Mechatronics and Intelligent Transportation Systems
https://www.acadlore.com/journals/MITS

Optimising AGV Routing in Container Terminals: Nearest Neighbor vs. Tabu Search

Adis Puška[1], Jurica Bosna[2], Nikola Petrović[3*], Saša Marković[3]

[1] Department of Public Safety, Government of Brčko District, 76100 Brčko, Bosnia and Herzegovina
[2] Department of Economics, University of Zadar, 23000 Zadar, Croatia
[3] Faculty of Mechanical Engineering, University of Nis, 18000 Niš, Serbia

[*] Correspondence: Nikola Petrović (petrovic.nikola@masfak.ni.ac.rs)

Received: 08-29-2024 **Revised:** 09-30-2024 **Accepted:** 10-08-2024

Citation: A. Puška, J. Bosna, N. Petrović, and S. Marković, "Optimising AGV routing in container terminals: Nearest neighbor vs. Tabu search," *Mechatron. Intell Transp. Syst.*, vol. 3, no. 4, pp. 203–211, 2024. https://doi.org/10.56578/mits030401.

Abstract: Automated Guided Vehicles (AGVs) represent a transformative advancement in the automation of transport operations, facilitating unmanned mobility within a wide array of environments, including production lines, warehouses, freight hubs, and terminal operations. In container terminals, where AGVs are increasingly deployed, the routing of these vehicles is a critical task aimed at minimising operational inefficiencies such as travel time, fuel consumption, and overall transportation costs. Routing in this context refers to the determination of optimal paths for a fleet of AGVs, which must satisfy a variety of operational constraints while also adhering to predefined user requirements. Given the high complexity of these problems, characterised by a large solution space, finding exact solutions is computationally intractable for most scenarios. As a result, heuristic methods are commonly employed to approximate optimal solutions. Among the various heuristic techniques, the nearest neighbor algorithm and Tabu search have been identified as promising approaches for determining efficient AGV routes in container terminal environments. These methods are applied to identify paths that minimise travel distance and time, enhancing resource utilisation and improving the overall reliability of goods delivery. The application of these algorithms is expected to lead to a significant reduction in the number of kilometres travelled by AGVs, thereby lowering operational costs and improving service efficiency. This paper examines the efficacy of the "nearest neighbor" and Tabu search algorithms in the context of AGV routing at container terminals, highlighting their potential to optimise fleet operations in the face of complex logistical challenges. Emphasis is placed on the comparative analysis of both algorithms, with a focus on their ability to approximate optimal solutions in dynamic and highly constrained environments.

Keywords: Automated Guided Vehicles (AGVs); Routing; Nearest neighbor algorithm; Tabu search

1 Introduction

AGVs are unmanned vehicles that operate using an automatic control system. AGVs are capable of loading, transporting, and unloading materials without human intervention. Also, AGVs are commonly used in production plants, warehouses, transshipment terminals, and distribution centers. With them, it can transport a wide range of loads, from small items weighing a few kilograms to heavy loads exceeding 100 tons.

The use of AGVs offers numerous advantages that significantly enhance the efficiency and quality of business processes. One key benefit is the system's flexibility, allowing for easy adjustments to transport paths and a quick response to changes in production demands. The installation of AGV systems is relatively straightforward, and users can increase transport capacity by simply adding more vehicles, enabling optimal adaptation to their needs.

Moreover, AGVs help reduce the risk of damage to goods due to their precise and controlled movements, which can deviate by only ±10 mm. Implementing AGVs also improves working conditions, especially in environments that involve hazardous materials or harsh conditions, making the workplace safer and more user-friendly for employees. The automation of processes with AGVs allows for continuous and reliable system management, with the ability to integrate complex information technologies.

Additionally, AGVs enhance the organization and optimization of production and logistics systems, thereby decreasing the need for a large workforce. AGVs have the capability to operate 24 hours a day, seven days a week, without breaks, AGVs are particularly well-suited for intensive and continuous processes. All these advantages make AGV systems an exceptionally valuable solution for improving modern industrial and logistics operations.

The main components of an AGV include the navigation system, the control and communication system, the security system, the energy system, and the cargo handling system, as shown in Figure 1.

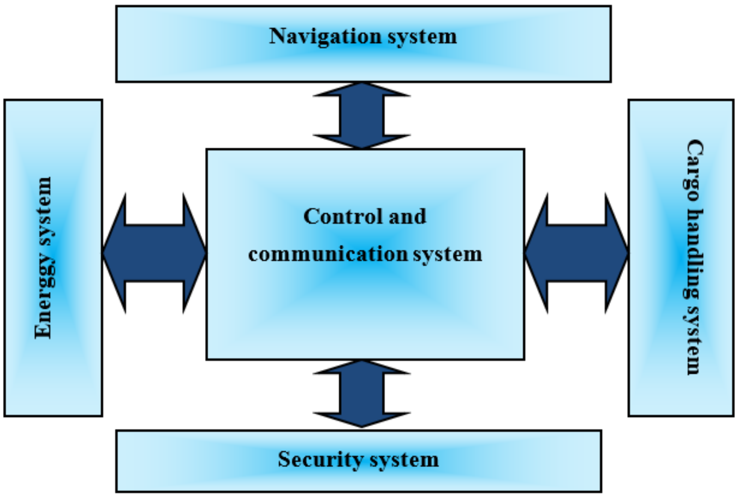

Figure 1. Schematic view of AGV main parts

Communication is the way the control and monitoring system can contact the vehicles. Messages can be of the type where, when, when to stop, when to slow down, etc. It may also contain error reports. Depending on the application, there are four basic types of data transmission through which communication is achieved: Radio connection, Infrared connection, Wired connection, and Connection via inductive connection.

There are two basic principles on which AGV navigation is based: fixed path and free navigation (laser, inertial, GPS and grid).

AGVs are powered by battery energy sources. The technology itself that is used depends on the working conditions in which the vehicle works. These can be the temperature of the environment, the number of state changes, the type of vehicle, the weight of the load, as well as the specific requirements of the user.

The purpose of the AGV obstacle detection system is to detect all obstacles on the road, in time, to stop the vehicle in time, while the path is blocked. When the path is clear again, the AGV will automatically continue. Immovable objects (shelves, walls, racks, etc.) must be detected to avoid collision or collision with them.

There are the following systems at AGV for the safe performance of tasks: bumpers, laser scanners, side protection, E-stop, rotating lights and sound warning system, and anti-shock sensors.

The implementation of Automated Guided Vehicle Systems (AGVS) requires specially trained personnel to oversee both the operation and maintenance of the vehicles and their accompanying equipment. This includes tasks such as online traffic management, adapting to real-time changes in system requirements, and ensuring smooth system performance. One of the most common challenges in AGVS applications is traffic management, particularly at critical points where AGV flows separate, merge, or cross. The key aspects of AGVS design and management involve designing the movement paths, determining vehicle requirements or the number of vehicles needed, planning vehicle deployment and operation schedules, positioning vehicles, managing battery usage, optimizing vehicle routing, and preventing or resolving traffic jams [1].

Beamon [2] described several important AGVS performance criteria such as: vehicle travel time, vehicle utilisation, queue length, and material handling costs.

AGVS consists of two interconnected subsystems: hardware and software. Hardware consists of physical components such as the AGV vehicle, tracks, controllers, sensors, guidance devices, etc. [3]. The software is a representation of approaches and algorithms for the systematic management of the AGV's hardware resources so that the entire system can function uniformly with the highest degree of efficiency. In the past four decades, hardware development has been far more intensive than software development. The software problem has become relevant in recently developed applications, where the time factor is of particular importance (for example, handling containers in ports, the need to perform a large number of operations in real time with the inclusion of a relatively large fleet of hundreds or more AGVs). Numerous problems, such as congestion and downtime of AGVs, are the result of inappropriate software.

Not all combinatorial optimisation problems belong to the class of NP-hard problems. While for some there are algorithms that solve them in polynomial time, many are NP-hard, where the solution requires searching the solution space whose size grows exponentially with the dimension of the problem. Heuristics often find suboptimal solutions in such cases. Heuristics are divided into constructive and improvement heuristics. Constructive heuristics gradually build a solution from an initial empty state according to predefined criteria and are faster. However, they usually produce lower-quality solutions than improvement heuristics, which systematically search the solution space and generate better suboptimal solutions until a stopping condition, such as a time limit or satisfactory accuracy, is met. Constructive heuristics are often used to initialise solutions in improvement heuristics. Various solution space search strategies have been developed to optimise this process, and these strategies, known as metaheuristics, provide a framework for creating efficient heuristics for specific problems [4–7].

AGV routing can be viewed as a specific variant of the vehicle routing problem (VRP), which is usually solved using optimisation techniques such as linear and integer programming. However, there are several key differences between these two problems, which make it necessary to approach them separately [8, 9]:

Road network: In VRPs, the analysed network mainly represents a large-scale urban network, where the length of the vehicle is negligible compared to the distance travelled, so the vehicles can be considered as points in motion. On the other hand, with AGVS, the distances between the starting and destination points are relatively short (approximately ten AGV lengths), which means that the road segment occupied by the vehicle cannot be neglected.

Road capacity and congestion: In classic VRP, road capacity is usually not a constraint, and congestion or collisions, except at intersections, are not taken into account. With AGVS, due to the limited possibilities of moving along the set paths, congestion or collisions may occur if the vehicles are not adequately distributed and routed.

Optimisation criteria: In VRP, the shortest path often corresponds to the shortest travel time, even if it is not the most profitable. However, with AGVS, due to the occupancy of certain routes, the shortest route is not always the fastest.

Network variability: The road network in VRP is usually considered static, while in AGVS the route layout can be adjusted to achieve more efficient vehicle routing and deployment.

In this paper, the routing process, i.e., the determination of the shortest path, of AGV vehicles at the container terminal using “nearest neighbour” and Tabu search algorithms will be discussed in detail. Finding optimal routing plans is an effective approach to enhance efficiency and minimize costs.

2 Literature Review

In the past few decades, considerable research work has been devoted to AGV routing problems. Numerous algorithms have been proposed to solve this problem. Heuristics are methods used to quickly find ”good” solutions, even when the optimal solution is not practical to achieve due to time or computational constraints. However, most of the obtained results are applicable only in situations with a small number of AGVs and a low degree of competitiveness. With the rapid increase in the number of AGVs, there is a need to find more efficient algorithms for solving AGV collision problems when using shared resources (tracks, loading, and unloading buffers).

Efficient route planning is crucial for optimizing AGV operations at container terminals. AGV routing is essential for optimizing operational efficiency and reducing transportation costs in automated container terminals (ACTs). However, uncertainties in the physical environment of ACTs pose challenges for determining these optimal plans. As decarbonization efforts progress in ACTs, Battery-Automated Guided Vehicles (B-AGVs) are increasingly being implemented. A critical concern is developing adaptable charging strategies that minimize operational downtime due to charging needs. Intelligent Transportation Systems (ITS) are vital in advancing fully automated vehicles. While considerable strides have been made in advanced driver assistance systems and automation technology, challenges such as enhancing traffic information, refining planning and control systems, and strengthening decision-making capabilities persist. These challenges have spurred significant industrial investment and research interest due to the substantial potential benefits of ITS in the automated driving sector.

The increasing integration of AGVs within container terminals and the coordinated scheduling of unmanned container trucks (UCTs) are crucial for the sustainable growth of port operations. The paper [10] explores the impact of AGVs in ACTs and the benefits of jointly scheduling UCTs to improve port efficiency. Comparative experiments were conducted to assess the feasibility of integrated scheduling approaches. Experimental results confirm the feasibility of addressing the joint scheduling challenge of AGVs and UCTs in ACTs. Compared to independent scheduling approaches, this integrated method shows superior performance in optimizing total operational costs, offering valuable insights for enhancing port scheduling practices.

The paper [11] introduces an anisotropic Q-learning approach that enables AGVs to find the shortest-time routes in a cross-lane guide-path network based on real-time vehicle conditions, including current and target positions, heading direction, and the density of vehicles in neighboring locations across four directions. Also, the paper explores AGV waiting times, proposing a method to estimate waiting time to refine the action-selection policy within the Q-learning framework. An enhanced anisotropic Q-learning routing algorithm is developed, incorporating this waiting-time estimation for improved decision-making. Simulation-based analysis of the algorithm’s parameters and performance reveals that the enhanced method provides stable, adaptive solutions for AGV routing, achieving a

9.5% increase in optimization efficiency.

Introduces a digital twin-driven AGV scheduling and routing framework designed to address such uncertainties are given in the paper [12]. By leveraging digital twins, uncertain factors can be detected and managed through interactions between physical and virtual environments. An improved artificial fish swarm algorithm combined with Dijkstra (IAFSA-Dijkstra) is proposed to find the optimal AGV scheduling and routing solution, which is verified within the virtual space before being applied in the real world to guide AGV operations. For resolving predicted conflicts, a conflict resolution method based on the Yen algorithm is employed to refine the scheduling plan. Case studies demonstrate that this approach effectively enhances efficiency and lowers costs in AGV scheduling and routing for ACTs.

The authors in paper [13] are investigates an optimal workstation assignment method for tandem automated guided vehicle systems (TAGVS) featuring multiple AGVs operating in high-load transport zones. The objective is to allocate workstations to specific zones in a way that minimizes both inter-zone and intra-zone flow while preventing collisions among AGVs. In traditional TAGVS setups, each zone is limited to one AGV, which can lead to delays or downtime in case of vehicle overloads or malfunctions. To address this issue, the paper introduces a new approach that allows for multiple AGVs in high-demand zones. This method optimizes workstation assignments based on building layout and transport flow requirements within the manufacturing process, reducing flow within and between zones and avoiding AGV collisions in overloaded areas. The effectiveness of the proposed approach is demonstrated in a real-world application at a Textbook Print Shop, where it outperforms traditional TAGVS configurations.

The paper [14] introduces a simulation-based optimization approach for AGV charging in U-shaped ACTs, utilizing an enhanced Proximal Policy Optimization (PPO) algorithm. Experimental results show that the enhanced PPO approach achieves faster convergence than both standard PPO and Deep Q-Network (DQN) methods. Additionally, when comparing the adaptive charging strategy derived from this method to a fixed charging threshold across six scenarios with varying charging rates, the proposed strategy demonstrated superior adaptability in two-thirds of the scenarios, adjusting more effectively to varying terminal conditions.

The paper [15] introduces a methodology using state space search to integrate planning knowledge, aiming to equip planning systems with the information needed for tasks involving lateral and longitudinal control, path following, trajectory generation, arbitration, and behavior execution by positioning the vehicle according to a high-level road plan. The study examines state-of-the-art methods for efficiently identifying the K nearest neighbors in high-dimensional road plans based on traffic data from high-definition maps. Experimental results show promising real-time performance for fast KNN algorithms, resulting in a robust tree index-based methodology that integrates path planning, trajectory tracking, trajectory generation, knowledge aggregation, and precise vehicle control for decision-making in self-driving vehicles.

3 AGV Route Optimization

The transport network of the container warehouse at the port terminal is usually organised as a rectangular (square) grid, that is, a network of movement paths [1]. The path taken by AGV from one buffer with coordinates [0,0] to another, adjacent buffer with coordinates $[x_1, x_2]$ can be represented as [8]:

$$d([0,0], [x_1, x_2]) = |x_1| + |x_2| \tag{1}$$

Equidistant contours, locations of all buffers at a distance d from the centre [0,0], where $d = |x_1| + |x_2|$, are squares oriented at an angle of 45° to the grid axes. This family of equidistant contours corresponds to the socalled Huygens construction, where a square at a distance d_2 from the centre can be obtained by translating the one at a distance d_1 and so on (Figure 2).

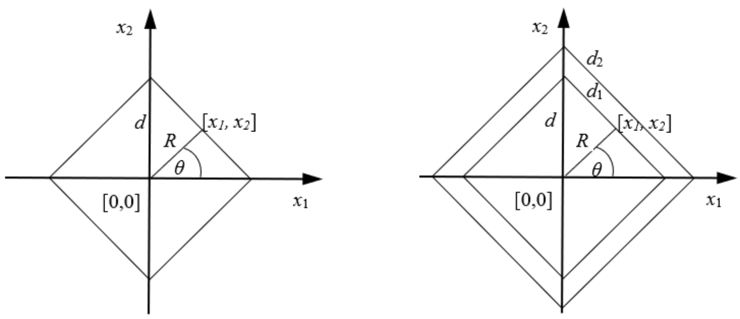

Figure 2. Equidistant contour and Huygens construction for a square network of movement paths

The distance between two buffers can also be expressed in polar coordinates as follows [8]:

$$d\left([0,0],[x_1,x_2]\right) = R\left|\cos\theta\right| + \left|\sin\theta\right| \tag{2}$$

where, $x_1 = R\cos\theta$ and $x_2 = R\sin\theta$. If the AGV moves along any coordinate direction at an angle $\theta = j\pi/2$ for j=0,1,2,3,..., the distance d is equal to the Euclidean distance R. If we assume that the AGV moves from one buffer to another at an angle of 45° in relation to the track network, that is, at an angle $\theta = j\pi/4$ for j=1,3,5,7,..., then the distance d is equal to $2R$ or 1.414 Euclidean distance. If, from a theoretical point of view, the trajectories of the AGV were uniformly distributed in all directions in relation to the center of the observed coordinate system [0,0], that is at an angle $0 \leq \theta \leq 2\pi$, then their average length would be [8]:

$$d = 2/\pi \int_0^{\pi} \left(\cos\theta + \sin\theta\right) d\theta = 4/\pi R \approx 1.27R \tag{3}$$

This means that the path travelled by an AGV between two adjacent buffers in a square network of paths can be approximated with satisfactory accuracy if twenty-seven percent is added to the Euclidean distance between the two buffers, measured directly on the terminal layout. In the numerical examples that follow, this kind of approximation was used, where it was shown that the deviation between real and approximated distances is negligible, while the procedure for determining the distance between neighbouring buffers is significantly simplified [8].

In accordance with the previously described procedure of approximating the length of the path that the AGV travels between two adjacent buffers and the nearest neighbour and Tabu search algorithms in solving the TSP, the optimal circular path of the AGV at the container terminal is determined here. In a circle, cycle, or loop of the shortest length, the AGV should go around a certain number of buffers arranged arbitrarily along the container stacks or at their corners and return to the starting point (a certain place in the port terminal, usually the foot of the container crane's overflow branch on the side of the operational shore).

3.1 The Nearest Neighbour Algorithm for the Construction of a Travelling Salesman's Route

The authors of the mentioned algorithm are Rosenkrantz et al. [16]. It belongs to the class of greedy algorithms and is used for route construction. The algorithm consists of the following steps [17]:

Step 1: Arbitrarily select the start node of the route.

Step 2: Find the node closest to the last one included in the route. Include this nearest node in the route.

Step 3: Repeat step 2 until all nodes are included in the route. Join the first and last node of the route.

3.2 Tabu Search

The Tabu search technique was proposed by Glover [18]. Tabu search is a metaheuristic algorithm that iteratively searches for solutions as in local search but has adaptive memory. The algorithm consists of the following steps [17, 18]:

Step 1: Generating the initial solution (route length) of the travelling salesman problem using one of the heuristic algorithms.

Step 2: Generate a solution by simply permuting the order of visiting two nodes in the travelling salesman's route. The number of permutations of the order of visiting two nodes is limited by the number of nodes.

Step 3: Calculation of the newly obtained solutions (route length).

Step 4: Calculating the savings when switching the order of visiting nodes.

Step 5: Sort the best permutations of the order of visiting nodes (arbitrary number of permutations). In the case of determining the minimum length of the route, if the savings are negative, the substitutions cannot lead to a reduction in the length of the route (end of the algorithm); otherwise, continue with the algorithm.

Step 6: Prohibit (T) certain permutations of the order of visiting nodes (flexible memory). The number of iterations during which the substitution is taboo is determined arbitrarily (it decreases with increasing number of iterations).

The tabu restriction does not apply in situations where a certain prohibition to change positions leads to the best currently discovered solution. Glover and Laguna [19] proposed the following way of showing the taboo status of certain moves. Each element of the matrix represents the number of remaining iterations during which a particular substitution is not allowed.

4 Implementation of the Approach and Discussion of the Results

When solving the traveling salesman's problem, it is necessary to determine the shortest closed, circular route of AGV vehicles in a container block with dimensions of 520 × 800 feet [8]. The buffer positions (B_1, B_2, B_3, B_4)

are presented in Figure 3. For ease of notation, the following equalities $B_1 = 1, B_2 = 2, B_3 = 3, B_4 = 4$ have been introduced in the following text.

Distances between the buffers can be specified as real (real) and as approximate. Real distances can be determined by summing the horizontal and vertical segments of paths that the AGV traverses through the network. In contrast, approximate distances can be determined by adding 27% to the Euclidean distance between adjacent buffers on the circular path of the AGV, measured directly on the container block layout. The calculation procedure and final values of real and approximate distances between adjacent buffers in the block are given in Table 1 in hundreds of feet [8].

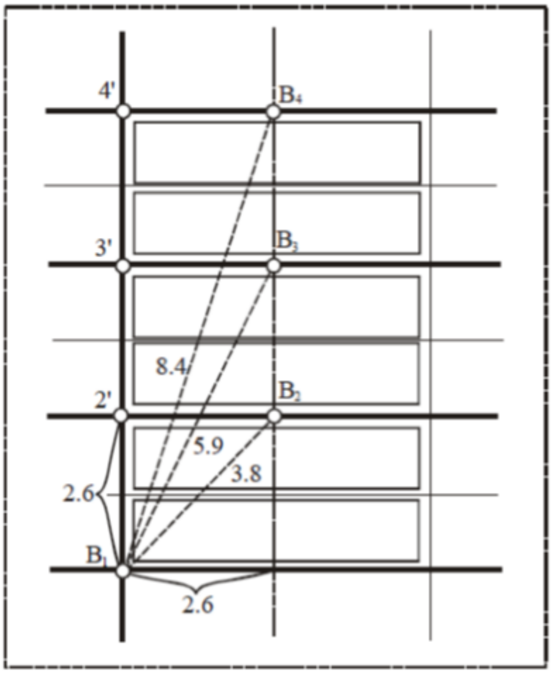

Figure 3. The buffer positions

Table 1. Real and approximative distances between adjacent buffers in a container block

Direction	Real Distances [$\times 10^2$ feet]	Approximative Distances [$\times 10^2$ feet]
1-2	2 × 2.6 = 5.200	3.8 × 1.27 = 4.826
1-3	3 × 2.6 = 7.800	5.9 × 1.27 = 7.493
1-4	4 × 2.6 = 10.400	8.4 × 1.27 = 10.668
2-3	3 × 2.6 = 7.800	3.8 × 1.27 + 2.6 = 7.426
2-4	4 × 2.6 = 10.400	5.9 × 1.27 + 2.6 = 10.033
3-4	3 × 2.6 = 7.800	3.8 × 1.27 + 2.6 = 7.426

Table 2. Real and approximative distances between adjacent buffers in a container block

	Real Distances [$\times 10^2$ feet]					Approximative Distances [$\times 10^2$ feet]			
	1	2	3	4		1	2	3	4
1	∞	5.200	7.800	10.400	1	∞	4.826	7.493	10.668
2	5.200	∞	7.800	10.400	2	4.826	∞	7.426	10.093
3	7.800	7.800	∞	7,800	3	7.493	7.426	∞	7.426
4	10.400	10.400	7.800	∞	4	10.668	10.093	7.426	∞

The distances between all pairs of buffers in the observed block of containers are given in Table 2. The distances are expressed in hundreds offeet, where it is not possible to move the AGV from one buffer to itself, and therefore it is: $d_{11} = \infty$, $d_{22} = \infty$, $d_{33} = \infty$, and $d_{44} = \infty$ [8].

Applying the nearest neighbor and Tabu search algorithms determined the shortest path of an AGV vehicle at a container terminal within a certain container block. The problem is simplified in the sense that only one, a specific

AGV, is assigned to a specific block so that conflicts and stoppages are avoided to the greatest extent possible. This is only possible at container terminals where a small number of AGVs are employed and where good software has been developed in the function of terminal traffic manager. In conditions of increased traffic, AGVs can follow each other on a pre-defined optimal circular path, but at an appropriate mutual distance to avoid possible head-on rear-end collisions of AGV vehicles.

Nearest neighbor" algorithm The route must start at node 1. Based on the distance between nodes given in Table 2, the node closest to node 1 is node 2. Let's include node 2 in the route, which for now reads (1, 2). The node closest to node 2 is node 3; including node 3 in the route that reads (1, 2, 3). It can be seen that node 4 is closest to node 3, so it is also included in the route. After connecting node 4 and node 1, the final route is (1, 2, 3, 4, 1), i.e. (1, 2', 2, 2', 3', 3, 3', 4', 4, 4', 1). The length of this route is 3034.6 feet.

Tabu search: Starting from step number 1 of the described algorithm, where the layout and length of the route obtained by the nearest neighbor algorithm (1, 2, 3, 4, 1) can be used as an initial solution to the traveling salesman's problem. It should be emphasized that the AGV starts its task from node 1. After completing the task, the AGV returns to node 1. The current solution and tabu statuses of individual replacements of the order of visiting two nodes are respectively equal:

1	2	3	4	1

Tabu status of possible permutations:

	2	3
1	0	0
	2	0

Permuting the sequence can yield a total of 3 different solutions. Table 3 shows the permutations that are performed, the routes that are generated based on the performed permutations, the length of the routes, and the savings in length that are achieved about the length of the existing routes.

The greatest savings would be achieved in the case of switching the order of visiting nodes 3 and 4. Also, the decision that the order of visiting nodes 3 and 4 must not be replaced in the next iteration.

Table 3. Replacements of the order of visiting nodes, routes, route lengths, and savings

Replacement	The Route that is Obtained	Route Lengths [feet]	Savings [feet]
(2, 3)	(1, 3, 2, 4, 1)	3568	3034.6 – 3568 = – 533.4
(2, 4)	(1, 4, 3, 2, 1)	3034.6	3034.6 – 3034.6 = 0
(3, 4)	(1, 2, 4, 3, 1)	2983.8	3034.6 – 2983.8 = 50.5

After the replacement, a new solution is obtained (the length of the AGV route is 2983.8 feet), and the tabu status of possible order replacements is determined. The new solution reads:

1	2	4	3	1

Tabu status of possible permutations:

	2	3
1	0	0
	2	1

Table 4. Replacements of the order of visiting nodes, routes, route lengths, and savings after the second iteration

Replacement	The Route that is Obtained	Route Lengths [feet]	Savings [feet]	Taboo Status
(2, 3)	(1, 3, 4, 2, 1)	2983.8	2983.8 – 2983.8 = 0	
(2, 4)	(1, 4, 2, 3, 1)	3568	2983.8 – 3568 = – 584.2	
(3, 4)	(1, 2, 3, 4, 1)	3034.6	2983.8 – 3034.6 = – 50.5	T

Table 4 shows the permutations that can be performed, the routes that are generated based on the performed permutations, the length of the routes, and the savings in length that are realised about the length of the newly obtained route.

It can be observed that replacing the order of visiting nodes 2 and 3 leads to the same solution as the initial solution (1, 3, 4, 2, 1) except that the direction of the order of visiting the nodes is reversed. The optimal solution is 1, 2, 4, 3, 1 or 1, 3, 4, 2, 1.

Algorithm code developed in MATLAB enables quick and efficient solving of "nearest neighbor" and Tabu search algorithms and various analyzes regardless of the number of buffers.

5 Conclusions

In the operation of a large number of means with a cyclic effect, the influence of the human factor on the efficiency and accuracy of work during the implementation of certain parts of the cycle is significant. This affects both the time (parts and/or the whole) of the cycle, as well as the reliability of work, the availability of funds, the level of work errors, etc. For this reason, solutions were sought that could reduce or eliminate this impact. The routing algorithms developed so far treat a small number of AGVs in the observed system. With the increase in the number of AGVs, the probability of collisions and work stoppages increases. These problems are current since studies have shown that the exploitation of a fleet of AGVs at container terminals is profitable only with the technology that is currently available. This area, which is intensively worked on, is related to the following areas of research: automatic vehicle guidance, intelligent vehicles, and intelligent navigation mechanisms, while a small number of works are devoted to AGV routing problems.

The paper shows the application of Tabu search in the optimization of routes of AGVs at the container terminal. The initial solution was obtained by the nearest neighbor algorithm. The shortest closed, circular route of an AGV vehicle in a container block has been determined. Distances between buffers can be calculated in two ways: as real or approximate. The actual distances are obtained by summing the horizontal and vertical segments of paths that the AGV traverses within the network. On the other hand, approximate distances are determined by adding 27% to the Euclidean distance between adjacent buffers on the circular route of the AGV, where the distance is measured directly on the layout of the container block. Replacing the order of visiting nodes 2 and 3 leads to the same solution as the initial solution (1, 3, 4, 2, 1), only the direction of the order of visiting nodes is reversed, i.e. 1, 2, 4, 3, 1 or 1, 3, 4, 2, 1.

In addition to the traditionally applied metaheuristic methods for solving routing problems, such as simulated annealing (SA), genetic algorithms (GA), and Tabu search (TS), in recent times, interest in the development and application of metaheuristics based on the principles of collective intelligence (SI swarm intelligence) has increased. These metaheuristics are based on artificial colonies whose functioning is inspired by the behavior of colonies of social insects in nature (ants, bees, termites...). Optimization with collective intelligence, as a direction of future research, offers the possibility of solving the problem of routing a large number of AGVs.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgements

This research was financially supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia (Grant No.: 451-451-03-65/2024-03).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Le-Anh and M. B. M. De Koster, "A review of design and control of automated guided vehicle systems," *Eur. J. Oper. Res.*, vol. 171, no. 1, pp. 1–23, 2006. https://doi.org/10.1016/j.ejor.2005.01.036

[2] B. M. Beamon, "Performance, reliability, and performability of material handling systems," *Int. J. Prod. Res.*, vol. 36, no. 2, pp. 377–393, 1998. https://doi.org/10.1080/002075498193796

[3] V. Jovanovic and D. Janošević, "Analysis and regulation of mechatronic systems in advanced mobile machines," *J. Eng. Manag. Syst. Eng.*, vol. 2, no. 3, pp. 140–149, 2023. https://doi.org/10.56578/jemse020301

[4] S. Zhao, P. Li, and Q. Li, "The vehicle routing problem considering customers' multiple preferences in last-mile delivery," *Tech. Gaz.*, vol. 31, no. 3, pp. 734–743, 2024. https://doi.org/10.17559/TV-20230717000807

[5] D. Marković, A. Stanković, D. Marinković, and D. Pamučar, "Metaheuristic algorithms for the optimization of integrated production scheduling and vehicle routing problems in supply chains," *Tech. Gaz.*, vol. 31, no. 3, pp. 800–807, 2024. https://doi.org/10.17559/TV-20240207001318

[6] D. Saha, R. Nidhi, B. K. S. Roy, and M. Mukherjee, "Performance comparison of PSO, HGSO, and DE optimization techniques for computation of directional overcurrent relay coordination in power systems," *Eng. Rev.*, vol. 44, no. 2, pp. 69–86, 2024. https://doi.org/10.30765/er.2429

[7] O. Mbah and Q. Zeeshan, "Optimizing path planning for smart vehicles: A comprehensive review of metaheuristic algorithms," *J. Eng. Manag. Syst. Eng.*, vol. 2, no. 4, pp. 231–271, 2023. https://doi.org/10.56578/jemse02040

[8] S. Bauk, "Intelligent information systems to the route optimization in maritime and port transportation," Ph.D. dissertation, Traffic Faculty, University of Belgrade, 2005.

[9] D. Marković, S. Marković, D. Marinković, and D. Pamučar, "Dynamic vehicle routing problem for smart waste collection," *Facta Univ. Ser. Mech. Eng.*, 2024. https://doi.org/10.22190/FUME240322021M

[10] L. Chu, Z. Gao, S. Dang, J. Zhang, and Q. Yu, "Optimization of joint scheduling for automated guided vehicles and unmanned container trucks at automated container terminals considering conflicts," *J. Mar. Sci. Eng.*, vol. 12, no. 7, p. 1190, 2024. https://doi.org/10.3390/jmse12071190

[11] P. Zhou, L. Lin, and K. H. Kim, "Anisotropic Q-learning and waiting estimation based real-time routing for automated guided vehicles at container terminals," *J. Heuristics*, vol. 29, pp. 207–228, 2023. https://doi.org/10.1007/s10732-020-09463-9

[12] P. Lou, Y. Zhong, J. Hu, C. Fan, and X. Chen, "Digital-twin-driven AGV scheduling and routing in automated container terminals," *Math.*, vol. 11, no. 12, p. 2678, 2023. https://doi.org/10.3390/math11122678

[13] R. G. Cha, C. Ji, and G. H. Nam, "Research on workstation-to-zone assignment for tandem automated guided vehicle system with multiple AGVs in overload zone of transport," *Int. J. Adv. Manuf. Technol.*, vol. 135, pp. 4653–4668, 2024. https://doi.org/10.1007/s00170-024-14495-7

[14] Y. Yang, J. Liang, and J. Feng, "Simulation and optimization of automated guided vehicle charging strategy for U-shaped automated container terminal based on improved proximal policy optimization," *Syst.*, vol. 12, no. 11, p. 472, 2024. https://doi.org/10.3390/systems12110472

[15] D. Yagüe-Cuevas, M. Paz-Sesmero, P. Marín-Plaza, and A. Sanchis, "Organizing planning knowledge for automated vehicles and intelligent transportation systems," *IET Intell. Transp. Syst.*, 2024. https://doi.org/10.1049/itr2.12583

[16] D. J. Rosenkrantz, R. E. Stearns, and P. M. L. II, "An analysis of several heuristics for the traveling salesman problem," *SIAM J. Comput.*, vol. 6, no. 3, pp. 563–581, 1977. https://doi.org/10.1137/0206041

[17] D. Teodorović, *Transportation Networks: A Quantitative Treatment*. Routledge, 2021.

[18] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986. https://doi.org/10.1016/0305-0548(86)90048-1

[19] F. Glover and M. Laguna, "Tabu search," in *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publications, 1992, pp. 70–150.