Autonomous Topological Optimisation for Multi-robot Systems in Logistics

Zuyuan Zhu
Gautham P. Das
Marc Hanheide
{zzhu,gdas,mhanheide}@lincoln.ac.uk
Lincoln Centre for Autonomous Systems (L-CAS), University of Lincoln
Lincoln, Lincolnshire, UK

ABSTRACT

Multi-robot systems (MRS) are currently being introduced in many in-field logistics operations in large environments such as warehouses and commercial soft-fruit production. Collision avoidance is a critical problem in MRS as it may introduce deadlocks during the motion planning. In this work, a discretised topological map representation is used for low-cost route planning of individual robots as well as to easily switch the navigation actions depending on the constraints in the environment. However, this topological map could also have bottlenecks which leads to deadlocks and low transportation efficiency when used for an MRS. In this paper, we propose a resource container based Request-Release-Interrupt (RRI) algorithm that constrains each topological node with a capacity of one entity and therefore helps to avoid collisions and detect deadlocks. Furthermore, we integrate a Genetic Algorithm (GA) with Discrete Event Simulation (DES) for optimising the topological map to reduce deadlocks and improve transportation efficiency in logistics tasks. Performance analysis of the proposed algorithms are conducted after running a set of simulations with multiple robots and different maps. The results validate the effectiveness of our algorithms.

CCS CONCEPTS

• **Computing methodologies** → **Simulation evaluation**; **Machine learning**; **Multi-agent planning**;

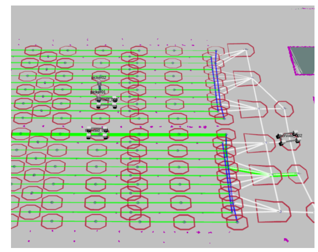
KEYWORDS

robot traffic planning, multi-robot systems, agri-robotics, topological optimisation, discrete event simulation, genetic algorithm

ACM Reference Format:
Zuyuan Zhu, Gautham P. Das, and Marc Hanheide. 2023. Autonomous Topological Optimisation for Multi-robot Systems in Logistics. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27 - March 31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3555776.3577666

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SAC '23, March 27 - March 31, 2023, Tallinn, Estonia
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/10.1145/3555776.3577666

(a) Real robot



(b) Simulated robots

Figure 1: Practical and simulated environments of the logistic robots used in strawberry farm.

1 INTRODUCTION

The agri-food industry is the UK's largest manufacturing sector, twice the size of the automotive and aerospace industries combined [11]. It is also a truly international industry that employs 3.9 million people and underpins the Gross Value Added (GVA) of £113 billion. Although there is widespread agreement that robotics will transform the food and agriculture industry in the next few years, much research and development is still needed. Deployment of a robotic fleet in a farming system have the potential to integrate multiple autonomous robotic and machine learning applications into commercial food production, which can improve the current heavily human dependent practices. This will also form a critical foundation for the transformation of food supply chains.

In navigation of multiple robots sharing an environment (also known as multi-robot path planning, MRPP), deadlock avoidance is a crucial problem since navigation deadlocks degrade the performance of the MRS by introducing extra time delay or completely blocking the navigation and task execution of the involved robots in the worst case scenario. However, deadlocks are usually difficult to predict due to the existence of higher-order deadlocks, especially when many robots moving around in a given environment with cooperation.

In this paper, deployment of a robotic fleet for logistics operations in a commercial strawberry production facility, consisting of polytunnels with raised tables of plants, to aid human pickers and improve their productivity is considered. This environment is convenient for human workers for tasks such as picking and crop-care operations. A topological map abstracting the topological constraints of the environment into a discretised graph representation of nodes and edges is used for the route planning of robots

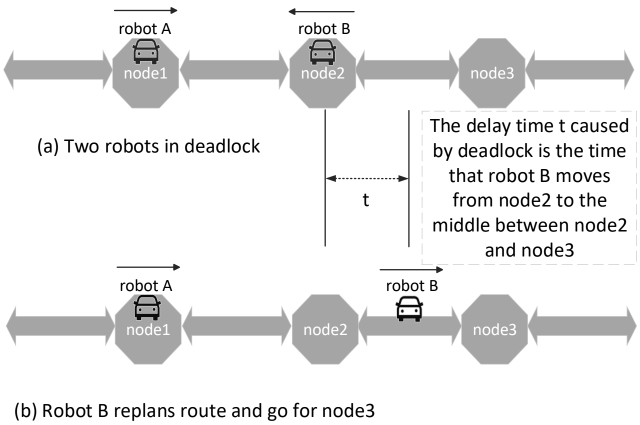


Figure 2: Time delay caused by deadlock in a multi-robot system.

in this work. However, autonomous robotic navigation in this environment is challenging as the rows of plants are usually long (in 100s of metres) and narrow without a chance for the robot to get around one another, see Fig 1. Therefore, collision avoidance and deadlock resolution are critical to ensure the robots navigating efficiently in the environment and executing on-demand logistics tasks. Towards this, a reactive deadlock resolution algorithm, named Request-Release-Interrupt (RRI) algorithm is proposed as the first contribution in this paper.

While deploying a robotic fleet into such an environment, the primary objective is to ensure the fleet works autonomously in the existing environment. However, further optimisation in the autonomous operation of the fleet is possible by identifying and addressing the navigation bottlenecks and identifying changes in the topology. With a well-designed topological map, deadlocks could be reduced and robot's travelling routes could also be shortened, thereby improving the efficiency of the MRS. A GA based autonomous topology optimisation to address this is the second contribution in this paper.

2 RELATED WORK

2.1 Deadlock Avoidance

A *deadlock* in multi-robot path planning is a situation in which no member of the group can proceed as each entity waits for others to take action. Usually, this is caused by robots blocking each other's routes, and the coordinator or route planner cannot find a solution for the robots to avoid each other, even though the solution exists. When a deadlock occurs in a multi-robot system, delay to the overall system will be inevitable due to resolving the deadlock. The delay time t is the time that one of the robot used to make away as shown in Figure 2. In logistics systems with multiple robots, deadlocks are almost inevitable in such long autonomy tasks. Based on the planning strategies used in the system, the methods to resolve deadlocks are quite different.

2.1.1 Centralised Strategies. In a centralised strategy, one coordinator plans the entire route for all robots. A key advantage of the centralised approach is the capability to find optimal paths for all robots [20, 22]. To use the centralised approach, the coordinator must know the current states of all the robots as well as the desired goal states in order to generate conflict-free or deadlock-free routes for each robot.

To guarantee conflict-free or deadlock-free routes, the central planner needs to progressively refine routes of individual robots using several constraint solvers to get multiple sets of constraints on routes [21], thereby increasing the computational complexity. Moreover, in order to obtain deadlock-free routes for all the robots, some robots may have to wait much longer to let the other conflicting robots pass through the critical points, resulting in slow task execution times.

However, centralised planners still have the common issue on scale when increasing the number of agents. Meanwhile, planning for large scale of agents simultaneously usually requires a large amount of computational resources.

2.1.2 Decentralised Strategies. In a distributed strategy, every agent has its own route planner and system configurations are usually not shared to among all agents. Agents can communicate and negotiate independently with each other so the decision-making is decentralised, which allows the multi-agent systems to deal with dynamic environments and moving obstacles [2]. With some combined heuristics to give priority to the robot which needs the path most, the decentralised approach can find sub-optimal solutions, which can also be complete and scalable [14]. As the agents have limited information about the nearby environment, reservation-based procedure is a good way to avoid conflicts and deadlocks. The early work of [10] proposes a first-come first-serve reservation principle shared by all agents when crossing a crossroad.

When it comes to large-scale and complex systems, decentralised strategy is necessary to reduce the computational complexity [8]. However, decentralised approaches usually only have local information or limited states of the neighbouring agents, so the solution usually is suboptimal instead of globally optimal [12].

2.1.3 Task Coordinator. Due to obstacles like traffic jams or deadlocks, the shortest path in logistics systems with multiple robots does not always correspond to the shortest journey time. To find the shortest route with conflict-free or deadlock-free, the coordinator needs to use mathematical modeling [25] (like Lagrangian relaxation [19]), combinatorial scheduling [5, 13, 18], or fuzzy logic [16]. Shortest time or travelling distance is not the only objective to be optimised; constraints like agents utilisation, loading, unloading, queueing, energy cost and so on are also targets to be optimised by the coordinator [9, 19, 24, 25].

In this paper, we use a centralised strategy for initial route planning with reservations on the current node occupied by the robot. However, if a deadlock happens, the involved robots use a decentralised strategy for route replanning based on the information of the neighbour topological nodes and nearby robots' states. The proposed algorithm aims to detect and resolve deadlocks in logistics tasks.

2.2 Topological Optimisation

Topological optimisation has been widely studied in the literature as the performance of tasks such as localisation and navigation depends closely on the topological map [4]. To improve the navigation

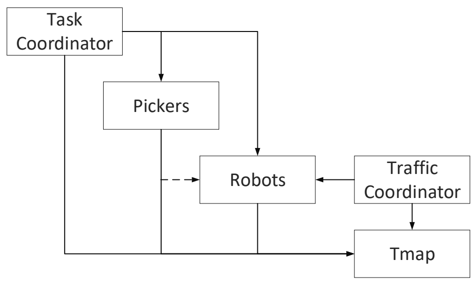


Figure 3: Multi-robot control system. *Task Coordinator* allocates tasks to pickers and robots. *Pickers* performs picking task. *Robots* performs logistics task. *Tmap* processes the navigation and environment information of the topological map. *Traffic Coordinator* coordinates the deadlocks and manages the access of storage resource. Solid arrow from module A to module B means module A has full access to module B and dashed arrow means only read access.

efficiency, firstly the robotic system needs an efficient localisation method, which can be achieved by using image clusters based on appearance similarities [23] or laser scanners [3].

In the environment like large corridor-based buildings, the three-dimensional problem can be transformed into a one-dimensional simple problem [15]. While in the situation where the robots have limited field of view, i.e., the sensing interactions are asymmetric, optimal topologies that yield stable coordination of multi-robot systems are needed [17].

In the logistics field, where robots need to go through long and narrow rows for transporting, there are many time slots that could be shortened like the time pikers has to wait for a robot to come after calling the robot and the time the fruits are kept outside. In this paper, we use a Genetic Algorithm (GA) to optimise the topological locations where the robots should be waiting to improve the robotic traffic. The proposed algorithm aims to improve the overall transport efficiency of MRS in logistics tasks.

3 METHODOLOGY

Our multi-agent system consists of 5 main parts: ***Pickers***, ***Robots***, ***Tmap***, ***Task Coordinator***, and ***Traffic Coordinator***, as shown in Fig. 3. ***Pickers*** and ***Robots*** are agents who perform picking tasks and logistics tasks respectively. The ***Task Coordinator*** has the highest information level, controlling the two agents: ***Pickers*** and ***Robots***, as well as the topological map environment ***Tmap***. The pickers are assigned the tables to pick fruits from. Whenever a picker needs a robot to transport the picked fruits, the picker makes a call which is registered in the ***Task Coordinator***. The ***Task Coordinator*** will assign an idle robot to the picker. All the agents' routes are planned by A* algorithm on a ***Tmap*** (topological map) which is shared by all agents and the two coordinators. All the topological nodes, including the storage node, are modelled as resource objects which require reservation before an agent can access them. The capacity of all topological nodes is set as 1 in this work. When robots have route conflicts with each other, the ***Traffic Coordinator*** interrupts one of the robot to replan using RRI algorithm (see Section 3.2).

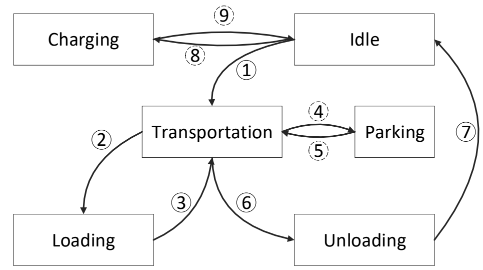


Figure 4: State diagram of a robot assistant: (1) Go to picker; (2) Loading from picker; (3) Go to base; (4) Start parking; (5) Go to storage; (6) Unloading at storage; (7) Idle if no new call from picker; (8) Charging battery; (9) Back to idle. The dashed steps only happens in certain conditions: (8) and (9) happen when battery is low; (4) and (5) happen when the storage is occupied by other robots.

This multi-agent system is simulated using a Discrete Event Simulation (DES) framework which models the operations of the system as a sequence of events in time [7]. WIth the DES as the simulation framework with low computational complexity, a GA based Topology Optimisation (GATO) is proposed to modify the topological map to optimise the picking and logistics process (see Section 3.3).

3.1 Robot states

In our simulation scenario, a robot is modeled as an agent with six states, as presented in Figure 4. In the simulation of picking and transporting tasks collaborated with pickers, the robot keeps switching between these six states in sequence. When one state is triggered by an event, the robot goes into the next state. For example, the robot starts at state *Idle* and goes to state *Transportation* when the picker makes a call and the ***Task Coordinator*** allocates the task to the robot, which is represented as transition 1 in Figure 4. Generally, a complete cycle of robot states consist of at least 4 states (*Idle*, *Transportation*, *Loading*, *Unloading*) and 5 transitions (1, 2, 3, 6, 7). Transition 8, 9 and state *Charging* will be performed when the robot's battery is lower than a fixed threshold (here, 40%). Transition 4, 5 and state *Parking* will be triggered when the *storage* is occupied at the moment when the robot trying to use the storage resource.

3.2 Request-Release-Interrupt (RRI) Algorithm

The flowchart in Fig 5 shows the coordinator planning path for a robot and coordinating the traffic among all robots in the RRI algorithm. When a robot has to navigate from its current location (start node) to a target location (goal node), an initial path is planned using the A* algorithm, without considering the locations of the other agents in the environment.

3.2.1 Request and Release. A robot first estimates the *occupy_time* that how long the robot plans to occupy the next node in its path. The robot must send a request to the topological node resource container with the *occupy_time* (and join the queue if the node already holds one entity/agent) before reserving and start navigating to the node. In normal operation without any conflicts, when the robot leaves, the node is released and becomes available to upcoming robots. Each robot will be using its local navigation algorithm to navigate from one node to the next. When the upcoming node from the planned route is reserved or occupied by other robots, the robot pauses its navigation and replans the route (to get a *new_route*) considering the other robot as an obstacle. Then it will decide whether to wait or use the *new_route* (when it's livelock) based on the *occupy_time* of the robot which holds the reservation to the blocked node.

3.2.2 Interrupt. If two or more robots stuck in deadlock (see Fig 6), the deadlock coordinator will rank the involved robots based on their dodging priorities and *interrupts* one of the deadlocked robot (dashed lines in Fig 5). The dodging priority is the priority of tasks they are performing. For example, a robot has the highest priority to pass through when heading to or backing from storage station and no neighbouring nodes to move to. In this case, the other robots must replan and use a different route. If the dodging priorities of all robots are the same, other factors such as the robots' task stages, number of neighbouring nodes, waiting time, and queue position can also be considered. As a result, the robots with lower priorities will be interrupted and forced to replan, thereby clearing the deadlocked path for the highest priority robot to use.

3.3 Genetic Algorithm based Topological Optimisation (GATO) Algorithm

When a mobile robotic fleet has to be introduced to an environment, the easiest option is to develop the robot navigation autonomy without modifying the topology of the environment. However, the performance of the MRS will be suboptimal, as the topology was not designed with robot-operations as a design criteria. On the other end of the spectrum is building the environment topology from the ground up, making it optimal for the robots to work. This, however, is challenging and expensive. An alternate, between the two ends is to make minor modifications to the topology so that the efficiency of the fleet operation is improved. Even so, finding the exact set of changes to adapt the topology from the possible modifications is still challenging.

Towards this, we propose a Genetic Algorithm based Topological Optimisation (GATO) algorithm for MRS in logistics tasks. In the GATO algorithm, a GA iterative approach is used to identify high-quality topological modifications. In order to check the quality (fitness) of the topology modifications (candidate solutions), the DES of picking and logistics operations is used as computationally low-cost fitness function. For brevity and clarity, we limit the goal of the GATO algorithm to search for the best locations of base nodes within a specified area, see the green area in Fig. 7a. However, more complex topology modifications can be solved using the GATO algorithm.

The pseudocode of the GATO algorithm is given in Algorithm 1, where the termination condition is decided by a fixed number of iterations. When generating the initial population, each individual is a list of one dimension list with a fixed number of genes (topological nodes). With each individual, we run DES to simulate the picking and transporting tasks, returning a result consisted of the simulation time and the deadlock number. Then the result is multiplied by a negative 2×1 weight to get the fitness of the individual as in (1).

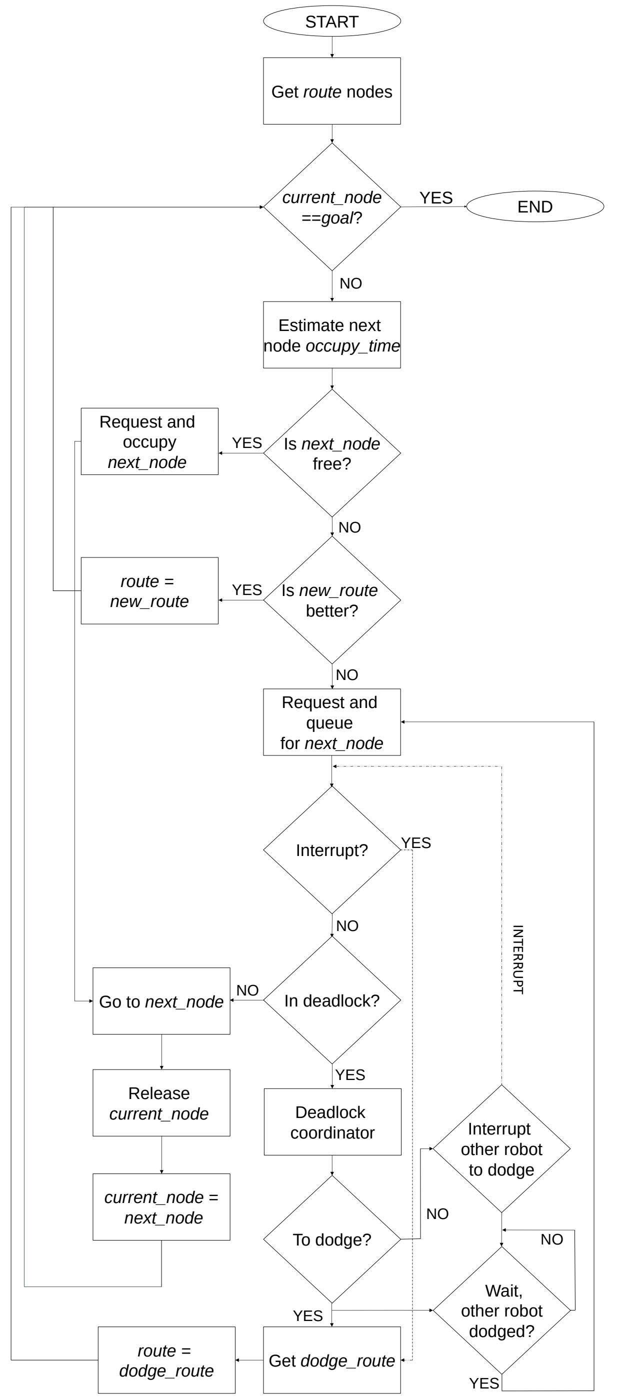


Figure 5: Flowchart of RRI algorithm for route traversing in multi-robot systems.

$$f = \begin{bmatrix} t_{sim} & n_{dl} \end{bmatrix} \times \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T \quad (1)$$

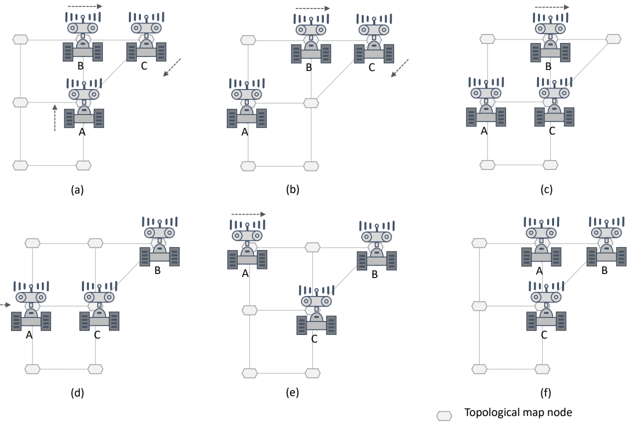


Figure 6: Schematic of three robots in deadlock (a) and stages (b-f) of resolving deadlock by making robot A step aside.

Algorithm 1 GATO

$t \leftarrow 0; t \in N$
$GenerateInitialPopulation(P(t))$
while $iteration < allowedIteration$ **do**
  $evaluateInDES(P(t))$
  $P'(t) \leftarrow selectBestIndividual(P(t))$
  $P"(t) \leftarrow selectPopForCrossover(P'(t))$
  $P(t+1) \leftarrow mutatePopulation(P"(t))$
  $t \leftarrow t+1$
end while
return $fittestMemeberOfPopulation(P)$

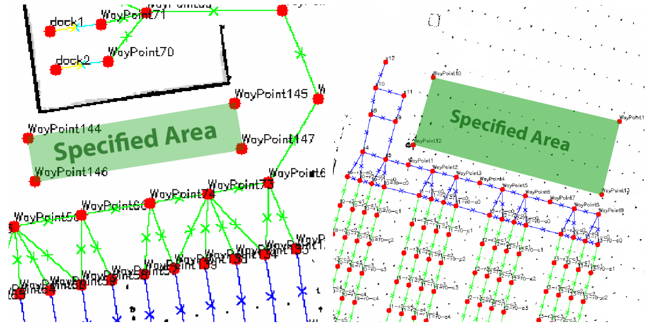
where f is fitness, t_{sim} is the simulation time, n_{dl} is the number of deadlock, $[w_1\ w_2]$ is the multi-objectives optimisation weight vector, which is negative as we want to minimise the simulation time and deadlock number. In the mutation or crossover process, the topological nodes of the individual are changed or exchanged.

4 EVALUATION

We used GATO for automatic topological optimisations in two strawberry farms: a small trial farm with two polytunnels of approximately 25m in length (referred later as Riseholme map) and another with four polytunnels of approximately 130m in length in a commercial production field (referred later as Clockhouse map). Within these environments, GATO is used to find the best locations in a pre-defined area for robot base nodes where the robots wait for new tasks when idle. The DES model simulates the full picking process, and is used here to evaluate the operational model corresponding to the topology modifications at fleet scale. For each DES simulation, Riseholme map takes about 3 seconds, Clockhouse map takes about 7 seconds. The parameters used for the DES are given in Table 1, refer to [7] for details of the DES.

4.1 Experimental setup

In this implementation, the GATO algorithm minimises the simulation time and deadlock numbers as defined in formula (1), both values normalised to [0, 1]. It should also be noted that other metrics



(a) Riseholme map specified area **(b) Clockhouse map specified area**

Figure 7: Specified rectangle area (green) for adding nodes as base stations.

Table 1: DES parameters.

Parameter	Value
picker_picking_rate	[0.4, 0.02]m/s
picker_transportation_rate	[1.0, 0.04]m/s
picker_max_n_trays	4
picker_car_n_trays	2
picker_unloading_time	[10.0, 0.2]s
tray_capacity	1200g
n_robots	2/4/6/8
robot_transportation_rate	1.0m/s
robot_max_n_trays	4
robot_unloading_time	10.0s
base_station_nodes	2/4/6/8
n_polytunnels	2/4
row_nodes	8 * 12/45 * 16
yield_per_node	[200.0, 5.0]

could also be used as the fitness value depending on the target we aim to optimise. For example, metrics such as robot travel distance, picker waiting time, and robot parking time are other possible fitness values for the picking and logistics process considered here. The other parameters of GA are presented in Table 2.

The experiments were validated with the Riseholme map with 168 nodes and the Clockhouse map with 743 nodes. As shown in Fig. 7, green areas are specified for adding extra topological nodes used for waiting nodes. The Risehome map is to be modified with the addition of four new nodes: *WayPoint144*, *WayPoint145*, *WayPoint146*, and *WayPoint147*, as shown in Fig. 7a. Similarly, the Clockhouse map is to be modified by the addition of four new nodes: *WayPoint10*, *WayPoint11*, *WayPoint12*, and *WayPoint13*, as shown in Fig. 7b. In both these maps, individuals of the initial population are created by GA as the location of four base nodes randomly placed within the specified area with constraint that the distance between any generated node is not less than a threshold σ_r from the nodes already created. In this paper, σ_r is 1.0m. Also, each newly added node is connected to its nearest three nodes including the existed nodes and the newly generated nodes.

Parameter	Value	Note
N_POP	128	the number of initial individual population
N_GEN	20/50	the number of generations to run
INDPB	0.05	the probability of mutate each attribute/gene of the individual
N_BASE	6	the number of base that each individual consists
CXPB	0.5	the probability with which two individuals are crossed
MUTPB	0.10	the probability for mutating an individual
N_TOU	10	the parameter for selecting individuals for breeding the next generation: each individual of the current generation is replaced by the 'fittest' (best) of N_TOU individuals drawn randomly from the current generation
WEIGHTS	[-1.0, -4.0]	$[w_1\ w_2]$ in formula (1), used to vary the importance of each objective one against another, a minimizing fitness is built using negatives weights

Table 2: GA parameters.

Table 3: Comparison of simulation time (defined in equation 1) between RRI and Fragment Planner with the Riseholme map.

Scenario	Time (s)
picker=2, robot=2, Fragment Planner	1328.8 ±77.4
picker=2, robot=2, **RRI**	**1267.7 ±54.5**

4.2 Experimental results

Initially, the robustness of the RRI algorithm for MRPP and deadlock resolution is evaluated with the original Riseholme map (without the new nodes) and by comparing against the performance with another MRPP algorithm named Fragment Planner [6]. The Fragment Planner algorithm is deadlock-free by ensuring all except one robot are blocked until the robot which is not blocked passes through the overlapping section of the route. The testing plan was to run the simulations for different number of pickers and robots for both MRPP algorithms. However, the Fragment Planner resulted in much larger delays in the logistics process when the number of pickers and robots were increased. So the comparison is limited to one test case with two pickers and two robots. Table 3 shows that RRI contributes to shorter completion time than the Fragment Planner.

Further evaluations of the GOTA algorithm were run only with the RRI algorithm. Table 4 indicates that the proposed RRI algorithm is robust enough on deadlock resolving. In these evaluations, the first test started with 2 pickers and 2 robots, performed all 8916 completed trials in all GOTA iterations, reaching a success rate of 100%. Within these simulations, the RRI achieved 1569202 route planning goals and resolved 20827 deadlocks. As the number of picker and robot increasing, the number of deadlocks increased dramatically from 20827 to 132572 for 8 pickers and 8 robots. However, the success rate only drops by 0.86%, from 100% to 99.14%.

Fig. 8a shows that the GA starts converging from the 5th generation with the *fitness* dropping to (0.32), which is very fast. The metrics analysed on the robot's performance are defined as the following:

(1) Service distance: The route distance between the robot's location when being called and the position when arriving picker's location.
(2) Delivery distance: The route distance between the picker's location and the position when arriving at the drop-off point (storage).
(3) Service time: The time from the robot being called by a picker to the time that the robot arriving at the picker's location.
(4) Delivery time: The time from the picker finishing loading to the time that the robot arriving at the drop-off point.
(5) Service speed: The average speed that the robot travelling from current location when being called to the goal location, i.e., picker's location.
(6) Delivery speed: The average speed that the robot travelling from picker's location to the drop-off point.
(7) Work time: the sum time of transporting,loading, and unloading.
(8) Rest time: the sum time of waiting at the base, including parking and idle.
(9) Waiting time: the sum time of parking at the base.

From Fig. 8a top left *Service Distance* we can find that both the actual delivery distance and service distance are longer than the corresponding shortest distance. That is because the robot has to replan often due to route conflicts with other robots. The gap between actual and shortest delivery distance is much bigger than the gap between the actual and shortest service distance means that the robot usually have to wait at the base. *Work and Rest Time* at the bottom left proves that the robot spend no time at the base for waiting the calls from pickers. From Fig. 8a top right *Service Time* we know that the actual (~610s) service time is much longer than the shortest (~380s) service time to picker, which means that there is long delays or frequent route replan happening. While the actual delivery time (~750s) is much longer than the shortest delivery time (~450s) due to long waiting time for using the storage.

From Fig. 8a bottom left *Service Speed* we know that the robot's max service speed and max delivery speed are all equal to the preset value (1.0m/s), but the actual service speed (~0.78m/s) and actual delivery speed (~0.68m/s) are much lower. Again, this is due to the long waiting time during delivery.

From Fig. 8a bottom right *Work and Rest Time* we know that the robot actually works a long time (~1800s) and never in idle mode (0s). But spend a lot of time on waiting as shown in rest time (~200s). It means there is room to improve the utilisation rate of robots.

Table 4: Results of RRI on deadlock resolving

Scenario	Goals Success	Goals Fail	Deadlocks	Trials	Success rate
picker=2, robot=2	1569202	0	20827	8916	100.0%
picker=4, robot=4	1420544	5	42158	9012	99.94%
picker=6, robot=6	1317200	12	50116	9051	99.87%
picker=8, robot=8	1293508	76	132572	8841	99.14%

Fig. 8a shows the best result from each generation. In general, the values (simulation time, deadlock number and fitness) are converging towards steady values that are same level as the best values from Fig. 8a. Though the simulation time and deadlock numbers vibrate through generations, the trend of the their combined fitness values indicates that the overall performance of individuals are improving owing to the evolutionary processes of GA which keeps selecting the fittest individuals for reproduction to produce offspring of the next generation.

Fig. 8b shows the heatmap of the best individuals for the Riseholme map. The rectangle block represents the topological node located by coordinate x and y. The heatmap value shows how fit the individual is. 1.0 means the node as a base station has the $Min(simulation\_time + n\_deadlock)$ while 0.0 means the max of $simulation\_time + n\_deadlock$. The figure indicates that the best individuals are not unique. As one node is not best individual with some nodes but could be best when combined with other nodes. It should be noted that even if a gene of the individual (i.e., base node) has 0.00 fitness this generation, next generation it could have higher fitness if mixed up with other genes. It explains why sometimes we don't see the minimum valid genes that have fitness of 0.01. The figure gives us an indicate of where the best individual could be. To find out the best individual all over the generations, Fig. 8c presents one of the best individual who consists of 3 topological nodes. The position of these 3 nodes are presented in Fig. 8d.

Fig. 9 shows the metric evolutionary process of the GA with the Clockhouse map. Similarly, the GA starts converging towards a stable solution after 4 generations by dropping to around 0.35. Compared to Fig. 8a, Fig. 9a has a bigger vibration due to the Clockhouse map being much larger than the Riseholme map and introduces more uncertainties.

Table 5: Results of GATO compared with default methods

Scenario	Time (s)	Deadlocks
picker=4, robot=0	5155.6 ± 112.7	-
picker=4, robot=4, Head	4038.2 ±134.5	9.9
picker=4, robot=4, **GATO**	**3979.5 ±83.5**	**3.9**

Table 5 compares the results of GATO with default method on the Clockhouse map. All scenarios run 10 times with random Gaussian noise to imitate picking speed variations. The first scenario does not use robots for logistics task, the pickers perform picking and transporting. We can see it costs the longest simulation time (defined in equation 1) for picking and transporting. The second scenario uses 4 pickers for picking and same number of robots for transporting. The base stations are set at the head of the rows by default. The third scenario uses same pickers and robots as second scenario, but the base stations are generated by GATO within the specified area. We find that the GATO achieves best result on minimum simulation time (3979.5s) and minimum deadlocks (3.9). Also, the minimum STD (83.5) indicates that the simulation time is more stable than other scenarios.

5 CONCLUSIONS AND FUTURE WORKS

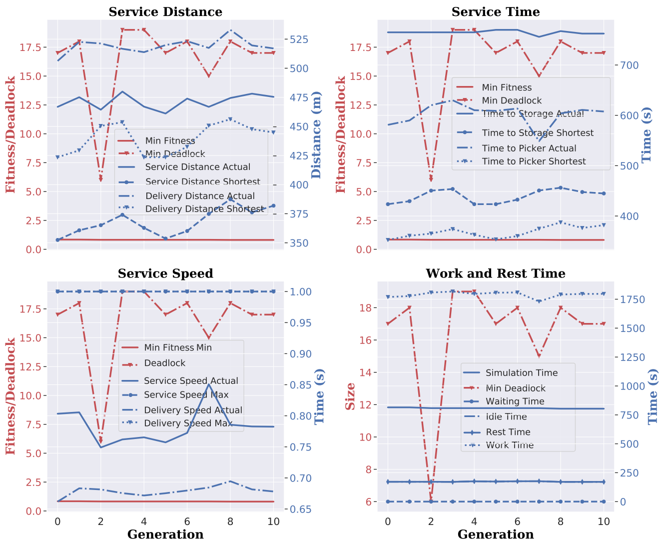
This paper investigates the use of distributed coordination approach, based on on-demand reservation table, for deadlock resolving in MRS. Furthermore, this paper implemented an algorithm for automatic topological optimisation for multi-robot systems in logistics, based on GA. Specifically, we consider the deadlock resolving problem related to soft fruit logistic operations, where robots must transport soft fruit from picker's loading place to drop-off point while minimising deadlocks and overall operating time. Crucially, we devise a specific RRI algorithm for our deadlock resolving problem, where waiting time and new route are compared for a better planning. This allows us to use the RRI approach to efficiently solve our deadlock resolving and collision avoidance problem. Moreover, we integrate the RRI algorithm along with GA for automatically optimising the topological map to improve the efficiency of multi-robot systems in logistics. Specifically, we specify a rectangle area for randomly generate new base stations for robot to wait while the storage node is occupied. A group of topological nodes are encoded as a GA individual and fittest individuals are selected for reproduction in order to produce offspring of the next generation. Then we get a best individual among all the individuals which is also a good quality solution for the topological map.

Finally, we empirically evaluate our RRI and GATO in discrete event simulations with random environments, comparing with manually optimised topological map. Results show that RRI provides superior performances in deadlock resolving and GATO is good in finding high quality solution for multi-robot systems in logistics.

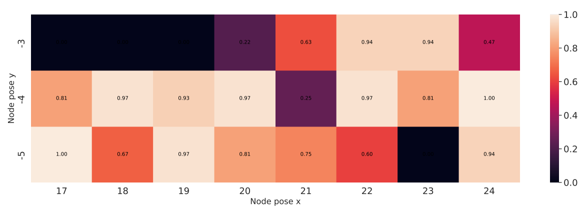
We believe that our work takes a first important step towards the use of automatic topological optimisation method for MRS in logistic scenarios, opening up a novel promising direction for MRS coordination in industrial domains, where existing works use GA for optimise task assignment [1]. In the future, we plan to consider the real-time distribution of deadlocks by adding penalty to those nodes where deadlocks often happen, as shown in Figure 11.

ACKNOWLEDGEMENTS

This paper was developed from the Robot Highways project funded by InnovateUK as part of the ISCF TFP Science and Technology into Practice: Demonstration call (grant number 51367).

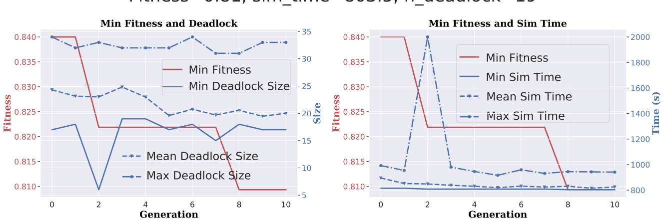


(a) Metrics analysed on the robot's performance: distance, time, speed.

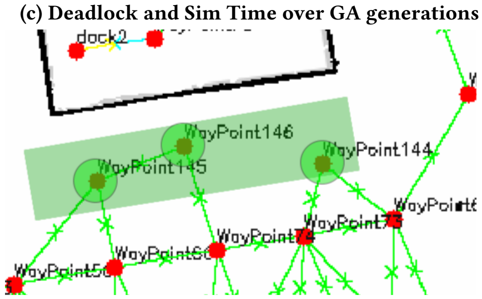


(b) Base nodes heatmap shows the location of the added base stations and their fitness. The value 1.00 means the node has minimum fitness, i.e., the sim time is small and the deadlocks are few.

Best Individual=[{'y': -3.7, 'x': 24.2},{'y': -4.2, 'x': 17.7}, {'y': -3.2, 'x': 20.2}]
Fitness=0.81, sim_time=803.5, n_deadlock=19

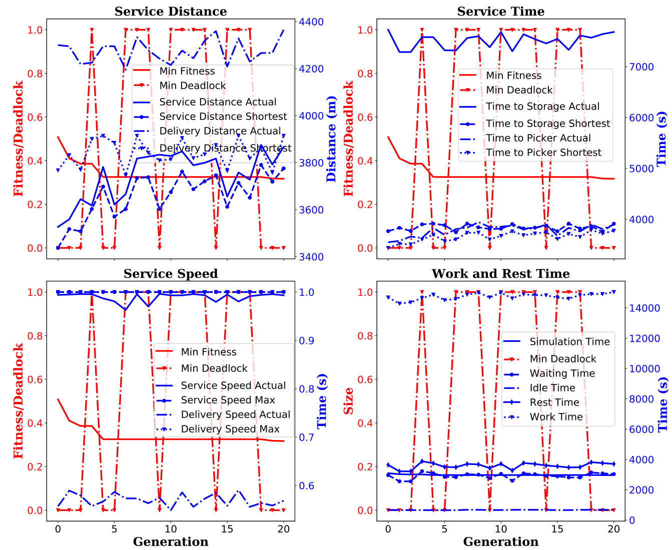


(c) Deadlock and Sim Time over GA generations

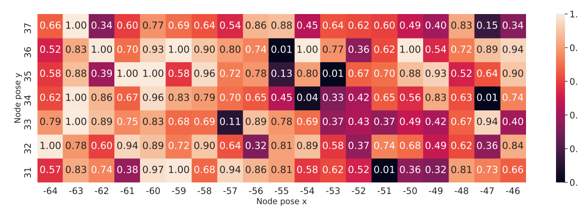


(d) Visualisation of the best base nodes generated by GA

Figure 8: Autonomous topological optimisation for minimising sim time and deadlocks by adding nodes in the specified area generated by GA and evaluated by DES with 3 robots on Riseholme map.

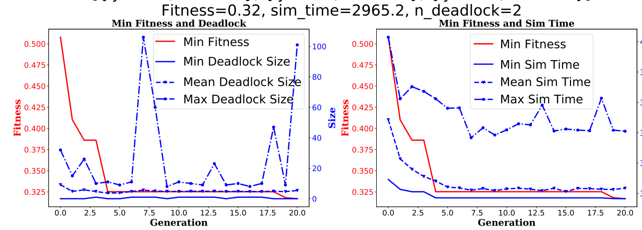


(a) Metrics analysed on the robot's performance: distance, time, speed.

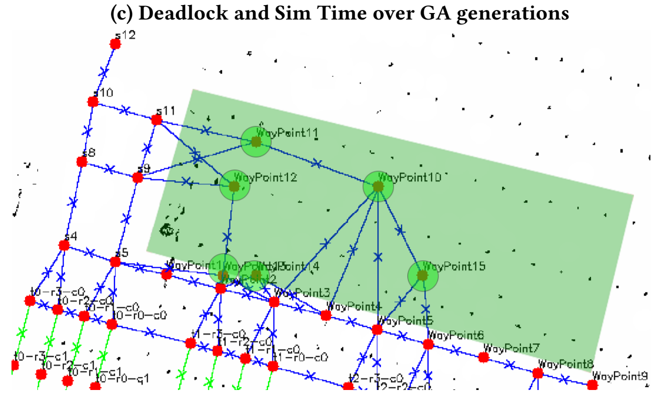


(b) Base nodes heatmap shows the location of the added base stations and their fitness. The value 1.00 means the node has minimum fitness, i.e., the sim time is small and the deadlocks are few.

Best Individual=[{'y': 35.7, 'x': -60.9}, {'y': 33.7, 'x': -62.9}, {'y': 36.7, 'x': -53.9}]
[{'y': 35.2, 'x': -59.4}, {'y': 36.2, 'x': -58.9}, {'y': 30.7, 'x': -62.4}]
Fitness=0.32, sim_time=2965.2, n_deadlock=2



(c) Deadlock and Sim Time over GA generations



(d) Visualisation of the best base nodes generated by GA

Figure 9: Autonomous topological optimisation for minimising sim time and deadlocks by adding nodes in the specified area generated by GA and evaluated by DES with 6 robots on Clockhouse map.

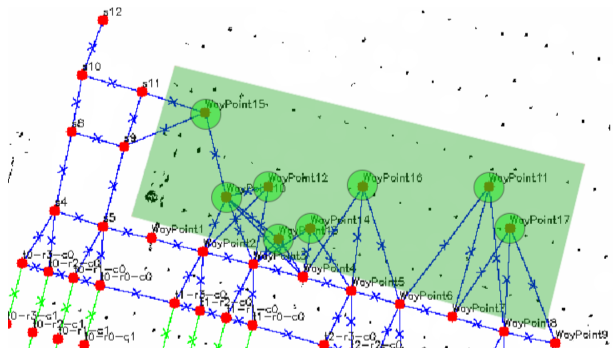


Figure 10: Visualisation of the best base nodes generated by GA and evaluated by DES with 8 robots on Clockhouse map.

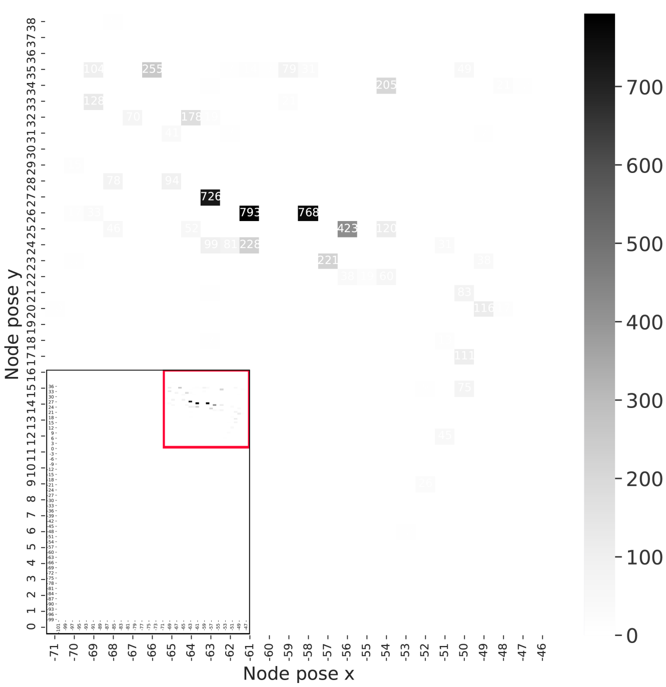


Figure 11: Heatmap distribution of the accumulated deadlocks over the GA generations with the Clockhouse map, showing most of the deadlocks locate in the head lanes and the entrance of storage, especially the junction area between them (large number and dark colour). The mini heatmap at the left corner is of the full map and the red area is where the deadlocks often happen and is enlarged.

REFERENCES

[1] Xiaoshan Bai, Ming Cao, Weisheng Yan, and Shuzhi Sam Ge. 2020. Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles. *IEEE Trans. Autom. Sci. Eng.* 17, 1 (2020), 248–260.

[2] Subhrajit Bhattacharya and Vijay Kumar. 2011. Distributed optimization with pairwise constraints and its application to multi-robot path planning. *Robot. Sci. Syst. VI* 177 (2011).

[3] A. Binch, G.P. Das, J.P. Fentanes, and M. Hanheide. 2020. Context Dependant Iterative Parameter Optimisation for Robust Robot Navigation. In *Proc. 2020 IEEE Int. Conf. Robot. Autom.*

[4] O Booij, B Terwijn, Z Zivkovic, and B Krose. 2007. Navigation using an appearance based topological map. In *Proc. 2007 IEEE Int. Conf. Robot. Autom.* 3927–3932.

[5] Ayoub Insa Corréa, André Langevin, and Louis-Martin Rousseau. 2007. Scheduling and routing of automated guided vehicles: A hybrid approach. *Comput. & Oper. Res.* 34, 6 (2007), 1688–1707.

[6] Gautham Das, Grzegorz Cielniak, Francesco Del Duchetto, Zuyuan Zhu, James Heselden, Johann Dichtl, Marc Hanheide, Simon Pearson, Jaime Pulido Fentanes, Adam Binch, Michael Hutchinson, and Pal From. 2022. A Unified Topological Representation for Robotic Fleets in Agricultural Applications. *[Manuscript Submitt. Publ.* (2022).

[7] Gautham P. Das, Grzegorz Cielniak, Johan From, and Marc Hanheide. 2018. Discrete Event Simulations for Scalability Analysis of Robotic In-Field Logistics in Agriculture – A Case Study. In *ICRA 2018 Work. Robot. Vis. Action Agric.* Brisbane.

[8] M De Ryck, M Versteyhe, and F Debrouwere. 2020. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *J. Manuf. Syst.* 54 (2020), 152–173.

[9] Ivica Draganjac, Tamara Petrović, Damjan Miklić, Zdenko Kovačić, and Juraj Oršulić. 2020. Highly-scalable traffic management of autonomous industrial transportation systems. *Robot. Comput. Integr. Manuf.* 63 (2020), 101915.

[10] Kurt Dresner and Peter Stone. 2004. Multiagent traffic management: A reservation-based intersection control mechanism. In *Auton. Agents Multiagent Syst. Int. Jt. Conf.*, Vol. 3. IEEE Computer Society, 530–537.

[11] Tom Duckett, Simon Pearson, Simon Blackmore, Bruce Grieve, Wen-Hua Chen, Grzegorz Cielniak, Jason Cleaversmith, Jian Dai, Steve Davis, Charles Fox, and Others. 2018. Agricultural robotics: the future of robotic agriculture. *arXiv Prepr. arXiv1806.06762* (2018).

[12] Giuseppe Fragapane, René de Koster, Fabio Sgarbossa, and Jan Ola Strandhagen. 2021. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* 294, 2 (2021), 405–426.

[13] Amir Hossein Gharehgozli, René de Koster, and Rick Jansen. 2017. Collaborative solutions for inter terminal transport. *Int. J. Prod. Res.* 55, 21 (2017), 6527–6546.

[14] James R Heselden and Gautham P Das. 2021. CRH*: A Deadlock Free Framework for Scalable Prioritised Path Planning in Multi-robot Systems. In *Towar. Auton. Robot. Syst.*, Charles Fox, Junfeng Gao, Amir Ghalamzan Esfahani, Mini Saaj, Marc Hanheide, and Simon Parsons (Eds.). Springer International Publishing, Cham, 66–75.

[15] Reza Javanmard Alitappeh, Guilherme AS Pereira, Arthur R Araújo, and Luciano CA Pimenta. 2017. Multi-robot deployment using topological maps. *J. Intell. & Robot. Syst.* 86, 3 (2017), 641–661.

[16] O A Joseph and R Sridharan. 2011. Evaluation of routing flexibility of a flexible manufacturing system using simulation modelling and analysis. *Int. J. Adv. Manuf. Technol.* 56, 1 (2011), 273–289.

[17] Pratik Mukherjee, Matteo Santilli, Andrea Gasparri, and Ryan K Williams. 2020. Optimal Topology Selection for Stable Coordination of Asymmetrically Interacting Multi-Robot Systems. In *2020 IEEE Int. Conf. Robot. Autom.* 6668–6674.

[18] Tatsushi Nishi, Yuichiro Hiranaka, and Ignacio E Grossmann. 2011. A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Comput. & Oper. Res.* 38, 5 (2011), 876–888.

[19] Tatsushi Nishi, Kenichi Shimatani, and Masahiro Inuiguchi. 2009. Decomposition of Petri nets and Lagrangian relaxation for solving routing problems for AGVs. *Int. J. Prod. Res.* 47, 14 (2009), 3957–3977.

[20] Roberto Olmi, Cristian Secchi, and Cesare Fantuzzi. 2008. Coordination of multiple AGVs in an industrial application. In *2008 IEEE Int. Conf. Robot. Autom.* 1916–1921.

[21] Federico Pecora, Marcello Cirillo, and Dimitar Dimitrov. 2012. On mission-dependent coordination of multiple vehicles under spatial and temporal constraints. In *2012 IEEE/RSJ Int. Conf. Intell. Robot. Syst.* IEEE, 5262–5269.

[22] Cristian Secchi, Roberto Olmi, Fabio Rocchi, and Cesare Fantuzzi. 2015. A dynamic routing strategy for the traffic control of AGVs in automatic warehouses. In *2015 IEEE Int. Conf. Robot. Autom.* IEEE, 3292–3297.

[23] Aleš Štimec, Matjaž Jogan, and Aleš Leonardis. 2008. Unsupervised learning of a hierarcht of topological maps using omnidirectional images. *Int. J. Pattern Recognit. Artif. Intell.* 22, 04 (2008), 639–665.

[24] Altan Yalcin, Achim Koberstein, and Kai-Oliver Schocke. 2019. An optimal and a heuristic algorithm for the single-item retrieval problem in puzzle-based storage systems with multiple escorts. *Int. J. Prod. Res.* 57, 1 (2019), 143–165.

[25] Yongsheng Yang, Meisu Zhong, Yasser Dessouky, and Octavian Postolache. 2018. An integrated scheduling method for AGV routing in automated container terminals. *Comput. & Ind. Eng.* 126 (2018), 482–493.