This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



**IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan. Cite as T-ASE paper.**

# Simulation-Based Approach for Automatic Roadmap Design in Multi-AGV Systems

Tena Žužek, *Member, IEEE*, Rok Vrabič, *Member, IEEE*, Andrej Zdešar, Gašper Škulj, Igor Banfi, Matevž Bošnak, Viktor Zaletelj, and Gregor Klančar

***Abstract*— This paper addresses the problem of establishing efficient intralogistic systems, focusing on the generation of roadmaps on a given layout and the coordination of multiple Automated Guided Vehicles (AGVs). A simulation-based approach for automatic roadmap design is proposed. An event-based simulator is developed that uses ant-colony inspired optimization to generate roadmaps tailored to the specific characteristics of a given intralogistic problem, i.e., the plant layout, fleet size, statistical description of tasks, dispatching algorithm, etc. The generated solutions are evaluated with a Multi-Agent Path Finding (MAPF) simulator that uses a Safe Interval Path Planning (SIPP) algorithm. By analysing the system throughput, the optimal fleet size for the system is proposed. The approach is validated through various examples and benchmarked against existing methods in the literature.**

***Note to Practitioners*—Creating efficient roadmaps plays a key role in improving the efficiency of multi-AGV systems. Usually, roadmaps are designed by experts, which is time consuming and may lead to suboptimal solutions due to the complexity of the problem. The main objective of this work is to develop a methodology that allows the automatic generation of efficient roadmaps taking into account the specific characteristics of an intralogistic problem. The main advantage is the fast generation of solutions and their validation in terms of throughput. At the same time, the presented methodology allows a quick correction of the solution if one of the problem characteristics changes, without the need to change the layout itself. This provides industry practitioners with an effective tool that facilitates their work in setting up multi-AGV systems, thus saving time and costs.**

***Index Terms*— Multi-AGV systems, roadmap design, multi-robot coordination, event-based simulations.**

## I. Introduction

AUTOMATED guided vehicles (AGVs) are increasingly used to improve the flexibility and efficiency of intralogistics in modern manufacturing and warehousing systems. AGVs are typically constrained to move along paths specified by a magnetic tape attached to the floor. These paths form a roadmap, the design of which in a given layout is one of the key parameters for logistics performance.

In most cases, roadmaps are still designed manually by human experts. This process is not only time-consuming, but can also lead to a solution that is far from optimal [1]. Moreover, the quality of the roadmap strongly depends on a number of parameters of the intralogistic problem, such as the number of AGVs, the distribution and parameters of the tasks, the task dispatching algorithm, the pathfinding algorithm, etc. As soon as one of the parameters changes, which is often the case in today's unpredictable environment, the roadmap needs to be adjusted accordingly to maintain an efficient traffic flow. Therefore, automating the design process could significantly reduce the initial setup time and cost, increase flexibility, and help experts design better roadmaps.

Several approaches can be found in the existing literature. Most of them address single robot motion planning in a static environment. For this purpose, random sampling techniques and probabilistic roadmap methods have proven successful [2], [3]. Some of these strategies have been upgraded for dynamic [4] and multi-robot environments [5], [6].

For managing large robot fleets, [7] proposed an adaptive roadmap optimization approach based on a linear programming formulation. The approach computes an optimal roadmap configuration according to environmental constraints and is able of adapting the roadmap whenever the environment or station demand changes. In [8], the coordination of large fleets is improved by a reinforcement learning approach that enables automatic generation of routes on a given layout. The approach aims to avoid conflict situations between vehicles and thus improve system performance. Reference [9] proposed an approach to optimize large-scale fleet management by combining multi-agent reinforcement learning with a graph neural network. Reference [10] addressed the problem of cooperative pathfinding through the idea of direction maps that can learn about the movement of vehicles and thus generate implicit cooperation. In [11], a method for building a directed roadmap graph is proposed that allows for an effective collision avoidance in multi-robot navigation. This is achieved through the optimization of vertex positions and edge directions by stochastic gradient descent.

In recent years, there has been a growing focus on developing redundant roadmaps to significantly enhance multi-AGV system efficiency. By offering multiple path choices, the

Manuscript received 27 July 2023; accepted 3 October 2023. This article was recommended for publication by Associate Editor A. Parisio and Editor J. Yi upon evaluation of the reviewer's comments. This work was supported in part by the Slovenian Research Agency (ARRS) under Grant L2-3168, Grant P2-0219, and Grant P2-0270; and in part by Epilog d.o.o., Ljubljana, Slovenia. *(Corresponding author: Rok Vrabič.)*

Tena Žužek, Rok Vrabič, and Gašper Škulj are with the Faculty of Mechanical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia (e-mail: rok.vrabic@fs.uni-lj.si).

Andrej Zdešar, Matevž Bošnak, and Gregor Klančar are with the Faculty of Electrical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia.

Igor Banfi and Viktor Zaletelj are with Epilog d.o.o., 1000 Ljubljana, Slovenia.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2023.3323099.

Digital Object Identifier 10.1109/TASE.2023.3323099

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.

number of collisions between AGVs decreases, and unproductive time (congestions due to waiting) is reduced. Digani et al. [1] presented an approach for automatic roadmap generation that defines a set of near-optimal paths in such a way that the coverage, the redundancy, and the connectivity are maximized. The approach was extended to include a coordination strategy that is obtained as a combination of centralized and decentralized approaches (i.e., shared resources and local negotiation) [12]. Based on [12], a multi-layer traffic manager software was developed [13] to address the typical challenges of industrial roadmaps (e.g., loading and unloading locations, lack of redundancy or bidirectional corridors), and handle dynamic obstacles and communication failures in real industrial scenarios. The approach presented in [1] has also been adapted and integrated with a semi-automated mapping methodology, which not only speeds up the installation process and reduces setup costs, but also enables the creation of more redundant roadmaps with higher connectivity compared to manually created ones [14].

Redundancy and robustness measures of roadmaps have also been a research focus of [15]. They proposed an approach for automated roadmap creation for large robot fleets that is able to utilize as much of the available space as possible. The routing performance of the generated roadmaps has been evaluated using different Multi-Agent Path Finding (MAPF) algorithms. Compared to [1], the proposed approach is able to cope with more cluttered environments.

However, existing approaches that prioritize maximizing redundancy can generate numerous unnecessary paths, resulting in low road utilization. The computation time, although not explicitly mentioned in any of the works, is also assumed to be relatively high [16]. Furthermore, introducing potential improvements by the experts can be difficult as the developed system is hard to understand.

These drawbacks have been addressed to some extent by [16], who combined advantages of mathematical optimization with human expertise using fuzzy logic. The proposed expert system enables automated generation of roadmaps that are not only mathematically efficient, but also applicable to real production systems. The approach takes into account that an efficient algorithm has to consider specific problem inputs and optimization goals, the expert knowledge should be incorporated, and the process time should be short. The proposed method is however still hard to understand by system planners who are not experienced with fuzzy logic and any changes of the rules also require changes in the code. Also, formatting the inputs into a format that the proposed approach is able to handle can be quite daunting and can significantly increase the preparation time [16].

The aforementioned approaches address the automated generation of roadmaps, but focus only on the layout and do not address other specifics of the intralogistic problem, such as fleet size, task distribution, task dispatching algorithm, path finding algorithm, etc. Additionally, they can be computationally expensive and therefore not applicable for systems with constantly changing environment. This paper proposes a simulation-based approach to automatic roadmap design that uses a combination of ant colony optimization (ACO) and SIPP-based MAPF to quickly generate, simulate, and evaluate solutions to given specifics of an intralogistic problem. Even though heuristic methods, such as ACO, typically do not guarantee global optimality, they provide good-quality solutions that are useful for real-world applications, especially in complex, large-scale problems where exact methods may be impractical. ACO-based algorithms have already proven to efficiently solve various path planning problems [17], [18], [19].

In the context of our proposed method, guaranteeing the global optimum is computationally too demanding due to the complexity of the fleet management problem as a whole. Instead of aiming for the global optimum, our approach focuses on rapidly generating good solutions that exhibit high throughput compared to other existing methods. The result of the proposed approach is a roadmap tailored to the parameters of the problem and a proposal for the optimal number of AGVs. This is achieved by 1) a grid-based discrete-time simulation and ACO-inspired grid graph optimization that results in improved movement patterns encoded in the form of directed costs, 2) a transformation of the generated directed grid graph into a roadmap, and 3) a task execution simulation that applies SIPP-based MAPF to evaluate the solutions. Using the proposed method, different types of conflicts can be effectively addressed, thus ensuring improved traffic flow: ACO-inspired roadmap optimization focuses on mitigating congestions through generation of such movement constraints that vehicles can effectively avoid each other (e.g. one way roads in narrow hallways), while SIPP algorithm addresses conflict resolution at the task-level based on task priorities. One should note that the assignment of task priorities is not a subject of this paper and is assumed to be managed by an external warehouse or manufacturing execution system.

## II. Approach Overview

The goal of the proposed approach is to create a roadmap based on a given intralogistic problem. The **intralogistic problem** is defined by:

1) the fleet size $N_{AGV}$,
2) the layout of the plant including geometric information, locations of obstacles, pick-up, and drop-off locations,
3) a description of the tasks, i.e., a statistical description of task arrival rates and due date distributions, and
4) the dispatching algorithm that assigns the tasks to the AGVs

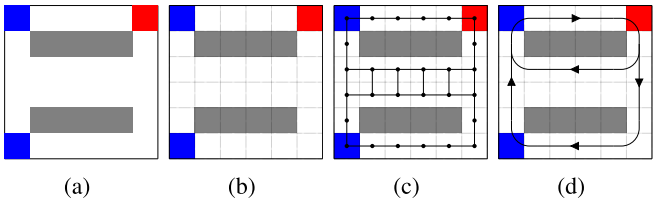
Given this description, the goal is to create a feasible roadmappoi and propose the size of the fleet that will allow the AGVs to complete the tasks on time, with little or no delay, and with fewer conflicts such as collisions.

The plant layout defines the workspace $W$ (Fig. 1a) which is discretized and represented as a 2-dimensional gridmap $G$ consisting of free cells and obstacles (Fig. 1b). Each grid cell has its own set of coordinates $[x, y]$, where $x, y \in \mathbb{N}_0$ represent the indices of the cells. The cells $[x_i, y_i]$ and $[x_j, y_j]$ are neighbors ($[x_i, y_i] \sim [x_j, y_j]$) if ($|x_i - x_j| = 1$ and $y_i = y_j$) or ($x_i = x_j$ and $|y_i - y_j| = 1$), which means that one of the cells is either North, South, West, or East of the other.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

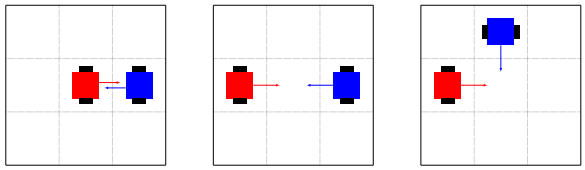


IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.



(a) (b) (c) (d)

Fig. 1. (a) Example of a simple layout - workspace $W$. (b) Workspace discretized into a gridmap $G$. (c) Weighted directed grid graph $\mathcal{G}$ superimposed on the gridmap $G$. (d) Resulting roadmap for AGVs. Gray squares represent obstacles, blue squares pick-up locations, and red squares drop-off locations.



(a) (b) (c)

Fig. 2. (a) Head-on edge conflict. (b) Head-on node conflict. (c) Side node conflict.

The size of the grid cell is determined by the size of an AGV: the cell is large enough to accommodate one AGV. It is assumed that two AGVs can meet on a single cell, but their movement is hindered to resolve the conflict. Two types of conflicts are distinguished:

1) edge conflict: two AGVs cross edges in opposite directions between the same two cells at the same time (Fig. 2a);
2) node conflict: two or more AGVs plan to occupy the same cell at the same time (Fig. 2b and Fig. 2c).

For the purpose of pathfinding, a weighted directed grid graph $\mathcal{G} = (V, E, w)$ is superimposed over the gridmap (Fig. 1c), where:

- $V = (v_1, \ldots, v_n)$ represents a set of graph vertices (nodes) that correspond to the centers of the free cells ($v_i$ corresponds to cell $[x_i, y_i]$);
- $E = (e_1, \ldots, e_m)$ represents a set of ordered pairs of vertices, called edges, that correspond to the transition between neighbor free cells: $E \subseteq \{(v_i, v_j) \mid (v_i, v_j) \in V^2, v_i \sim v_j\}$;
- $w = (w_1, \ldots, w_m) \in \mathbb{R}^{+m}$ represents a set of positive real weights that determine the costs of moving along the individual edges. If the weight $w_k = \infty$ is assigned to the edge $e_k$ connecting vertex $v_i$ to vertex $v_j$, moving from vertex $v_i$ (cell $[x_i, y_i]$) to vertex $v_j$ (cell $[x_j, y_j]$) is not possible.

In a grid graph, a route is defined as a sequence of vertices $r = (v_1, v_2, \ldots, v_R)$ in which each next vertex is a neighbor to the previous one. The cost of the route $c$ is the sum of all weights of the edges $w_k(e_k)$ connecting the vertices: $c = \sum_k w_k(e_k)$. The task of the pathfinding algorithm is to find such a route from the start vertex (start cell) to the goal vertex (goal cell) that its cost is minimized.

Tasks are defined as a triple of pick-up location, drop-off location, and task due time: $T = (p_{\text{pick-up}}, p_{\text{drop-off}}, t_{due})$. Tasks are generated based on the statistical description of arrivals and assigned to AGVs by a given dispatching algorithm. The AGVs then follow the route determined by the pathfinding algorithm.

The first subproblem can now be formalized as: *Given the set of problem inputs, find such weights $w$ of the weighted directed grid graph $\mathcal{G}$ so that the throughput (executed tasks per time unit) increases.*

Once the weights of the grid graph are determined, the second subproblem is as follows: *Transform the resulting weighted grid graph into a coherent roadmap that can be used to navigate the AGVs in the plant.* A roadmap (Fig. 1d) is a higher-level graph that abstracts the connectivity described by the weighted grid graph as a set of connections (edges) and key vertices that typically appear at the junction of connections. Connections are designed to be consistent with the kinematics constraints of the AGVs and to assure feasible motion. The roadmap also enables computationally efficient path planning as it compactly describes the grid graph with fewer vertices and edges. A roadmap is coherent if all vertices are reachable [20], i.e., for all vertices exists at least one path to every other vertex $\forall(v_i, v_j)\ \exists\ r = (v_1, \ldots, v_R) : v_i = v_1$ and $v_j = v_R;\ i \neq j$.

The last subproblem is the validation and evaluation of the quality of the generated roadmap through a series of experiments. This is treated as a classical multi-agent pathfinding problem on the created roadmap.

## III. Optimization and Validation Methodology

The methodology consists of three main steps: *A.* Grid graph optimization, *B.* Weighted grid graph to roadmap transformation, and *C.* MAPF simulation. In the following, each of the steps is described in detail.

### A. Grid Graph Optimization

To optimize the grid graph weights $w$, an event-based simulator was developed. Time $t$ is discretized. Each event occurs at a particular instant in time called a tick $\tau$ and marks a change of state in the system.

The inputs to the simulation are:

- the characteristics of the intralogistic problem,
- simulation duration $t_{sim}$, and
- the optimization parameters.

The first step of the grid graph optimization is the preprocessing of the workspace $W$ (the plant layout). During preprocessing, $W$ is converted into a gridmap $G$ composed of the following cell types: $o$ - obstacle, $f$ - free space, $p$ - pick-up, and $d$ - drop-off. In addition, $pd$ - a temporary storage unit is introduced that acts as both a pick-up and a drop-off cell type.

A weighted directed grid graph $\mathcal{G}$ is superimposed over $G$, whose edges connect traversable cells ($f$, $p$, $d$, and $pd$). At the beginning of the optimization, the weights of all edges are equal to $w_{init}$, which is $w_{max}/2$ by default. The weights are clamped to the interval $[1, w_{max}]$ at all times.

AGVs are represented as homogeneous agents. Agents $A = (a_1, \ldots, a_{N_{AGV}})$ are randomly placed on the starting positions defined by cells $p$, $d$, and $pd$. The following is assumed:

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.

- each cell is large enough to accommodate one agent,
- each agent has prior knowledge of the environment,
- the environment is static,
- agents do not communicate with each other, and
- at each time step each agent can either wait or move to one of its neighboring cells, i.e., one cell to the North, South, West, or East.

Each agent $a_i$ moves on $\tau_{a_i}$ determined by its velocity $v_{a_i}$. At each move, the tick is updated to be $\tau_{a_i} \leftarrow \tau_{a_i} + 1/v_{a_i}$. The maximum velocity of an agent is equal to 1 cell per tick.

To simulate the task arrival process, task generators $TG$ are introduced. Each task generator $TG_j \in TG$ generates a new task $T = (p_{\text{pick-up}}, p_{\text{drop-off}}, t_{due})$ at $t = \tau_{TG_j}$, where $\tau_{TG_j}$ is determined by the arrival rate. The task parameters depend on the characteristics of an individual task generator. When a new task is generated, it is dispatched to one of the agents according to the dispatching algorithm $TD$ (e.g., assign the task to the closest free AGV) and added to its set of tasks $\mathcal{T}_{a_i}$.

When a task is assigned to an agent, a plan $P_{a_i}$ to reach the destination is generated according to a path finding algorithm $PF$. The complete plan consists of the route from the agent's starting position to the pick-up position and the route from the pick-up to the drop-off position. Since the workspace is represented in the form of a weighted grid graph, any algorithm that can find paths on such graphs can be used, e.g., A*, R*, JPS, etc.

Each agent follows the steps of its plan and may collide with one or more other agents at each time step. To reduce the amount of collisions, a cost update algorithm inspired by ant-colony optimization (ACO) is proposed. The pheromone levels coincide with costs, i.e. the weights of the directed grid graph used by A*. These costs depend not directly on the path length as in traditional ACO approaches, but on the success of task execution. Specifically, if a task is executed on time, the costs decrease (higher pheromone level), and if there is a delay, the costs increase. By adopting this strategy, we establish a direct connection between the intralogistic problem and the pathfinding problem.

To encourage ant-like behaviour, the collisions are handled as follows. When a collision occurs, the involved agents need additional time to resolve it, so $\tau_a$ increases by the predefined value $t_{\text{penalty}}$ for all agents in collision. On each time step $t$ when an agents completes the task, the costs of all the cell edges on its route are updated as shown in Eq. (1).

$$\forall e_i = (v_j, v_{j+1}) \in r : w_i \leftarrow w_i + t - t_{due} \tag{1}$$

To prevent convergence to a potentially suboptimal path, an appropriate pheromone evaporation rate $\lambda_{evp}$ and evaporation value $w_{evp}$ must be chosen at the beginning of the simulation. All weights are then updated on $\tau_{evp}$ every $1/\lambda_{evp}$ steps as shown in Eq. (2).

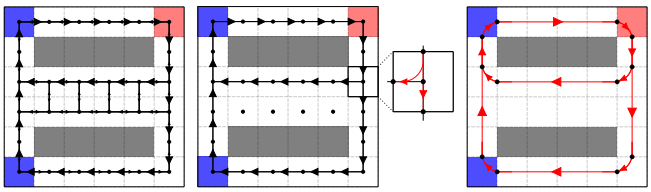
$$w_i \leftarrow \begin{cases} \min(w_i + w_{evp}, w_{init}) & \text{if } w_i < w_{init} \\ \max(w_i - w_{evp}, w_{init}) & \text{if } w_i > w_{init} \\ w_i & \text{otherwise} \end{cases} \tag{2}$$

The complete pseudocode of the proposed ACO-inspired grid graph optimization is presented in Algorithm 1.

**Algorithm 1** ACO-Inspired Grid Graph Optimization

**Input:** $W$, $N_{AGV}$, $TG$, $TD$, $PF$, $t_{sim}$, $\tau_{evp}$, $w_{evp}$
**Output:** $\mathcal{G}$

load the plant layout $W$, create gridmap $G$ and the corresponding weighted grid graph $\mathcal{G}$;
$\forall w_i \in \mathcal{G} : w_i = w_{init}$
generate $N_{AGV}$ agents on random starting positions;
**for** $t = 1, \ldots, t_{sim}$ **do**
  **for each** $TG_j \in TG$ with $\tau_{TG_j} = t$ **do**
    generate task according to $TG_j$;
    assign task to agent according to $TD$;
  **for each** $a_i \in A$ with $\tau_{a_i} = t$ **do**
    **if** $P_{a_i} = \emptyset$**and**$\mathcal{T}_{a_i} = \emptyset$ **then**
      stay
    **else if** $P_{a_i} = \emptyset$**and**$\mathcal{T}_{a_i} \neq \emptyset$ **then**
      generate $P_{a_i}$;
    **if** $P_{a_i} \neq \emptyset$ **then**
      move according to $P_{a_i}$;
    **if** $\mathcal{T}_{a_i}$ = executed **then**
      $\forall e_i = (v_j, v_{j+1}) \in r : w_i \leftarrow w_i + t - t_{due}$;
    $\tau_{a_i} \leftarrow \tau_{a_i} + 1/v_{a_i}$;
  **for each** pair $(a_i, a_j)$ **do**
    **if** in collision **then**
      $\tau_{a_i} \leftarrow \tau_{a_i} + t_{penalty}$;
      $\tau_{a_j} \leftarrow \tau_{a_j} + t_{penalty}$;
  **if** $\tau_{evp} = t$ **then**
    update $w_i$ according to Eq. (2);
    $\tau_{evp} \leftarrow \tau_{evp} + 1/\lambda_{evp}$
**return** $\mathcal{G}$



(a) (b) (c)

Fig. 3. (a) The weighted grid graph $\mathcal{G}$. (b) The pre-processed grid graph. (c) The final roadmap.

To analyze how the proposed algorithm improves the traffic flow, the following metrics are observed:

- the throughput (number of executed tasks per unit of time), and
- the conflict resolution metrics (number of collisions).

### B. Weighted Grid Graph to Roadmap Transformation

The weighted grid graph $\mathcal{G}$ (Fig. 3a) is first pre-processed. The paths between all possible pairs of $p$, $d$, and $pd$ are found using A*. Edges that are not on these paths are removed (Fig. 3b). This effectively minimizes the number of edges while preserving coherency, i.e., that all stations are still reachable.

Each node of the graph is then analysed in terms of its inbound and outbound edges to determine the geometric primitives of the roadmap. These include vertical or horizontal

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.

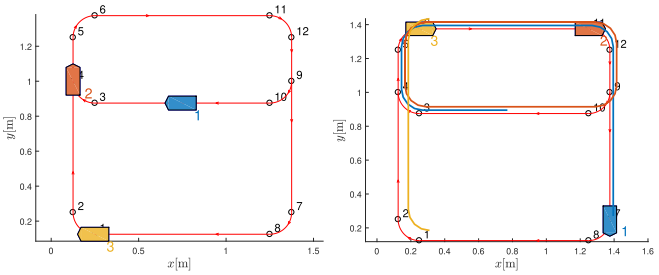



Fig. 4. MAPF simulator - example for the roadmap in Fig. 3c with 3 AGVs. (a) Initial locations. (b) Paths and final locations.

lines and circular 90 degree arcs. To position the geometric primitives, new nodes must be added at midpoints of the edges as illustrated in Fig. 3b. To create the final roadmap, the number of edges is minimised by joining sequential vertical and sequential horizontal edges (Fig. 3c).

### C. MAPF Simulation

To validate the efficiency of the generated roadmaps a MAPF simulator was developed. Using the simulator, realistic simulations of an intralogistic system can be carried out, as it considers the shape and the kinematics of the AGVs for detecting and resolving collisions.

The MAPF simulator extends and upgrades the SIPP-based prioritized path planner [21], so that it is applicable to the multiagent pathfinding problem. While the basic SIPP operates on a discrete graph, the developed MAPF simulator solves continuous problems where agents can be at any location at any time. To make MAPF problem computationally efficient, priorities are introduced into SIPP graph search. Without prioritized tasks MAPF is computationally too demanding and can be solved in a reasonable amount of time only for a low number of tasks and AGVs [22], [23].

Given a task, an AGV needs to plan a collision-free path from its current location to the pick-up location and then to the drop-off location. Tasks are assigned priorities in an oldest-first manner. To obtain conflict-free planning and identify successor states in graph search, we integrated a collision test with occupancy interval overlaps. AGVs use the SIPP planner according to the task priorities to assign occupancy time intervals for vertices and edges along the planned paths. The planner assumes that conflicts can happen along the same edges and vertices which makes collision checks very efficient. Additionally, collisions are also detected when AGV traverse different nearby edges or vertices that are close enough for the AGVs to collide due to their size. Virtual edges and vertices are introduced and assigned a nearby flag, indicating that occupying one such edge or vertex occupies the nearby one as well.

An example of the MAPF simulator for the roadmap in Fig. 3c with 3 AGVs is shown in Fig. 4. AGV $a_1$ (blue) with the highest priority needs to deliver to node 7, AGV $a_2$ (orange) to node 11, and AGV $a_3$ (yellow), with the lowest priority, to node 6. In Figs. 4a and 4b, the initial and the final locations of the AGVs are shown, along with the travelled paths.

Using the MAPF simulator the roadmaps can be evaluated through the analysis of the following metrics:

- throughput, determined by the time it takes to execute a set of tasks,
- coordination time, i.e., the time the AGVs spend on waiting for the occupied edges or vertices to free up, and
- the total distance travelled by all AGVs.

## IV. Validation and Results

To evaluate the proposed approach, we tested the methodology on three different plant layouts:

- a real production layout $W_{real}$ (Fig. 5a, taken from [24]),
- a fictional production layout $W_{fict}$ (Fig. 5b, taken from [24]), and
- a typical industrial warehouse $W_{wrhs}$ (Fig. 5c, taken from [12]).

The developed event-based simulator was used to generate different weighted grid graphs for each layout that were then transofrmed into roadmaps using the weighted grid graph to roadmap transformation. The generation of variants depends on a number of parameters and initialization. The simulation parameters influence the task execution delay and, in turn, the weights. When a delay occurs, the weights of the cell edges on the particular path increase, which means that it is less likely that the next AGV will take the same path. The path with fewer collisions will be preferred over the shortest path. In general, the probability of delays increases with a greater number of AGVs $N_{AGV}$, larger penalty $t_{penalty}$, and shorter task due time $t_{due}$. AGVs are also more likely to explore alternative paths if the initial weights of cell edges $w_{init}$ are low (below $w_{max}/2$). The parameters of the ACO optimization, the evaporation rate $\lambda_{evp}$ and the evaporation value $w_{evp}$, also guide the path selection. Lastly, the resulting movement patterns also depend on the initialization. The tasks that are generated first and the locations of the AGVs to which these tasks are assigned, influence the final solution.

### A. Analysis of the Generated Roadmaps

Different solutions for the real production layout $W_{real}$ from Fig. 5a are presented in Fig. 6. Compared to Scenario 1 (Fig. 6a), Scenario 2 (Fig. 6b) was obtained using lower $N_{AGV}$, smaller $t_{due}$, lower $w_{init}$, and smaller $\lambda_{evp}$, to increase the likelihood of agents exploring different paths. In Scenario 3 (Fig. 6c), $w_{init}$ was even lower, and the collision penalty $t_{penalty}$ was increased. With these optimization parameters the preference is shifted towards paths with fewer collisions over shorter ones. In Fig. 6d, the solution proposed in the literature is shown for comparison. Solutions for a fictional layout $W_{fict}$ and a typical industrial warehouse $W_{wrhs}$ are provided in the Appendix.

The generated roadmaps were evaluated using MAPF simulations implemented in Matlab. The scenarios proposed in the literature were also analysed. For each layout, the roadmaps were compared using the following metrics: the task execution time $t_{fin}$, the total coordination time $t_{coo}$ (waiting time in

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.

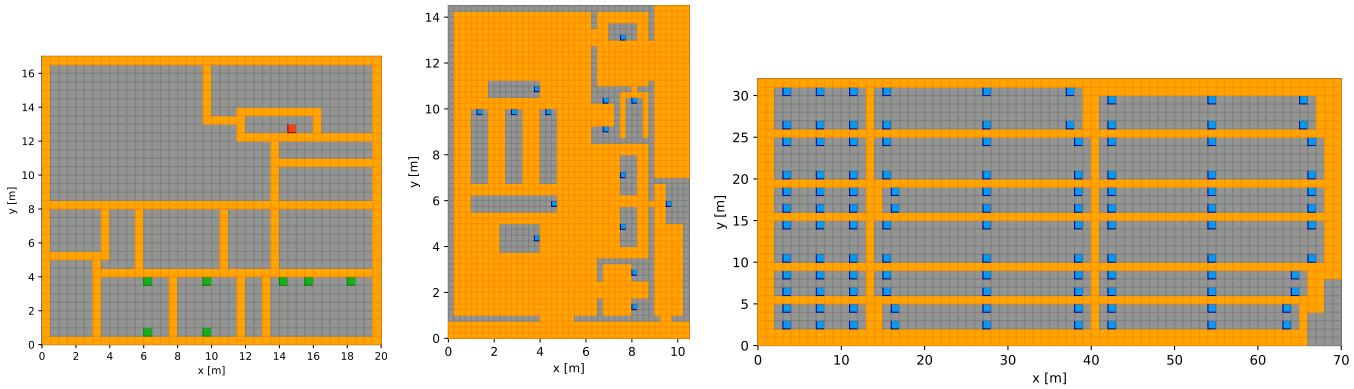



Fig. 5. (a) A real production layout $W_{real}$ (taken from [24]). (b) A fictional layout $W_{fict}$ (taken from [24]). (c) A typical industrial warehouse layout $W_{wrhs}$ (taken from [12]). The orange cells represent free space, grey cells obstacles, green cells pick-up locations, red cells drop-off locations, and blue cells pick-up and drop-off locations.

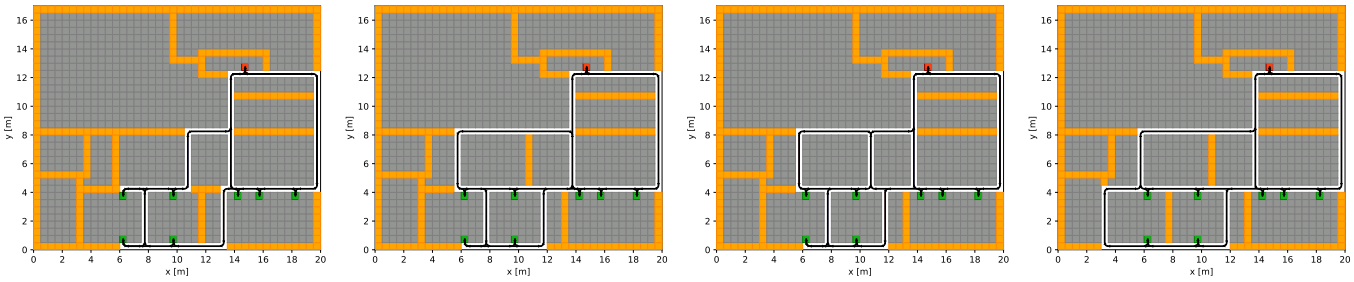



Fig. 6. Different solutions for a real production layout $W_{real}$: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3, and (d) Scenario L from the literature [24].

nodes to prevent collisions), and the total distance travelled by all AGVs $d_{tot}$.

Analyses have been carried out according to the following conditions and assumptions:

- For each roadmap, $N_{AGV}$ is set to 10 and 20.
- The same set of 1000 randomly generated tasks and initial positions of the AGVs are used for different roadmaps (solutions) of the same layout.
- Tasks are always in stock, as soon as an AGV delivers a task, it is assigned a new one (simplified dispatching).
- SIPP-based prioritized path planner is used.
- Loading/unloading time is simulated with 2 s of waiting at pick-ups and drop-offs.
- AGV velocity is 1 m/s and size 0.48 m $\times$ 0.24 m.
- The simulation is stopped when all the tasks have been completed.

The results of the evaluation are shown in Table I. The best scenarios for each layout are identified based on the highest throughput, i.e., the lowest total time to complete all tasks $t_{fin}$ (in Table I denoted in bold). In all cases, except for the $W_{fict}$ layout with 10 AGVs, the throughput is improved compared to the roadmaps proposed in the literature (for $W_{real}$ 2% and 4% for 10 and 20 AGVs, respectively, and for $W_{wrhs}$ 9% and 10% for 10 and 20 AGVs, respectively). In the $W_{fict}$ layout with 10 AGVs, our optimized scenarios are marginally worse in terms of throughput (around 3%), however, the coordination times are significantly lower (almost 65%), which indicates that with the higher number of AGVs, the proposed scenarios would outperform the scenario from the literature. This is confirmed by the simulations with 20 AGVs, where 12% higher throughput is achieved.

One can also see that there is a correlation between the highest throughput, i.e. lowest $t_{fin}$ and the shortest total distance travelled by the AGVs $d_{tot}$ (except for the $W_{fict}$ layout for 20 AGVs), while the coordination time $t_{coo}$ is usually not the lowest for the scenario with the highest throughput. The reason for that is that in order to prevent a conflict, it is usually more efficient for AGVs to wait at a crossroad instead of driving around on a bypass, which results in a larger $t_{coo}$. This is not a priori true for all cases, as both $t_{coo}$ and $d_{tot}$ heavily depend on the roadmap itself, i.e. the length of the paths between individual stations, location of intersections, etc. A direct correlation between individual metrics is therefore not absolute and can deviate from general trends due to roadmap-specific characteristics.

In the $W_{fict}$ layout, the $t_{coo}$ of the Scenario L proposed in the literature is much larger than in any other solution obtained with the proposed optimizer. The reason for that is that other solutions mostly consist of single way connections (except where the layout does not allow that, e.g. dead-end connections at the stations), while the Scenario L consists of two-way connections only. It is worth mentioning that the solution from the literature was a bit unclear [24], therefore the two-way connections have been assumed by the authors.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



**IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.**

TABLE I

PERFORMANCE METRICES FOR DIFFERENT ROADMAPS AND LAYOUTS OBTAINED WITH MAPF SIMULATIONS

| | $N_{AGV}$ | Scenario | $t_{fin}$ [s] | $t_{coo}$ [s] | $d_{tot}$ [m] |
|---|---|---|---|---|---|
| $W_{real}$ | 10 | 1 | **4825** | 2806 | 41192 |
| | | 2 | 5290 | 3226 | 45330 |
| | | 3 | 4828 | 2568 | 41448 |
| | | L | 4911 | 2964 | 41857 |
| | 20 | 1 | **3414** | 19757 | 41761 |
| | | 2 | 3821 | 17762 | 45514 |
| | | 3 | 3420 | 19319 | 41954 |
| | | L | 3535 | 18865 | 42292 |
| $W_{fict}$ | 10 | 1 | 2682 | 1286 | 21361 |
| | | 2 | 2655 | 1653 | 20406 |
| | | 3 | 2787 | 1187 | 22723 |
| | | L | **2592** | 3638 | 17697 |
| | 20 | 1 | **1458** | 2848 | 21703 |
| | | 2 | 1473 | 3512 | 20802 |
| | | 3 | 1525 | 2866 | 23266 |
| | | L | 1639 | 7607 | 18958 |
| $W_{wrhs}$ | 10 | 1 | **14112** | 128 | 136353 |
| | | 2 | 15565 | 103 | 150945 |
| | | 3 | 14585 | 87 | 140933 |
| | | L | 15415 | 103 | 148806 |
| | 20 | 1 | **7082** | 257 | 135938 |
| | | 2 | 7941 | 172 | 152457 |
| | | 3 | 7211 | 235 | 138161 |
| | | L | 7769 | 185 | 148479 |

TABLE II

PERFORMANCE METRICES FOR DIFFERENT ROADMAPS AND LAYOUTS OBTAINED WITH GRID GRAPH SIMULATIONS

| | $N_{AGV}$ | Scenario | $N_{tasks}$ | $N_{collisions}$ |
|---|---|---|---|---|
| $W_{real}$ | 10 | 1 | **2283** | 1661 |
| | | 2 | 2092 | 1326 |
| | | 3 | 2277 | 1672 |
| | | L | 2264 | 1402 |
| | 20 | 1 | 3773 | 20160 |
| | | 2 | 3690 | 14333 |
| | | 3 | 3773 | 19907 |
| | | L | **3796** | 18443 |
| $W_{fict}$ | 10 | 1 | 1378 | 766 |
| | | 2 | 1466 | 889 |
| | | 3 | 1432 | 777 |
| | | L | **1543** | 2686 |
| | 20 | 1 | 2735 | 3491 |
| | | 2 | 2656 | 3767 |
| | | 3 | 2468 | 3232 |
| | | L | **2832** | 11303 |
| $W_{wrhs}$ | 10 | 1 | **1421** | 181 |
| | | 2 | 1279 | 167 |
| | | 3 | 1400 | 163 |
| | | L | 1310 | 139 |
| | 20 | 1 | **2867** | 888 |
| | | 2 | 2546 | 749 |
| | | 3 | 2817 | 744 |
| | | L | 2664 | 656 |

While for a lower number of AGVs a solution with two-way roads is very efficient, with increasing number of vehicles the coordination time is too high and other solutions enable higher throughput.

To check the consistency of the results, similar simulations have been carried out using the grid graph event-based simulator. Because the event-based simulator is much faster than the MAPF simulator, random task sequences can be simulated to avoid potential bias. For each scenario 20000 steps were simulated and the simulations were repeated 10 times to obtain the averages. The following metrics were observed: the number of tasks executed in a given time $N_{tasks}$ and the number of collisions between AGVs $N_{collisions}$. The results are shown in Table II. The best scenarios for each layout (in Table II denoted in bold) are identified based on the highest number of executed tasks $N_{tasks}$. One can see that the results are comparable to those obtained in the MAPF simulations, where the $N_{tasks}$ is analog to the $t_{fin}$ (there are some discrepancies for the $W_{real}$ and the the $W_{fict}$ layouts with 20 AGVs). Similarities can also be drawn between $N_{collisions}$ and $t_{coo}$. These results indicate that for a quick initial evaluation of the roadmap quality the event-based simulator can be used that is much faster than the MAPF simulator. For a more reliable evaluation, however, the MAPF simulator has to be used that enables more realistic simulations.

### B. Optimal Fleet Size

To determine the optimal fleet size $N_{opt}$ for an individual layout, throughput analysis was applied. $N_{opt}$ is estimated

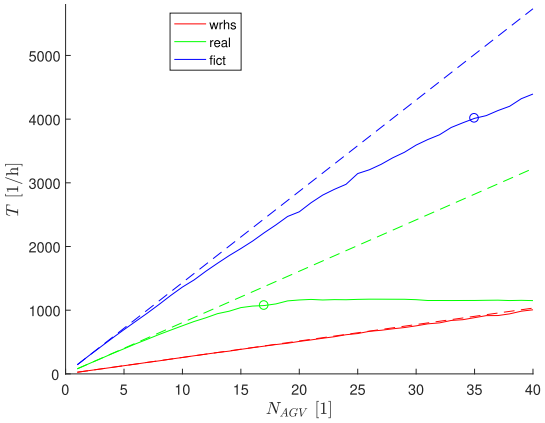



Fig. 7. Throughput analysis for the Scenario 1 of each layout. Solid lines are the obtained performances, dashed lines are ideal performances, and the circles mark a 20% drop of performance.

based on a 20% drop of throughput $T$ (tasks per hour) according to the ideal performance at increasing number of AGVs [25]. In Fig. 7, the throughput analysis is shown for the Scenario 1 of each layout. The solid lines represent the obtained performances, the dashed lines the ideal performances, and the circles mark the $N_{opt}$.

In Table III, a comparison of all scenarios for all layouts is shown: for each generated roadmap, the throughput for 10 AGVs $T_{10}$ and 20 AGVs $T_{20}$, the optimal fleet size $N_{opt}$, and the estimated optimal throughput $T_{opt}$ are given. The results show that for different roadmaps of the same layout, the optimal number of AGVs $N_{opt}$ is different. For example, for

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.

TABLE III

THROUGHPUT FOR 10 AND 20 AGVS AND ESTIMATE OF OPTIMAL NUMBER OF AGV AND THROUGHPUT FOR EACH ROADMAP

| | Scenario | $T_{10}$ [1/h] | $T_{20}$ [1/h] | $N_{opt}$ | $T_{opt}$ [1/h] |
|---|---|---|---|---|---|
| $W_{real}$ | 1 | 746 | 1054 | 17 | 1073 |
| | 2 | 691 | 942 | 17 | 1012 |
| | 3 | 745 | 1053 | 18 | 1124 |
| $W_{fict}$ | 1 | 1342 | 2469 | 35 | 4012 |
| | 2 | 1356 | 2444 | 35 | 3867 |
| | 3 | 1292 | 2361 | 40 | 4305 |
| $W_{wrhs}$ | 1 | 256 | 514 | 154 | 3339 |
| | 2 | 227 | 454 | 183 | 3506 |
| | 3 | 230 | 456 | 186 | 3556 |

the $W_{wrhs}$ layout, the optimal fleet size for different roadmaps varies between 154 and 186.

It can also be observed that one roadmap of the same layout performs best for a lower number of AGVs, while another roadmap performs better for a higher number of AGVs. In the case of $W_{fict}$ layout, Scenario 2 leads to the highest throughput when 10 AGVs are used, while Scenario 1 is best for 20 AGVs. Thus, based on the system requirements and resource availability, one can choose the best combination of roadmap and fleet size. For example, in the case of the $W_{fict}$ layout, if the highest throughput is desired and the efficiency of the AGVs is of secondary importance (there are enough AGVs available and their operating costs are of secondary importance), then Scenario 3 with $N_{opt} = 40$ should be chosen (see Table III). If, on the other hand, a fixed number of 20 AGVs is available, Scenario 1 should be used.

### C. Nonuniform Task Distribution

To further validate the approach and show that it enables optimization of the roadmap for specific characteristics of the intralogistic problem, a nonuniform task distribution is analysed for the $W_{fict}$ layout. To simulate the nonuniform tasks distribution, the tasks between the left station ($x = 3.9$ m, $y = 10.9$ m) and the right station ($x = 7.6$ m, $y = 5.8$ m) are assumed to be twice as frequent as between other stations. The solution obtained with the event-based optimizer, Scenario N, is shown in Fig. 10e (see Appendix). The MAPF simulator was used to evaluate the quality of the optimized solution.

The comparison of the optimized Scenario N with non-optimized scenarios, Scenario 1 and Scenario L from Subsection IV-A (scenarios for uniform task distribution), is shown in Table IV. Again, 1000 tasks have been simulated for both 10 and 20 AGVs. The results show that with the optimized roadmap the tasks are delivered much faster. Compared to the Scenario 1, $t_{fin}$ is shorter for 14% and 15% for 10 AGVs and 20 AGVs, respectively, while the improvement compared to the Scenario L is even greater. Additionally, both the coordination time $t_{coo}$ and the total distance travelled by AGVs $d_{tot}$ are significantly lower.

TABLE IV

A COMPARISON OF OPTIMIZED AND NONOPTIMIZED SCENARIOS FOR A NONUNIFORM TASK DISTRIBUTION

| | $N_{AGV}$ | Scenario | $t_{fin}$ [s] | $t_{coo}$ [s] | $d_{tot}$ [m] |
|---|---|---|---|---|---|
| $W_{fict}$ | 10 | 1 | 2573 | 2345 | 19439 |
| | | L | 2756 | 4652 | 17118 |
| | | N | **2207** | 2069 | 16013 |
| | 20 | 1 | 1865 | 10552 | 21111 |
| | | L | 2378 | 12553 | 17492 |
| | | N | **1582** | 9595 | 16987 |

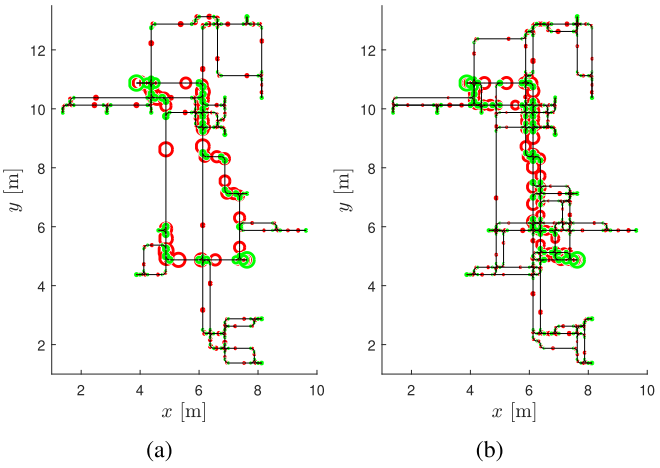



Fig. 8. Comparison of road (red circles) and intersection occupancy (green circles) for nonuniform task distribution. (a) Roadmap optimized for uniform task distribution (Scenario 1). (b) Roadmap optimized for nonuniform task distribution (Scenario N).

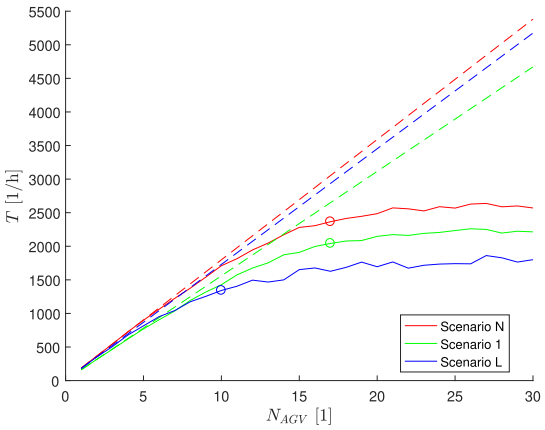



Fig. 9. Throughput analysis for a nonuniform task distribution. The solid lines are the achieved performances, the dashed lines are the ideal performances, and the circles mark the 20% drop of performance.

In Fig. 8, a visual comparison is shown between the non-optimized roadmap (Scenario 1, Fig. 8a) and the optimized roadmap (Scenario N, Fig. 8b). The size of the red and green circles indicates the occupancy of the roads and nodes, respectively. In the optimized roadmap, there are more vertical and horizontal connections for AGVs to choose between when delivering tasks. This facilitates coordination and consequently leads to a shorter coordination time and a shorter total path length.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.

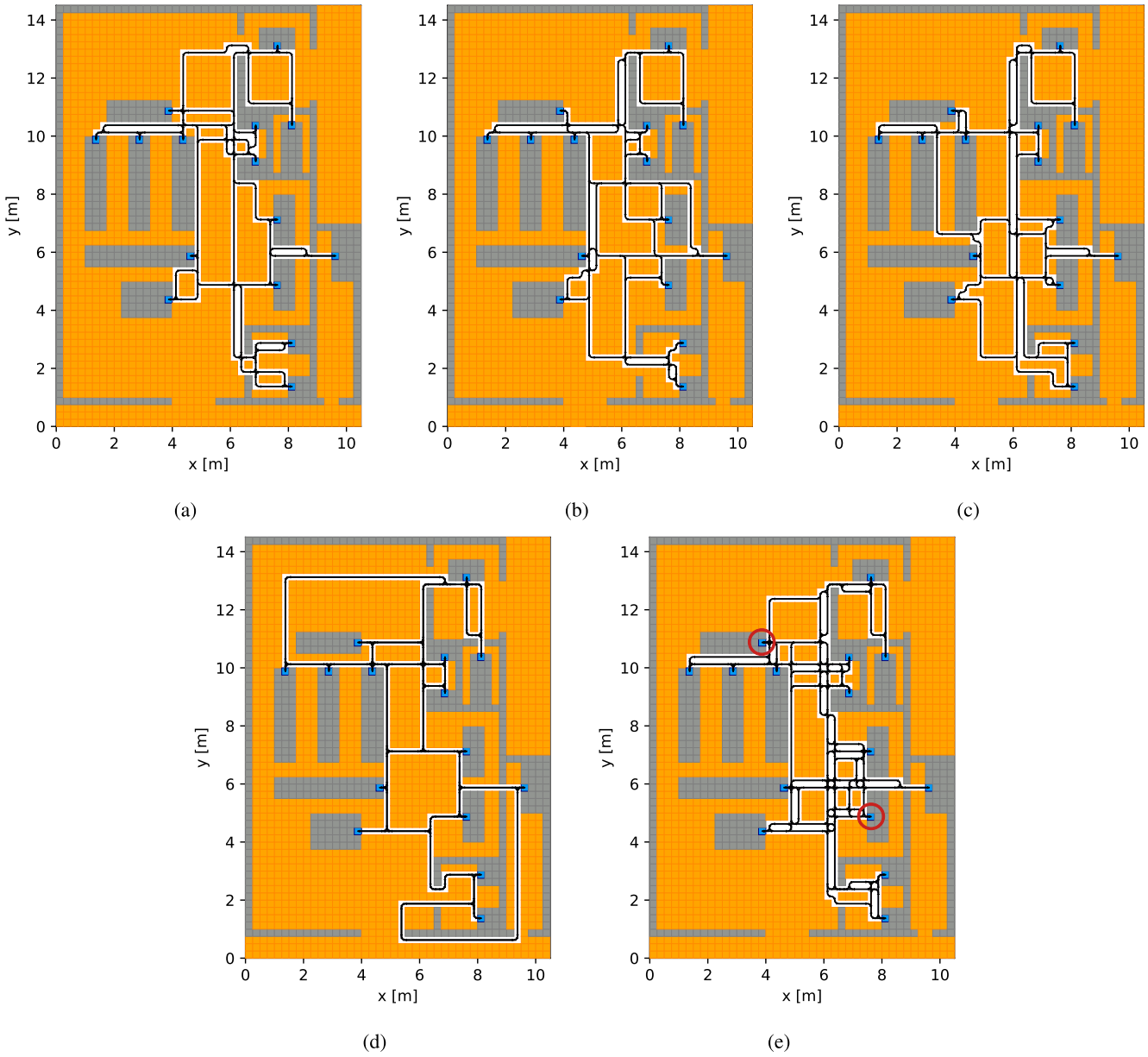



Fig. 10. Different solutions for a fictional layout $W_{fict}$: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3, (d) Scenario L from the literature [24], and (e) Scenario N for nonuniform task distribution. The frequency of the tasks is higher between the two stations marked with circles.

Fig. 9 shows the throughput analysis for nonuniform task distribution for all three scenarios. The results show that the highest performance is indeed obtained with the optimized roadmap (Scenario N). The analysis also showed that the optimal fleet size for this specific intralogistic problem is 17, which is much lower than in the case of uniform task distribution of the same layout, where optimal number of AGVs is 35 (see Table III). The main reason for this is in the increased traffic between the two stations which causes congestion to appear sooner, i.e., at a lower number of AGVs.

## V. Conclusion

In this paper, we proposed a simulation-based approach for automatic roadmap design in intralogistic systems. Our approach combines an ACO-inspired optimization algorithm and MAPF to generate, simulate, and evaluate roadmaps tailored to the specific characteristics of the intralogistic problem. Using discrete-time simulation to minimize conflicts and improve movement patterns, a weighted directed grid graph is first generated and transformed into a roadmap. Then, the MAPF simulator employing a SIPP-based prioritized path planner is used to evaluate the generated roadmaps in terms of throughput and coordination time. Compared to existing approaches, our methodology enables the generation of roadmaps that take into account not only the layout but also other characteristics of the system such as the fleet size, the statistical description of the tasks, and dispatching algorithm. The results show that taking all these characteristics into account from the beginning can significantly increase the efficiency of the system, leading to higher throughput

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



**IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.**

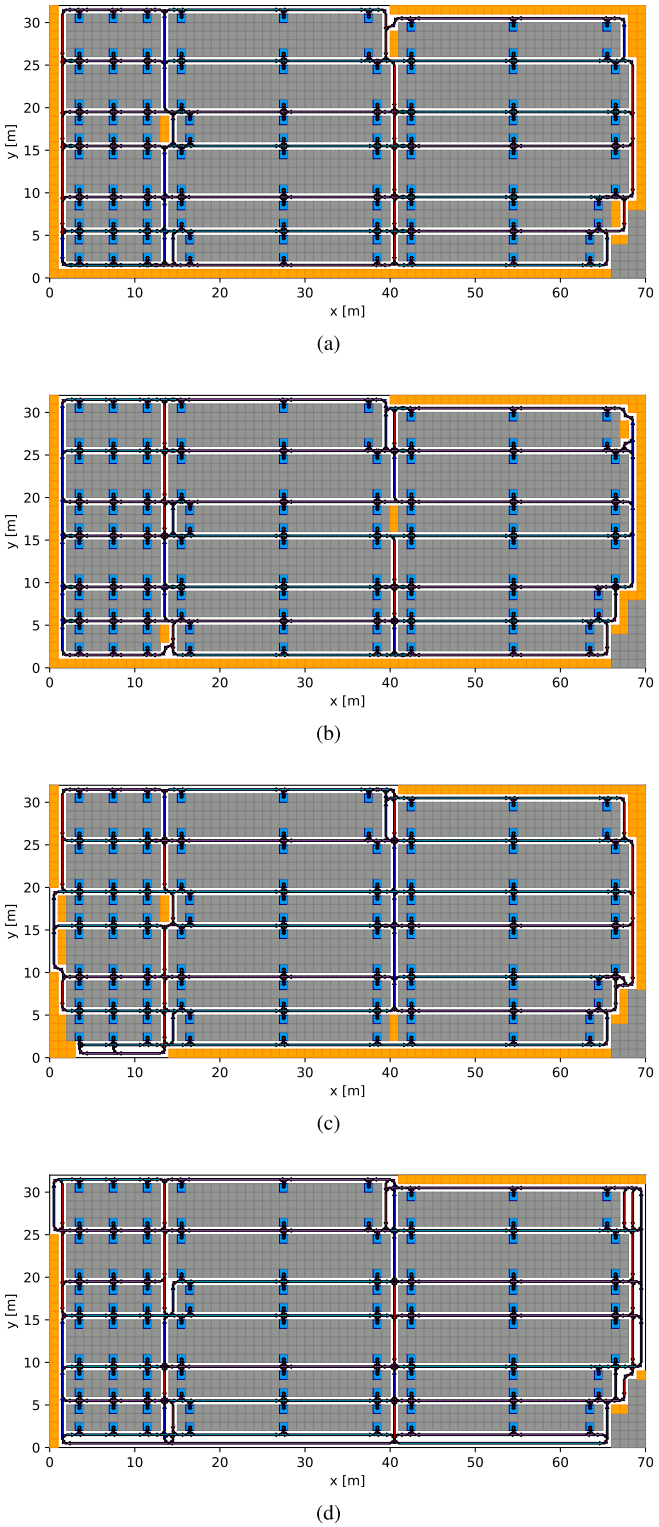



Fig. 11. Different solutions for a typical warehouse layout $W_{wrhs}$: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3, and (d) Scenario L from the literature [12].

compared to the solutions proposed in the literature (from 2% to 15%).

One of the main advantages of our approach is its speed. Using the proposed optimization algorithm, different solutions can be generated in a few seconds for a simple layout and in a few minutes for a more complex layout. Combining the proposed approach with the MAPF simulator and analyzing the system throughput, the optimal fleet size for each intralogistic problem can also be quickly determined.

Despite the promising results, our approach does have some limitations. Firstly, the resulting movement patterns depend on the optimization parameters, which have to be selected in advance by human experts. Second, the solution also depends on the initialization, which currently incorporates some randomness. In the future, this randomness will be mitigated by adapting the optimization algorithm to ensure that the solution converges to the same weighted grid graph, and thus the same roadmap, when using the same input parameters.

It is worth emphasizing that the solution to the roadmap design problem depends on multiple interconnected parameters. If any of them change (e.g., the number of AGVs, the MAPF algorithm, etc.), the solution changes as well. It should also be noted that the analyses were performed for the case where there is no idle time - a task is always assigned to an AGV. The management of the idle time will be a part of future work. Our future work will also address the effectiveness of other optimization algorithms for generating movement constraints (e.g., using reinforcement learning) and exploring different MAPF algorithms. The proposed approach will also be extended to consider the battery level of the vehicles and the positioning of the charging stations, which will allow the system to use resources more efficiently.

## APPENDIX
## SOLUTIONS OBTAINED BY GRID-GRAPH OPTIMIZER

### A. A Fictional Layout

In Fig. 10, different solutions for a fictional layout $W_{fict}$ are presented. Compared to solution 1 (Fig. 10a), solution 2 (Fig. 10b) was obtained using fewer AGVs but with shorter $t_{due}$ and lower evaporation rate $\lambda_{evp}$. In solution 3 (Fig. 10c), initial cell value $w_{init}$ was additionally lowered. The solution from the literature [24] is shown in Fig. 10d. In Fig. 10e, the solution for a nonuniform task distribution is shown.

### B. A Typical Warehouse Layout

In Fig. 11, different solutions for a typical industrial warehouse $W_{wrhs}$ are presented. Compared to solution 1 (Fig. 11a), solution 2 (Fig. 11b) was obtained using less AGVs but with shorter $t_{due}$, lower evaporation rate $\lambda_{evp}$ and evaporation value $w_{evp}$. In solution 3 (Fig. 11c), initial cell value $w_{init}$ was additionally lowered. In Fig. 11d, the solution proposed in the literature is shown [12].

## REFERENCES

[1] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "An automatic approach for the generation of the roadmap for multi-AGV systems in an industrial environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1736–1741.

[2] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1748–1758, Oct. 2020.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.






[3] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Springer Tracts in Advanced Robotics*. Berlin, Germany: Springer, 2004, pp. 43–57.

[4] J. P. van den Berg, D. Nieuwenhuisen, L. Jaillet, and M. H. Overmars, "Creating robust roadmaps for motion planning in changing environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 2415–2421.

[5] S. Kumar and S. Chakravorty, "Multi-agent generalized probabilistic RoadMaps: MAGPRM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3747–3753.

[6] A. A. Neto, D. G. Macharet, and M. F. M. Campos, "Multi-agent rapidly-exploring pseudo-random tree," *J. Intell. Robotic Syst.*, vol. 89, nos. 1–2, pp. 69–85, Jan. 2018.

[7] A. Kleiner, D. Sun, and D. Meyer-Delius, "ARMO: Adaptive road map optimization for large robot teams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 3276–3282.

[8] D. Kozjek, A. Malus, and R. Vrabič, "Reinforcement-learning-based route generation for heavy-traffic autonomous mobile robot systems," *Sensors*, vol. 21, no. 14, p. 4809, Jul. 2021.

[9] J. Kim and K. Kim, "Optimizing large-scale fleet management on a road network using multi-agent deep reinforcement learning with graph neural network," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 990–995.

[10] M. R. Jansen and N. R. Sturtevant, "Direction maps for cooperative pathfinding," in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, vol. 4, 2021, pp. 185–190.

[11] C. Henkel and M. Toussaint, "Optimized directed roadmap graph for multi-agent path finding using stochastic gradient descent," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 776–783.

[12] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Ensemble coordination approach in multi-AGV systems applied to industrial warehouses," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 922–934, Jul. 2015.

[13] F. Pratissoli, N. Battilani, C. Fantuzzi, and L. Sabattini, "Hierarchical and flexible traffic management of multi-AGV systems applied to industrial environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 10009–10015.

[14] P. Beinschob, M. Meyer, C. Reinke, V. Digani, C. Secchi, and L. Sabattini, "Semi-automated map creation for fast deployment of AGV fleets in modern logistics," *Robot. Auto. Syst.*, vol. 87, pp. 281–295, Jan. 2017.

[15] J. Stenzel and L. Schmitz, "Automated roadmap graph creation and MAPF benchmarking for large AGV fleets," in *Proc. 8th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2022, pp. 146–153.

[16] S. Uttendorf, B. Eilert, and L. Overmeyer, "Combining a fuzzy inference system with an A* algorithm for the automated generation of roadmaps for automated guided vehicles," *At Automatisierungstechnik*, vol. 65, no. 3, pp. 189–197, Mar. 2017.

[17] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[18] A. Kulatunga, D. Liu, G. Dissanayake, and S. B. Siyambalapitiya, "Ant colony optimization based simultaneous task allocation and path planning of autonomous vehicles," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Jun. 2006, pp. 1–6.

[19] P. Bedi, N. Mediratta, S. Dhand, R. Sharma, and A. Singhal, "Avoiding traffic jam using ant colony optimization–A novel approach," in *Proc. Int. Conf. Comput. Intell. Multimedia Appl. (ICCIMA)*, Dec. 2007, pp. 61–67.

[20] B. Sobota, C. Szabo, and J. Perhac, "Using path-finding algorithms of graph theory for route-searching in geographical information systems," in *Proc. 6th Int. Symp. Intell. Syst. Informat.*, Sep. 2008, pp. 1–6.

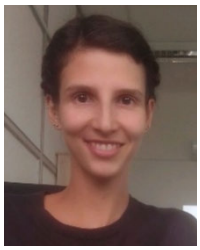
[21] M. Phillips and M. Likhachev, "SIPP: Safe interval path planning for dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 5628–5635.

[22] A. Andreychuk, K. Yakovlev, P. Surynek, D. Atzmon, and R. Stern, "Multi-agent pathfinding with continuous time," *Artif. Intell.*, vol. 305, Apr. 2022, Art. no. 103662.

[23] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artif. Intell.*, vol. 219, pp. 40–66, Feb. 2015.

[24] S. Uttendorf, B. Eilert, and L. Overmeyer, "A fuzzy logic expert system for the automated generation of roadmaps for automated guided vehicle systems," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2016, pp. 977–981.

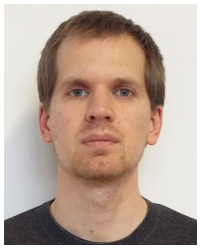
[25] S. Rajotia, K. Shanker, and J. L. Batra, "Determination of optimal AGV fleet size for an FMS," *Int. J. Prod. Res.*, vol. 36, no. 5, pp. 1177–1198, May 1998.

**Tena Žužek** (Member, IEEE) received the Ph.D. degree in mechanical engineering from the Faculty of Mechanical Engineering, University of Ljubljana, in 2022. She is currently a Research Assistant with the Laboratory for Mechatronics, Production Systems and Automation, University of Ljubljana. Her main research interests include autonomous mobile robots, fleet management, multi-agent simulations, and reinforcement learning.

**Rok Vrabič** (Member, IEEE) received the Ph.D. degree in 2012. He is currently an Assistant Professor with the Faculty of Mechanical Engineering, University of Ljubljana, and the Research Lead of the Laboratory for Mechatronics, Production Systems, and Automation. He teaches courses on robotic systems and mechatronics. His research interests include interplay between robotics and artificial intelligence. He is an Associate Member of the International Academy for Production Engineering CIRP.

**Andrej Zdešar** received the Ph.D. degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 2015. He is currently a Teaching Assistant with the Laboratory of Control Systems and Cybernetics. His research interests include autonomous mobile systems, visual servoing, machine vision, trajectory tracking control, model predictive control, probabilistic state estimation, and fuzzy systems.



**Gašper Škulj** received the Ph.D. degree from the Faculty of Mechanical Engineering, University of Ljubljana, in 2016. He is currently an Assistant Professor with the Laboratory for Mechatronics, Production Systems, and Automation, Faculty of Mechanical Engineering, University of Ljubljana. His research interests include mechatronic systems, robotics, distributed systems, and advanced production systems, with emphasis on self-organization.

**Igor Banfi** is currently pursuing the M.Sc. degree with the Faculty of Mechanical Engineering, University of Ljubljana. During his studies, he participated in several student competitions, including the AIAA DBF Competition and the Renesas MCU Car Rally. He is also a Backend Developer with Epilog d.o.o. His research interests include fleet management, multi-agent simulations, and optimization modeling.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



**IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.**

**Matevž Bošnak** received the Ph.D. degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 2013. He is currently pursuing the Ph.D. degree with the Laboratory of Control Systems and Cybernetics. He is also an Assistant Professor with the Laboratory of Control Systems and Cybernetics. His research interests include autonomous mobile systems, the modeling of dynamic systems, embedded systems, sensor data fusion, indoor localization, and system control.

**Viktor Zaletelj** received the Ph.D. degree from the Faculty of Mechanical Engineering, University of Ljubljana, in 2008. He is currently the Head of Robotics with Epilog d.o.o., Ljubljana, Slovenia. His research interests include intralogistics in advanced manufacturing systems, adaptive manufacturing systems modeling, and industrial computing system design.

**Gregor Klančar** received the Ph.D. degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 2003. He is currently an Associate Professor with the Faculty of Electrical Engineering, University of Ljubljana. His research interests include autonomous mobile robots, motion control, trajectory tracking, path planning, localization, and agent-based behavior systems.

**IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2024, Yokohama, Japan.**