



Article

A Dual-Layer Weight-Leader-Vicsek Model for Multi-AGV Path Planning in Warehouse

Shiwei Lin , Ang Liu and Jianguo Wang *

Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia; shiwei.lin-1@student.uts.edu.au (S.L.); ang.liu@student.uts.edu.au (A.L.)

* Correspondence: jianguo.wang@uts.edu.au

Abstract: Multiple automatic guided vehicles are widely involved in industrial intelligence. Path planning is crucial for their successful application. However, achieving robust and efficient path planning of multiple automatic guided vehicles for real-time implementation is challenging. In this paper, we propose a two-layer strategy for multi-vehicle path planning. The approach aims to provide fast computation and operation efficiency for implementation. The start–destination matrix groups all the vehicles, generating a dynamic virtual leader for each group. In the first layer, the hybrid A* algorithm is employed for the path planning of the virtual leaders. The second layer is named leader–follower; the proposed Weight-Leader-Vicsek model is applied to navigate the vehicles following their virtual leaders. The proposed method can reduce computational load and achieve real-time navigation by quickly updating the grouped vehicles' status. Collision and deadlock avoidance is also conducted in this model. Vehicles in different groups are treated as dynamic obstacles. We validated the method by conducted simulations through MATLAB to verify its path-planning functionality and experimented with a localization sensor.

Keywords: path planning; Vicsek model; leader–follower; hybrid A*; multiple automatic guided vehicles



Citation: Lin, S.; Liu, A.; Wang, J. A Dual-Layer Weight-Leader-Vicsek Model for Multi-AGV Path Planning in Warehouse. *Biomimetics* **2023**, *8*, 549. <https://doi.org/10.3390/biomimetics8070549>

Academic Editors: Ameer Hamza Khan, Shuai Li and Danish Hussain

Received: 20 October 2023

Revised: 11 November 2023

Accepted: 14 November 2023

Published: 15 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of robotic technologies, mobile robots are implemented in commerce and industry. Automated Guided Vehicles (AGVs) enhance transportation efficiency with less cost [1,2]. They are utilized in the industry as a part of industrial intelligence, intelligent logistics, and intelligent factories [2,3]. AGVs comprise the industry's unit load vehicles, towing vehicles, forklifts, and pallet trucks [4]. AGVs are applied to diminish labour costs and improve safety for the high demands in a production environment [5,6]. Servo handling systems, warehousing systems, logistics, and storage industries employ AGVs in various areas [6,7], including transportation, transshipment, distribution, and material handling in manufacturing [8].

AGVs can transfer products at high speeds in a chaotic situation. They are implemented in production lines with flexible development in modern manufacturing, incorporating automated intelligent control systems [4,6,9]. With sensors' help, detecting obstacles and automatically eliminating problems ensure the intelligence and adaptability of AGVs [6]. AGV navigation in the industrial environment usually implements the fixed line or the coupled approach, which adapts the single-robot navigation methods. It lacks the flexibility and the possibility of real-time implementation. It has highly demanded that AGV have a quick and flexible route setting and adapt to the dynamic environment. The motivation of this study is to provide fast computed path planning with flexibility and scalability for multi-AGV systems, addressing most situations.

This paper introduces a new algorithm for this purpose. Its main contribution is as follows:

- A novel path-planning approach for multi-AGV systems based on the improved Vicsek model with a leader–follower structure. The bio-inspired approaches are widely used in multi-robot path planning. The Vicsek model is adapted for this case because the AGVs aim to move as a swarm operation and provide quick computation for the entire system.
- Offering the fast path setting for multiple AGVs in one calculation step, which differs from the other path-planning algorithms that repeat the algorithms for every robot.
- For real-time implementation, it provides faster computation and adaption to the environment.

Instead of navigating all the AGVs directly, a set of virtual leaders is introduced in the AGV swarms to navigate all the AGVs, combining search and intelligent algorithm advantages. Biological patterns and the leader's principle are the main concepts in this model, and it integrates the coupled and decoupled approach. The hybrid centralized decentralized is proposed for determining the leader's path while providing flexibility for AGVs. Each AGV follower collects data from its neighbours and is led by the virtual leaders without the restriction of the current group. One of the significant advantages of the proposed multi-AGV path-planning system is that it requires less computational load for real-time implementation.

The Leader-Vicsek model was published in [10], and this paper is an extension. The published paper introduces the concept of the novel Weight-Leader-Vicsek model, but it uses simple, straightforward path planning for the leader and traditional Vicsek update equations for followers. Nevertheless, this paper uses a dual layer to improve the model as an enhanced Weight-Leader-Vicsek model (WLVM). It generates the dynamic virtual leader by the hybrid A* algorithm and uses a start–destination matrix to determine the swarms' integration, and collision avoidance is achieved by priority. Also, the followers' position and angle updates are improved by adding the weight for the average angle and considering the current leader's angle. The simulations are conducted in a new scenario, and WLVM is compared with other algorithms.

This paper offers a multi-AGV path-planning approach for optimizing automatic transportation in commercial or industrial warehouses. The paper is organized as follows. Section 2 reviews path-planning algorithms and the multi-AGV navigation approaches. Weight-Leader-Vicsek model is proposed in Section 3 for multi-AGV path planning and navigation. Experiment results are demonstrated in Section 4 to validate the approach, and the conclusion is in Section 5.

2. Related Work

The Automatic Guided Vehicle (AGV) plays a crucial role in the intelligent transportation system. AGVs are the primary automated equipment that carry materials and process unmanned distribution and sorting in an unmanned storage environment [11]. Path planning has been the most crucial consideration of mobile robot navigation, which plans the path from the start to the target for mobile robots [3].

Obstacle avoidance functions must be developed to operate AGVs when considering dynamical limitations and dynamical safety [1,12]. Meta-heuristic-based methods [13], graph search-based methods [2,14], mathematical optimization-based methods [1], and potential field and navigation-based methods [15] are the four main categories for navigation algorithms.

Additionally, adapting the classic graph search algorithm is implemented widely for AGV path planning, such as the A* and Dijkstra [16]. The limitation of the improved A* algorithm is that the path cannot be guaranteed optimal and is conducted under simulation. Different algorithms can be combined; for example, the Dijkstra algorithm is for initial static path planning, and the virtual potential function algorithm is for dynamic path planning [17]. The drawback of this study is that static or dynamic obstacles have yet to be tested in real-life testing. Sampling-based methods are also significant for single AGV path planning, such as the Voronoi graph and rapidly exploring random trees (RRT)

methods [18]. Future research on the improved RRT must combine obstacle avoidance and data-driven path planning.

The mathematical optimization-based approach also consists of open-loop and closed-loop strategies [1]. A nonlinear model predictive control (MPC) algorithm has been proposed for large-size AGVs with onboard LIDAR sensors and a 14 DoF vehicle dynamics model [1]. Artificial potential field (APF) methods [19] and the probabilistic road map [20] are also proposed for motion planning. The limitations of these studies include vehicle control, moving obstacles, and real-time implementation.

The AGV path-planning algorithms introduced above are mainly focused on individual AGVs. For the venues with multiple AGVs operated simultaneously, the dynamic environment due to other AGVs and people needs to be handled safely and efficiently.

Multi-AGV systems have become more popular because of their powerful task-solving complexity, continuous operation, reduced maintenance and operational costs, and broad convergence, flexibility, and versatility [21,22]. Multi-AGV optimization scheduling can improve the logistics transportation system structure and system operation efficiency and reduce transportation costs [23,24].

Finding the best path quickly and avoiding collision is worth studying in an AGV operation [25]. AGVs obtain the paths from the multi-AGV scheduling system, and they sense the surroundings independently and communicate with others by sending poses [26]. AGVs are usually guided by optical, electromagnetic, and laser navigation technologies or combinations of them, following the arranged path and avoiding collisions [27]. The research on multi-AGV routing can be classified as semi-dynamic, fully-dynamic, and static routing [28].

Vehicle dispatching, positioning, vehicle routing, scheduling, and collision and deadlock avoidance are considered for designing vehicle transportation systems [29]. Current multi-AGV systems commonly implement the centralized control architecture to perform tasks, such as motion coordination, mission allocation, and path planning [30]. The centralized control system is referred to as the warehouse management system (WMS) [31]. However, each AGV plans its path for the distributed approaches and resolves deadlocks or collisions by communicating with its neighbours. Vehicle autonomy and distributed computation are characteristics of decentralized methods [21].

Moreover, a decentralized approach for determining the shortest paths and motion coordination based on nonholonomic vehicle constraints is presented in [21]. It needs to consider the other vehicles' motion plans or locations in future work. A regional control model is introduced for distributed control for the multi-AGV system in [32] to minimize the complexity of scheduling issues. The limitations are related to real manufacturing companies and distributed control mechanisms.

An ant-agent optimized by a repulsive potential field is developed to combine centralized and decentralized control and avoid path conflicts with stability and efficiency [28]. The approach's limitations are that it cannot determine whether the method is optimal or considers the working condition of AGV. A multi-AGV path-planning method improves ant colony algorithms according to prioritized planning, considering battery management in [33] as a decentralized algorithm, whereas it only uses simulation to verify the results.

The metaheuristic algorithms are widely used for optimization problems for AGV systems, such as task allocation [24] and path planning [34]. Multi-agent path-planning algorithms can be divided into rule-based, search-based, and learning-based [35]. Rule-based algorithms use mature solutions for path planning by transferring the problem to other problems; search-based algorithms implement heuristic search algorithms and are classified as decoupled search, and coupled search algorithms [36], and learning-based algorithms obtain optimal solutions from suboptimal solutions [35].

A multi-AGV probabilistic time-constrained path-planning algorithm is also based on the A* heuristic algorithm with dynamic stochastic network theory in [22]. An improved A* path-planning algorithm is introduced for a grid-shaped network, ensuring the locating and execution of motion commands [25]. The unidirectional directed graph method combined

with the A* algorithm for AGV path planning in a multi-AGV scheduling system is presented [37]. The limitations of these improved A* algorithms consist of the iterative update which is not adaptive of path arc time consumption [22], lacking experiments [22,25,37], dynamic scheduling algorithms, or path optimality [37].

Heuristic information and elastic time window are considered in the improved ant colony algorithm [38], and the conflict resolution strategies are based on the priority of AGV task scheduling. The drawbacks of the algorithm include transportation equipment, actual operation, and path conflict. A hybrid genetic algorithm-particle swarm optimization is proposed for multi-AGV path planning with a fuzzy logic controller in [36], combining scheduling and path planning. The shortcoming focuses on dynamic real-time scheduling in a large-scale system. A genetic algorithm is improved to consider the highest charging utilization rate and the shortest path to plan the optimal path for multi-AGV, whereas it lacks real operation [11].

Deadlock avoidance is the primary consideration during multi-AGV path planning. Nodes describe the physical locations, whereas the grids are independent spaces in the environment [23]. The node-based coordination strategies strictly avoid the AGVs occupying a common node. The authors of [23] propose deadlock strategies by combining nodes and grids, and future work will be integrated with a distributed framework for scalability. A structural online control policy is proposed for multi-AGV deadlock resolution based on analyzing the system as discrete events [39]. The topological graph and roadmap work for the AGVs' subsequent coordination by local negotiation and shared resources as a holistic approach in industrial warehouses [40]. The shortcomings focus on the task allocation mechanism and real factors of the graph weights.

However, the multi-agent algorithms plan the path independently and lack consideration of moving obstacles and real-time implementation. Most evolution-based and swarm-based algorithms are bio-inspired, and the biological pattern is considered when developing the new algorithm. This paper proposes a Weight-Leader-Vicsek model algorithm, which incorporates the advantages of decentralized and centralized approaches. Each AGV collects data from relevant AGVs, determines its path, and achieves collision avoidance. At the same time, a central decision-maker assigns the multi-AGV groups and virtual leaders for the defined groups. AGV control variables can be gained with faster computational speed and less complexity. This model provides path-planning functionality simultaneously for grouped AGVs and treats each group as different swarms.

3. WLVM with Virtual Leaders and Weight

3.1. Preliminary Knowledge: Vicsek Model

The Vicsek model is introduced for particles with biologically motivated interaction with self-ordered motion and is proposed in [41]. The Vicsek model can be expressed by (1) and (2), and the algorithm is shown in Algorithm 1. Biological subjects spin in the same direction and move as their neighbourhood for interaction in an $L * L$ square region [41,42]. For a multi-agent system, the Vicsek model has been improved by taking a fixed number of neighbours and a percentage of neighbours into account as the remote neighbours' strategy [43]. The hierarchical Weighting Vicsek model is proposed for flocking navigation, and it assigns different layers for the drones with weights to enhance the convergence speed, analyzing the involved parameters [44].

$$x_i(t + 1) = x_i(t) + v_i(t)\Delta t \quad (1)$$

$$\theta_i(t + 1) = \langle \theta(t) \rangle_r + \Delta \theta \quad (2)$$

where $x_i(t + 1)$ is the location of the particle i , the velocity v_i is gained through an absolute velocity, and the angle is represented by $\theta_i(t + 1)$. $\langle \theta(t) \rangle_r$ represents the average angle of the neighbouring particles within a circle with radius r . $\Delta \theta$ denotes a random number from $[-\frac{\eta}{2}, \frac{\eta}{2}]$ as noise.

Algorithm 1: Vicsek model

```

1 Initialize parameters
2 for time ← 1 to timemax do
3   Calculate averageTheta
4    $x \leftarrow x + vel * \cos(theta) * dt$ 
5    $y \leftarrow y + vel * \sin(theta) * dt$ 
6    $theta \leftarrow averageTheta + noise$ 
7 end
    
```

3.2. Problem Statement

The basic components of AGV path planning include the start, the target, and the environment. Path planning aims to generate the path from the start to the target without collisions. Figure 1 indicates an example of AGV path planning. As the occupancy map, the map uses the binary number to represent the locations in a 2D space. It can be transformed into a grid map or nodes for performing the heuristic or other path-planning algorithms. The obstacles or the walls are annotated in black.



Figure 1. Path planning in the map.

The AGV is supposed to move from the start to the target, and green points demonstrate the path points. The obstacles or walls are treated as static obstacles, and the generated path should not be overlapped with the static obstacles. The path points are represented by $r_1^k, r_2^k, \dots, r_n^k$. r is the position of the current iteration k for n th AGV, and r_n^k is (x_n^k, y_n^k) .

Collision and deadlock avoidance are necessary for the multi-AGV system. Each AGV is assumed to communicate with other AGVs treated as dynamic obstacles. The deadlock of AGVs should be avoided for great system performance, and the adjusting progress is based on priority. The adjusted position of AGV n is represented by r_{n-new} .

3.3. Algorithm Description

3.3.1. Overview

WLVM is proposed for a multi-AGV path-planning algorithm to improve the Vicsek model because the Vicsek model cannot achieve practical path planning in the industry. WLVM assigns the virtual leaders to collaborate with the grouped AGVs and guide the followers to reach their destinations, considering collision avoidance.

Figure 2 describes the process of WLVM. According to a real industrial environment, the storage map is generated. Points represent the map; 0 is open space for movement, whereas 1 represents the wall or obstacles. The number of AGVs, velocities, angles, and locations are set for model initialization. The AGVs are divided into swarms based on their locations and destinations with assigned virtual leaders. The positions and angles of virtual leaders are computed by the hybrid A* algorithm. The follower-AGVs use the status of the leader in the current group to obtain the average angle within the defined path for WLVM. The AGVs implement a segment delay function to be separated by a certain distance for optimal arrangements. The AGVs avoid vehicle congestion and deadlock. Ref. [10] presents the design of the model for the AGV system.

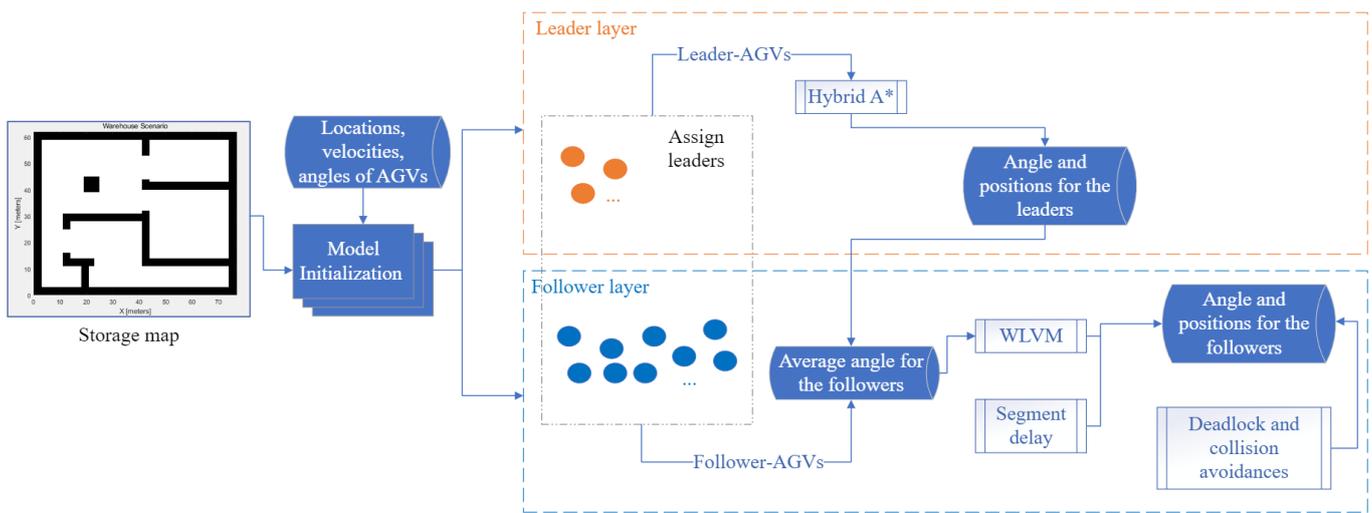


Figure 2. WLVM process.

Table 1 saves the positions and angles of each swarm during the path-planning process. N stands for the number of the AGVs, and *Virtual Leader* represents the leader in the group. WLVM saves the positions and angles for follower-AGVs and virtual leaders in each iteration and is used for further updating.

Table 1. AGV navigation data in the same group.

Iteration	AGV 1	AGV 2	...	AGV n	Virtual Leader
1	$\begin{bmatrix} x_1^1 \\ y_1^1 \\ \theta_1^1 \end{bmatrix}$	$\begin{bmatrix} x_2^1 \\ y_2^1 \\ \theta_2^1 \end{bmatrix}$...	$\begin{bmatrix} x_n^1 \\ y_n^1 \\ \theta_n^1 \end{bmatrix}$	$\begin{bmatrix} x_L^1 \\ y_L^1 \\ \theta_L^1 \end{bmatrix}$
2	$\begin{bmatrix} x_1^2 \\ y_1^2 \\ \theta_1^2 \end{bmatrix}$	$\begin{bmatrix} x_2^2 \\ y_2^2 \\ \theta_2^2 \end{bmatrix}$...	$\begin{bmatrix} x_n^2 \\ y_n^2 \\ \theta_n^2 \end{bmatrix}$	$\begin{bmatrix} x_L^2 \\ y_L^2 \\ \theta_L^2 \end{bmatrix}$
⋮	⋮	⋮	⋮	⋮	⋮
k	$\begin{bmatrix} x_1^k \\ y_1^k \\ \theta_1^k \end{bmatrix}$	$\begin{bmatrix} x_2^k \\ y_2^k \\ \theta_2^k \end{bmatrix}$...	$\begin{bmatrix} x_n^k \\ y_n^k \\ \theta_n^k \end{bmatrix}$	$\begin{bmatrix} x_L^k \\ y_L^k \\ \theta_L^k \end{bmatrix}$

3.3.2. Dynamic Virtual Leader

Dynamic virtual leaders are implemented in WLVM for shorter paths, faster convergence, and more accurate direction for planning the path to arrive at the destination. Each multi-AGV group has one virtual leader. Figure 3 demonstrates the principle of

the dynamic virtual leader. When a new AGV joins the current group, the AGVs of the multi-AGV group will treat it as part of the current group, and the group dynamically generates the virtual leader. The virtual leaders are generated in a static environment. When the AGVs aim for different areas, the start–destination matrix makes the separation, which refers to Section 3.3.3.

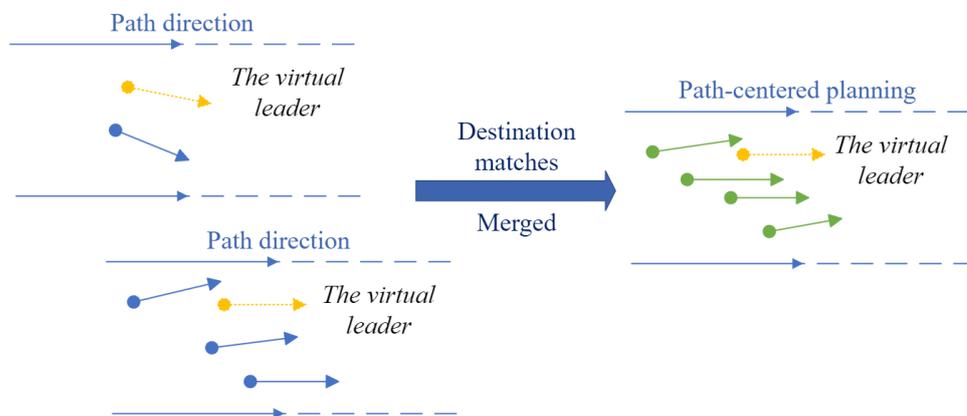


Figure 3. Principle of WLVM.

The positions for each AGV are calculated by (3). The equations for calculating the angles for follower AGVs are as (4).

$$\begin{pmatrix} x_i^{k+1} \\ y_i^{k+1} \end{pmatrix} = \begin{pmatrix} x_i^k \\ y_i^k \end{pmatrix} + v\Delta t \cdot \begin{pmatrix} \cos\theta_i^k \\ \sin\theta_i^k \end{pmatrix} \tag{3}$$

$$\theta_i^{k+1} = \omega_1 \cdot \arctan \frac{\langle \sin(\theta_i^k) \rangle_p}{\langle \cos(\theta_i^k) \rangle_p} + \omega_2 \cdot \theta_i^{k+1} + \eta_i^k \tag{4}$$

where the average direction $\arctan \frac{\langle \sin(\theta_i^k) \rangle_p}{\langle \cos(\theta_i^k) \rangle_p}$ is estimated along the path p , and the travel distance is represented by $v\Delta t$. η_i^k denotes the noise. w_1 is a random number in $(0, 1)$, and the sum of w_1 and w_2 is 1. The swarm only considers the AGVs in the defined path in the same direction. The hybrid A* algorithm is integrated to calculate the position and angles for virtual leaders, and θ_i^{k+1} denotes the angle of the leader in the current iteration. It is adapted for multi-task implementation for virtual leaders.

When an AGV joins a new group, the virtual leader of that group will change if the AGV’s destination aligns with the group’s destination. If not, the group’s structure and leader remain unchanged. The leaders are predefined, while the followers keep gaining statuses from the neighbourhoods and then updating their status. It provides the possibility of real-time implementation.

For virtual leaders, angles and positions are generated by the hybrid A* algorithm, and the pseudo-code is indicated in Algorithm 2. A* is a widely used graph traversal algorithm because of optimal efficiency and completeness [45]. The hybrid A* algorithm is proposed in [46], which guarantees kinematic feasibility and continuous nature [47]. The heuristics are the maximum non-holonomic-without-obstacles and obstacle map, ignoring the nonholonomic nature [46]. The MATLAB navigation toolbox has the function for the hybrid A* algorithm.

Algorithm 2: Hybrid A*

```

1 Initialization of openset, closeset
2 openset.push(start)
3 while openset is not empty do
4   xcurrent ← openset.popMinCostNode
5   if exists RS path then
6     | return the path
7   end
8   for xnext calculated by kinematic equation do
9     Collision avoidance
10    if xnext not exists in closeset then
11      g ← g(xcurrent) + l(xcurrent, xnext)
12      if xnext not exists in openset or g < g(xnext) then
13        g(xnext) ← g
14        h(xnext) ← Heuristic(xnext, xgoal)
15        Pred(xnext) ← xcurrent
16        if xnext not exists in openset then
17          | openset.push(xnext)
18        else
19          | openset.update(next node)
20        end
21      end
22    end
23  end
24 end

```

The hybrid A* and WLVM combination is shown in Figure 4, and the proposed WLVM is in Algorithm 3. The leaders are generated for each swarm, and they are unique. For the generated path for followers, the path needs to be smooth by the Spline curve for AGVs to operate. The leaders will be regenerated if the environment or group formation changes. The statuses of AGVs are defined as (5). The statuses of AGVs are dynamically assigned based on their roles in the current swarm. Once the AGV arrives at the destination, the status is −1.

$$status = \begin{cases} 1, & \text{if leader} \\ 0, & \text{if follower} \\ -1, & \text{if arrives} \end{cases} \quad (5)$$

WLVM

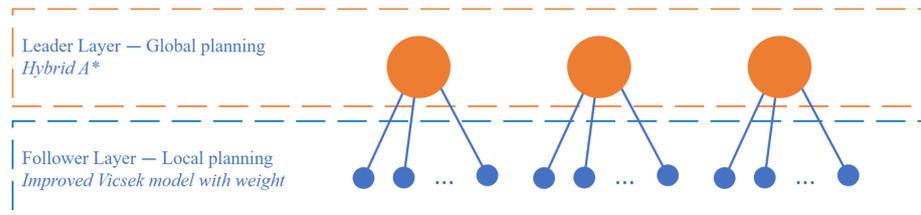


Figure 4. Dual layer of WLVM.

Algorithm 3: WLVM.

```

Data: x, y, leaderStart, leaderTarget
1 Initialize parameters
  // n - the number of particles
2 n ← size(x,2) + 1
3 dt ← 1
4 change ← 1
5 x ← [x leaderStartx]
6 y ← [y leaderStarty]
7 theta ← zeros(1,n)
8 for i ← 1 to n do
9   | theta(i) ← leaderStarttheta
10 end
11 while change > 0 do
12   | change ← change - 1
13   for each changed swarm do
14     | // getting the positions and angles for the virtual leader
15     | [LeaderPos, LeaderAngle] ← HybridAstar
16     | for time ← 1 to timemax do
17     |   Calculate averageTheta
18     |   x ← x + vel * cos(theta) * dt
19     |   y ← y + vel * sin(theta) * dt
20     |   theta ← w1 * averageTheta + w2 * leaderAngle(time) + noise
21     |   Smoothen path
22     |   Deadlock and collision avoidance
23   end
24   Segment delay
25 end
26 Swarm evaluation
27 if swarm changes then
28   | change ← change + 1
29 end

```

3.3.3. Start–Destination

WLVM can handle the multi-agent motion directed to different areas, as shown in Figure 5. The start–destination matrix is saved in Table 2. It assigns the multi-AGV groups, determines each group’s destination, treats them as other swarms, and does not integrate them. *M* stands for the number of leader-AGVs, which is much less than the number of follower-AGVs. *Start* and *Destination* represent each leader’s origin and destination locations.

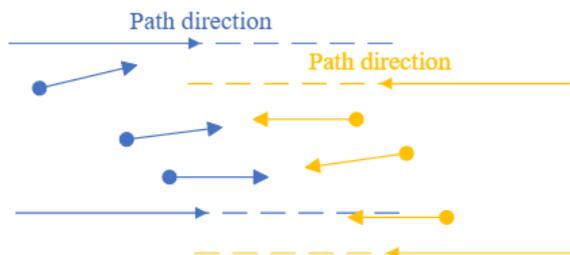


Figure 5. Groups with a different direction.

Table 2. Start–destination matrix.

Location	Virtual Leaders			
	Leader ₁	Leader ₂	...	Leader _M
Start	Start ₁	Start ₂	...	Start _M
Destination	End ₁	End ₂	...	End _M

Keep updating positions and directions within the defined group based on the destination flag. When AGVs are in operation, they only consider the AGVs along the path in the same direction. Even if the other path with a different direction is closer, the AGVs would not be generated. It only concerns the AGVs on the current path.

3.3.4. Segment Delay

The proposed WLVM enables AGVs to travel as a swarm and reach the target, which results in inefficiency for loading and unloading, so segment delay is introduced to solve this problem. Segment delay allows for the spacing of AGVs to avoid them arriving simultaneously. This facilitates organized loading and unloading operations. Figure 6 indicates the operation of segment delay, and it sets segment delay as two segments. Each AGV follows the defined path with a different segment delay for departure in each swarm. The segment delay is set as five segments in the computational experiments to provide the buffer area among AGVs.

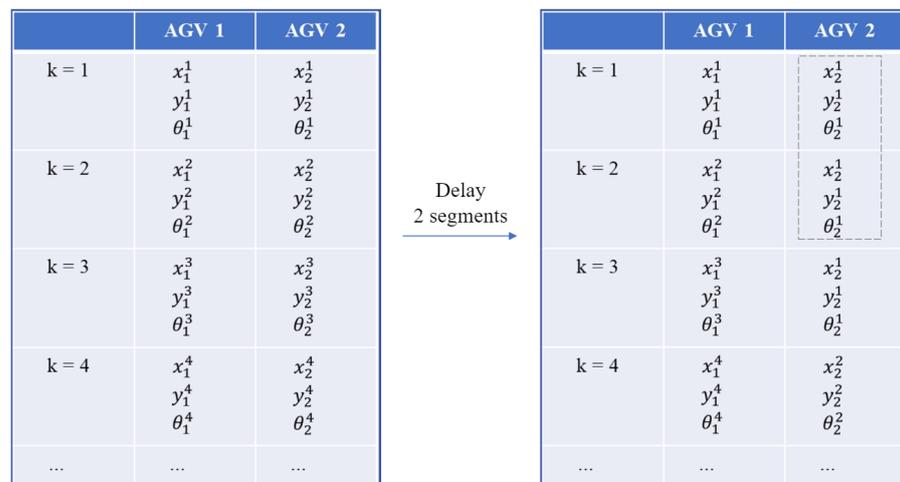


Figure 6. The positions after setting segment delay as two segments.

3.3.5. Collision Avoidance

Obstacle Avoidance

Collision avoidance is necessary during the updating iterations to keep AGVs safe. There usually are some obstacles on the map in the practical implementation, so collision avoidance with the obstacles should be achieved. Collision avoidance of the virtual leaders is achieved by the hybrid A* algorithm. The follower-AGV utilizes the leader angle in the next iteration to determine the direction of the movement.

Figure 7 demonstrates the movement of the AGV. Each obstacle or wall sets the buffer area as 1m. The leader angle θ indicates the movement of the path, and AGV is represented by i . If a path point represented by $r_i(x_i, y_i)$ overlaps with the buffer area or the restricted area as the obstacles, it requires adjusting positions. The steps for achieving obstacle avoidance are as follows.

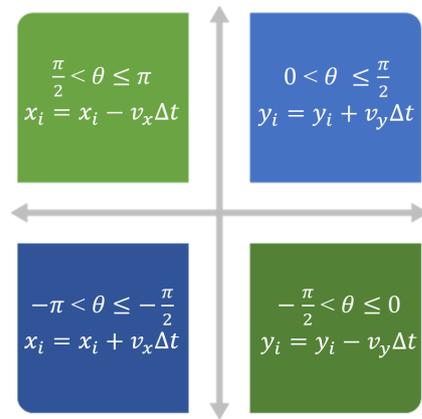


Figure 7. The AGV’s new position after collision avoidance.

First, comparing θ with 0. If $\theta \geq 0$ means the path is aimed to move upper/forward, then $y_i = y_i + v_y \Delta t$. Although $\theta < 0$ means the path moves lower, then $y_i = y_i - v_y \Delta t$. The AGVs follow the dotted line to change the Y locations. Second, comparing θ with $\frac{\pi}{2}$ or $-\frac{\pi}{2}$. If $\theta \geq \frac{\pi}{2}$ means the path is aimed to move left, then $x_i = x_i - v_x \Delta t$. Although $\theta \leq -\frac{\pi}{2}$, which means the path is moving right, then $x_i = x_i + v_x \Delta t$. The AGVs follow the dotted line to change the X locations.

Deadlock Avoidance

The other AGVs in the predictable path or the moving obstacles are treated as dynamic obstacles. As shown in Figure 8, vehicle congestion must be avoided. The target location reserves only one vehicle in each iteration to avoid deadlock. The strategies to deal with the moving obstacles, except for other AGVs, refer to the previous section.

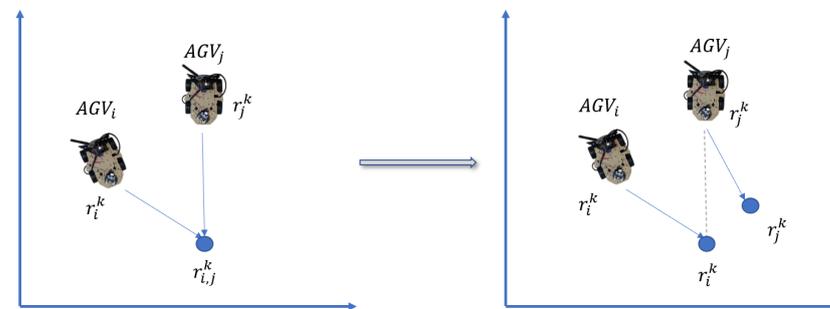


Figure 8. The AGV’s new position after deadlock avoidance.

Deadlock avoidance involves priorities for AGVs, and high priority is assigned when AGVs carry goods close to the destination or have urgent tasks. The AGV with higher priority remains following the predicted path, whereas the other AGV moves to an available position. The priority is calculated based on (6), and the new position is gained by (7). The new position is updated in the robot’s path after ensuring it is available.

$$priority_i = \omega_1 \cdot priority_{task} + \omega_2 \cdot distance \tag{6}$$

$$r_{i-new}^{k+1} = r_i^{k+1} + rand \tag{7}$$

where i stands for the current robot, and $distance$ is the distance from the current position to the destination. $priority_{task}$ represents the priority of each assigned task, and if the task is more urgent than others, $priority_{task}$ is higher. w_1 and w_2 are the weight of each factor, the sum of w_1 and w_2 is 1. r_{i-new} stands for a new position of the robot i .

3.4. Comparison

The proposed WLVM implements dynamic swarms and virtual leaders to ensure accurate direction and faster convergence, considering collision avoidance, the start–destination matrix to distinguish the destinations of swarms and the applications of multi-objective algorithms in the industrial environment. Figure 9 shows the advanced functionality of the WLVM. It also provides flexibility because of the dynamic swarms involved, and the follower-AGV can join or disconnect from the current group. The AGVs are assumed to exchange information during operations; if one AGV enters another area, it becomes a member of the new group.

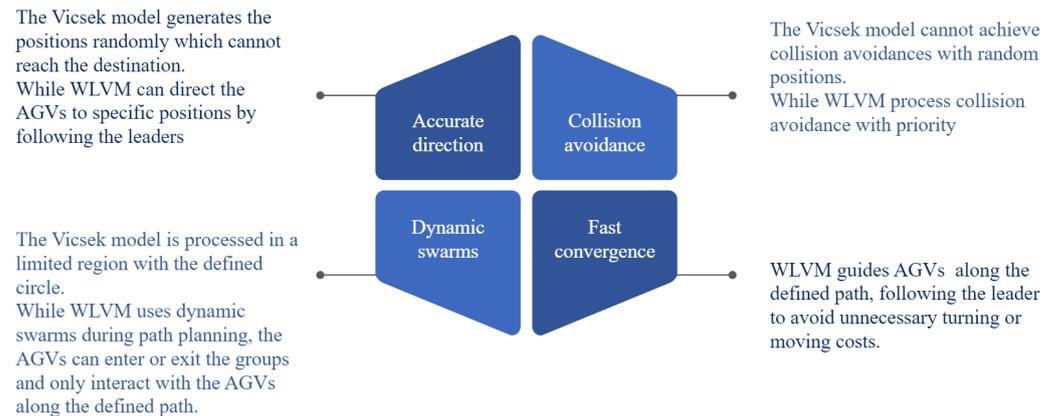


Figure 9. Comparison of Vicsek model and WLVM.

The WLVM is novel for multi-AGV navigation, and it adapts the biological pattern because it achieves the path planning of several AGVs in one step. The traditional Vicsek model can describe the multi-agent movement, which is enhanced to improve WLVM. Following the biological pattern, AGVs move automatically as their neighbours do in the same direction if operating the same task. It is typical to involve several AGVs for one task; the improved WLVM achieves fast multi-AGV path planning by updating the positions and angles. It only requires calculation for the virtual leaders in the leader layer and one calculation step in the follower layer. Even though the number of AGVs is large, it obtains the path with a quick calculation.

4. Computational Experiments

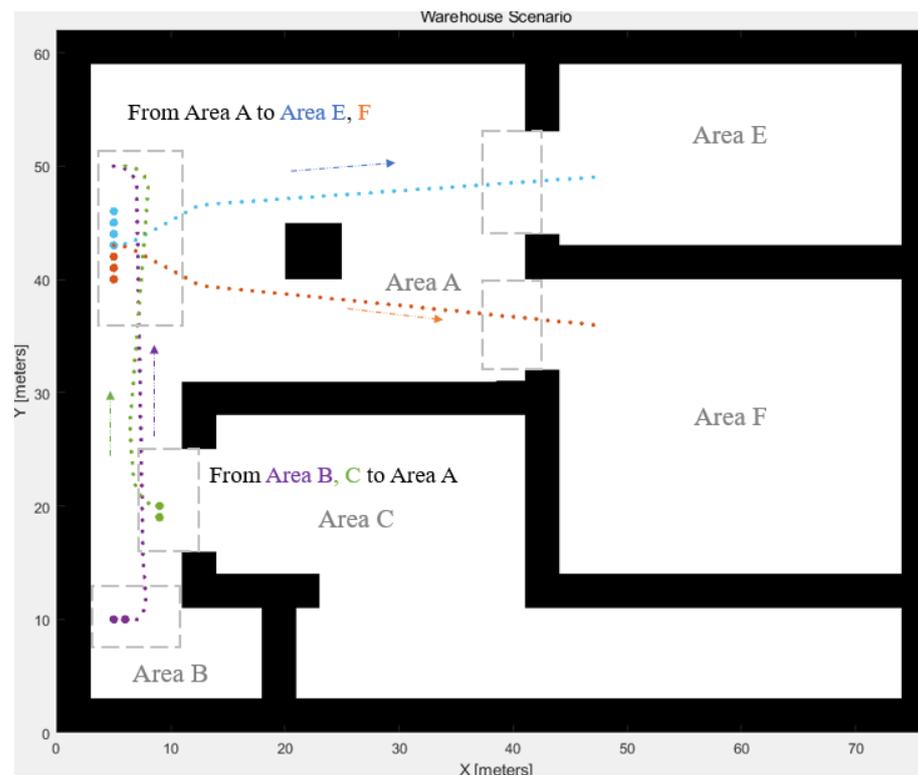
4.1. Experiment Settings

Figure 10 generates the warehouse map and denotes the initial locations of AGVs; different colours indicate the different swarms. WLVM is validated through MATLAB. The start–destination matrix separates the swarms. Each delivery group is operating in Area A.

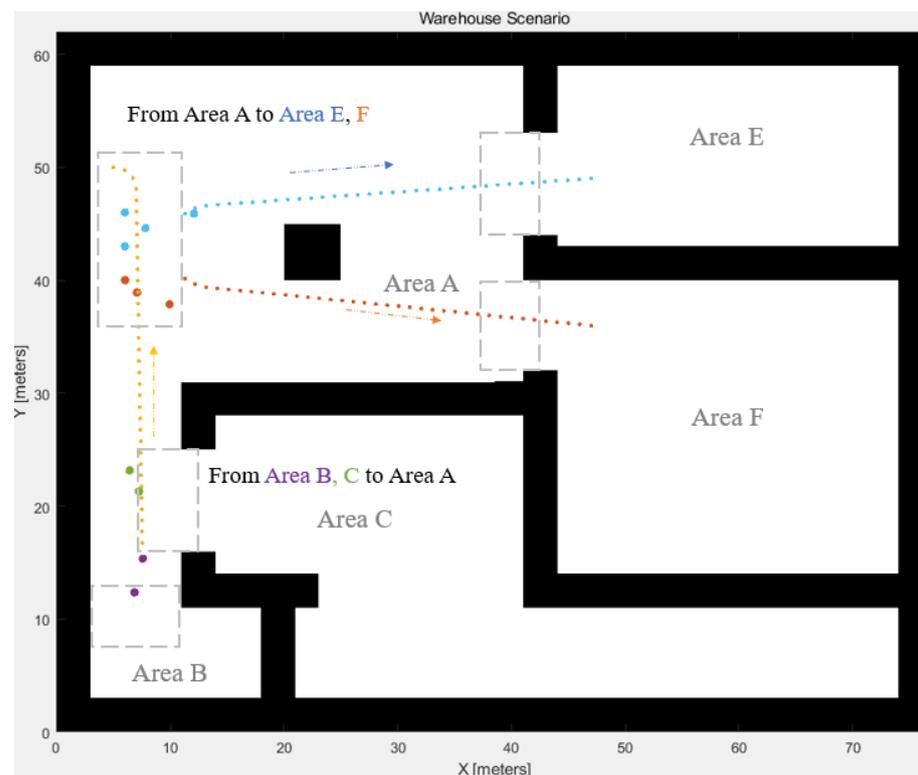
The multi-AGV system is engaged for deliveries from Area A because the three-dimensional storage system is implemented in the primary storage rooms: Area B to Area F. The materials are placed on the platform to transfer to the defined location by the pallet in the three-dimensional storage system.

Assumptions in the simulation are as follows:

- 7 AGVs depart from Area A to different storage areas for processing tasks, and 4 AGVs move to Area A for parking;
- Set segment delay for five segments in each swarm;
- AGVs with an absolute velocity of 1 m/s;
- Each AGV is equipped with a board that can operate WLVM;
- Each AGV has onboard sensors for localization and obstacle detection;
- Each AGV communicates with other AGVs, sending its positions, angles and statuses.



(a) After $t = 0$ (1st group change.)



(b) After $t = 8$ (2nd group change.)

Figure 10. The path for virtual leaders.

The groups' settings are listed in Table 3.

Table 3. Group settings.

Group No.	Group	Priority	Number of AGVs	Color
1	Area A to E	1	4	Blue
2	Area A to F	2	3	Orange
3	Area B to A	2	2	Purple
4	Area C to A	3	2	Green

4.2. Results

4.2.1. Simulation

Figure 10 displays the path for the virtual leaders generated by the hybrid A* algorithm after the group changes. The directions are indicated along with the path. It also outlines the path area for each swarm. The blue path is from Area A to Area E, and the orange path is from Area A to Area F. The purple path is from Area B to Area A, and the green path is from Area C to Area A. Groups 1 and 2 from Area A are separated based on the simulation’s destination. Groups 3 and 4 are merged into one group when they are close to each other and aim for the same destination.

Figure 11 shows the AGVs’ paths. When two delivery groups pass the same area, collision avoidance ensures that AGVs operate safely. The start–destination matrix has played a role in distinguishing the multi-AGV group. The groups of inbound delivery are aimed at different main storage rooms with different virtual leaders. When the group arrives at the destination, the group is reminded of the current state, and the moving AGVs achieve collision avoidance based on the priorities.

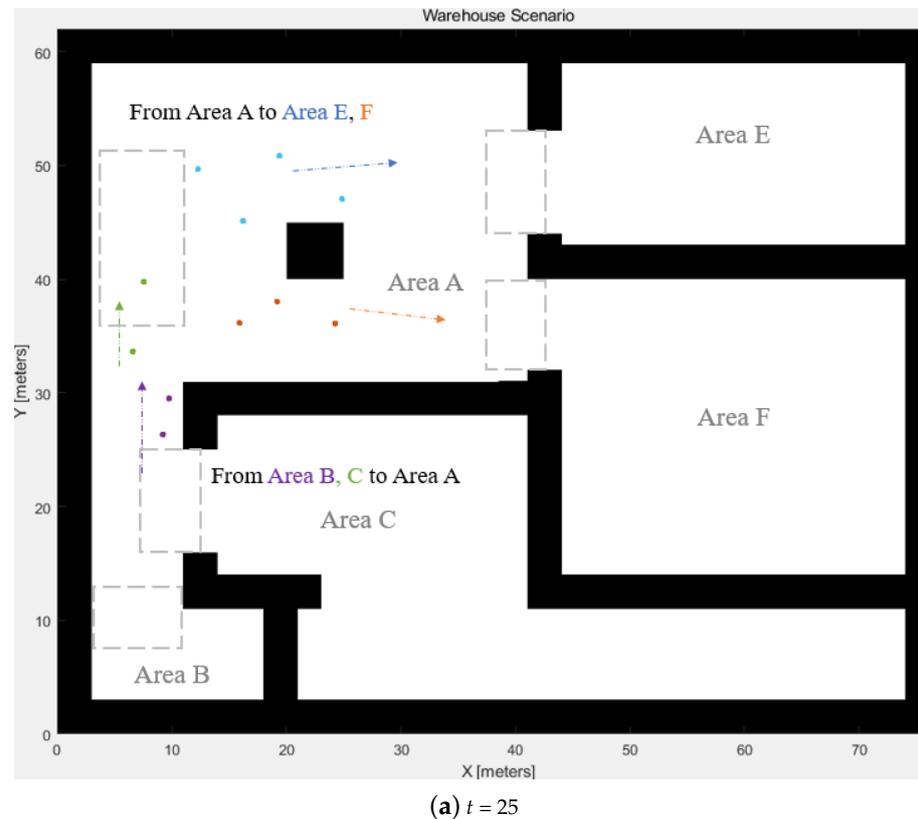


Figure 11. Cont.

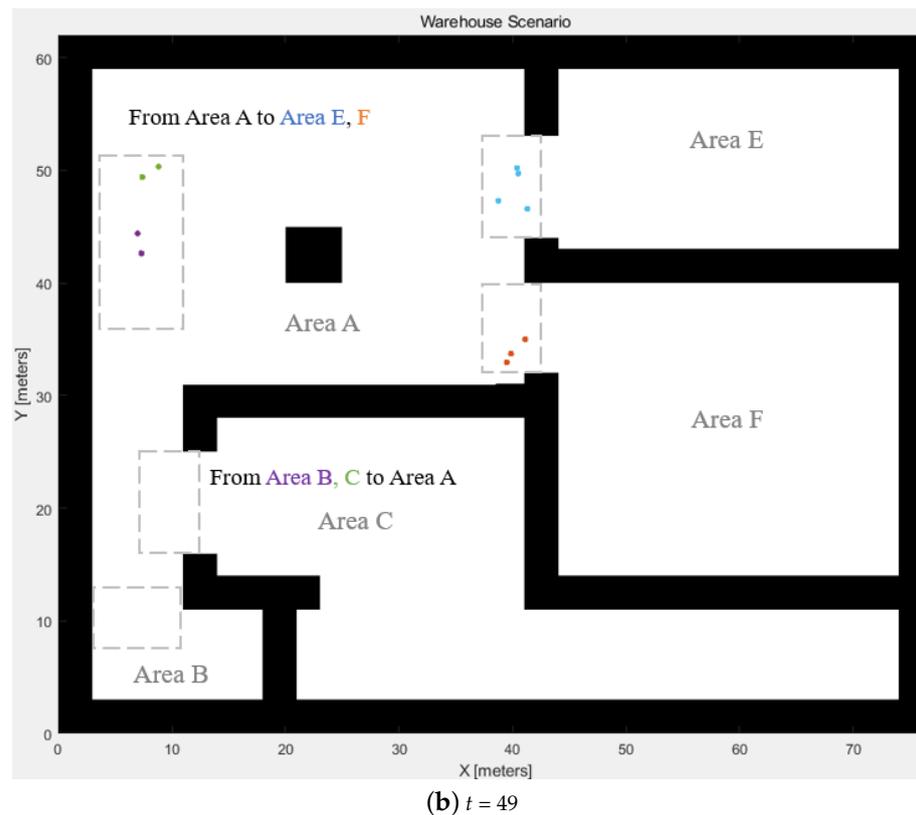


Figure 11. AGV positions during the path-planning process and the grey areas indicate the packing areas.

The following Table 4 lists the performance measurements. The group number is assigned according to the respective swarms. The distance is the travelling distance for each AGV in the group, which refers to the distance each AGV must travel to accomplish its assigned task. The time is the consumed time for each swarm to complete the task.

Table 4. Performance measurements.

Group No.	Group	Time	Distance	Group Changes
1	Area A to E	49 s	34.84, 36.46, 35.84, 34.02	Separated to Group 1 and 2 when $t = 0$
2	Area A to F	44 s	36.13, 37.36, 35.97	
3	Area B to A	37 s	33.00, 34.21	Group 3 and 4 merged when $t = 8$
4	Area C to A	35 s	30.82, 30.86	

4.2.2. Experiment

The results of the simulation section are validated by the experiment with the Raspberry Pi robot and Ultra-Wide Band (UWB) for positioning. UWB provides centimetre-level positioning and high positioning accuracy in the indoor environment. The robot follows the designed path gained by the simulation.

The experiment used the AGVs from Group 1, the inbound delivery group from Area A to Area E, with 4 AGVs. They followed the defined paths and gained positioning data from the positioning sensor. The positioning results were collected from the Decawave UWB sensors, and the robot carries the target for getting locations. The iterations of the results shown in Figure 12 are $t = 0$, $t = 25$, and $t = 49$. The start positions are indicated as $t = 0$, and the destinations are shown when $t = 49$. The locations have some bias due to the sensor accuracy, which can be fixed by sensor fusion in future work.

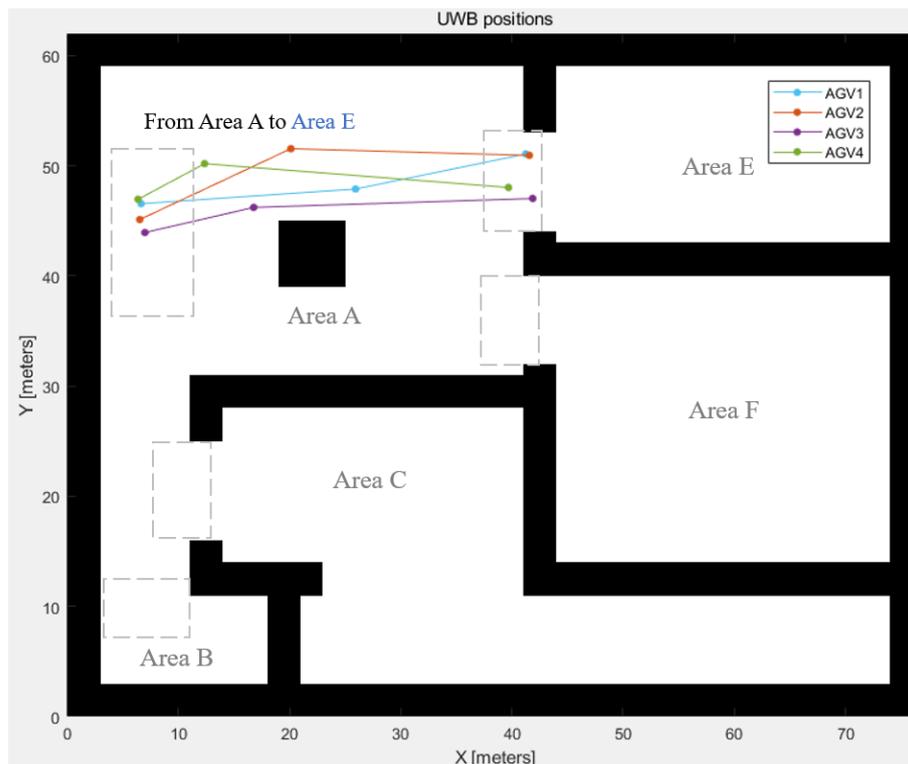


Figure 12. UWB positions following the generated path.

4.3. Comparison

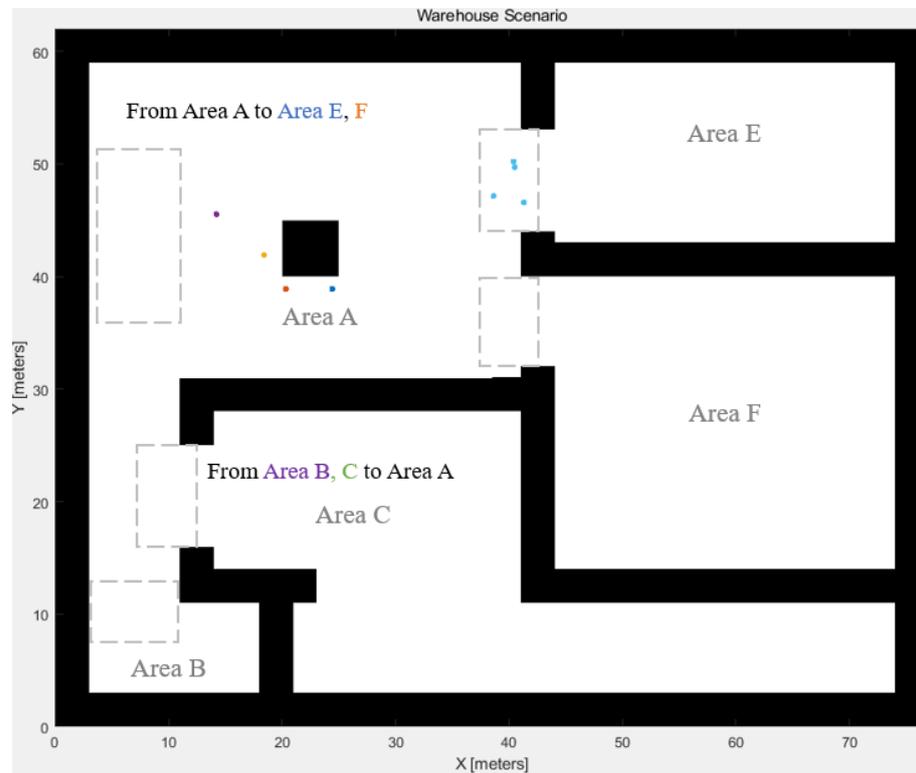
The proposed WLVM is compared with the RRT* algorithm [48] and an improved APF with deterministic annealing (DA-APF) [49] for path planning in Group 1, with a segment delay of five segments. The comparison of the runtime is listed in Table 5. RRT is a sampling-based algorithm that is popular for multi-constrained and high-dimensional path planning [50]. Sampling-based planners connect samples by constructing trees iteratively from the sampling distributions, and the planners deterministically or probabilistically draw random samples. APF uses attractive and repulsive potentials and treats the robot as an affected object. Wu et al. [49] apply a deterministic annealing strategy to improve the classical APF algorithm for path planning, changing the temperature parameters.

Table 5. Runtimes for the RRT*, DA-APF, and WLVM algorithms.

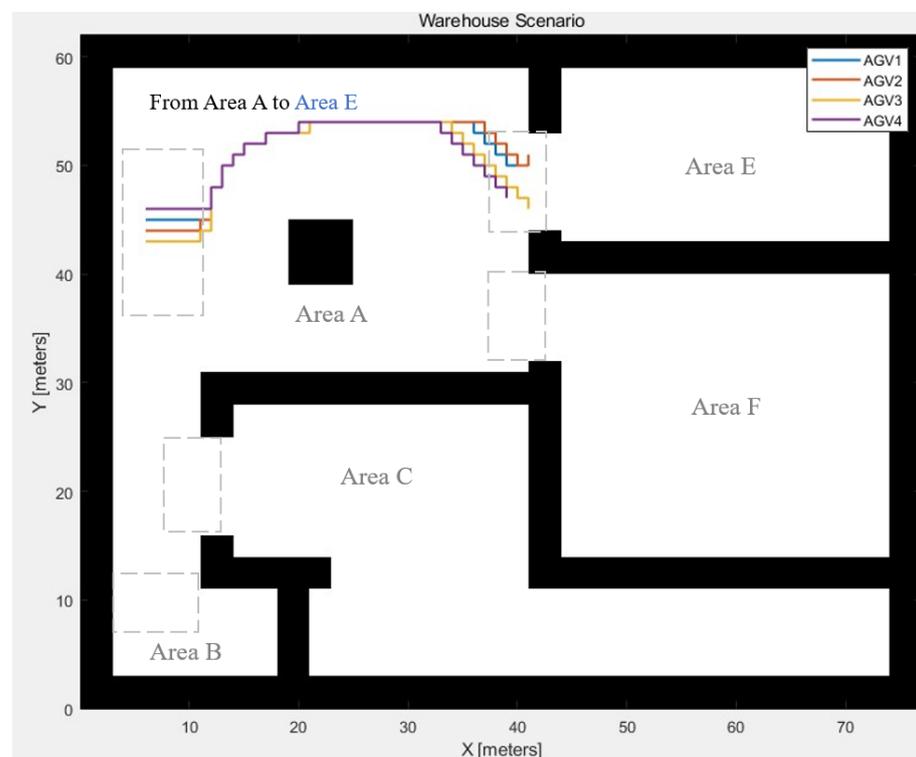
Algorithm	Runtime	Total Runtime
WLVM	Leader: 0.0757 s AGV 1–4: 0.0056 s	0.0813 s
RRT*	AGV 1: 1.5522 s AGV 2: 1.1777 s AGV 3: 1.1550 s AGV 4: 1.1533 s	5.0382 s
DA-APF	AGV 1: 0.0419 s AGV 2: 0.0381 s AGV 3: 0.0372 s AGV 4: 0.0374 s	0.1546 s

The RRT* and DA-APF algorithms need to repeat the calculation for each AGV, whereas WLVM calculates the path for all AGVs at one step. WLVM provides fast path planning due to the computational speed. Figure 13 shows the AGV positions generated by the RRT* and DA-APF algorithms. Figure 13a shows the positions generated by the RRT* algorithm for $t = 25$ that are marked by different colours and the last positions when $t = 64$ that are marked by blue. If the number of AGVs dramatically rises, the computation time for the RRT* algorithm will rise rapidly, whereas it would not affect WLVM. Figure 13b

indicates the path generated by the DA-APF algorithm. Although DA-APF achieves fast computation, it is still slower than WLVM for grouped AGVs.



(a) RRT* algorithm.



(b) DA-APF algorithm.

Figure 13. AGV positions generated by other algorithms.

5. Conclusions

The WLVM is proposed to improve the Vicsek model, and it develops dynamic virtual leaders and a start–destination matrix, considering the leaders' direction with weight. Also, it is capable of providing scalability and flexibility for multi-AGV systems with fast and flexible path settings. The path-planning problem is formulated as a 2D space with the start and target locations, and it avoids static and dynamic obstacles. From the literature, most path-planning approaches plan the path independently. However, the proposed WLVM algorithm can offer the path settings in one calculation step for swarms.

The WLVM implements the virtual leader to navigate the follower-AGV in each multi-AGV group, and the leader's positions and angles are generated by the hybrid A* algorithm. The proposed approach updates the statutes of AGVs with iterations, and the angles of AGVs consider the neighbour and the leader. Model and system initialization and multi-AGV group formation are completed through a centralized method, whereas it achieves the dynamic decentralized approach for each AGV.

It computes the follower-AGVs' path with a quick computation, even though the number of AGVs is large. Unnecessary turning costs and path segments are avoided in this model. Each AGV only considers its neighbours in the path and tends to move as its neighbours. For swarm integration or separation, the start–destination matrix plays a role. It determines whether the multi-AGV group is aimed at the same destination and makes changes in the decentralized follower layer. Segment delay is implemented for optimal arrangement between AGVs for loading.

The proposed WLVM has the following benefits: accurate direction, dynamic swarms, fast convergence, and collision avoidance. It can achieve real-time implementation due to its computational speed and robust implementation. For the computational experiment, four groups with different settings demonstrate swarm separation and integration. The proposed algorithm is compared with other algorithms for path planning of four AGVs; the WLVM saves 98.39% and 47.41% computational time than the RRT* and DA-APF algorithms, respectively. WLVM is robust and simple for implementation during the AGVs' or robots' operations. The proposed algorithm can be applied to various scenarios involving the system of multiple robots, such as warehouses, logistics systems, ports, and airports.

The limitation of the proposed approach is that it does not consider the cost value during the path planning as the heuristic methodologies. Therefore, the generated path cannot be measured or estimated with the specific costs to determine whether the path is globally optimal. The robot's dynamics and different speed scenarios must be considered during the real-world application. With the following considerations, WLVM would be more practical in the industrial environment for future work. Mission planning and task allocation can be included in the further improvement of this model. The multi-AGV system would be more practical if it involves fault tolerance during implementation. The combination of sensors and the sensor-fusion algorithm could be considered in further real experiments to estimate the angle and the position and detect and avoid obstacles. Implementing a neural network for the path-planning approach can be considered a potential improvement.

Author Contributions: Conceptualization, S.L.; methodology, S.L.; software, S.L.; validation, S.L.; formal analysis, S.L.; investigation, S.L.; resources, S.L.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, S.L., A.L., and J.W.; visualization, S.L.; supervision, J.W.; project administration, S.L.; funding acquisition, S.L. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AGV	Automatic guided vehicles
Multi-AGV	Multiple automatic guided vehicles
WLVM	Weight-Leader-Vicsek Model
RRT	Rapidly-exploring random trees
WMS	Warehouse management system
DA-APF	APF algorithm based on deterministic annealing

References

- Liu, J.; Jayakumar, P.; Stein, J.L.; Ersal, T. A nonlinear model predictive control formulation for obstacle avoidance in high-speed autonomous ground vehicles in unstructured environments. *Veh. Syst. Dyn.* **2017**, *56*, 853–882. <https://doi.org/10.1080/0042314.2017.1399209>.
- Yuan, Z.; Yang, Z.; Lv, L.; Shi, Y. A Bi-Level Path Planning Algorithm for Multi-AGV Routing Problem. *Electronics* **2020**, *9*, 1351. <https://doi.org/10.3390/electronics9091351>.
- Guo, X.; Ren, Z.; Wu, Z.; Lai, J.; Zeng, D.; Xie, S. A Deep Reinforcement Learning Based Approach for AGVs Path Planning. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 6833–6838. <https://doi.org/10.1109/cac51589.2020.9327532>.
- Lynch, L.; Newe, T.; Clifford, J.; Coleman, J.; Walsh, J.; Toal, D. Automated Ground Vehicle (AGV) and Sensor Technologies—A Review. In Proceedings of the Twelfth International Conference on Sensing Technology (ICST), Limerick, Ireland, 4–6 December 2018; pp. 347–352.
- Cramer, M.; Cramer, J.; De Schepper, D.; Aerts, P.; Kellens, K.; Demeester, E. Benchmarking low-cost inertial measurement units for indoor localisation and navigation of AGVs. *Procedia CIRP* **2019**, *86*, 204–209.
- Kostov, M.; Kostova, V.; Markoska, R. AGV guidance system simulations with a programmable robotics kit. *Int. J. Reason.-Based Intell. Syst.* **2015**, *7*, 42–46.
- Cui, Z.; Xu, H.; Chen, Z.; Yang, H.; Huang, S.; Gong, M. Design of a novel AGV with automatic pick-and-place system based on scissor lifting platform. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 4435–4440. <https://doi.org/10.1109/cac51589.2020.9327003>.
- Zacharia, P.T.; Xidias, E.K. AGV routing and motion planning in a flexible manufacturing system using a fuzzy-based genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2020**, *109*, 1801–1813. <https://doi.org/10.1007/s00170-020-05755-3>.
- Li, P.; Xu, Y.; Shen, T.; Bi, S. INS/UWB integrated AGV localization employing Kalman filter for indoor LOS/NLOS mixed environment. In Proceedings of the International Conference on Advanced Mechatronic Systems (ICAMechS), Kusatsu, Japan, 26–28 August 2019; pp. 294–298.
- Lin, S.; Liu, A.; Kong, X.; Wang, J. Development of Swarm Intelligence Leader-Vicsek-Model for Multi-AGV Path Planning. In Proceedings of the 2021 20th International Symposium on Communications and Information Technologies (ISCIT), Tottori, Japan, 19–22 October 2021; pp. 49–54. <https://doi.org/10.1109/ISCIT52804.2021.9590578>.
- Wang, J.; Pan, J.; Huo, J.; Wang, R.; Li, L.; Nian, T. Research on Optimization of Multi-AGV Path Based on Genetic Algorithm Considering Charge Utilization. *J. Phys. Conf. Ser.* **2021**, *1769*, 12052. <https://doi.org/10.1088/1742-6596/1769/1/012052>.
- Yadav, A.; Gaur, A.; Jain, S.M.; Chaturvedi, D.K.; Sharma, R. Development Navigation, Guidance & Control Program for GPS based Autonomous Ground Vehicle (AGV) using Soft Computing Techniques. *Mater. Today Proc.* **2020**, *29*, 530–535. <https://doi.org/10.1016/j.matpr.2020.07.309>.
- Cekmez, U.; Ozsiginan, M.; Sahingoz, O.K. A UAV path planning with parallel ACO algorithm on CUDA platform. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 347–354. <https://doi.org/10.1109/ICUAS.2014.6842273>.
- Digani, V.; Hsieh, M.A.; Sabattini, L.; Secchi, C. Coordination of multiple AGVs: A quadratic optimization method. *Auton. Robot.* **2018**, *43*, 539–555. <https://doi.org/10.1007/s10514-018-9730-9>.
- Kong, H.; Sun, J.; Hu, J. Real-time Motion Planning Based on Layered Cost Map for AGV Navigation. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 7624–7628. <https://doi.org/10.1109/cac51589.2020.9327401>.
- Xu, Z.; Liu, X.; Chen, Q. Application of Improved Astar Algorithm in Global Path Planning of Unmanned Vehicles. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019. <https://doi.org/10.1109/CAC48633.2019.8996720>.
- Almurib, H.A.F.; Nathan, P.T.; Kumar, T.N. Control and path planning of quadrotor aerial vehicles for search and rescue. In Proceedings of the SICE Annual Conference 2011, Tokyo, Japan, 13–18 September 2011; pp. 700–705.
- Shi, Y.; Li, Q.; Bu, S.; Yang, J.; Zhu, L. Research on Intelligent Vehicle Path Planning Based on Rapidly-Exploring Random Tree. *Math. Probl. Eng.* **2020**, *2020*, 5910503. <https://doi.org/10.1155/2020/5910503>.
- Huang, T.; Huang, D.; Qin, N.; Li, Y. Path Planning and Control of a Quadrotor UAV Based on an Improved APF Using Parallel Search. *Int. J. Aerosp. Eng.* **2021**, *2021*, 5524841. <https://doi.org/10.1155/2021/5524841>.

20. Xu, J.; Tian, Z.; He, W.; Huang, Y. A Fast Path Planning Algorithm Fusing PRM and P-Bi-RRT. In Proceedings of the 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), Jinan, China, 23–25 October 2020; pp. 503–508. <https://doi.org/10.1109/PHM-Jinan48558.2020.00098>.
21. Draganjac, I.; Miklic, D.; Kovacic, Z.; Vasiljevic, G.; Bogdan, S. Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 1433–1447. <https://doi.org/10.1109/tase.2016.2603781>.
22. Lian, Y.; Xie, W.; Zhang, L. A Probabilistic Time-Constrained Based Heuristic Path Planning Algorithm in Warehouse Multi-AGV Systems. *IFAC-PapersOnLine* **2020**, *53*, 2538–2543. <https://doi.org/10.1016/j.ifacol.2020.12.293>.
23. Chen, X.; Wu, W.; Hu, R. A Novel Multi-AGV Coordination Strategy Based on the Combination of Nodes and Grids. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6218–6225. <https://doi.org/10.1109/lra.2022.3164754>.
24. Xia, P.; Xu, A.; Zhang, Y. A Multi-AGV Optimal Scheduling Algorithm Based on Particle Swarm Optimization. In *Artificial Intelligence and Security, Proceedings of the 6th International Conference, ICAIS 2020, Hohhot, China, 17–20 July 2020*; Communications in Computer and Information Science; Springer: Singapore, 2020; Chapter 47, pp. 527–538. https://doi.org/10.1007/978-981-15-8083-3_47.
25. Lian, Y.; Xie, W. Improved A* Multi-AGV Path Planning Algorithm Based on Grid-Shaped Network. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 2088–2092. <https://doi.org/10.23919/ChiCC.2019.8865840>.
26. Yang, Q.; Lian, Y.; Liu, Y.; Xie, W.; Yang, Y. Multi-AGV Tracking System Based on Global Vision and AprilTag in Smart Warehouse. *J. Intell. Robot. Syst.* **2022**, *104*, 42. <https://doi.org/10.1007/s10846-021-01561-5>.
27. Xing, L.; Liu, Y.; Li, H.; Wu, C.C.; Lin, W.C.; Chen, X. A Novel Tabu Search Algorithm for Multi-AGV Routing Problem. *Mathematics* **2020**, *8*, 279. <https://doi.org/10.3390/math8020279>.
28. Chen, J.; Zhang, X.; Peng, X.; Xu, D.; Peng, J. Efficient routing for multi-AGV based on optimized Ant-agent. *Comput. Ind. Eng.* **2022**, *167*, 108042. <https://doi.org/10.1016/j.cie.2022.108042>.
29. Solichudin, S.; Triwiyatno, A.; Riyadi, M.A. Conflict-free dynamic route multi-AGV using dijkstra floyd-warshall hybrid algorithm with time windows. *Int. J. Electr. Comput. Eng. (IJECE)* **2020**, *10*, 3596–3604. <https://doi.org/10.11591/ijece.v10i4.pp3596-3604>.
30. Zhao, Y.; Liu, X.; Wu, S.; Wang, G. Spare zone based hierarchical motion coordination for multi-AGV systems. *Simul. Model. Pract. Theory* **2021**, *109*, 102294. <https://doi.org/10.1016/j.simpat.2021.102294>.
31. Cardarelli, E.; Digani, V.; Sabattini, L.; Secchi, C.; Fantuzzi, C. Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses. *Mechatronics* **2017**, *45*, 1–13. <https://doi.org/10.1016/j.mechatronics.2017.04.005>.
32. Zheng, K.; Tang, D.; Gu, W.; Dai, M. Distributed control of multi-AGV system based on regional control model. *Prod. Eng.* **2013**, *7*, 433–441. <https://doi.org/10.1007/s11740-013-0456-4>.
33. Zhang, Y.; Wang, F.; Fu, F.; Su, Z. Multi-AGV Path Planning for Indoor Factory by Using Prioritized Planning and Improved Ant Algorithm. *J. Eng. Technol. Sci.* **2018**, *50*, 534–547. <https://doi.org/10.5614/j.eng.technol.sci.2018.50.4.6>.
34. Cao, X.; Zhu, M. Research on global optimization method for multiple AGV collision avoidance in hybrid path. *Optim. Control. Appl. Methods* **2021**, *42*, 1064–1080. <https://doi.org/10.1002/oca.2716>.
35. Tao, L.; Zhang, S.; Chen, S.; Zheng, N. Multi-AGV pathfinding for automatic warehouse applications. In Proceedings of the 2021 China Automation Congress (CAC), Beijing, China, 22–24 October 2021. <https://doi.org/10.1109/cac53003.2021.9728597>.
36. Zhong, M.; Yang, Y.; Dessouky, Y.; Postolache, O. Multi-AGV scheduling for conflict-free path planning in automated container terminals. *Comput. Ind. Eng.* **2020**, *142*, 106371. <https://doi.org/10.1016/j.cie.2020.106371>.
37. Liu, C.; Tan, J.; Zhao, H.; Li, Y.; Bai, X. Path planning and intelligent scheduling of multi-AGV systems in workshop. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 2735–2739. <https://doi.org/10.23919/ChiCC.2017.8027778>.
38. Yang, Y.; Zhang, J.; Liu, Y.; Song, X. Multi-AGV Collision Avoidance Path Optimization for Unmanned Warehouse Based on Improved Ant Colony Algorithm. In *Bio-Inspired Computing: Theories and Applications, Proceedings of the 14th International Conference, BIC-TA 2019, Zhengzhou, China, 22–25 November 2019*; Communications in Computer and Information Science; Springer: Singapore, 2020; Chapter 41, pp. 527–537. https://doi.org/10.1007/978-981-15-3425-6_41.
39. Zajac, J.; Malopolski, W. Structural on-line control policy for collision and deadlock resolution in multi-AGV systems. *J. Manuf. Syst.* **2021**, *60*, 80–92. <https://doi.org/10.1016/j.jmsy.2021.05.002>.
40. Digani, V.; Sabattini, L.; Secchi, C.; Fantuzzi, C. Ensemble Coordination Approach in Multi-AGV Systems Applied to Industrial Warehouses. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 922–934. <https://doi.org/10.1109/tase.2015.2446614>.
41. Vicsek, T.; Czirók, A.; Ben-Jacob, E.; Cohen, I.; Shochet, O. Novel Type of Phase Transition in a System of Self-Driven Particles. *Phys. Rev. Lett.* **1995**, *75*, 1226–1229. <https://doi.org/10.1103/PhysRevLett.75.1226>.
42. Lu, X.; Zhang, C.; Huang, C.; Qin, B. Research on swarm consistent performance of improved Vicsek model with neighbors' degree. *Physica A* **2022**, *588*, 126567. <https://doi.org/10.1016/j.physa.2021.126567>.
43. Lu, X.; Zhang, C.; Qin, B. An improved Vicsek model of swarm based on remote neighbors strategy. *Physica A* **2022**, *587*, 126553. <https://doi.org/10.1016/j.physa.2021.126553>.
44. Liu, X.; Xiang, X.; Chang, Y.; Chao, Y.; Tang, D. Hierarchical Weighting Vicsek Model for Flocking Navigation of Drones. *Drones* **2021**, *5*, 74. <https://doi.org/http://dx.doi.org/10.3390/drones5030074>.

45. Li, H.; Chu, Z.; Fang, Y.; Liu, H.; Zhang, M.; Wang, K.; Huang, J. Source-seeking multi-robot team simulator as container of nature-inspired metaheuristic algorithms and Astar algorithm. *Expert Syst. Appl.* **2023**, *233*, 120932. <https://doi.org/10.1016/j.eswa.2023.120932>.
46. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Practical Search Techniques in Path Planning for Autonomous Driving. *Ann Arbor* **2008**, *1001*, 18–80.
47. Petereit, J.; Emter, T.; Frey, C.W.; Kopfstedt, T.; Beutel, A. Application of Hybrid A* to an Autonomous Mobile Robot for Path Planning in Unstructured Outdoor Environments. In Proceedings of the ROBOTIK 2012, 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; p. 1.
48. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. <https://doi.org/10.1177/0278364911406761>.
49. Wu, Z.; Dai, J.; Jiang, B.; Karimi, H.R. Robot path planning based on artificial potential field with deterministic annealing. *ISA Trans.* **2023**, *138*, 74–87. <https://doi.org/10.1016/j.isatra.2023.02.018>.
50. Wang, J.; Chi, W.; Li, C.; Wang, C.; Meng, M.Q.H. Neural RRT: Learning-Based Optimal Path Planning. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1748–1758. <https://doi.org/10.1109/TASE.2020.2976560>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.