An Industrial Perspective on Multi-Agent Decision Making for Interoperable Robot Navigation following the VDA5050 Standard

Niels van Duijkeren, Luigi Palmieri, Ralph Lange and Alexander Kleiner

Abstract— **This paper provides a perspective on the literature and current challenges in Multi-Agent Systems for interoperable robot navigation in industry. The focus is on the multi-agent decision stack for Autonomous Mobile Robots operating in mixed environments with humans, manually driven vehicles, and legacy Automated Guided Vehicles. We provide typical characteristics of such Multi-Agent Systems observed today and how these are expected to change on the short term due to the new standard VDA5050 and the interoperability framework OpenRMF. We present recent changes in fleet management standards and the role of open middleware frameworks like ROS2 reaching industrial-grade quality. Approaches to increase the robustness and performance of multi-robot navigation systems for transportation are discussed, and research opportunities are derived.**

I. INTRODUCTION

Multi-robot navigation encompasses an ever-tighter integration of a vast number of disciplines and research as in most of the robotics research. It is largely about the decision-making stack from translating high-level tasks into individual actuator commands for multiple robots [1]. Where Automated Guided Vehicles (AGVs) have been the main go-to solution for a very long time, Autonomous Mobile Robots (AMRs) are rapidly gaining popularity driven by their flexibility and increasingly advanced capabilities. An example of such an AMR is the ActiveShuttle by Bosch Rexroth depicted in Fig. 1, primarily deployed for the automotive industry supply chain. Another example are autonomous cleaning machines for e.g., offices and supermarkets that are currently coming onto the market. Although these use cases are completely different, the decision-making stacks have much in common. In this paper we present an industrial perspective on research on decision making in multi-agent systems and the current challenges for a settings in *mixed environments*, namely transportation tasks in factories. Mixed environments here refer to the setting where the environment is shared with other "uncontrolled" agents such as humans and legacy (disconnected) AGVs. The requirements for a multi-agent systems are, as usual, motivated in a top-down fashion. We will however present the current view on the decision-making stack in a bottom-up order, that is from a single-agent behavior and motion planning stack to the task assignment problem.

Transportation tasks in factories are commonly to provision production machinery and assembly lines, either from storage or directly between different machines, and to take

All authors are with Robert Bosch GmbH, Corporate Research, Stuttgart, Germany {niels.vanduijkeren, luigi.palmieri, ralph.lange, alexander.kleiner}@bosch.com

Fig. 1. The ActiveShuttle by Bosch Rexroth, an AMR for the automotive supply chain.

finalized components to storage locations. The factory is structured by driving lanes that, as mentioned, are shared between vehicles and humans.

In a multi-agent setting, one robot is typically assigned to one transportation task at a time and the robots share the available resources (primarily space, but also infrastructure like charging stations). The requirements from an operator perspective for industrial transportation systems is that they are efficient and reliable. Higher efficiency helps keeping costs low by requiring fewer robots. Reliability is needed to be able to effectively plan tasks for the Multi-Agent System (MAS), and thereby the overall factory operations. Reliability implies robustness against expected disturbances and uncertainties, however can practically also refer to the ability to provide timely status updates and thereby facilitate reactive planning. Notice that these top-down requirements propagate through the entire decision-making stack.

Industry is steadily evolving towards more integrated planning and control mechanisms with the advent of Industry 4.0. Production machinery in factories are adopting Internet of Things (IoT) principles such as open APIs to facilitate this. AGVs have been abundant for several decades, initially blindly following guidance systems in the floor. The next trend were stand-alone AMRs with an isolated/proprietary form of coordination, but more flexible due to e.g., laser-based navigation.

Now new vendor-independent protocols and standards for communication between fleet management and single robots such as VDA5050 [2] and OpenRMF [3] are being developed. VDA5050 is a standard pushed by the German Automotive-Industry Association (VDA) and the German Mechanical Engineering Industry Association (VDMA) and
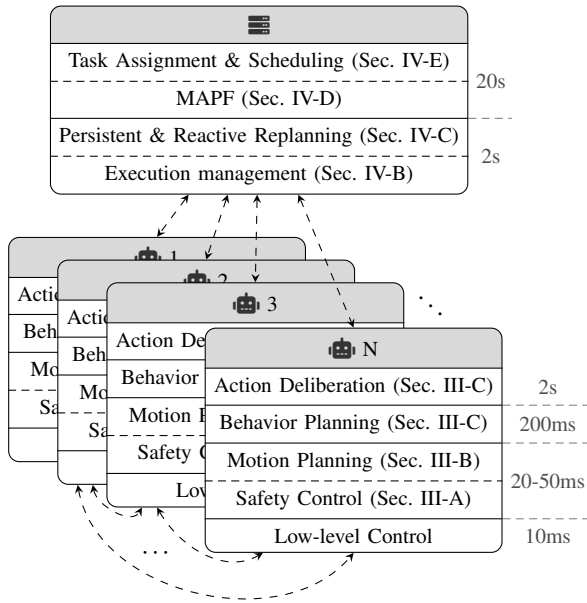
Fig. 2. Schematic overview of a generic industrial multi-agent navigation stack with typical cycle times.

is breaking open the market for fleet management and control of AMRs and creating new business opportunities. Similarly, with ROS2 an open industrial-grade framework for building robotic systems is available, and by the Nav2 stack [4] and OpenRMF [3] also single-robot navigation and fleet management tools are provided.

Contributions: (a) We present typical qualitative problem dimensions faced in industrial MASs. *(b)* A reference industrial multi-agent decision-making stack is presented with currently common solution approaches. Furthermore, we survey how recent developments in literature fit in and from that derive *research opportunities*. *(c)* The role of the VDA5050 standard for industrial MASs is discussed and so are the opportunities arising from ROS2 and the accomanying tools.

Paper Structure: In Section II we present a reference transportation case used in this paper. Subsequently, in Section III the single-agent decision stack is briefly discussed focussing on mechanisms and challenges to achieve robustness. Thereafter, in Section IV the centralized multi-agent coordination pipeline is detailed. In Section V we discuss integration aspects and developments therein, and the role of benchmarking such integrated systems. We conclude the paper with a short summary and research opportunities in Section VI.

II. INDUSTRIAL MULTI-AGENT SYSTEMS

Fig. 2 depicts a typical decision-making stack for autonomous navigation of a multi-agent robotic system in industry.

Remark 1: We consider the same structure of the stack to be applicable for transportation and coverage tasks. The environments for coverage tasks are more diverse and still mostly an unexplored domain.

Number of agents in plant	50-200
Number of agent types [1]	4-8
AMR speed (m/s)	0.7-1.5
Duration of tasks (minutes)	3-20
Typical delay per task (minutes)	0-5
Time between tasks (minutes) [2]	1+

We consider AMRs (possibly in groups of different types and vendors) operating in a semi-structured environment with a known roadmap of driving lanes. The above system is effectively modeled by agents moving on a directed graph. The agents operate in mixed environments, in which also other autonomous agents participate with uncertain behavior:

Human-driven vehicles: For example forklifts, pallet movers, and tugger trains.
- The motions of these vehicles are constrained and assumed predictable.
- The vehicles are often significantly faster than AMRs, it is therefore desirable that AMRs let human-driven vehicles pass if the situation allows.
- The vehicles may occasionally halt on the driving lanes to load and unload. Although undesirable, the vehicles may also be parked for extended periods of time and partially block lanes.
- Personnel is instructed to give priority to AGVs at intersections and not to disturb their motion.

Human workers: Moving solo or in groups.
- Behavior and motion is hard to predict reliably, robots ought to move carefully around humans to minimize risk of causing injuries.
- Personnel is instructed to give way to AGVs at all times.

Legacy AGVs and AMRs: Groups of AGVs and AMRs that use a legacy non-cooperative fleet management system.
- The systems are usually the least flexible, and it is desirable to not interfere with their motions.
- The systems are safe and will stop the for anything that blocks their path, AMRs might—unlike AGVs—bypass the obstacle.

There is much potential in restructuring a warehouse or factory such that the environment is more certain [5], this also allows much faster vehicles if the environment is free from humans. For many small to medium enterprises and existing factories, the trend is that automation is introduced more gradually and dealing with uncertainty is highly relevant. Fortunately the problem setting reveals enough structure such that a decomposition in subtasks is often trivial, and with easily-verifiable assumptions guarantees can be obtained. Moreover, as Table I reveals, the typically observed problem dimensions are manageable.

---

[1] For transporting loads of different types and sizes; vehicles are often from different vendors.

[2] Machine-provisioning tasks are usually planned one by one as they arrive, if pre-planning of factory operations is possible it is common that tasks arrive in batches.

III. SINGLE-AGENT PLANNING STACK

The goal for the single-agent planning stack is to execute short motion tasks with a horizon in the order of up to 10 seconds. A single-agent planning stack provides information about the robot capabilities and constraints. The stack needs to ensure that a task is executed safely and reliably and needs to be able to provide feedback in case task execution is delayed or stuck and provide possible solutions.

A. Safety Control

In mixed environments, the layer that connects motion planning with the low-level actuators is a safety control unit. Often a dedicated Programmable Logic Controller (PLC) is paired with one or multiple certified safety Lidar scanners to monitor that pre-programmed, velocity-dependent, areas remain clear of obstacles and the PLC intervenes if they do not. More modern approaches aim to not employ discrete switching behavior anymore [6]. AGVs rely on these systems to brake for obstacles in the way, the aim for AMRs is to keep these areas free of obstacles in the motion planning stage. Safety controllers override any motion commands if necessary and safety constraints have a significant influence on the robot behaviors and efficiency.

B. Motion Planning

The task of the motion planner is to drive the robot fast and smooth to the given goal. This layer concerns itself with the vehicle dynamics, action limits, load stability, and user-defined bounds on velocity and acceleration. In principle a reference path can be assumed to be given, however an explicitly-specified maximum deviation serves to allow avoiding obstacles by planning in a corridor [7]. Model Predictive Control (MPC) is a popular approach to the motion planning problem with enabling tools freely available [8]. Through extended models it is avoided that robots carrying heavy loads require excessive currents to drive due to the orientation of caster wheels [9]. Motion planning is typically restricted to making decisions in continuous spaces with ideally differentiable objectives and constraints.

C. Behavior Planning and Action Deliberation

Behavior planning in constrast is concerned with making logical decisions, such as whether to yield at an intersection, overtake vehicles, or facilitate being overtaken. In the architecture of Fig. 2 the function is separated in two layers: (1) a behavior control layer that implements reactive behavior and executes sequences of actions, (2) an action deliberation layer that considers the interaction with other agents, the interaction with the multi-agent planning stack, and plans action sequences to optimize performance. For behavior planning, behavior trees have rapidly spread in from the gaming industry to robotics [10]. They are a flexible representation of robot behaviors that can be programmed by hand, synthesized from formal methods such as Linear Temporal Logic (LTL) [11], or obtained from reinforcement learing or learning from demonstration. The action deliberation layer can for instance be the high-level scoring and validation of several possible scenarios, committing to the best scenario and steer the behavior planning accordingly. Although we will not enter into much detail of this aspect, but we note that humans might interfere with the tasks that were allocated by the multi-agent planning stack. Robots typically have multiple modes of user interaction that override the autonomous task execution.

D. Distributed Coordination and Planning

There exist many works on distributed coordination and cooperative planning for most of the above functional layers and may help to increase robustness in multi-agent plan execution. Approaches include precisely coordinating continuous trajectories [12] and avoid collisions between agents while deviating from the prescribed path [13]. Distributed negotiation over resources becomes relevant when simultaneous movements are not possible due to resource constraints [14]. This type of coordination is sometimes referred to as *swarm intelligence* [15], but unfortunately no open standards have been developed for this purpose yet. This functionality allow robots to for instance collaboratively execute tasks that could not be performed by a single robot.

IV. MULTI-AGENT PLANNING STACK

The multi-agent planning stack is responsible for orchestrating the motions of single agents such that high-level tasks are completed. In this paper one can assume that the transportation or cleaning tasks are already decomposed such that they can principally be executed by single agents.

A. VDA5050 Standard for Transportation

The VDA5050 standard for fleet management is developed and promoted by major industry associations in Germany. Its primary scope is to control wheeled AMRs moving in a 2D model of the world. The text of the standard is available on a GitHub repository and contributions can be made via pull requests [2]. The VDA5050 protocol assumes there is one of a so-called *master control* that controls all vehicles in a plant. The standard is primarily a definition of the communication between the master control and the AGVs/AMRs. Hereby, the vehicles of different vendors can be planned and controlled by a single fleet management system. Similarly, the fleet management system can be freely replaced by another if desired. There are facilities in the standard to allow intermediate proprietary control layers for legacy systems, for this it is assumed that all relevant information of the vehicles is relayed to the VDA5050-based control. The aim of the standard is to support MASs consisting of up to a few thousand vehicles.

The VDA5050 standard prescribes usage of Message Queuing Telemetry Transport (MQTT) protocol as a means of communication via wireless networks and the messages are packed in a JSON format. The plant operator defines a roadmap with constraints for the edges and properties of the nodes. Each AGV communicates a factsheet about its properties and abilities. The AGVs are expected to be able to perform single-agent tasks and that they transmit

their status frequently. The *master control* is responsible for Multi-Agent Planning (MAP), conflict resolution and traffic control, charging jobs, communication with the infrastrure (e.g., doors), and resolving communication problems.

The control concept of VDA5050 is to send task orders to the agents. These task orders are decomposed in two parts, the *base* and the *horizon*. The base is the part of the task that the AGV is expected to execute immediately and is not assumed to be cancellable. The horizon is not yet confirmed and therefore only informative and may change. A typical implementation will order a base that is quite short where the AGV has to drive up to e.g. 6 meters, where the horizon may be the entire route to the goal or just a few edges beyond the base. The order message consists of over 50 fields and includes the target position and orientation, the nodes to traverse, the actions to perform on each node, the shape of path between nodes as straight lines or by Non-Uniform Rational Basis Spline (NURBS) curves, the maximum deviation (e.g., to allow driving around obstacles), the maximum velocity, maximum height (for high loads), et cetera. There is a predefined set of actions (pick, place, start/stop charging, etc.), but it is foreseen that vendors can extend these based on the needs of an application. The status message sent by the robot includes the current state, but also informs the multi-agent planning stack and indirectly the operator about contingencies to e.g. diagnose delayed or stuck robots.

B. Execution Management

Note that the VDA5050 is tailored to a specific, but abundant, use case often seen in the automotive and mechanical engineering industry. The standard focusses on the interaction between a central fleet manager and the robotic fleet, to define the means for orchestrating the robot motion, which naturally implies that a Multi-Agent Path Finding (MAPF) component is present. However, at the execution-management level of the planning and coordination stack, it is assumed that the desired trajectories for all robots are available. The main responsiblity is to ensure robust execution of these plans through real-time monitoring and allocation of physical resources to agents. The procedure is typically implemented by a reactive resource-reservation system. Based on the current state of the robot or fleet, one robot receives the clearance to move along a certain edge. This may be as simple as a first-come-first-serve assignment, but more suitably a method that avoids deadlocks among the agents participating in the MAPF plan [16], [17], [18]. Inspired by optimization-based street traffic management approaches [19], one might consider dealing with deadlocks and congestions implicitly but thereby account for humans [20] and autonomous agents too. On a lower level, execution management optimizations are on precisely timing how agents pass through intersections [21], [22]. Finally, in particular when robots need to temporarily deviate from their assigned corridors, for instance to circumvent blockages or to overtake slower vehicles, coordination on the level of continuous trajectories is appropriate [23].

C. Persistent & Reactive Replanning

Unmodeled disturbances in the execution of an MAPF plan can have mild consequences on the efficiency of the multi-agent system but may also be dramatic in case a deadlock occurs. Motivated hereby, the Action Dependency Graph (ADG) was proposed to maintain and enforce an execution schedule that can be derived from most MAPF plans [17]. A follow-up work presents the so-called Switchable Action Dependency Graph (SADG) and a method to optimally change the order of planned actions in the MAPF to react on disturbances while guaranteeing to preserve deadlock-freeness [24]. There is even more potential in promptly rerouting vehicles in the event of unforeseen congestions [25]. One can consider the problem as to continously locally repairing MAPF plans [26], but also to continuously process incoming jobs such as in [27]. All the above is ideally executed at the time scales of several seconds, a rapid response is critical and temporary suboptimality is rather unproblematic. Conflict-Based Search (CBS) methods [28] seem naturally suited for continuous updates reactively replanning based on observed real conflicts as suggested in e.g. [18].

D. Multi-Agent Path Finding

The goal of the MAPF component is to find approximately optimal routes for tasks that are not yet being executed. These tasks are typically in a queue of tasks to be started from about half a minute up to 30 minutes in the future. The MAPF is presented separately from task assignment because it remains to be commonly implemented this way. MAPF is a well-established research area [29] and has a plethora of highly performant algorithms. The algorithms are sufficiently fast for problem sizes presented in Tab. I and for the scaling expected over the next few years and the scope of VDA5050. Optimality in MAPF is usually defined as either the plans that minimize the *makespan* or those that minimizes the *sum-of-costs*. The latter is often chosen for the heuristic that it is typically appreciated that any single robot finishes their task sooner if possible to be available for new tasks. The MAPF problem is often still approached by the classical prioritized planning methods [30] and single-agent plans are generated sequentially in the order that the paths for tasks are being planned. For larger scale use cases practically improved completeness [31], or complete methods such as Safe Interval Path Planning (SIPP) [32], CBS [28] are appreciated. So is the speed-up that is attained through follow-up contributions [33], [34], [35], and very recently [36]. It is remarked that concepts where plant operators can improve the search by defining e.g., highways [35] can be very useful in practice if properly exploited. Robust path-finding methods and the framework presented in [18] are highly relevant to the industrial use case and form a basis for the work presented here. Finally, we note that methods to combine data-driven models for heuristics with suboptimal search hold a lot of potential [37].

E. Task Assignment and Scheduling

Task assignment is the problem of deciding which agent should execute which task in the queue, or if necessary what combination of robots ought to be used. One straightforward approach is to select the robot that can reach the starting position of the task the soonest. This approach often works quite well, but suffers from the usual shortcomings of a greedy decision-making approach. It does confirm that for transportation use cases the task assignment problem is closely related to the MAPF problem. It is therefore no surprise that it is proposed to solve the optimal MAPF and task-assignment problem simultaneously [38]. Task assignment and scheduling interface with cooperative multi-agent task planning mechanisms upwards [39], [40], these methods are out of scope for this paper but are becoming increasingly relevant for industry.

V. INTEGRATION AND BENCHMARKING

The integration of a complex robotic system is a challenging engineering endeavor. Combining many sensors, actuators, communication hardware, and complex software is not a new domain. However, these robots operate with little human supervision in uncertain environments, while they are expected to autonomously resolve conflicts in a rationally legible manner. All required technologies to make this happen move forward with a fast pace, and testing and measuring these advances in an integrated system is essential. It can be frustrating to realize that one little thing that goes wrong in the decision-making stack, may quickly result in classical AGVs to outperform the more modern systems.

A. Robot Operating System 2 (ROS2)

ROS and ROS2 are the de-facto standard frameworks in academia for developing any robotic system, with ROS2 it steadily gains traction in industry to fulfill that same role. The core of ROS2 is to provide a framework for intra- and inter-robot communication and interaction. It is complemented with a large ecosystem of algorithms and tools to quickly implement, simulate, and deploy robotic systems. The active involvement of industry in the devopment of ROS2 has lead to a very high software-quality standard, it becomes increasingly rare that a problem in the ROS2 core is a cause for problems in a robotic system. The same is true for the most popular tools in ROS2, such as *(a) Navigation2*, the highly flexible plugin-based stack for single-robot navigation [4]; *(b) PlanSys2*, a PDDL-based planning system to conveniently write, deploy, and troubleshoot AI-planning based decision-making algorithms [41]; *(c) OpenRMF*, a framework for implementing fleet management and multi-agent planning systems [3].

OpenRMF is a relatively new project that is not based on an already popular ROS1 counterpart like Navigation2 and PlanSys2 had. It brings new tools to build web interfaces, connecting to automated task dispatching systems, connecting to entire fleet management systems or single devices, and the core components of facilitating auctions, traffic coordination, and scheduling. Any useful tool implemented in OpenRMF is expected to gain rapid adoption in both academia and perhaps eventually industry.

Another relevant ROS2 community effort highly relevant to this paper is the `vda5050_msgs` package [42] to conveniently couple the VDA5050 standard with ROS2 via an MQTT bridge.

B. Benchmarking

Whether your robotic problem is really solved by an engineered solution is hard to decide. Bringing a competitive product to the market ultimately means that the costs of the solution are justifiable for the customer. The desirable high efficiency and reliability are two key criteria which one can attempt to measure. However, increasing modularization and open standards like VDA5050 make this problem hard to do right. In practice, a multi-agent planning stack should work well with any mixture of robot teams from ideally every possible combination of vendors. The development of highly-automated simulation-based benchmarking suites with an ever-growing portfolio of cases and scenarios is indispensable. In order to avoid that every company needs to repeat these efforts, especially open-source activities in this direction are highly valuable [43]. This latter open-source project based on ROS2 and OpenRMF was recently started and hopefully gathers momentum in the near future.

VI. CONCLUSIONS AND OPPORTUNITIES

In this paper we gave a perspective on Multi-Agent Systems for industrial navigation. By presenting the problem setting in detail we could confirm the industrial relevance of many recent advances in research. We would like to conclude this paper with a list of application-driven research challenges we have identified:

Research Opportunity 1: The main goal for the persistent replanning function detailed in Sec. IV-C is to maintain an optimized flow on the roadmap graph. How can controlled agents and uncertain uncontrolled agents be optimally guided to avoid congestions and deadlocks? What role can data-driven models play to this end?

Research Opportunity 2: We present MAPF (Sec. IV-D) to mainly be about planning new paths and replanning (Sec IV-C) about dealing with disturbances. How can these steps be arranged such that they operate on a common plan and e.g., existing plans are modified based on new task information.

Research Opportunity 3: Roadmap graphs considered in Sec. IV-D represent a restricted view on physically possible motions and in case of a blockage, an AMR may be able to drive around it. How can we model temporary free space motions in a persistent MAPF planning scheme?

Research Opportunity 4: MAPF planning in factories is highly repetitive, we can potentially obtain rich data-driven models to improve the quality of plans over time. How can such data-driven models be effectively used in the planning stage without imposing a restrictive computational burden.

Research Opportunity 5: How can industrial companies co-operate on a common benchmarking platform to facilitate optimization of complementary products by other vendors without sharing sensitive intellectual property.

REFERENCES

[1] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robotics: Automation Magazine*, vol. 14, pp. 61–70, mar 2007.

[2] German Association of the Automotive Industry, "VDA 5050 AGV Communication Interface," Tech. Rep. Version 2.0, Jan. 2022. https://github.com/vda5050/vda5050.

[3] Open Source Robotics Foundation, "Open-RMF: A common language for robot interoperability." https://www.open-rmf.org/.

[4] S. Macenski, F. Martin, R. White, and J. Ginés Clavero, "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[5] P. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Magazine*, vol. 29, pp. 9–20, 03 2008.

[6] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[7] N. van Duijkeren, *Online Motion Control in Virtual Corridors - For Fast Robotic Systems*. PhD thesis, KU Leuven, 2019.

[8] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, pp. 147–183, oct 2021.

[9] J. Arrizabalaga, N. van Duijkeren, M. Ryll, and R. Lange, "A caster-wheel-aware MPC-based motion planner for mobile robotics," in *2021 20th International Conference on Advanced Robotics (ICAR)*, IEEE, dec 2021.

[10] M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, "A survey of behavior trees in robotics and AI," *Robotics and Autonomous Systems*, vol. 154, p. 104096, aug 2022.

[11] M. Colledanchise, R. M. Murray, and P. Ogren, "Synthesis of correct-by-construction behavior trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, sep 2017.

[12] R. Van Parys and G. Pipeleers, "Distributed MPC for multi-vehicle systems moving in formation," *Robotics and Autonomous Systems*, vol. 97, pp. 144–152, nov 2017.

[13] R. Firoozi, L. Ferranti, X. Zhang, S. Nejadnik, and F. Borrelli, "A distributed multi-robot coordination algorithm for navigation in tight environments,"

[14] C. Liu, C.-W. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 3, pp. 18–29, mar 2018.

[15] Agilox Services GmbH, "The AGILOX X-SWARM Technology for maximum efficiency and flexibility." Online available, Sept. 2021. https://www.agilox.net/en/blog/the-agilox-x-swarm-technology-for-maximum-efficiency-and-flexibility/ (accessed on 2022/09/10).

[16] H. Ma, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with delay probabilities," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, feb 2017.

[17] W. Hönig, S. Kiesel, A. Tinka, J. Durham, and N. Ayanian, "Persistent and robust execution of MAPF schedules in warehouses," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1125–1131, 2019.

[18] D. Atzmon, R. Stern, A. Felner, G. Wagner, R. Barták, and N.-F. Zhou, "Robust Multi-Agent Path Finding and Executing," *Journal of Artificial Intelligence Research*, vol. 67, pp. 549–579, 2020.

[19] S. Lin and B. De Schutter and Y. Xi and H. Hellendoorn, "Fast model predictive control for urban road networks via MILP," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 846–856, Sept. 2011.

[20] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[21] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, mar 2008.

[22] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *2015 American Control Conference (ACC)*.

[23] F. Pecora, H. Andreasson, M. Mansouri, and V. Petkov, "A loosely-coupled approach for multi-robot coordination, motion planning and control," in *Twenty-eighth international conference on automated planning and scheduling*, 2018.

[24] A. Berndt, N. V. Duijkeren, L. Palmieri, and T. Keviczky, "A feedback scheme to reorder a multi-agent execution schedule by persistently optimizing a switchable action dependency graph," in *Proceedings of the Distributed and Multi-Agent Planning (DMAP) Workshop at ICAPS*, Oct. 2020.

[25] K. Vedder and J. Biswas, "x∗: Anytime multi-agent path finding for sparse domains using window-based iterative repairs," *Artificial Intelligence*, vol. 291, p. 103417, feb 2021.

[26] A. Felner, R. Stern, J. S. Rosenschein, and A. Pomeransky, "Searching for close alternative plans," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 3, pp. 211–237, 2007.

[27] H. Ma, J. Li, S. Kumar, and S. Koenig, "Lifelong multi-agent path finding for online pickup and delivery tasks," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 837–845, 2017.

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

[29] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, E. Boyarski, and R. Bartak, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of the Twelfth Anual Symposium on Combinatorial Search*, (Napa, California, USA), Sept. 2019.

[30] M. Erdmann and T. Lozano-Pérez, "On multiple moving objects," *Algorithmica*, vol. 2, pp. 477–521, nov 1987.

[31] M. Cáp, P. Novák, A. Kleiner, and M. Selecký, "Prioritized planning algorithms for trajectory coordination of multiple mobile robots," *IEEE Transactions on Automation Science and Engineering*, vol. 12, 2015.

[32] M. Phillips and M. Likhachev, "SIPP: safe interval path planning for dynamic environments," in *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pp. 5628–5635, IEEE, 2011.

[33] M. Barer, G. Sharon, R. Stern, and A. Felner, "Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem," in *Seventh Annual Symposium on Combinatorial Search*, 2014.

[34] E. Boyarski, A. Felner, R. Stern, G. Sharon, D. Tolpin, O. Betzalel, and E. Shimony, "Icbs: Improved conflict-based search algorithm for multi-agent pathfinding," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[35] L. Cohen, T. Uras, T. K. S. Kumar, H. Xu, N. Ayanian, and S. Koenig, "Improved solvers for bounded-suboptimal multi-agent path finding," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, p. 3067–3074, AAAI Press, 2016.

[36] J. Li, Z. Chen, D. Harabor, P. J. Stuckey, and S. Koenig, "MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 10256–10265, jun 2022.

[37] M. Spies, M. Todescato, H. Becker, P. Kesper, N. Waniek, and M. Guo, "Bounded suboptimal search with learned heuristics for multi-agent systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 2387–2394, jul 2019.

[38] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, "Conflict-based search with optimal task assignment," in *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '18, (Richland, SC), p. 757–765, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[39] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," in *Cooperative Robots and Sensor Networks 2015*, pp. 31–51, Springer International Publishing, 2015.

[40] A. Torreño, E. Onaindia, A. Komenda, and M. Štolba, "Cooperative multi-agent planning," *ACM Computing Surveys*, vol. 50, pp. 1–32, nov 2018.

[41] F. Martín, J. Ginés, F. J. Rodríguez, and V. Matellán, "Plansys2: A planning system framework for ros2," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - October 1, 2021*, IEEE, 2021.

[42] Fraunhofer IPA, "vda5050_msgs." https://github.com/ipa320/vda5050_msgs, 2022.

[43] Bosch Research, "mrp_bench." https://github.com/boschresearch/mrp_bench, 2022.