



HAL
open science

Collective obstacle avoidance strategy - an agent-based simulation approach

Juliette Grosset, Alain-Jérôme Fougères, Moïse Djoko-Kouam, Jean-Marie Bonnin

► **To cite this version:**

Juliette Grosset, Alain-Jérôme Fougères, Moïse Djoko-Kouam, Jean-Marie Bonnin. Collective obstacle avoidance strategy - an agent-based simulation approach. ASPAI 2022: 4th International Conference on Advances in Signal Processing and Artificial Intelligence, Oct 2022, Corfu, Greece. hal-03958445

HAL Id: hal-03958445

<https://imt-atlantique.hal.science/hal-03958445>

Submitted on 26 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Type of Presentation:

Oral: In-person:
Poster: Virtual in Zoom:
The same:

Topic: Applied Artificial Intelligence

Collective obstacle avoidance strategy – an agent-based simulation approach

J. Grosset^{1,2}, **A.-J. Fougères**², **M. Djoko-Kouam**^{2,3} and **J.-M. Bonnin**¹

¹ IMT Atlantique, IRISA, Rennes, France

² ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

³ IETR, CentraleSupélec, Rennes, France

E-mail: juliette.grosset@ecam-rennes.fr

Summary: The context of factory 4.0 leads more and more to decentralised solutions, as centralisation shows its limits. One of the research areas of Industry 4.0 is the use of autonomous guided vehicles (*AGVs*), autonomous industrial vehicles (*AIVs*). We want to show that cooperation is useful and necessary to increase their autonomy. We propose in this paper an agent model to test scenarios in Industry 4.0 environments with a fleet of *AIVs*. In addition, we are interested in the resolution of global obstacle avoidance by *AIVs* with a collective strategy. The results of the simulation will be evaluated by performance indicators such as distance and time in order to compare the different proposed approaches.

Keywords: Agent modelling, cooperative strategy, simulation, industry 4.0, autonomous industrial vehicles, obstacle avoidance

1. Introduction

One of the fundamental axes of Industry 4.0 is the search for decentralised solutions in the decision-making process because centralisation quickly shows its limits [1]. Indeed, if an industry is centrally controlled, then if a technical failure, e.g., a network failure occurs, everything will come to a standstill. Decentralised solutions allow for more flexible control, which is preferable.

The use of autonomous guided vehicles (*AGVs*) or autonomous industrial vehicles (*AIVs*) is a strategic research area in the context of Factory 4.0 and Logistic 4.0. Currently, vehicles have autonomy over their navigation linked to rails, physical or virtual markers that allow them to get to their goal. They follow predetermined trajectories and they are autonomous in their navigation after having received an order from a central, a supervisor. If they detect a problem, or an obstacle they can report the information to the central, but it is the control centre that will decide for the other vehicles.

Our point of view to improve the autonomy of *AIVs* is to move towards decentralised solutions with collective strategies taken by the fleet of vehicles. To increase their intelligence and their individual and collective decision-making, they need to become more cooperative [2, 3, 4]. Thus, the *AIVs* will be more efficient in performing the five core tasks they have to control (task allocation, localisation, path planning, motion planning and vehicle management) [5].

Thus, they could carry out their missions with a collective strategy. This translates for example into a fleet of *AIVs* that communicate and share information. In the first level of strategy, there is the shared perception of their vision of the environment. A higher level that would achieve a collective global strategy for

the realisation of missions would be the allocation, scheduling and distribution of tasks between them in real-time. The aim is for robots to be more efficient, waste less time and use less energy to accomplish all their tasks.

Our objective is to improve the *AIV* autonomy integrated into a fleet based on collective intelligent strategies. Among the problems to be solved to make *AIVs* more autonomous, we can particularly identify 1) the avoidance of collisions between vehicles or with static or dynamic obstacles [6, 7], and 2) the path planning or path finding when an obstacle hinders the passage of *AIVs* or obstructs a predefined path [8, 9, 10].

The problem of avoidance is an individual problem that is addressed in a more global and collective way. There are three phases: perception of the obstacle (detection), rerouting or trajectory planification (avoidance) and the overall strategy. In the following, we will deal with avoidance through the choice of the route, the path but not the trajectory. This chosen path can be costly when an aisle is completely obstructed. This can be solved by enhanced cooperation [11, 12], i.e., the ability of autonomous industrial vehicles (*AIVs*) to exchange relevant information with each other on their traffic conditions.

This is how we augmented the Bannes' algorithm [11], in a previous work [13], to take into account obstacle avoidance, in addition to communication for avoidance at intersections.

Solutions are needed for the second problem, especially in the case where an obstacle completely obstructs a lane. This prevents a mission from not being completed because the main route defined by an *AIV* is compromised by an obstruction. Path planning or path-finding algorithms [9, 14] make it possible to

re-plan their route, depending on whether the target of the mission has been reached or not.

In this paper, we will propose a global obstacle avoidance with shared perception. Indeed, we will not use a local avoidance solution to improve autonomy. But rather a collaboration between the different vehicles of the fleet to move towards the basis of a global collective strategy in our future work. The proposed approach is adaptable and flexible for centralised or decentralised solutions. In other words, we are dealing with a distributed system where each robot decide by itself, augmented with communication between vehicles to share information about the environment. The communication may also go through a central supervisor before being redistributed.

The paper is organised as follows: section 2 describes the implementation of a collective strategy consequent to the detection with agent modelling, the exchange of messages to cooperate and the different approaches we will compare. The description of the simulation environment, the implementation of the

experiments and the results are analysed in section 3. A discussion on the cooperative infrastructure and how fuzzy logic can help to qualify the obstacles are also made in section 3. Finally, conclusions and future work are presented in section 8.

2. Collective strategy consequent to the detection

2.1. Modelling agent

In order to validate the use of collective strategy algorithms to increase the autonomy of *AIVs*, we rely on a methodological framework consisting of 3 phases: (1) agent modelling, (2) verify in simulations the correct performance of the algorithm, and (3) experiment with TurtleBot3 Burger robots in perspective.

We propose an agent model in Fig. 1 that adapts to different situations and scenarios that are tested in the context of Industry 4.0.

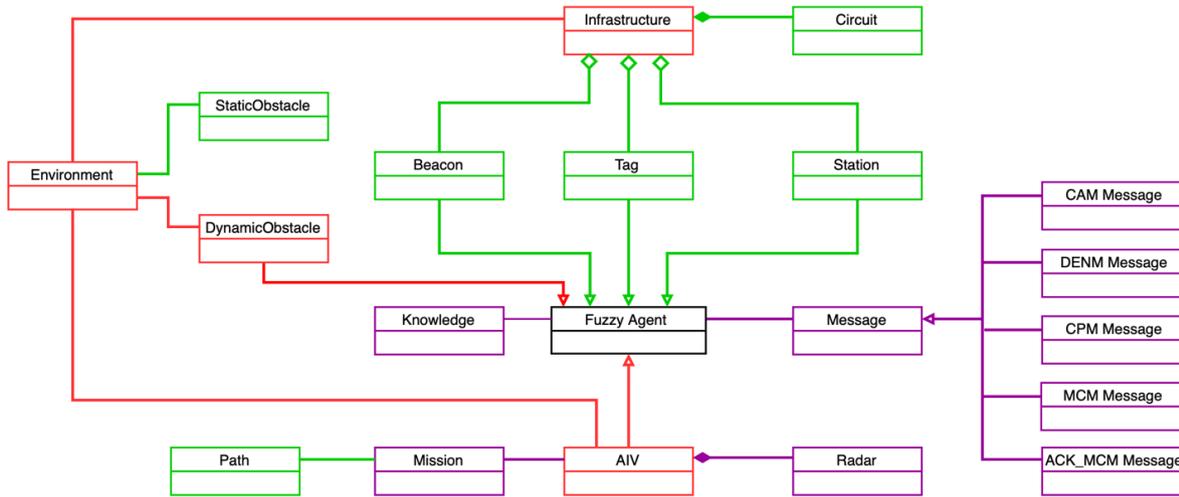


Fig. 1. Modelling agents adaptable to different scenarios for Industry 4.0

Infrastructure is deployed in the environment and is composed of a traffic plan, and active elements such as beacons, tags and stations. Since they are active, these elements of infrastructure are modelled as software agents. These agents are defined as fuzzy agents because they can take fuzzy decisions from uncertain knowledge or perception, but these possibilities are not illustrated in this paper.

Industrial vehicles, which are also modelled as autonomous agents, carry out missions defined by routes and sent by a supervisor of the environment. They are equipped with radar and thanks to their knowledge of the environment they can avoid collisions with static obstacles (goods in the aisles), or dynamic obstacles (other vehicles or operators). *AIVs* can also cooperate by communicating together. We

choose to use different types of standardised messages transposed from ETSI for ITS.

2.2. Exchanging messages to cooperate

The solutions proposed in the literature make it possible to solve an avoidance problem locally. In order to be able to carry out all the missions assigned to the vehicles, it is necessary that they cooperate and share information on their perception of the environment. Indeed, if an obstacle is perceived by a vehicle, the sending of a *Cooperation Perception Message* (CPM) (ETSI TR 103 562 standard [15]) makes it possible to warn and help the other vehicles to make decisions about their route. This means that if an unavoidable obstacle is perceived on its route, it can recalculate and plan a new route to be able to complete its task.

2.3. Obstacle avoidance approaches

We propose to study three approaches to obstacle avoidance by *AIVs* that will be tested in the simulation. Each robot has knowledge of the environment, i.e., the position of aisles, intersections, *AIVs* stations, and mission points. These various important traffic points of the circuit will be called nodes in the following, and are identified by respective numbers shown in Fig. 3.

In the first approach, agents do not have the ability to re-plan a route and change their route when faced with an obstacle.

In the second approach, vehicles are able to change routes and plan a new route when an obstacle is detected, but they do not cooperate with other vehicles. This means that vehicles will enter routes where obstacles are present and will have to change their route, even though another vehicle has already perceived this. They are able to calculate the path to their destination.

The path planning algorithm chosen for these experiments is Dijkstra. The objective is not to optimise a path planning algorithm (each *AIV* agent having to execute this algorithm), but to use an efficient algorithm to measure the performance of a collective strategy compared to an individual approach in carrying out missions in an industrial environment.

The last approach will be a collective strategy upon detection of an obstacle. This strategy is established according to two points of view:

- The vehicle detects an obstacle, which determines the level of obstruction. Then, it communicates the description (position, level of obstruction, type – static or dynamic) of the obstacle to the other vehicles. It can react according to its static or dynamic type to avoid it if possible, or it can replan another path to

reach its objective and accomplish its mission. This algorithm is described in Fig. 2: algorithm 1.

- The vehicle receives information about an obstacle in the environment. It first tries to determine whether the obstacle is on its path to accomplishing its mission. If this is the case and the obstacle is obstructing its planned path, it can replan its route. If the obstacle can be avoided, it can act and anticipate this static or dynamic obstacle by arriving in the vicinity of the received position of the obstacle. This improves the safety of the area and prevents accidents. This algorithm is described in Fig. 3: algorithm 2.

3. Simulation results and discussion

3.1. Environment simulation

The environment chosen to test the different approaches is the warehouse presented by Bechtsis *et al.* [16]. The directions of circulation have been adapted for our experiments and are identified in Fig. 3. We work with five *AIVs* corresponding to the five parking spaces available in the environment.

AIVs must perform missions in the environment shown in Fig. 3. A mission consists of travelling from its car park to a storage point and back to its car park. Thus, the *AIVs* applying the collective intelligent strategy have two route calculations to perform with the Dijkstra algorithm. The first one is to calculate the shortest path to the storage point. The second one is to calculate the shortest route back to their assigned parking space.

Algorithm 1: AIV detects an obstacle ahead	Algorithm 2: AIV receives an information about an obstacle
<p>Input : AIV attributes, Radar Function Communication to other AIVs</p> <pre> 1 if AIV detects an obstacle then 2 Determine level of obstruction of the obstacle 3 Send message to other AIVs - 4 (presence of an obstacle, position and level of obstruction) 5 if Obstacle is static then 6 if Possibility to bypass the static obstacle then 7 Bypass the static obstacle 8 else 9 Dijkstra algorithm - 10 calculate a new path to continue its mission 11 end 12 else 13 Slowing down - 14 increasing caution (dynamic obstacle: pedestrian, etc.) 15 if Possibility to bypass the dynamic obstacle then 16 Bypass the dynamic obstacle 17 else 18 Dijkstra algorithm - 19 calculate a new path to continue its mission 20 end 21 end 22 end </pre>	<p>Input : AIV attributes, Radar Function, Communication to other AIVs</p> <pre> 1 if Obstacle is on its path then 2 if Obstacle is static then 3 if Possibility to bypass the static obstacle then 4 Bypass the static obstacle 5 else 6 Dijkstra algorithm - 7 calculate a new path to continue its mission. 8 end 9 else 10 Slowing down - increasing caution 11 (dynamic obstacle: pedestrian, etc.) 12 if Possibility to bypass the dynamic obstacle then 13 Bypass the dynamic obstacle 14 else 15 Dijkstra algorithm - 16 calculate a new path to continue its mission 17 end 18 end 19 else 20 Continue its path and its mission 21 end </pre>

Fig. 2. Collective intelligent strategy algorithm

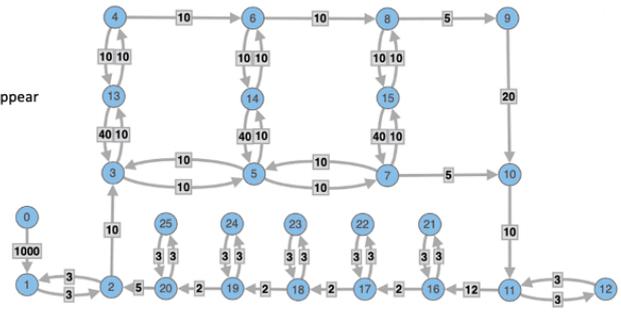
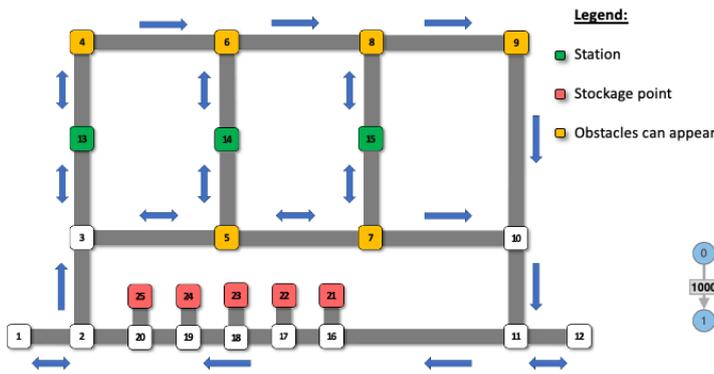


Fig. 3. Warehouse environment [16] with a) adapted direction of traffic, and b) its oriented graph with costs.

The costs in distance between the different nodes of the circuit have been chosen and applied to favour certain directions of traffic. For example, vehicles should go from node 14 to node 6, which has a cost of 10, rather than to node 5, which has a cost of 40. These costs between the different nodes are shown in the directed graph in Fig. 3. These costs are used by Dijkstra's algorithm to find the path that costs the least distance, and therefore the fastest.

To simulate these different experiments, we implemented a graphical application with different functionalities. The different classes presented in Fig. 1 have been implemented in Python from scratch. We use and develop this simulator for various experiments in the laboratory, and for teaching to engineering students. Its interface is shown in Fig. 4. In the central part, we have the representation of the warehouse presented by Bechtis *et al.* [16], which we have adapted in Fig. 3. The different nodes present in Fig. 3 and Fig. 4 are represented by white squares in the interface. The vehicles are visualised by small coloured circles, and the obstacles are coloured black with different sizes corresponding to the obstruction levels in the aisle.

In the following, we will denominate the scenarios *sc1*, *sc2* and *sc3*. The missions will be named *mi* and the nodes *ni*.

The functionalities proposed by our application (Fig. 4) are to:

- Stop the simulation.
- Place obstacles randomly in terms of size T1 to T4 (T4 corresponding to a size that obstructs an entire aisle – see Fig. 4) and position at the six possible nodes shown in yellow in Fig. 3.
- Launch a scenario: the agents must then carry out missions given by the supervisor. That is, go from their parking place to one of the three green nodes in Fig. 3, before returning to their parking place. On the other hand, pressing the scenario 1, 2 or 3 button allows them to choose which strategy (*sc1*, *sc2* or *sc3*) they can perform.

For instance, in Fig. 4, an avoidable obstacle with a size *T2* in *n6* is present, and an obstacle obstructs the whole passage with a size *t4* in *n8*. The size of the obstacle in the interface corresponds to its obstruction level.

These different approaches were detailed in subsection 3.3. *Sc1*, therefore, does not allow the agents to use Dijkstra's algorithm. *Sc2* does, and *sc3* allows agents to communicate with each other about the description of a perceived obstacle.

3.2. Setting up the experiments

The three approaches were tested in six different experiments. Each experiment corresponded to an obstacle completely obstructing an aisle on one of the six nodes: *n4*, *n5*, *n6*, *n7*, *n8* and *n9* shown in orange in Fig.3. During these six experiments, the agents had to perform 1 of 3 missions consisting in going from the car park to one of the nodes *n13*, *n14* and *n15* randomly chosen by the supervisor, and then returning to their parking place. They were equipped with one of the three approach capabilities described in subsection 3.3. That is, three experiments were performed with *sc1*, *sc2* and *sc3* with an obstruction at the same node *ni* to compare the approaches.

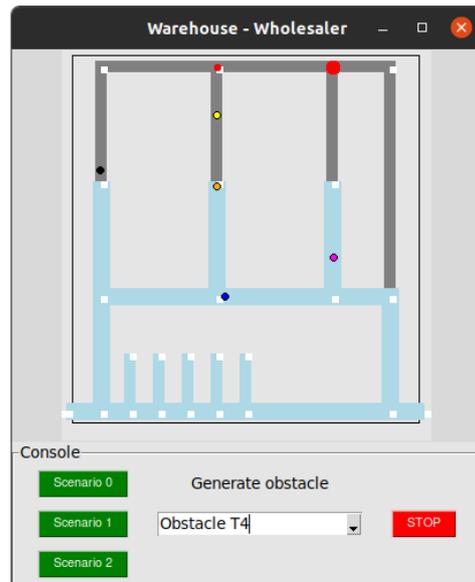


Fig. 4. Application interface to simulate missions in the warehouse environment

The performance indicators that were used to define the efficiency and performance of these approaches are:

- The validity of the set of missions performed by the five agents.
- The distance travelled to carry out the different missions.
- The overall time is taken to complete the various tasks.

Distance and time are performance indicators that were chosen because they are strongly related to the energy used by a robot, an *AIV*. Thus, this indicates the energy impact of the different approaches.

3.3. Experiments: comparisons with performance indicators

Table 1 summarises the different results of the agents in the six experiments according to *sc1*, *sc2* or *sc3*. *Sc1* does not allow the agents to complete all the missions, which is explained by the fact that they are not able to recalculate their routes. Thus, if any agent was blocked by an obstruction on a node, it remained without the ability to react and the agents' overall mission failed.

Conversely, *sc2* and *sc3* applied by the agents allowed in both cases to fulfil all the missions thanks to their ability to adapt and to plan a new route (with the Dijkstra algorithm) when encountering an obstacle obstructing a path.

The distance and overall execution time to complete all the missions have therefore been quantified in Table 2 and Table 3 respectively.

The analysis that emerges is that *sc3* allows the agents to travel less distance in 4 out of 6 experiments and allows them to complete the missions more quickly in all six experiments.

The cooperative perception allows the agents to not go directly to the obstacle if it is on their way, but to redefine with the help of Dijkstra another path to reach their goal. Thus, logically, this collaboration takes precedence over the individual strategy, and communication in a fleet of *AIVs* thus increases their autonomy and their ability to adapt to new situations. Over the six experiments, there is a 3% gain in overall distance and a 9% gain in time for the whole fleet of *AIVs*. This cooperative perception is an essential basis for a future collective global strategy for the sharing and organisation of distributed missions within a fleet.

Table 1. Completed missions in relation to the strategy number of agents

Completed missions							
Obstacle node	N4	N5	N6	N7	N8	N9	
Strategy number							
1	False						
2	True						
3	True						

Table 2. Total distance travelled by agents in relation to the strategy number

Total distance							
Obstacle node	N4	N5	N6	N7	N8	N9	
Strategy number							
2	600	740	660	660	740	755	
3	600	700	640	660	710	715	

Table 3. Total execution time to carry out the missions in relation to the strategy number

Total execution time							
Obstacle node	N4	N5	N6	N7	N8	N9	
Strategy number							
2	489.75	677.56	521.23	559.59	582.97	567.52	
3	482.14	662.82	456.60	542.04	466.35	470.40	

3.4. Cooperative infrastructure

After, the different results mentioned in the previous section, we will discuss the possible cooperation of the infrastructure. Indeed, it could help the collective *AIVs* strategy with local information. This knowledge of the environment could be shared by active elements of the infrastructure such as cameras. This could help to qualify an obstacle or even detect an obstacle before a vehicle enters an area with an unavoidable obstacle.

3.5. Fuzzy logic to qualify obstacles

In perspective, the *AIV* agents of our model presented in Fig. 1 may become fuzzy. Indeed, the agents will have incomplete, fragmented, fuzzy and uncertain knowledge in certain situations. The fuzzy logic could allow the agents to be able to recalculate or not their itinerary according to whether an obstacle seems avoidable or not, depending on the qualification of the disturbance caused by the obstacle.

4. Conclusions and perspectives

In the context of Industry 4.0, mobile robots must become increasingly autonomous in order to avoid obstacles in an intelligent way. We are interested in the possible impact of cooperative perception between the vehicles of a fleet during a mission. Our aim is to move to a higher level of collaboration with the possibility of mission sharing and reorganisation in a distributed way.

The comparison of the three approaches evaluated using distance, execution time and whether or not a mission has been successfully achieved, showed that it is essential that vehicles have the ability to re-plan their routes while necessary. In addition, the last approach, which allowed agents to communicate the description of perceived obstacles, allowed for gains in distance and execution time of overall missions. Indeed, it allows *AIVs* to avoid traffic around an area where a

static or dynamic obstacle (human operator, for instance) is present in an aisle. It also allows operators or other robots to be safe in that area: for example, to remove a static obstacle.

This very preliminary study already shows the interest of collaboration to increase the collective and individual efficiency of the vehicles in a fleet. It opens the door to more advanced global collective strategies with the possibility of the allocation, scheduling and distribution of tasks between them in real-time after the perception of an obstacle. Furthermore, the tracking of obstacles by the robots, or assisted by the infrastructure, i.e. a cooperative perception, would also optimise the efficiency of the fleet in carrying out these missions.

The perspectives that emerge from this work are to increase the work of the analysis with other performance indicators, but especially more experiments. Indeed, we can define several missions in a row by robots with several obstacles obstructing the lanes at different places. Another strong perspective to our work is to experiment the collective strategy with the TurtleBot3 Burger robots.

In addition, the notions of cooperative infrastructure and the use of fuzzy logic to qualify obstacles are also planned to be further developed in future work.

Acknowledgements

The authors would like to thank the region Bretagne- France for funding the VIASIC project as part of the ARED-2021-2024 call for projects concerning the strategic innovation area: Economics for industry for intelligent production.

References

- [1] A. R. Boccella, P. Centobelli, R. Cerchione, T. Murino, and R. Riedel, (2020). Evaluating Centralized and Heterarchical Control of Smart Manufacturing Systems in the Era of Industry 4.0. *Applied Sciences*, 10(3): 755.
- [2] U. Cao, A. S. Fukunaga, and A. B. Kahng, (1995), Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4: 7–27.
- [3] Z. H. Ismail, N. Sariff, and E. G. Hurtado, (2018). A survey and analysis of cooperative multi-agent robot systems: challenges and directions. In *Applications of Mobile Robots*, IntechOpen, pp. 8–14.
- [4] Y. Rizk, M. Awad, and E. W. Tunstel, (2019). Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys*, 52(2): 1–31.
- [5] M. De Ryck, M. Versteijhe, K. Shariatmadar (2020). Resource management in decentralised industrial automated guided vehicle systems. *Journal of Manufacturing System*, 54: 152–173.
- [6] S. Haddadin, A. De Luca, and A. Albu-Schäffer, (2017). Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6): 1292–1312.
- [7] A.M. Nascimento, L.F. Vismari, C.B.S.T. Molina, P.S. Cugnasca, J.B. Camargo, J.R. de Almeida, ..., and A.Y. Hata, (2019). A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. *IEEE Transactions on Intelligent Transportation Systems*, 21(12): 4928–4946.
- [8] V. Kunchev, L. Jain, V. Ivancevic, A. Finn, (2006). Path planning and obstacle avoidance for autonomous mobile robots: a review. 10th international conference on knowledge-based & intelligent information & engineering systems (KES2006), pp. 537–44.
- [9] H. Ma, S. Koenig, (2016). Optimal target assignment and path finding for teams of agents. In *Int. Conf. On Autonomous Agents and Multi Agents Systems (AAMAS)*, pp. 1144-1152.
- [10] D.H. Kim, N.T. Hai, W.Y. Joe, (2018). A guide to selecting path planning algorithm for automated guided vehicle (AGV). *Lect Notes Electr Eng*, Vol. 465, pp. 587–96.
- [11] N. Bahnes, B. Kechar, and H. Haffaf, (2016). Cooperation Intelligent Autonomous Vehicles to enhance container terminal operations. *Journal of Innovation in Digital Ecosystems*, 3(1): 22-29.
- [12] M. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, (2013). Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3): 1162–1175.
- [13] J. Grosset, A.-J. Fougères, M. Djoko-Kouam, C. Couturier, J.-M. Bonnin, (2022). Simulation of a collision and obstacle avoidance algorithm for cooperative industrial autonomous vehicles. 2nd IFSA Winter Conf. on Automation, Robotics & Communications for Industry 4.0 (ARCI'2022), Andorra la Vella, Andorra, February 2-4, pp. 88–91.
- [14] K. Karur, N. Sharma, C. Dharmatti, J. E. Siegel, (2021). A Survey of Path Planning Algorithms for Mobile Robots. *Vehicles*, 23: 448–468.
- [15] Intelligent Transport Systems (ITS). Vehicular Communications; Basic set of Applications; Analysis of the Collective Perception Service (CPS), Release 2, ETSI TR 103 562 V2.1.1, 2019.
- [16] D. Bechtsis, N. Tsolakis, D. Vlachos, J. S. Srai, (2018). Intelligent autonomous vehicles in digital supply chains: a framework for integrating innovations towards sustainable value networks. *Journal of cleaner production*, 181: 60–71.