

Improving AGV path planning efficiency using genetic algorithms with hamming distance-based initialization

Breznikar, Ž.^a, Gotlih, J.^a, Artič, Ž.^b, Brezocnik, M.^a

^aUniversity of Maribor, Faculty of Mechanical Engineering, Maribor, Slovenia

^bUniversity of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia

ABSTRACT

This paper presents a Genetic Algorithm (GA) framework for warehouse navigation as a Travelling Salesman Problem (TSP) variant for Automated Guided Vehicles (AGVs). The warehouse layout is represented as a graph, where pick-up locations serve as terminal nodes. A distance matrix, computed via Breadth-First Search (BFS) enables efficient route evaluation. To promote diversity in the initial population, a Hamming distance-based vectorized initialization strategy is employed, ensuring that the chromosomes are maximally distinct. The GA balances exploration and exploitation by dynamically adjusting the fitness function. Early generations emphasize diversity, while later ones focus on solution refinement, improving convergence and avoiding premature stagnation. Our key contribution demonstrates that the Hamming distance-based approach achieves comparable or better results with significantly fewer chromosomes. This reduces computational cost and runtime, making the method well-suited for real-time AGV routing in warehouses. The framework is adaptable to structured environments and shows strong potential for integration into real-world logistics and robotics applications. Future work will focus on optimizing the algorithm and integrating it into the ROS 2 environment. The simplified version of the algorithm can be accessed at: <https://github.com/IntoTheVoid-61/Warehouse-Pathfinder>.

ARTICLE INFO

Keywords:

Automated guided vehicles (AGV);
Warehouse routing;
Genetic algorithms (GA);
Combinatorial optimization,
Hamming distance initialization;
Robot operating system 2 (ROS 2)

*Corresponding author:

ziga.breznikar@student.um.si
(Breznikar, Ž.)

Article history:

Received 6 July 2025
Revised 24 October 2025
Accepted 27 October 2025



Content from this work may be used under the terms of the Creative Commons Attribution 4.0 International Licence (CC BY 4.0). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

1. Introduction

The rapid expansion of automated manufacturing has brought increasing demand for intelligent logistics and autonomous systems in warehouses. Among these, Automated Guided Vehicles (AGVs) play a crucial role in transporting goods efficiently within complex environments. A considerable body of research has investigated the application of AGV systems across a wide range of environments and operational contexts, supported by diverse methodological approaches. In recent years, studies have increasingly adopted metaheuristic techniques to enhance system performance and optimization [1–10]. As warehouse layouts become more intricate and dynamic, determining optimal routes for AGVs becomes a critical challenge, essential for reducing delivery times and operational costs.

Genetic Algorithms (GAs), inspired by the principles of natural selection and evolution, are well suited for solving complex optimization problems such as AGV routing. By evolving population of candidate solutions through operations, inspired by biological evolution, GAs can efficiently explore large solution spaces and efficiently convergence towards near-optimal solutions even in the presence of multiple constraints and non-linearities. Their flexibility and robustness

have made them a popular choice for various combinatorial problems, including the well-known Travelling Salesman Problem (TSP) [11, 12], which closely parallels AGV routing in warehouses.

Traditional or uninformed search strategies, such as brute-force or blind random search, quickly become impractical as the scale of the problem increases [13]. In high-dimensional warehouse environments, the combinatorial explosion of possible routes leads to significant computational costs and sub-optimal outcomes. This underscores the need for heuristic or metaheuristic approaches that can guide the search process intelligently.

In this study, we propose a GA-based approach for warehouse routing, tailored for a top-down two-dimensional warehouse representation. Our algorithm encodes AGV routes as sequence of terminal locations and evolves these sequences to minimize travel distance. A central focus of this work is the effect of initial population diversity on GA performance. Specifically, we compare two initialization strategies: the conventional random initialization and a Hamming distance-based approach [14] explicitly designed to maximize population diversity, thereby enhancing exploratory capabilities in the early stages of the GA.

To evaluate the efficiency of the proposed method, extensive experiments were conducted using both initialization strategies under controlled settings. Results demonstrate that Hamming distance-based initialization performs comparably or better than standard approach, even when using significantly fewer individuals per generation. This indicates that strategic population design can reduce computational costs while maintaining solution quality—a critical insight for real-time AGV routing applications in operational warehouse systems.

The remainder of this paper presents the implementation of our GA-based routing algorithm, the design of the Hamming-based initialization, and detailed statistical analysis of the results obtained through multiple experimental runs.

2. Related work

The Travelling Salesman Problem (TSP) is a classic benchmark in combinatorial optimization and has been extensively addressed using GAs due to their ability to explore large, complex search spaces and avoid local optima through evolutionary operators. Standard GA implementations typically start with randomly generated populations, which help avoid early convergence but still frequently lead to premature stagnation in complex or highly constrained problems.

In the context of warehouse logistics, the TSP is frequently adapted to model route optimization for AGVs, where efficient sequencing of pick-up and delivery tasks is critical. Several works have proposed heuristic and metaheuristic-based solutions, including Ant Colony Optimization (ACO) and GA-based frameworks [11]. However, many approaches assume idealized conditions or rely on brute-force exploration, which becomes computationally expensive as the number of tasks and constraints grow. As a general approach, the authors in [15] proposed an enhanced GA for the TSP, in which the population is initialized using the Iterative Approximate Method, significantly improving solution efficiency and convergence speed.

Recent advancements have focused on enhancing diversity within GA population to improve exploration and convergence stability. For example, some authors have introduced adaptive mutations rates or hybridized GAs with local search techniques to maintain population diversity. Hamming distance, a measure of dissimilarity between binary strings, has been proposed for initializing populations that are maximally distinct [16]. While it has been applied in other domains, its use in structured, graph-based environments like warehouse routing remains relatively underexplored.

To our knowledge, few studies have empirically compared Hamming distance-based initialization with traditional random initialization in the context of AGV routing on warehouse graphs, employing statistically rigorous analysis to evaluate performance and convergence behaviour. This paper contributes to the existing literature by integrating a Hamming distance-based population initialization within a GA specifically designed for AGV navigation. By fostering high initial diversity, the algorithm reduces reliance on large population sizes, thus enhancing computational efficiency without sacrificing solution quality.

3. Environmental modelling and preprocessing

3.1 Matrix-based environment encoding

The warehouse environment is defined by three structural parameters:

- number of aisles,
- number of storage locations per aisle,
- number of storage blocks.

These parameters provide a compact and flexible description of the warehouse layout, which is essential for scalable simulations and graph-based modelling.

To facilitate algorithmic processing, the physical warehouse is first abstracted into a two-dimensional matrix. Each cell in the matrix corresponds to a discrete warehouse location and is assigned an integer label representing its functional role. Fig. 1 illustrates the top-down view of the warehouse layout and its matrix representation.

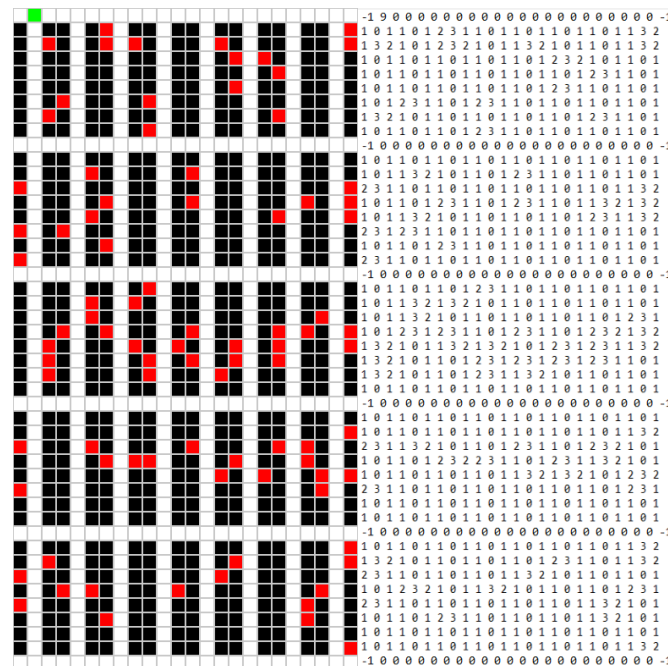


Fig. 1 Top-down view of the warehouse layout and its matrix representation

Table 1 Classification of colours and numerical values from Fig. 1

Value	Colour code	Description
0	White	Free space: Traversable
1	Black	Empty storage location: Untraversable
2	Red	Storage location: Untraversable
3	White	Pickup location: Traversable
9	Green	Start and end point: Traversable

Table 1 summarises the classification scheme used to assign numerical values to different elements of the warehouse.

Only the cells labelled 0, 3 and 9 are traversable. Among these, all nodes labelled as 3 represent mandatory pickup locations that the AGV must visit at least once. This matrix formulation allows a direct transformation into a graph, where each traversable cell becomes a node and edges represent valid moves between adjacent cells.

3.2 Graph-based preprocessing

To enable efficient path planning and distance evaluation, the matrix representation of the warehouse is transformed into a weighted graph [17]. In this graph, each traversable matrix cell (labelled as 0, 3 or 9) becomes a node. An undirected edge is created between every pair of adja-

cent traversable nodes. By default, edges are assigned a weight of 1, representing uniform movement cost. However, in real-world applications, factors such as bottlenecks, blocked or narrow passages, and other environmental constraints may increase the traversal difficulty. Such conditions are modelled by assigning higher weights to the affected edges.

Once the graph is constructed, all terminal nodes, marked with the label 3 in the matrix, are identified. These represent the pickup locations that the AGV must visit at least once. The set of terminals serves as the basis for solving a TSP-like optimization task.

To quantify distances between terminals, a distance matrix is computed using a BFS algorithm [18]. For each terminal node, BFS calculates the shortest path (in terms of total edge weight) to every other terminal. The resulting distance matrix is a symmetric square matrix where each element $D_{i,j}$ represents the shortest traversable distance between terminals i and j .

The distance matrix serves as a critical input to the optimization algorithm. It allows for rapid evaluation of the total route length of any candidate solution, without requiring real-time path-finding through the graph [19]. This preprocessing step thus transforms the original navigation problem into a purely combinatorial optimization task, significantly reducing computational overhead during the evolutionary search.

4. Genetic algorithm optimization

4.1 Chromosome representation and problem complexity

To apply a GA to the problem of AGV routing within a warehouse, we first define how a potential solution (chromosome) is represented. Each chromosome encodes a specific sequence in which the autonomous vehicle should visit all required terminal nodes, that is, the pickup locations identified during preprocessing.

A chromosome consists of a permutation of all terminal nodes, where each gene represents a single terminal node, and the order of genes determines the traversal path of the AGV. This is represented in Fig. 2, which illustrates 5 different potential solutions to the routing problem, given 8 pickup locations.

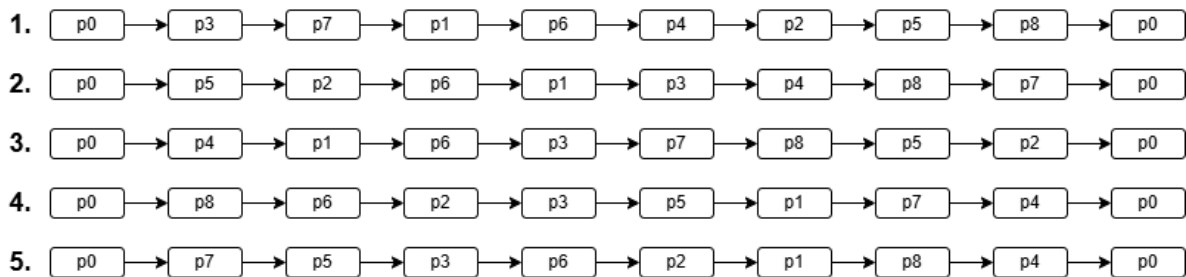


Fig. 2 Example of 5 different solutions to the routing problem with 8 terminal nodes.

This problem closely resembles the mentioned TSP, a well-known combinatorial optimization problem. For n terminal nodes, there are $n!$ possible permutations, making exhaustive search methods computationally infeasible, even for relatively small values of n . As the problem scales, brute-force methods become impractical due to factorial growth in complexity.

Given a set of terminals $T = \{p_1, p_2, \dots, p_n\}$, the objective is to find a permutation π , as shown in Eq. 1, that minimizes the total travel distance.

$$\pi = \sum_{i=1}^n D_{\pi(i), \pi(i+1)} \quad (1)$$

Since brute-force methods become computationally infeasible for large n , a GA is employed to address this challenge. GA are well-suited for permutation-based combinatorial problems and enable efficient exploration of large solution spaces by evolving a population of chromosomes through genetic operations.

4.2 Hamming distance-based population initialization

A critical step in ensuring the effectiveness of a GA is the initialization of a diverse population. Diversity promotes broad exploration of the solution space in early generations and helps avoid premature convergence to local optima. To systematically promote diversity, we employed a strategy based on the Hamming distance.

For two sequences of equal length, the Hamming distance is defined as the number of positions at which the corresponding elements differ [20]. In our context, each chromosome is a permutation of terminal nodes, and the Hamming distance between two chromosomes indicates the number of differing terminal positions in the visitation sequence.

The population is initialized by iteratively generating random permutations and comparing them against the already selected chromosomes. A candidate chromosome is accepted into the population only if a minimum Hamming distance from the set of existing chromosomes exceeds a decreasing threshold, starting from the maximum value n (the number of terminals). This ensures that the initial population is highly diverse, therefore promoting broad exploration of the solution space. If, after a predefined number of attempts, no chromosome meets the current Hamming threshold, the threshold is reduced by one and the process repeats. This adaptive mechanism balances diversity with feasibility. A high-level implementation of the described method is shown in Fig. 3.

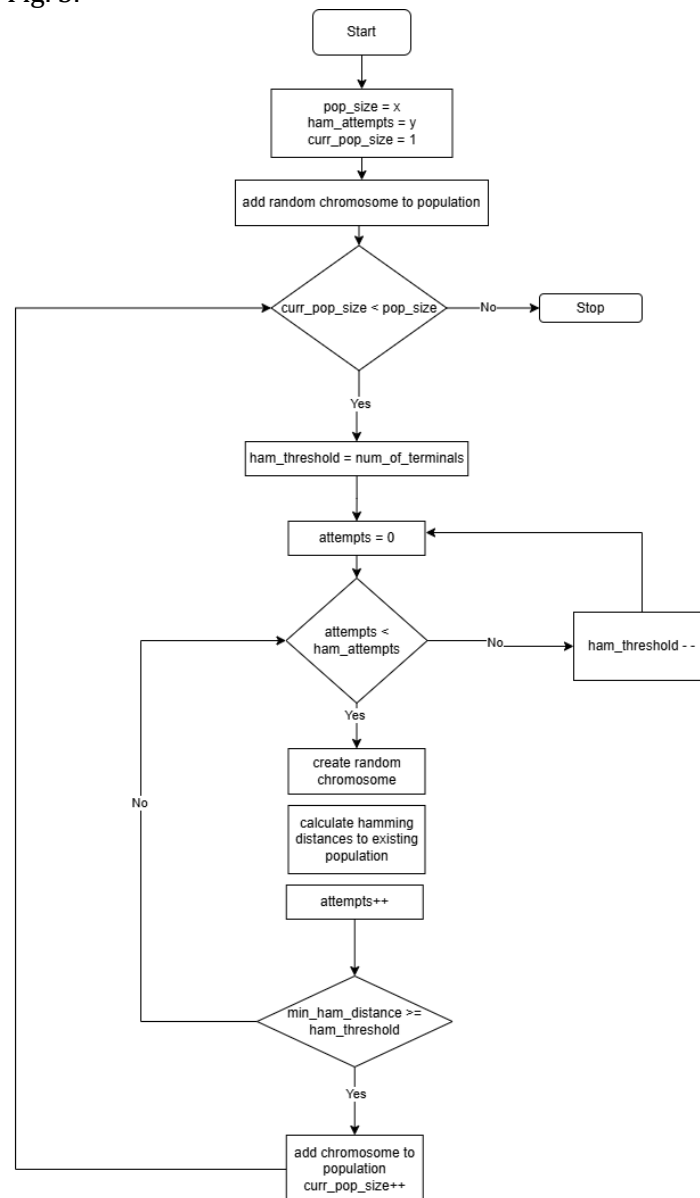


Fig. 3 Flowchart of a high-level hamming distance-based algorithm to ensure initial population diversity.

This Hamming distance-based algorithm ensures diversity in the starting population, enabling a broader and more efficient exploration of the solution space. The computational complexity of the initialization process increases approximately as $O(P^2 A n)$, where P denotes the population size, n the number of pickup locations, and A the number of random Hamming distance attempts per chromosome.

4.3 Evolutionary dynamics and evaluation

This section describes the key components of the GA, including selection mechanisms, elitism strategy, mutation operators, and the dynamic fitness evaluation. These mechanisms were designed to balance exploration and exploitation throughout the optimization process.

A dynamic tournament selection method is used to select chromosomes for reproduction or crossover. In each tournament, a subset of k chromosomes from the current population is randomly selected, and the best-performing individual is chosen. To balance exploration and exploitation over time, the tournament size is dynamically adjusted, as shown in Eq. 2.

$$k = \text{round}((k_{\max} - k_{\min}) \cdot \text{progress} + k_{\min}) \quad (2)$$

where *progress* is defined in Eq. 3.

$$\text{progress} = \frac{\text{current_generation}}{\text{max_generation}} \quad (3)$$

This allows smaller tournament sizes early on (encouraging diversity) and larger tournaments later (favouring selection pressure). Following the selection, each chromosome is either reproduced directly into the next generation with probability p , or undergoes crossover with a different chromosome with probability $1 - p$.

To preserve high quality solutions, elitism is applied by copying the top i chromosomes unaltered into the next generation. The number of elite chromosomes increases quadratically with the progress of generations as shown in Eq. 4.

$$i = \text{round}((i_{\max} - i_{\min}) \cdot \text{progress}^2 + i_{\min}) \quad (4)$$

This strategy ensures that more optimal solutions are retained as the algorithm converges.

Two types of mutation operators are employed to maintain genetic diversity:

- Single-gene swap: Two genes (positions) in the chromosome are randomly swapped [21].
- 2-opt-swap: A sub-sequence of genes is reversed, a common local optimization technique in TSP-like problems [22].

The mutation probability is dynamic and is sampled from a uniform distribution $U(a,b)$, where bounds a and b evolve as generations progress, as shown in Eq. 5 and Eq. 6.

$$a = (a_{\text{end}} - a_{\text{start}}) \cdot \text{progress} + a_{\text{start}}, \text{ where } a_{\text{start}} > a_{\text{end}} \quad (5)$$

$$b = (b_{\text{end}} - b_{\text{start}}) \cdot \text{progress} + b_{\text{start}}, \text{ where } b_{\text{start}} > b_{\text{end}} \quad (6)$$

This allows higher variability early on (favouring exploration), which gradually decreases as the algorithm approaches convergence.

The fitness function transitions from the exploration phase to the exploitation phase to balance global search and solution refinement. During the exploration phase, the fitness incorporates both solution quality and population diversity, as shown in Eq. 7.

$$f = \alpha \cdot \text{normalized_path_length} - (1 - \alpha) \cdot \text{normalized_hamming_distance} \quad (7)$$

Normalization is necessary to account for differing numerical scales. Path lengths are normalized relative to population statistics, while Hamming distances are normalized relative to the maximum possible value, corresponding to the chromosome length. The coefficient α balances the two terms and is defined by a sigmoid function [23] as shown in Eq. 8.

$$\alpha = \frac{1}{1 + e^{-x}} \quad (8)$$

where x is defined as shown in Eq. 9.

$$x = 20 \cdot \frac{\text{generation_counter}}{\text{exploration_end}} - 10 \quad (9)$$

In Eq. 9, *generation_counter* is an integer value representing the current generation number, while *exploration_end* is the predefined generation at which exploitation begins. During the exploitation phase, fitness is based solely on the provided path length of a chromosome. This phase focuses entirely on improving the solution quality.

A high-level schematic of the proposed GA framework is illustrated in Fig. 4.

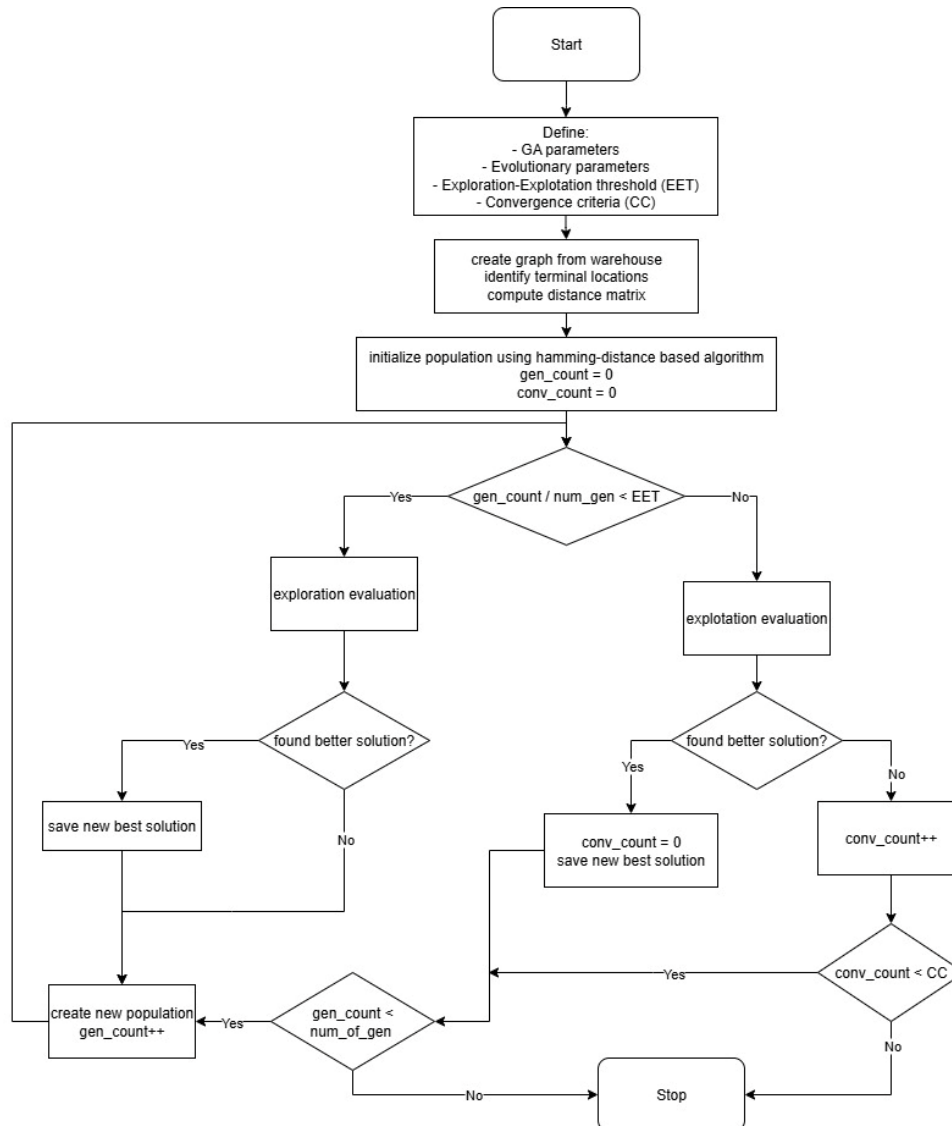


Fig. 4 Algorithmic structure of the GA-based navigation framework

5. Results and discussion

This section presents a comprehensive evaluation of the impact of Hamming distance-based population initialization (H) compared to random initialization (NH) in a GA designed for optimizing warehouse routing. The analysis spans four key performance metrics:

- runtime duration,
- initial population diversity,
- convergence speed,
- final solution quality (best tour length).

The experiments were performed over 20 independent runs per setting, using identical seeds and warehouse environments to ensure consistency.

5.1 Equal population size comparison

In the first series of experiments, both methods were executed under identical setting with 30 individuals per generation. Each configuration was repeated 20 times using the same seeds and warehouse environment. To validate the use of parametric tests, Shapiro-Wilk and D'Agostino-Pearson normality tests were conducted for all metrics [24]. Results indicated normal distribution across all groups. Table 2 summarizes the results.

Table 2 Statistical analysis of GA metrics: Hamming vs. Non-Hamming

Metric	Hamming (Mean $\pm$ SD)	Non-Hamming (Mean $\pm$ SD)	<i>p</i> -value	Significance
Runtime (s)	45.0283 $\pm$ 5.3	40.2011 $\pm$ 5.79	0.0057	significant
Initial population Hamming distance	80.3567 $\pm$ 0.03	78.2347 $\pm$ 0.05	< 0.0001	significant
Generations	6122.05 $\pm$ 692.09	5907.45 $\pm$ 739.28	0.2904	not significant
Best tour length	400.8000 $\pm$ 11.19	403.6000 $\pm$ 9.46	0.3915	not significant

Key takeaways:

- Population diversity: The Hamming-initialized population achieved significantly higher diversity, confirming the effectiveness.
- Runtime: Hamming initialization introduced a slight computational overhead due to pairwise distance calculations.
- Final Quality & Convergence: No statistically significant improvement was observed.

5.2 Exploring population size effects

Additional experiments were conducted to investigate whether Hamming distance-based initialization can effectively compensate for a reduced population size in GA applications. This evaluation was carried out on a structurally distinct warehouse layout to ensure generalizability of the findings. In this setting, we compared the performance of a GA configured with only 30 individuals initialized using the Hamming distance strategy (H30) against a GA employing 150 individuals initialized randomly (NH150). Despite the fivefold disparity in population size, the H30 configuration consistently demonstrated comparable solution quality to the NH150 setup in both final tour length and convergence behaviour.

Statistical analysis of the tour length distributions supported this finding, yielding a significant *p*-value [25] ($p < 0.05$) indicating that the performance difference favoured the H30 configuration. These results underscore the effectiveness of diversity-promoting strategies in evolutionary algorithms. Instead of compensating for insufficient diversity by brute-force scaling of the population size, initializing the population with maximally dissimilar chromosomes enables more efficient exploration of the solution space. This provides a strong argument for adopting informed initialization techniques, especially in resource-constrained environments where computational efficiency is critical, such as real-time AGV routing in dynamic warehouse settings.

6. Conclusion

This work explored the application of a GA to solve warehouse routing problem, a task critical to the efficiency of AGV system in logistics and manufacturing environments. Our results confirm that a GA-based approach is not only viable but effective for generating high quality routing solutions within reasonable computational budget. The simplified version of the algorithm can be accessed at: <https://github.com/IntoTheVoid-61/Warehouse-Pathfinder>. Future work of the algorithm will extend to its optimization and integration into the ROS2 environment.

In a targeted comparison, a GA with just 30 chromosomes initialized using Hamming-based method achieve statistically comparable results in tour quality relative to that of a GA with 150 randomly initialized individuals. This outcome highlights the central contribution of our study:

intelligent population seeding can significantly reduce the required population size without compromising performance.

This finding has tangible implications for real-world deployment. Reducing population size lowers computational time and memory usage, making approach more suited for embedded or real-time systems commonly used in AGV applications. Accordingly, this study provides a practical and scalable GA-based framework for warehouse routing, with added benefit of an initialization method that enhances performance under constrained resources.

Future work should focus on deploying the proposed system on real-world AGV platforms and evaluating its performance through experiments and field-testing. While the algorithm demonstrates strong results in two-dimensional navigation (based on top-down view of the warehouse), it does not yet account for three-dimensional considerations. Furthermore, as this study was conducted on an abstracted AGV model, specific characteristics of actual vehicle were not incorporated. Notably, the model assumed that AGV could transport an unlimited mass of cargo and did not require a return to the starting point for unloading. Future implementations on embedded system should therefore account for such practical constraints, including limited payload capacity and the need for cargo drop-off behaviour.

References

- [1] Löffler, M., Boysen, N., Schneider, M. (2021). Picker routing in AGV-assisted order picking systems, *INFORMS Journal on Computing*, Vol. 34, No. 1, 440-462, doi: [10.1287/ijoc.2021.1060](https://doi.org/10.1287/ijoc.2021.1060).
- [2] Agatz, N., Bouman, P., Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone, *Transportation Science*, Vol. 52, No. 4, 965-981, doi: [10.1287/trsc.2017.0791](https://doi.org/10.1287/trsc.2017.0791).
- [3] Han, Z., Wang, D., Liu, F., Zhao, Z. (2017). Multi-AGV path planning with double-path constraints by using an improved genetic algorithm, *Plos One*, Vol. 12, No. 7, Article Id. e0181747, doi: [10.1371/journal.pone.0181747](https://doi.org/10.1371/journal.pone.0181747).
- [4] Simon, J., Trojanová, M., Hošovský, A., Sárosi, J. (2021). Neural network driven automated guided vehicle platform development for Industry 4.0 environment, *Tehnički Vjesnik – Technical Gazette*, Vol. 28, No. 6, 1936-1942, doi: [10.17559/TV-20200727095821](https://doi.org/10.17559/TV-20200727095821).
- [5] Bostancı, B., Karaağaç, A. (2019). Investigating the shortest survey route in a GNSS traverse network, *Tehnički Vjesnik – Technical Gazette*, Vol. 26, No. 2, 355-362, doi: [10.17559/TV-20170924174221](https://doi.org/10.17559/TV-20170924174221).
- [6] Xu, L., Wang, Y., Liu, L., Wang, J. (2016). Exact and heuristic algorithms for routing AGV on path with precedence constraints, *Mathematical Problems in Engineering*, Vol. 2016, No. 1, Article No. 5040513, doi: [10.1155/2016/5040513](https://doi.org/10.1155/2016/5040513).
- [7] Wang, Y.J., Liu, X.Q., Leng, J.Y., Wang, J.J., Meng, Q.N., Zhou, M.J. (2022). Study on scheduling and path planning problems of multi-AGVs based on a heuristic algorithm in intelligent manufacturing workshop, *Advances in Production Engineering & Management*, Vol. 17, No. 4, 505-513, doi: [10.14743/apem2022.4.452](https://doi.org/10.14743/apem2022.4.452).
- [8] Wang, Y.D., Lu, X.C., Song, Y.M., Feng, Y., Shen, J.R. (2022). Monte Carlo Tree Search improved Genetic Algorithm for unmanned vehicle routing problem with path flexibility, *Advances in Production Engineering & Management*, Vol. 17, No. 4, 425-438, doi: [10.14743/apem2022.4.446](https://doi.org/10.14743/apem2022.4.446).
- [9] Boccia, M., Masone, A., Sterle, C., Murino, T. (2023). The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach, *European Journal of Operational Research*, Vol. 307, No. 2, 590-603, doi: [10.1016/j.ejor.2022.10.023](https://doi.org/10.1016/j.ejor.2022.10.023).
- [10] Micieta, B., Durica, L., Binasova, V. (2018). New solution of abstract architecture for control and coordination decentralized systems, *Tehnički Vjesnik – Technical Gazette*, Vol. 25, Supplement 1, 135-143, doi: [10.17559/TV-20160117100949](https://doi.org/10.17559/TV-20160117100949).
- [11] Wang, Y.D., Lu, X.C., Shen, J.R. (2021). Improved Genetic Algorithm (VNS-GA) using polar coordinate classification for workload balanced multiple Traveling Salesman Problem (mTSP), *Advances in Production Engineering & Management*, Vol. 16, No. 2, 173-184, doi: [10.14743/apem2021.2.392](https://doi.org/10.14743/apem2021.2.392).
- [12] Sui, J., Ding, S., Huang, X., Yu, Y., Liu, R., Xia, B., Ding, Z., Xu, L., Zhang, H., Yu, C., Bu, D. (2025). A survey on deep learning-based algorithms for the traveling salesman problem, *Frontiers of Computer Science*, Vol. 19, Article No. 196322, doi: [10.1007/s11704-024-40490-y](https://doi.org/10.1007/s11704-024-40490-y).
- [13] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2009). *Introduction to algorithms*, Third edition, MIT Press, Cambridge, Massachusetts, USA.
- [14] Leung, Y., Gao, Y., Xu, Z.-B. (1997). Degree of population diversity - a perspective on premature convergence in genetic algorithms and its Markov chain analysis, *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, 1165-1176, doi: [10.1109/72.623217](https://doi.org/10.1109/72.623217).
- [15] Alkafaween, E., Hassanat, A., Essa, E., Elmougy, S. (2024). An efficiency boost for genetic algorithms: Initializing the GA with the iterative approximate method for optimizing the Traveling Salesman Problem – Experimental insights, *Applied Sciences*, Vol. 14, No. 8, Article No. 3151, doi: [10.3390/app14083151](https://doi.org/10.3390/app14083151).

- [16] Cruz Chávez, M.A., Martínez-Oropeza, A. (2016). Feasible initial population with genetic diversity for a population-based algorithm applied to the vehicle routing problem with time windows, *Mathematical Problems in Engineering*, Vol. 2016, No. 1, Article No. 3851520, [doi: 10.1155/2016/3851520](https://doi.org/10.1155/2016/3851520).
- [17] Diestel, R. (2017). *Graph theory*, 5th edition, Springer, Berlin, Germany, [doi: 10.1007/978-3-662-53622-3](https://doi.org/10.1007/978-3-662-53622-3).
- [18] Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2014). *Data structures & algorithms in Java*, 6th edition, Wiley, Hoboken, New Jersey, USA.
- [19] Deza, M.M., Deza, E. (2013). *Encyclopedia of distances*, 2nd edition, Springer, Berlin, Germany, [doi: 10.1007/978-3-642-30958-8](https://doi.org/10.1007/978-3-642-30958-8).
- [20] Black, P.E. (2006). Hamming distance, In: Black, P.E. (ed.), *Dictionary of algorithms and data structures* [online], Black, P.E. (ed.), from <https://www.nist.gov/dads/HTML/HammingDistance.html>, accessed May 13, 2025.
- [21] Sivanandam, S., Deepa, S. (2008). Genetic algorithms, In: Sivanandam, S., Deepa, S. (eds.), *Introduction to genetic algorithms*, Springer, Berlin, Germany, 15-37, [doi: 10.1007/978-3-540-73190-0_2](https://doi.org/10.1007/978-3-540-73190-0_2).
- [22] Uddin, F., Riaz, N., Manan, A., Mahmood, I., Song, O.-Y., Malik, A.J., Abbasi, A.A. (2023). An improvement to the 2-Opt heuristic algorithm for approximation of optimal TSP tour, *Applied Sciences*, Vol. 13, No. 12, Article No. 7339, [doi: 10.3390/app13127339](https://doi.org/10.3390/app13127339).
- [23] Science Direct. Sigmoid Function, from <https://www.sciencedirect.com/topics/computer-science/sigmoid-function>, accessed April 4, 2025.
- [24] Altman, D.G., Bland, J.M. (1995). Statistics notes: The normal distribution, *The BMJ*, Vol. 310, Article No. 298, [doi: 10.1136/bmj.310.6975.298](https://doi.org/10.1136/bmj.310.6975.298).
- [25] Gao, J. (2020). P-values – a chronic conundrum, *BMC Medical Research Methodology*, Vol. 20, Article No. 167, [doi: 10.1186/s12874-020-01051-6](https://doi.org/10.1186/s12874-020-01051-6).