Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2026, 20(7), 325–334
https://doi.org/10.12913/22998624/220051
ISSN 2299-8624, License CC-BY 4.0

Received: 2026.01.14
Accepted: 2026.05.15
Published: 2026.06.01

Use of cobots in the optimization of workstations for the Type-1 assembly line balancing problem

Gustavo Erick Anaya-Fuentes[1], Eva Selene Hernández Gress[1*], Irving Barragán-Vite[1], Norberto Hernández-Romero[1]

[1] Engineering Academic Area, Universidad Autónoma del Estado de Hidalgo, México
[2] Engineering and Sciences School, Tecnológico de Monterrey, Mexico
* Corresponding author's e-mail: evahgress@tec.mx

ABSTRACT

The assignment of tasks to workstations in manufacturing and service processes, along with the minimization of these stations, has led to various assignment problems, including the simple assembly line balancing problem (SALBP). As an NP-complete problem, its computational complexity makes the use of exact methods infeasible for finding a solution. The new in this research analyzes is the implications of this problem by incorporating cobots to support workers at workstations, aligning with the human-centric pillar of Industry 5.0. To address the problem, a heuristic algorithm was employed to generate feasible solutions; while it does not guarantee an optimal value, this is characteristic of such methods. Results were compared with instance databases from literature, and a sensitivity analysis was conducted on time reductions due to cobot support, testing values of 20%, 30%, 40%, 50%, and 60%. The sensitivity analysis revealed a significant difference when a 40% reduction was considered. Consequently, we conclude that regardless of the weightings analyzed, cobots support consistently reduces the number of workstations required in organizational manufacturing and service processes.

Keywords: heuristics, industry 5.0, optimization, manufacturing, cobots.

INTRODUCTION

Organizations leverage resource optimization to increase profit margins. Therefore, process improvement actions focus on reducing input, time, tasks count, and the general resources used to produce goods or provide services. A significant approach to minimizing resource use lies in the optimal balancing of assembly lines, which gives rise to various problems addressed in the background section of this document. These include Simple Problem Types-1, -2, -E, and Type-F, distinguished primarily by their optimization goals. To reduce production times, both automated and semi-automated technologies, such as cobots, are utilized. Cobots are robots designed to support human workers in certain tasks without fully replacing them. This collaboration enhances efficiency in shared activities compared to those performed solely by humans.

The main contribution of this research is the sensitivity analysis of the time reduction percentages when cobots are integrated into workstations. This is based on ISO standards [9] which consider a 40% reduction rate for this respect. Additionally, this document aims to demonstrate the benefits of cobots in contributing to the "Human-Centric" pillar of Industry 5.0.

Assembly line balancing problem (ALBP)

The ALBP has its roots in the comparison of assignment problems such as the Traveling Salesman Problem (TSP), task scheduling. Originally, the ALBP [1] was defined as a set of tasks J1, J2, ..., Jn requiring Tk units of time, to be assigned to workstations while ensuring

compliance with precedence and cycle time constraints. Later, computational methods were developed to solve the ALBP, recognizing it as a computationally difficult problem. The ALBP can be classified by the number of workers collaborating at a workstation. When only one worker is assigned per station, it is referred to as a Simple Assembly Line, as shown in Figure 1, assigning two workers per workstation, is also possible Figure 1b, as is assigning up to three workers, known as a Multi-manned Assembly Line Figure 1c.

The problem has evolved to be classified based on its objective function. This classification comprises five types of problems: Type-1 seeks to minimize the number of workstations required to balance assembly lines within a predetermined cycle time. Type-2 aims to minimize the cycle time given a predetermined number of workstations. Type-3 strives to maximize smoothness by distributing task as evenly as possible across a fixed number of workstations. Type-4 focuses on maximizing the assignment of similar tasks to the same workstations under determined cycle time. Finally, Type-5 incorporates two objective functions: maximizing workload smoothness and maximizing the grouping of similar tasks [2]. This document focuses on commonly referred to as the Simple Assembly Line Balancing Problem Type-1 (SALBP-1). The following section analyzes existing approaches to the ALBP and its classifications:

Simple assembly line balancing problem type 1 (SALBP-1)

Genetic Algorithms have also been employed as a heuristic method to solve the SALBP-1 to optimize cycle times [3]. Other studies have emphasized the impact of automation on society from economic, philosophical, and human perspectives, while exploring the characteristics of human-robot collaboration [4], within the framework of Industry 4.0 [5]. Additionally, research has been conducted to identify potential opportunities for utilizing cobots through the Value Stream Mapping, simulating their implementation with software support. The objective was to maximize savings within the production process and increase average manufacturing volumes from an Industry 5.0 perspective [6]. Cobots have further been used to simulate the assignment of robots to workstations for the reconfiguration of assembly lines in the production of biomedical products [7]. While some literature focuses specifically on reviewing the state of the art regarding cobots [8], other studies propose databases for human-robot collaborative tasks based on established parameters [9], which facilitate the determination of collaborative times. Building on these foundations, the present research proposes a hybrid heuristic that, to our knowledge, has not previously been applied to this problem. This approach intends to minimize the number of workstations while satisfying precedence constraints within the Assembly Line Balancing Problem, specifically the SALBP-1.

Cobots y SALBP

The assembly line balancing problem involving collaborative robots is defined by the ability of humans and robots to perform tasks simultaneously at the same workstation, either in parallel or collaboratively. A mixed-integer programming model was formulated for line balancing and task assignment involving collaborative robots [3], as shown in Figure 2, where specific workstations are supported by cobots. This study investigates the optimal allocation of cobots to workstations. Additionally, it addresses the balancing of assembly lines for multi-product systems within families of similar products; in these systems, the assembly of each product requires sequencing a set of tasks across a series of workstations to which specific activities are assigned.

Solution techniques for SALBP

Various methodologies have been employed to solve task planning and scheduling problems, given their inherent complexity as NP-hard problems. These approaches range from traditional analytical techniques to advanced computational methods. Traditional methods rely on analytical techniques capable of finding optimal solutions; however, they are typically applicable only to small-scale problems due to high computational costs. Furthermore, they tend to be inflexible and inefficient when addressing large-scale, complex problems.

Conversely, heuristic methods can effectively address large problems by utilizing dispatching rules that generally provide near-optimal

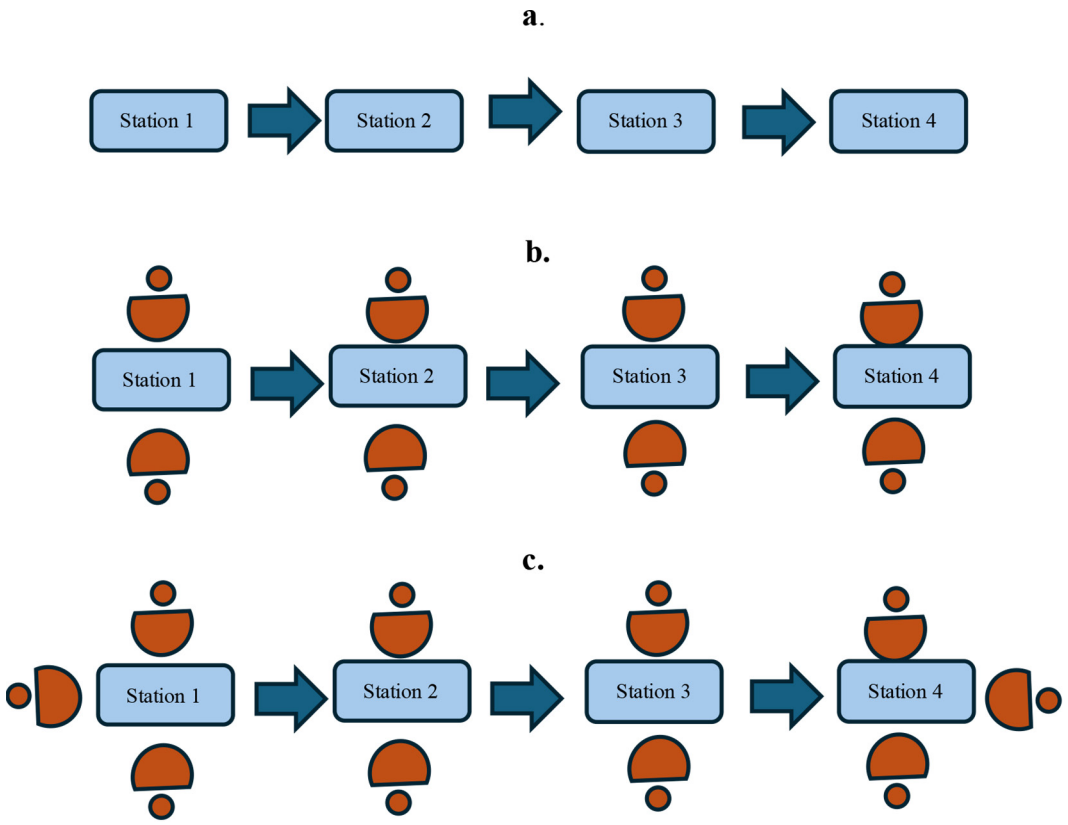

Figure 1. (a) Simple assembly line configuration, (b) two-sided assembly line configuration, (c) multi-manned assembly line configuration

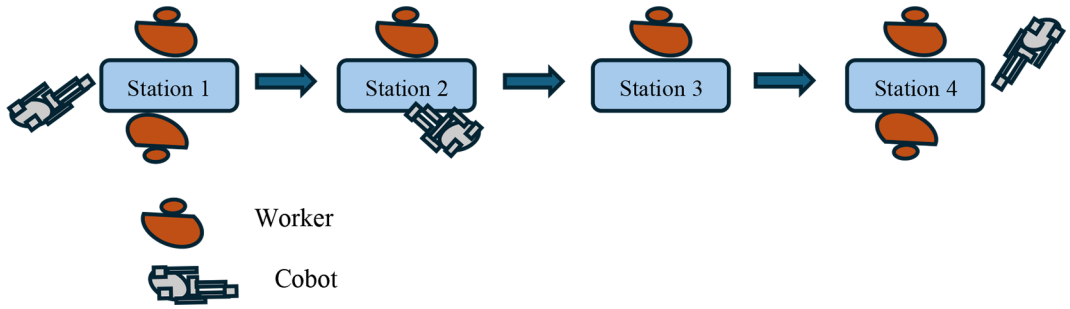

Figure 2. Cobots on stations

solutions, albeit without guaranteeing the absolute optimum. Finally, the third category comprises advanced methods. Analytical techniques include linear programming, integer programming, mixed-integer linear programming, dynamic programming, and branch and bound. Heuristic approaches encompass priority rules, dispatching rules, tabu search, and simulated annealing. lastly, advanced methodologies involve simulation with Petri nets, CAD modeling, discrete event simulation, and agent-based simulation, as well as artificial intelligence techniques such as genetic algorithms, ant colony optimization, fuzzy logic, bee colony optimization, and migrating birds optimization [10]. The present study utilizes an iterated local search approach.

Problem statement

Production costs impact parameters such as productivity; in this context, we aim to reduce costs by minimizing the number of workstations while considering one worker per workstation assisted by a collaborative robot or cobot. This work proposes a linear programming model for

this problem, which seeks to minimize workstations using cobots.

Objective function (1) seeks to minimize the number of workstations *Ws* for the given instance. Constraint (2) guarantees that each task is assigned to exactly one workstation. Constraint (3) is included to enforce cycle time limits and prevent assignments that exceed the allowed threshold. To satisfy task precedence relationships for $(i, j) \in E$, where i is a direct predecessor of j, constraint (4) is employed. Constraints (5) and (6) ensure the simultaneous involvement of both humans and robots in collaborative tasks; specifically, if a task $i \in I$ is performed collaboratively, both resources are engaged. Constraint (7) stipulates that a manual task $j \in I$ can only commence once the preceding manual task $i < j$ has been completed at the same workstation. Similarly, this logic applies to the robotic execution of task pairs $(i, j) \in I$. Constraint (8) accounts for the availability of both human and collaborative alternatives, while the total number of robots in the system is restricted by constraint (9). Finally, constraints (10), (11), and (12) define the domain and types of the decision variables. The model serves as a reference for the problem; however, this assignment problem is classified as NP-Hard, making it impractical to solve using exact methods. Therefore, this document proposes the use of a metaheuristic that offers feasible results without guaranteeing optimal solutions.

MATERIALS AND METHODS

The research design is a quantitative experimental computational study utilizing Python. Results were compared statistically using the evaluated parameters from standard literature instances. Statistical analysis was conducted through hypothesis tests for differences in means using Minitab 22 software. This research proposes an iterated local search method to address the Type-1 assembly line balancing problem, aiming to minimize the number of required workstations by utilizing cobots in some of them. The methodological process followed can be described as follows:

1. To generate high-quality, feasible solutions approximating the global optimum, standard process times (time matrix) and precedence constraints (precedence matrix) are necessary.
2. Collaborative task times are derived from manual production times, taking into account ISO parameters (International Organization for Standardization, 2010) that define maximum movement speeds of 1.6 m/s for human-machine interaction. Notably, speeds vary depending on the specific robot model and its power-saving modes. In realistic scenarios, the maximum speed is assumed to be between 0.5 and 1.0 m/s, implying efficiency improvements ranging from 60 to 220 percent. Consequently, a sensitivity analysis

$$Min \sum Ws \tag{1}$$

$$\sum \sum X_{ijk} = 1, \forall\, i \epsilon\ Z \tag{2}$$

$$S_i + \sum_{k \in K} \sum_{p \in P} t_{ip}.X_{ikp} \leq c\ ,\ \forall\ i\ \in I \tag{3}$$

$$S_i + \sum_{k \in K} \sum_{p \in P} t_{ip}.X_{ikp} \leq S_j + \underline{C}\left(Z_j - Zi\right)\ ,\ \forall\ (i,j) \in E \tag{4}$$

$$S_i + t_{ip}.X_{ikp} \leq\ S_j + \underline{C}\left(1 - \sum_{p \in P} X_{jkp}\right) + \underline{c}\left(1 - X_{ikpc}\right) + \underline{c}\left(1 - y_{ij}\right), \forall\ (i,j) \in I, k \in K \tag{5}$$

$$S_i + \sum_{p \in P} t_{ip}.X_{ikp} \leq\ S_j + \underline{c}\left(1 - X_{jkpc}\right) + \underline{c}\left(1 - y_{ij}\right), \forall\ (i,j) \in I, k \in K \tag{6}$$

$$S_i + \sum_{p \in P} t_{ip}.X_{ikp} \leq\ S_j + \underline{c}\left(1 - X_{ikp}\right) + \underline{c}\left(1 - X_{jkp}\right) + \underline{c}\left(1 - y_{ij}\right), \forall\ (i,j) \in I, k \in K, p \in \{P_H, P_R\} \tag{7}$$

$$X_{ikp} \leq r_k \forall\ i \in I, k \in K, p \in \{P_H, P_C\} \tag{8}$$

$$\sum_{k \in K} r_k \leq q \tag{9}$$

$$S_i, Z_i \geq 0\ \forall\ i\ \in I \tag{10}$$

$$r_k \in \{0,1\}\ \forall\ k \in K \tag{11}$$

$$y_{ij} \in \{0,1\}\ \forall\ i,j\ \in I, i \neq j \tag{12}$$

was performed to evaluate these specific reduction percentages.

3. To account for other factors that may influence this research, potential time improvements of 20%, 30%, 40%, 50%, and 60% were evaluated, leading to the proposal of a collaborative database based on these parameters. The Iterated Local Search starts by generating a random population of n individuals representing feasible solutions, subject to precedence constraints. To ensure feasibility, tasks with no preceding constraints are identified and added to a vector designated as the "available list". In an iterative process for each of the n individuals, tasks are selected randomly from this list. Once a task is assigned to a solution, the algorithm verifies whether it is a predecessor for other tasks; if so, those tasks are subsequently added to the available vector list.
4. To allocate a collaborative robot to a workstation, a probability threshold of 0.1 was employed to identify the optimal assignment of tasks and robots that minimizes the total number of workstations. This approach ensures that not all workstations are equipped with a collaborative robot. In addition to precedence constraints, the cycle time constraint was enforced to prevent assigning tasks that exceed the allowable cumulative time per station.
5. If assigning a task to a workstation violates the cycle time constraint, the activity is removed from that station and assigned to a new one, designated as $Ws + 1$. This process recurs until all tasks are assigned to a workstation for each of the n individuals in the population, collectively referred to as the stack. Subsequently, the top 50% of the results from the stack are

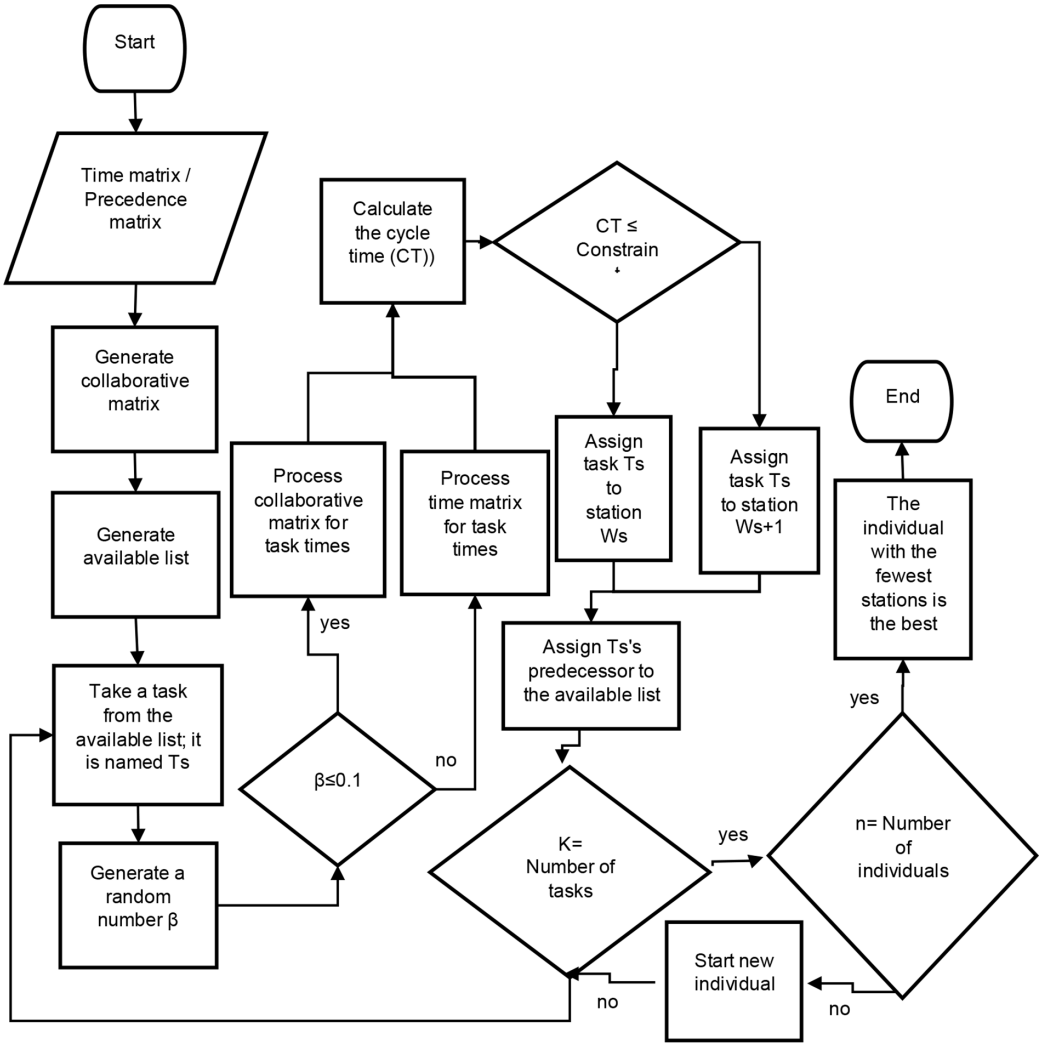

Figure 3. Flowchart of the proposed algorithm

selected to transition to the next generation. This iterative cycle continues for m generations or iterations.

6. Ultimately, the optimal solution identified across the m generations is designated as the best result. This outcome achieves a minimized number of workstations by efficiently assigning tasks and integrating collaborative robots into specific stations, as illustrated in Figure 3.

The following includes the pseudocode of the algorithm for its potential reproduction: The algorithm was implemented using the Python programming language, and the source code is available at https://github.com/Gustavo-Anaya/Cobots/issues/1.

RESULTS AND DISCUSSION

The algorithm proposed in this research was tested on various standard instances from the literature, including Merten (7), Bowman (8), Jaeschke (9), Jackson (11), Mansor (11), Mitchell (21), Heskia (28), Kilbridge (45), Tonge (70), Arcus (83), and Arcus (111), utilizing different cycle times for each instance. The primary objective is to minimize the number of workstations in the Simple Assembly Line Balancing Problem Type 1 (SALBP-1).

To evaluate the efficiency of the proposed algorithm against existing heuristics, a hypothesis test for the difference in proportions was conducted at a 95% confidence level. Specifically, the proposed method was compared with a genetic algorithm (GA) heuristic commonly applied to task allocation in line balancing. Out of 64 samples analyzed for each method, the GA outperformed the proposed algorithm 26 times, whereas the latter achieved superior results in 38 instances. Applying the Z-test for proportions, a calculated value of 2.1234 was obtained. When compared against the critical Z-value of 1.645, the null hypothesis is rejected. This indicates a statistically significant difference, confirming that the proportion of better results is higher for the algorithm proposed in this study.

The results are summarized in Table 1, which provides a comparative analysis of the outcomes obtained from the following methodologies: the HGA1 hybrid genetic algorithm [11], the HGA2 hybrid genetic algorithm [12], the SAA simulated annealing approach [13], the HALBP heuristic assembly line balancing procedure [14], and the MILP mixed-integer linear programming model [15]. The performance of these established algorithms is benchmarked against the proposed algorithm (PA) introduced in this study (Table 1). Workstations of the proposed algorithm compared with [12].

To conduct a statistical comparison of the results, Minitab Statistical Software 22 was utilized. The initial comparison was performed between the results of the Merten instance reported in [12–15], and those obtained by the proposed algorithm in this research.

In the comparative hypothesis test conducted for the Merten (7) instance the null hypothesis (H_0) states that there are no significant differences between the population means of the methods reported in the literature and the algorithm proposed in this study. Population 1 corresponds to the results found in the literature, while Population 2 represents the outcomes generated by the proposed algorithm.

The resulting p-value (0.032), leads to the rejection of the null hypothesis in favor of the alternative hypothesis. This implies that the number of workstations required by the methods documented in the literature is statistically significantly greater than that obtained using the proposed method. These findings highlight the effectiveness of the heuristic approach and support the relevance of incorporating collaborative robots (cobots) in the system design.

Subsequently, a similar comparison was performed using the Bowman (8) instance. The null hypothesis assumes (H_0) no difference in the number of workstations between methods, whereas the alternative hypothesis (H_1) indicated that the average number of workstations derived from literature-based approaches is greater than the average obtained by the proposed method. The results are consistent with those obtained for the Merten instance (7), reinforcing the superiority of the proposed approach.

Finally, the hypothesis testing conducted for the Kilbridge (45) instance further compares the population means of the proposed algorithm against those reported for other benchmark instances and methods in the literature**.** The results consistently demonstrate the improved performance of the proposed method in reducing the number of required workstations.

Table 2 provides a summary of the hypothesis tests conducted for each instance.

Table 1. Compared results

Instance	CT	HGA1	HGA2	SAA	HALBP	MILP	PA	
		Ns	Ns	Ns	Ns	Ns	Ns	Cobots
Merten (7)	6	---	3	3	6	3	3	2
	7	---	3	3	5	3	2	2
	8	---	3	3	5	3	2	2
	10	---	3	3	3	3	2	1
	15	---	2	2	2	2	1	1
	18	---	1	1	2	1	1	1
Bowman (8)	17	---	5	5	---	5	3	2
	20	---	4	4	5	4	2	2
	21	---	4	4	---	4	2	2
	24	---	4	4	---	4	2	2
	28	---	2	2	---	2	2	2
	31	---	2	2	---	2	2	1
Jaeschke (9)	6	---	5	6	8	5	3	3
	7	---	5	6	7	5	3	1
	8	---	5	5	6	5	3	2
	10	---	4	4	4	4	2	2
	18	---	2	2	3	2	1	1
Jackson (11)	7	---	5	6	7	5	4	3
	9	---	4	4	5	4	3	2
	10	---	4	4	6	4	2	2
	13	---	3	3	4	3	2	2
	14	---	3	3	4	3	2	1
	21	---	2	2	3	2	1	1
Mansor (11)	45	---	3	3	---	3	2	2
	54	---	3	3	---	3	2	2
	63	---	2	2	---	2	2	1
	72	---	2	2	---	2	2	1
	81	---	2	2	---	2	2	1
Mitchell (21)	14	---	7	7	9	7	4	3
	15	---	7	7	8	7	4	3
	21	---	4	5	5	5	3	2
	26	---	4	4	5	4	2	2
	35	---	3	3	3	3	2	1
	39	---	2	2	3	2	2	1
Heskia (28)	138	5	4	5	6	4	4	4
	205	4	3	4	6	3	3	2
	216	3	3	3	4	3	5	1
	256	3	3	3	5	3	2	2
	324	2	2	2	3	2	2	2
	342	2	2	2	3	2	2	1
Kilbridge (45)	57	6	5	6	8	5	6	4
	79	4	5	4	6	5	4	3
	92	4	4	4	5	4	4	3
	110	3	3	3	5	3	3	2
	138	3	3	3	4	3	2	2
	184	2	2	2	3	2	2	1

Tonge (70)	176	17	14	19	21	14	14	7
	364	6	5	7	9	5	5	5
	410	5	4	5	7	4	5	5
	468	5	4	4	7	4	4	4
	527	4	4	4	7	4	4	3
Arcus (83)	5048	11	11	11	16	11	9	6
	5853	10	10	10	13	10	8	5
	6842	8	8	8	10	8	6	5
	7571	9	7	10	11	7	6	4
	8412	8	6	8	10	6	5	4
	8998	5	6	7	8	6	4	4
	10816	6	5	5	8	6	4	3
Arcus (111)	5755		14	---	24	14	21	8
	8847	12	12	14	18	12	12	6
	10027	10	10	12	15	10	9	6
	10743	14	10	14	14	10	9	5
	11378	8	7	9	9	7	8	5
	17067	5	5	7	7	5	5	4

Note: The execution times of the algorithms have an expected value of 5 seconds.

Based on the statistical comparison table, it is evident that in 70% of the instances, the results improved significantly when employing both cobots and the method proposed in this study. In the remaining instances, no significant improvements were observed despite the integration of cobots and the evaluated algorithm.

The rejection of the null hypothesis confirms that the average number of workstations reported in the literature is significantly higher than the results obtained in this study. This strengthens the evidence supporting the adequacy of integrating cobots and the proposed algorithm for effective task scheduling. The following comparative graph illustrates a performance summary across all instances, contrasting the results from previous studies with those achieved by the algorithm proposed herein.

The objective of the heuristic algorithm is to reduce the number of workstations used (Ns). However, the sample results from the selected instances are not sufficient evidence due to the bias inherent in their sample nature. To provide support, statistical hypothesis tests were used, allowing for statistical inference on the population data. This analysis found a reduction in the number of workstations in instances that utilized cobots compared to the instances in the literature, which do not consider the use of cobots.

Furthermore, results demonstrate that the integration of cobots, their allocation, and task scheduling via Iterated Local Search is beneficial, as this approach reduces the number of workstations required for operational processes.

The 40% reduction in task times when using cobots is based on parameters established by the International Organization for Standardization and previous studies that have reported improvements in the range of 30% to 50% [5, 9]. To provide a comprehensive view, a sensitivity analysis is incorporated, considering reductions of 20%, 30%, 40%, 50%, and 60%. The results are shown in Table 3.

To complement the sensitivity analysis shown in Table 3, a one-way ANOVA with 5 levels was conducted for each analyzed percentage. The ANOVA results show P-value is 0.004, indicating that there is a significant difference between the levels analyzed.

For the sensitivity analysis, the groupings generated by Tukey's method are included with the intention of grouping the weightings. It was found that there is a significant difference between the 40% weighting and the rest of the evaluated values, with the 40% weighting having the lowest average number of workstations required.

Table 2. Results of the statistical comparisons

Instance	Hypothesis testing	Result and interpretation
Merten (7)	$H_0: \mu_1 - \mu_2 = 0$ $H_1: \mu_1 - \mu_2 > 0$	Ho is not rejected. The number of workstations required for the Merten (7) instance when applying the proposed method is lower than the results reported in the literature [12], [13], [14], and [15].
Bowman (8)		Ho is rejected. The number of workstations required for the Bowman (8) instance when applying the proposed method is lower than the results reported in the literature [12], [13], and [15].
Jaeschke (9)		Ho is rejected. The number of workstations required for the Jaeschke (9) instance when applying the proposed method is lower than the results reported in the literature [12–15].
Jackson (11)		Ho is rejected. For the Jackson (11) instance, the number of workstations achieved by the proposed method is lower than the results reported in the literature [12–15].
Mansor (11)		Ho is rejected. Regarding the Mansor (11) instance, the number of workstations achieved by the proposed method is lower than the results reported in the literature [12], [13], and [15].
Mitchell (21)		Ho is rejected. For the Mitchell (21) instance, the number of workstations achieved by the proposed method is lower than the results reported in the literature [12–15].
Heskia (28)		Ho is not rejected for the Heskia (28) instance. This indicates that the average number of workstations reported in the literature [11–15] is statistically equal to the average achieved by the proposed method.
Kilbridge (45)		Ho is not rejected for the Kilbridge (45) instance. This indicates that the average number of workstations reported in the literature [11–15] is statistically equal to the average achieved by the proposed method.
Tonge (70)		Ho is not rejected for the Tonge (70) instance. This indicates that the average number of workstations reported in the literature [11–15] is lower than the average achieved by the proposed method.
Arcus (83)	$H_0: \mu_1 - \mu_2 = 0$ $H_1: \mu_1 - \mu_2 > 0$	Ho is rejected for the Arcus (83) instance. This indicates that the average number of workstations reported in the literature [11–15] is statistically equal to the average achieved by the proposed method.
Arcus (111)		Ho is rejected for the Arcus (111) instance. This finding suggests that the average number of workstations reported in the literature [11–15] is statistically equivalent to the mean achieved by the proposed method.

Table 3. Sensitivity analysis

Instance	Workstation (20%)	Workstation (30%)	Workstation (40%)	Workstation (50%)	Workstation (60%)
Merten (7)	5	4	3	3	3
Bowman (8)	6	5	3	4	3
Jaeschke (9)	7	6	3	5	4
Jackson (11)	8	7	4	6	5
Mansor (11)	9	8	2	6	6
Mitchell (21)	11	10	2	7	7
Heskia (28)	13	11	2	9	9
Kilbridge (45)	15	13	11	10	10
Tonge (70)	20	17	4	14	13
Arcus (83)	22	19	6	16	14
Arcus (111)	25	22	5	18	17

CONCLUSIONS

In this study, the behavior of data related to a known problem, the assembly line balancing problem, and the incorporation of cobots into it was analyzed. Consequently, the behavior of the number of workstations in instances from the literature that do not use cobots was compared to those in the present investigation in which cobots are included to support the workforce, in accordance with the Industry 5.0 pillar called "Human-Centered". The International Organization for Standardization considers that tasks supported by cobots can reduce task times by 20 to 60%. Therefore, this study includes a sensitivity analysis to evaluate the impact of these percentages in the quest to optimize the Simple Assembly Line Balancing Problem Type-1 (SALBP-1) through

a task organization technique that facilitates task selection considering restrictive tasks with a heuristic algorithm specifically proposed to address this problem, using an Iterated Local Search approach to find good feasible solutions. The results demonstrate a reduction in the number of workstations with the use of cobots compared to instances without their collaboration, which is supported by statistical hypothesis tests. This can be interpreted as a reduction in the number of workstations required in production processes that use cobots versus those that do not.

In this study, it is assumed that the cost of workstations is higher than that of the collaborative robot, so the priority was to minimize the number of workstations. Future studies could consider the costs of cobots and workstations to determine the minimum cost, making it the objective function. It is also possible to seek the proper allocation of cobots to workstations to optimize the process according to initial conditions.

REFERENCES

1. Held, M., Karp, R. A dynamic programming approach to sequencing problems. ACM '61: Proceedings of the 1961 16th ACM national meeting. New York. 1961. p. 1–4. https://doi.org/10.1145/800029.808532
2. Kim, Y. K., Kim, Y. J., Kim, Y. Genetic algorithms for assembly line balancing with various objectives. CAIE. 1996; 30(3): 397–409.
3. Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., Spengler, T. Balancing of assembly lines with collaborative robots. Bus Res. 2020; 13: 93–132.
4. Lefranc, G., Lopez-Juarez, I., Osorio-Comparán, R., Peña-Cabrera, M. Impact of cobots on automation. Procedia Comput. Sci. 2022; 214(C): 71–78. https://doi.org/10.1016/j.procs.2022.11.150
5. Weiss, A., Wortmeier, A. K., Kubicek, B. Cobots in Industry 4.0: A Roadmap for Future Practice Studies on Human–Robot Collaboration. IEEE Transactions on HMS. 2021; 51(4): 335–345. https://doi.org/10.1109/THMS.2021.3092684
6. Pizón, J., Cioch, M., Kanski, L., Sánchez-García, E. Cobots Implementation in the era of Industry 5.0 using modern business and management solutions. Adv. Sci. Technol. Res. J. 2022; 16(6): 166–178.
7. Rossi, F., Pini, F., Carlesimo, A., Dalpadulo, E., Blumetti, F., Gherardini, F., Leali, F. Effective integration of Cobots and additive manufacturing for reconfigurable assembly solutions of biomedical products. Int J Interact Des Manuf. 2020; 14(3): 1085–1089. https://doi.org/10.1007/s12008-020-00682-9
8. Semeraro, F., Griffiths, A., Cangelosi, A. Human–robot collaboration and machine learning: A systematic review of recent research. RCIM. 2023; 79: 1–16. https://doi.org/10.1016/j.rcim.2022.102432
9. International Organization for Standardization. ISO. Retrieved 2023, from https://www.iso.org/obp/ui/#iso:std:iso:13855:ed-2:v1:en
10. Muñoz-Guevara. J.A., Toro-Ocampo, E. Vélez-Gallego, M.C. Robotic assembly systems planning and scheduling problems: A review. Int J Ind Eng Comp. 2024; 15: 845–870. https://doi.org/10.5267/j.ijiec.2024.8.001
11. Zamzam, N., Elakkad, A. Time and space multi-manned assembly line balancing problem using genetic algorithm. JIEM. 2021; 14(4): 733–749. https://doi.org/10.3926/jiem.3542
12. Zamzam, N., Sadek, Y., Afia, N., El-Kharbotly, A. Multi-manned assembly line balancing using genetic algorithm. IJERT. 2015; 4(12): 56–61.
13. Roshani, A., Roshani, A., Roshani, A., Mohsen, S., Esfandyari, A. A simulated annealing algorithm for multi-manned assembly line balancing problem. J. Manuf. Syst. 2013; 32(1): 238–247. https://doi.org/10.1016/j.jmsy.2012.11.003
14. Dimitriadis, S. G. Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation. COR. 2006; 33(9): 2757–2774. https://doi.org/10.1016/j.cor.2005.02.027
15. Fattahi, P., Roshani, A., Roshani, A. A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. Int J Adv Manuf Technol. 2011; 53: 1433–3015. https://doi.org/10.1007/s00170-010-2832-y