521-020-05610-2.

[21] A. Nourmohammadi, M. Fathi, and A. H. C. Ng, "Balancing and scheduling assembly lines with human-robot collaboration tasks," *Computers & Operations Research*, vol. 140, Apr. 2022, Art. no. 105674, https://doi.org/10.1016/j.cor.2021.105674.

[22] L. Borba, M. Ritt, and C. Miralles, "Exact and heuristic methods for solving the Robotic Assembly Line Balancing Problem," *European Journal of Operational Research*, vol. 270, no. 1, pp. 146–156, Oct. 2018, https://doi.org/10.1016/j.ejor.2018.03.011.

[23] D. Ogan and M. Azizoglu, "A branch and bound method for the line balancing problem in U-shaped assembly lines with equipment requirements," *Journal of Manufacturing Systems*, vol. 36, pp. 46–54, Jul. 2015, https://doi.org/10.1016/j.jmsy.2015.02.007.

[24] N. Pekin and M. Azizoglu, "Bi criteria flexible assembly line design problem with equipment decisions," *International Journal of Production Research*, vol. 46, no. 22, pp. 6323–6343, Nov. 2008, https://doi.org/10.1080/00207540701441988.

[25] J. Gao, L. Sun, L. Wang, and M. Gen, "An efficient approach for type II robotic assembly line balancing problems," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1065–1080, Apr. 2009, https://doi.org/10.1016/j.cie.2008.09.027.

[26] M. Rabbani, Z. Mousavi, and H. Farrokhi-Asl, "Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem," *Journal of Industrial and Production Engineering*, vol. 33, no. 7, pp. 472–484, Oct. 2016, https://doi.org/10.1080/21681015.2015.1126656.

[27] Z. Li, M. N. Janardhanan, Q. Tang, and P. Nielsen, "Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line," *Engineering Optimization*, vol. 50, no. 5, pp. 877–893, May 2018, https://doi.org/10.1080/0305215X.2017.1351963.

[28] M. N. Janardhanan, Z. Li, G. Bocewicz, Z. Banaszak, and P. Nielsen, "Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times," *Applied Mathematical Modelling*, vol. 65, pp. 256–270, Jan. 2019, https://doi.org/10.1016/j.apm.2018.08.016.

[29] J. M. Nilakantan, S. G. Ponnambalam, and P. Nielsen, "Application of Particle Swarm Optimization to Solve Robotic Assembly Line Balancing Problems," in *Handbook of Neural Computation*, P. Samui, S. Sekhar, and V. E. Balas, Eds. Academic Press, 2017, pp. 239–267.

[30] M. Sajid, F. A. Almufadi, and M. Jahanzaib, "Chaotic Behavior in a Flexible Assembly Line of a Manufacturing System," *Engineering, Technology & Applied Science Research*, vol. 5, no. 6, pp. 891–894, Dec. 2015, https://doi.org/10.48084/etasr.611.

[31] H. Mhalla, "An exact constructive algorithm for the knapsack sharing problem," *Optimization Methods and Software*, vol. 32, no. 5, pp. 1078–1094, Sep. 2017, https://doi.org/10.1080/10556788.2016.1240795.

AUTHORS PROFILE

Nilufer Pekin Alakoc graduated from Middle East Technical University, Türkiye with a major in Statistics and a minor in Operations Research. She received her MSc degree in Industrial Engineering and her Ph.D. in Statistics. She has combined 18 years of experience in teaching and statistical data analyses at universities in Kuwait and Türkiye. Her research interests mainly include statistical applications in engineering, optimization, scheduling, fuzzy logic in data analysis and quality control, regression, and time series analyses.

Hedi Mhalla is an Associate professor at the American University of the Middle East, Kuwait. He obtained his PhD in Computer Science from the Université de Paris 1 Panthéon-Sorbonne, France. He received his BS in Mathematics and his M.S in Statistics and Stochastic Models for Economics and Finance from the Université de Paris 7 Diderot, France. During the last fifteen years, he acquired a rich experience in both academic and non-academic areas and he participated in various research and academic projects.