Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Adaptive Tabu Search for Flexible Assembly Line Balancing

**Jose Luis Peinado-Portillo**

jose_peinado@utcj.edu.mx

Universidad Tecnologica de Ciudad Juarez https://orcid.org/0000-0002-2162-8118

**Jose Dario Vazquez-Martinez**

Universidad Tecnologica de Ciudad Juarez https://orcid.org/0009-0001-4851-3353

**Method Article**

**Keywords:** Assembly line balancing, Tabu Search, metaheuristics, optimization, automotive manufacturing

**Posted Date:** March 13th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-9098544/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** The authors declare no competing interests.

# Adaptive Tabu Search for Flexible Assembly Line Balancing

José Luis Peinado-Portillo[1*] and José Darío Vásquez Martínez[2**]

[1] Mechatronics Engineering, Universidad Tecnológica de Ciudad Juárez,

Av. Universidad Tecnológica, Cd. Juárez, 32697, Chihuahua, México.

[2] Industrial Engineering, Universidad Tecnológica de Ciudad Juárez, Av. Universidad Tecnológica, Cd. Juárez, 32697, Chihuahua, México.

*Corresponding author(s). E-mail(s): jose peinado@utcj.edu.mx;

Contributing authors: jose vasquezm@utcj.edu.mx;

**These authors contributed equally to this work.

## Abstract

This research presents the development and evaluation of the Adaptive Tabu Search for Flexible Assembly Line Balancing (ATS-FALB), a metaheuristic algorithm designed to optimize task assignment in flexible automotive production lines. Assembly line balancing problems are computationally complex and classified as NP-hard, making optimal solutions difficult to obtain in large-scale manufacturing systems. To address this challenge, the proposed algorithm integrates adaptive mechanisms based on dynamic long-term memory structures and advanced penalty strategies that guide the search process and reduce the risk of premature convergence.

Experimental validation was conducted using a benchmark dataset of 30 vehicle models representing realistic automotive production scenarios. The results indicate that the ATS-FALB algorithm consistently produces optimal or near-optimal solutions while maintaining stable convergence behavior. In particular, the method achieves

an average reduction of 0.14 hours in cycle times and reaches high-quality solutions in fewer than 100 iterations. These findings demonstrate the robustness and efficiency of the proposed approach for improving operational performance in manufacturing environments.

**Keywords:** Assembly line balancing, Tabu Search, metaheuristics, optimization, automotive manufacturing.

## Introduction

Assembly line balancing constitutes a fundamental aspect of mass production systems across various industries, particularly in the automotive sector. The inherent complexity of assigning tasks to workstations aiming to minimize total production time and cycle times has become a critical challenge in modern manufacturing.

A well-balanced production line, characterized by the optimal distribution of tasks among workstations, significantly enhances productivity, increases process speed, and optimizes operational efficiency. Consequently, this contributes to reduced operating costs and more efficient energy resource utilization, thereby fostering sustainable production practices (ElMolouk et al., 2025).

Traditionally, various optimization methods have been proposed to address assembly line balancing. One of the most prominent is the Branch and Bound algorithm, which utilizes a decision tree structure to systematically explore the space of potential solutions. This method begins with a task assignment and evaluates all feasible workstations, generating new branches that represent partial solutions. Each branch is subsequently assessed to determine its potential to surpass the current best solution, discarding those that fail to meet this criterion.

Essentially, assembly line balancing entails a combinatorial exploration subject to multiple constraints. Since this problem is classified as NP-hard, the number of possible combinations grows exponentially as the production line length and the number of task-to-station assignment variants increase. Consequently, exact

methods based on mathematical models become computationally inefficient when facing the vast solution space inherent in real-world scenarios (Sotskov, 2023).

Furthermore, real-world production environments are characterized by uncertainty regarding task processing times at each station, which further complicates the application of traditional solution methods. In this context, metaheuristic-based algorithms have gained prominence, as they enable the resolution of highly complex problems, providing near-optimal solutions within reasonable computational timeframes (Prasad et al., 2022).

Metaheuristics are widely recognized as effective methods for solving complex optimization problems. These intelligent approaches utilize iterative processes to navigate toward high-quality solutions. A metaheuristic acts as an optimization technique that extracts information from the search space to identify promising areas where high-quality solutions reside (Camacho-Villalón et al., 2023).

Tabu Search (TS) is a metaheuristic specifically designed to enhance local search by employing memory structures that prevent cycling through previously visited solutions (Imbien, 2025). TS functions as a local search method aiming to explore the solution space beyond local optima. This technique begins with an initial solution and progressively attempts to improve it by evaluating neighboring solutions. To prevent the algorithm from falling into cycles or repeating previously explored paths, a memory mechanism is incorporated, restricting certain moves for a predetermined number of iterations. These moves are stored in a structure known as the "tabu list," which temporarily prohibits their reuse, thereby promoting diversification throughout the search process (Arıkan, 2021).

**Proposed Model**

The proposed model is based on the Tabu Search (TS) metaheuristic. Unlike traditional implementations that rely solely on simple local search mechanisms, this model incorporates dynamic memory structures to enhance the exploration of the solution space. Through a memory mechanism, the algorithm restricts the return to

previously visited states via a "tabu list," thereby fostering diversification during the search process.

Furthermore, the model incorporates a long-term frequency memory mechanism, which modifies the cost surface of the original objective function. This approach prevents the algorithm from being constrained to local minima by promoting a systematic exploration of under-visited regions of the search space. This is achieved through a cumulative penalty scheme applied to recurrent solutions, which effectively avoids stagnation at local optima and enhances the algorithm's global exploration capability.

The proposed optimization model is grounded in an integer programming structure aimed at minimizing total workload. Below is the mathematical formulation that guides the behavior of the Adaptive Tabu Search for Flexible Assembly Line Balancing (ATS-FALB) algorithm.

**1. Problem Formulation**

This section defines the structural components of the mathematical model. First, the **sets** are established, representing the indices over which the decision variables operate, such as parts, workstations, or time periods. Subsequently, the **parameters** are defined; these correspond to the known constants of the system, including costs, capacities, and demands, which govern the model's constraints and shape the formulation of the objective function.

**Sets and Parameters**

$K$: Set of Automobiles, where $k \in \{1,. \ . \ ., K\}$.

$J$: Set of Workstations, where $j \in \{1,. \ . \ ., J\}$.

$P_j$: Set of parts assignable to Workstation $j$.

$t_{ijk}$: Processing time required for part $i \in P_j$ at workstation $j$ for automobile $k$.

## Decision Variables

$x_{ijk} \in \{0,1\}$: Binary variable that takes the value 1 if part $i$ is assigned to Workstation $j$ for automobile $k$, and 0 otherwise.

## Objective Function

The objective is to minimize the total workload across all stations and all automobiles:

$$\min Z = \sum_{k \in K} \sum_{j \in J} \sum_{i \in P_j} \left( t_{ijk} \cdot x_{ijk} \right)$$

## Constraints

Unique Assignment: Each workstation must process exactly one part for each automobile:

$$\sum_{i \in P_j} x_{ijk} = 1, \qquad \forall j \in J, \qquad \forall k \in K$$

### 2. Adaptive Evaluation Function

The proposed model avoids solving the problem via exact combinatorial models. Instead, it utilizes a dynamic evaluation function guided by the trajectory generated during the tabu search process. Unlike the original formulation, this function incorporates an additional memory-based penalty term, which modifies the objective function value based on the frequency with which specific solutions or components have been visited. Thus, the exploration of new regions within the search space is promoted, and the probability of premature convergence toward local optima is mitigated.

$$f(x) = Z(x) + \lambda \cdot H(X)$$

Where:

$Z(x)$ is the base objective function originally defined.

$H(x)$ represents the long-term memory mechanism; this function counts how many times part $i$ has been assigned to station $j$ throughout the search within a specific number of iterations, aiming to prevent cycling at a single point.

$\lambda$ is a penalty factor that enables the search for improved times and prevents entrapment in known optima.

When the model detects that the function $f(x)$ does not exhibit improvements after a predefined number of iterations, a restart mechanism is activated in order to prevent the algorithm from remaining trapped in a local optimum. In this process, the penalty function $H(x)$ assumes the role of a prohibitive penalty, significantly increasing the cost associated with previously explored solutions. In this way, the algorithm is forced to move toward new regions of the search space, promoting greater diversification in the exploration process.

## Experiments and Discussion

The experiment was designed to simulate a flexible automotive production line with a configuration of seven workstations. Each station represents different variants within the production processes, which directly impact the total cycle time, as shown in Table 1.

**Table 1. Relationship between Stations and Parts.**

| Workstation | Variations |
|---|---|
| Chassis | White, Black, Red |
| Engine | Complete engine |
| Transmission | Manual, Automatic |
| Wheels | Standard rims, Luxury rims |
| Electrical parts | Manual, Automatic |
| Interior coverings | Fabric, Vinyl, Microfiber |
| Body panels | Doors, Hood, Trunk |

The data used in the experiment were stored in a database with a NoSQL structure, specifically in MongoDB, which was used to feed the optimization model with synthetic times reported in recent literature. The database contains information corresponding to 30 automobile models, including the duration times associated with each stage of the production process. This made it possible to evaluate different production scenarios and analyze the behavior of the optimization model under various configurations, from which the results presented in the following section were obtained.

Figure 1 presents a comparative analysis between the proposed model based on tabu search and a traditional random search algorithm. For a single automobile model, the graph shows the evolution of the cycle time (in hours) over 500 iterations.

The results show a significantly faster convergence in the proposed model (red line), which reaches a cycle time of 7.72 hours within the first 100 iterations. This behavior suggests that the memory mechanism and the neighborhood structures implemented in the algorithm are highly

effective in guiding the exploration of the solution space, allowing the prioritization of previously unvisited regions and improving the efficiency of the search process.

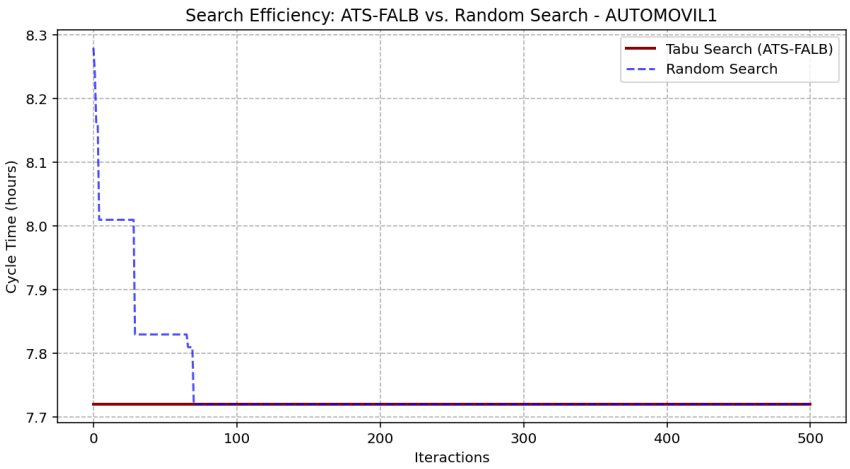



**Figure 1. Comparison of convergence between the proposed model and the traditional search.**

In Figure 2, the convergence analysis of the ATS-FALB algorithm applied to a sample of six vehicle models with different assembly configurations is presented. This experiment makes it possible to validate the generalization capability of the proposed method under product variability in a flexible production line.

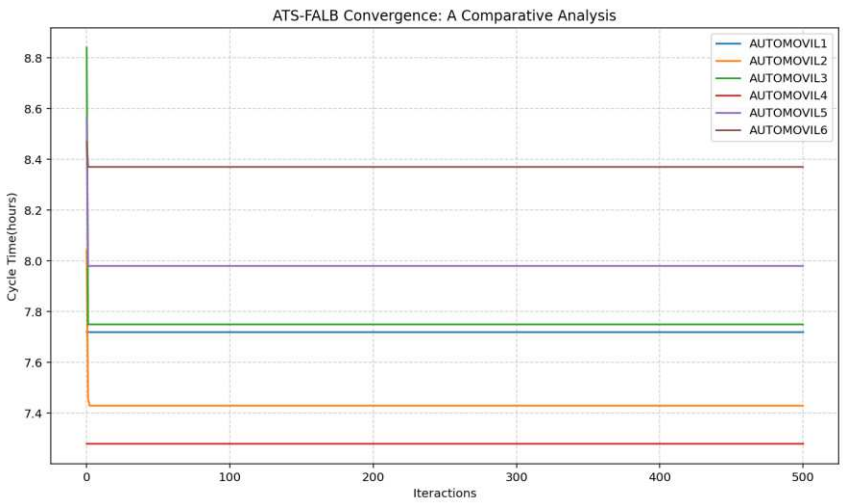



**Figure 2. Multimodel Robustness Analysis.**

Although each automobile model presents different cycle times due to variability in task durations at each station, the algorithm's convergence is consistently fast in all cases, generally stabilizing within the first 100 iterations.

In addition, the graph allows visualization of different levels of operational complexity associated with each automobile model. The figure presents only six vehicles for illustrative purposes; however, when the experiment is repeated with the thirty models considered in the database, only minimal variation in convergence times is observed.

It is worth noting that once the minimum cycle time is reached, the algorithm does not show significant changes or divergent behavior in subsequent iterations. This behavior confirms that the tabu list is being properly managed, preventing the ATS-FALB algorithm from entering repetitive cycles or converging to suboptimal solutions.

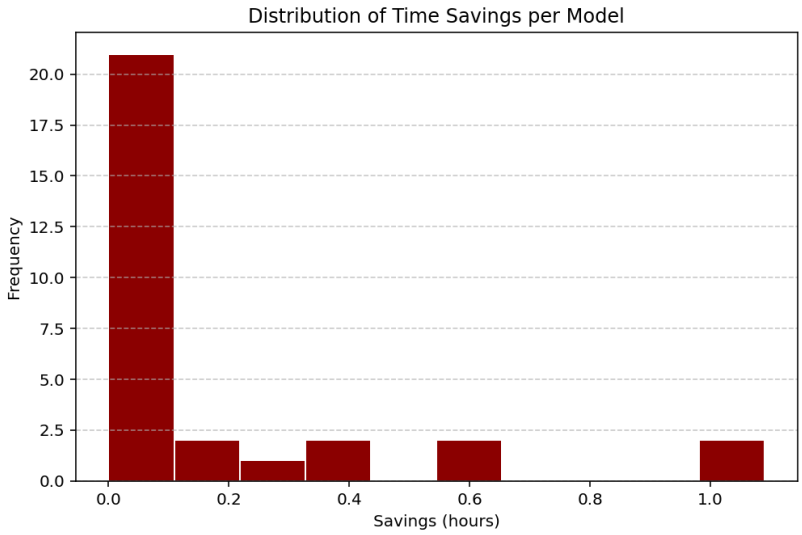



**Figure 3. Distribution of Time Savings**

In Figure 3, a histogram is presented that quantifies the direct impact of the ATS-FALB algorithm on the efficiency of the production line. This graph shows the frequency of the time savings obtained after optimization in an extended fleet of vehicles.

The distribution shows a generalized improvement, in which, in most cases, the algorithm achieves immediate reductions in cycle time. This behavior reveals a clear trend toward improved performance across all evaluated models.

On the other hand, the absence of negative values indicates that the ATS-FALB algorithm always finds a solution equal to or better than the initial solution, confirming its consistency as a search method. Consequently, these results validate its usefulness as an optimization tool for assembly line balancing problems.

**Table 2. Summary of Optimization Results (Representative Selection)**

| Vehicle Model | Savings (h) | Optimal Cycle Time (h) |
|---|---|---|
| AUTOMOBILE 3 | 1.09 | 7.75 |

| Vehicle Model | Savings (h) | Optimal Cycle Time (h) |
|---|---|---|
| AUTOMOBILE 30 | 1.02 | 7.33 |
| AUTOMOBILE 2 | 0.61 | 7.43 |
| AUTOMOBILE 5 | 0.58 | 7.98 |
| AUTOMOBILE 10 | 0.42 | 7.84 |
| AUTOMOBILE 1 | 0 | 7.72 |
| AUTOMOBILE 4 | 0 | 7.28 |
| Average (Total Fleet) | 0.14 | 7.79 |

To evaluate the performance of the proposed model, Table 2 summarizes the results obtained for the complete fleet of 30 vehicle models. As observed, the ATS-FALB algorithm achieves an average reduction of 0.14 hours in the total cycle time. It is important to highlight that while in some models the algorithm confirms the efficiency of the initial configuration by showing savings of 0.00 hours, as in the cases of AUTOMOBILE 1 and AUTOMOBILE 4, in more complex scenarios such as AUTOMOBILE 3 and AUTOMOBILE 30, significant improvements of more than 1.0 hour are achieved. This consistency in the results demonstrates that the tool is robust, as it always finds solutions equal to or better than the initial ones, ensuring stable operation in the production line.

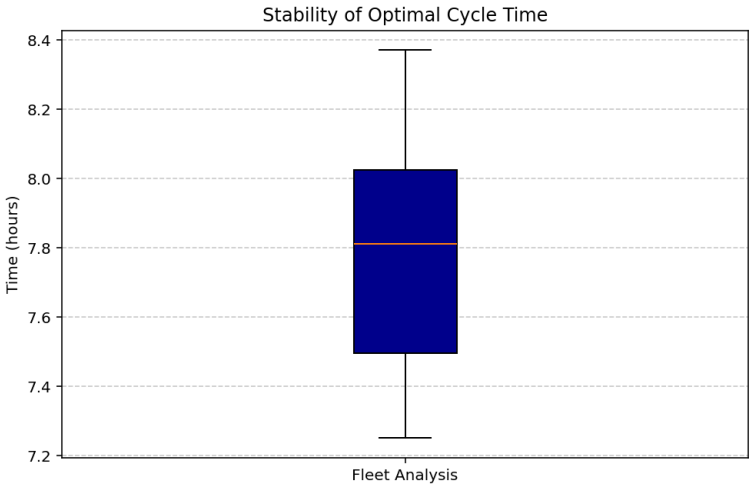



*Figure 4. Statistical Stability and Dispersion Analysis*

To conclude the analysis of results, Figure 4 presents a boxplot that summarizes the stability of the optimal cycle time obtained by the ATS-FALB algorithm across the entire fleet of evaluated vehicles.

The operational range of the optimal cycle times is mainly distributed between 7.2 and 8.3 hours, indicating that the ATS-FALB algorithm can adapt to different levels of complexity in the configuration of the production line while remaining within a manageable interval for production planning. Likewise, the graph suggests a relatively balanced distribution of times, with little bias toward extremely fast or excessively slow configurations.

On the other hand, the limited presence of whiskers in the diagram indicates a significant absence of outliers, suggesting that the algorithm exhibits highly robust behavior. Consequently, no erratic cycle times are generated for any of the 30 evaluated vehicle models, which helps ensure stable and predictable operation in production lines.

## Conclusion

This research demonstrates that the ATS-FALB algorithm (Assembly Line Balancing using Tabu Search) constitutes an effective and robust solution for operational management in automotive manufacturing environments. Based on the evaluation conducted on a sample of 30 vehicle models, superior efficiency is observed when compared with conventional methods. In particular, the long-term memory mechanism inherent to tabu search allows rapid convergence toward optimal or near-optimal solutions, representing a significant improvement in terms of computational cost.

Likewise, the model exhibits stable and consistent behavior, being capable of adapting to different assembly configurations while maintaining a uniform convergence process without significant fluctuations. This characteristic ensures the reliability of the algorithm under variations in the production line configuration. Additionally, although in some cases no significant reductions in cycle time were obtained—either because the global optimum was found quickly or was already very close to the initial random solution—in multiple scenarios the algorithm succeeded in reducing operation times, which directly impacts the efficiency of the production process.

Finally, the obtained results open the possibility for future applications in real industrial environments. In particular, the ATS-FALB algorithm could be integrated with real-time monitoring and data acquisition systems,

allowing it to dynamically respond to unexpected changes in the production line and facilitating self-optimization processes based on updated operational information.

References

Arıkan, M. (2021). Type-2 assembly line balancing with workload smoothing objective: A reactive tabu search algorithm. *Gazi University Journal of Science, 34*(1), 162–178. https://doi.org/10.35378/gujs.690131

Camacho-Villalón, C. L., Stützle, T., & Dorigo, M. (2023). Designing new metaheuristics: Manual versus automatic approaches. *Intelligent Computing, 2*. https://doi.org/10.34133/icomputing.0048

ElMolouk, R. S., El-Kharbotly, A. M. K., & Taha, R. B. (2025). Optimization of time and energy in straight one-sided robotic assembly lines. *Scientific Reports, 15*(1), Artículo 13227. https://doi.org/10.1038/s41598-025-94202-y

Imbien, J. R. (2025). *Optimizing collaborative edge-cloud computing: A task allocation strategy using hybrid genetic algorithm and tabu search* [Tesis de maestría, Oslo Metropolitan University].

Prasad, S. D., Sharma, V., Jain, A., & Sahu, D. K. (2022). Productivity improvement using DPE-assembly line balancing. *International Journal of Engineering Applied Sciences and Technology, 7*(2), 177–188. https://doi.org/10.33564/IJEAST.2022.v07i02.025

Sotskov, Y. N. (2023). Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and

perspectives. *Algorithms, 16*(2), 100. https://doi.org/10.3390/a16020100