Autonomous Robots (2019) 43:539–555
https://doi.org/10.1007/s10514-018-9730-9

Coordination of multiple AGVs: a quadratic optimization method

Valerio Digani[1] · **M. Ani Hsieh**[2] · **Lorenzo Sabattini**[3] · **Cristian Secchi**[3]

Received: 15 February 2017 / Accepted: 2 April 2018 / Published online: 7 April 2018

Abstract

This paper presents an optimization strategy to coordinate a fleet of Automated Guided Vehicles (AGVs) traveling on ad-hoc pre-defined roadmaps. Specifically, the objective is to maximize traffic throughput of AGVs navigating in an automated warehouse by minimizing the time AGVs spend negotiating complex traffic patterns to avoid collisions with other AGVs. In this work, the coordination problem is posed as a Quadratic Program where the optimization is performed in a centralized manner. The proposed method is validated by means of simulations and experiments for different industrial warehouse scenarios. The performance of the proposed strategy is then compared with a recently proposed decentralized coordination strategy that relies on local negotiations for shared resources. The results show that the proposed coordination strategy successfully maximizes vehicle throughput and significantly minimizes the time vehicles spend negotiating traffic under different scenarios.

Keywords Multi-robot coordination · Quadratic optimization · AGV systems · Path planning

1 Introduction

Automated warehouses are becoming an increasingly popular solution for end-of-line logistics due to improved efficiency and flexibility (Sabattini et al. 2013). In order to meet the increasing delivery throughput requirements, warehouses are employing more and more Automated Guided Vehicles (AGVs) such that it is now common to find hundreds of AGVs in a single warehouse. It is therefore not surprising that multi-vehicle planning and coordination is receiving even more attention by the robotics community in the last years, especially in the context of AGVs for industrial applications.

Existing work addressing the multi-vehicle planning and coordination problem for AGVs in industrial applications includes (Zhang and Mehrjerdi 2013; Hoshino and Seki 2013; Pecora et al. 2012; Olmi et al. 2011; Secchi et al. 2015) and references therein. Recently, a hierarchical strategy for the coordination of a fleet of AGVs in an automated warehouse was proposed in Digani et al. (2014b), Digani et al. (2015b). The hierarchical strategy first partitions the robots' environment into sectors. The sequence of sectors each AGV has to cross to reach its destination is computed using a Model Predictive Control (MPC) based centralized planner in order to maximize the performance of the entire fleet. Within each sector, a decentralized strategy is used to coordinate the AGVs to provide a simple and scalable traffic strategy. Each time an AGV enters a new sector, the central unit determines the traffic level in each sector by counting the number of AGVs it contains. Using the A^* (LaValle 2006) algorithm, the sequence of sectors the AGV needs to track is recomputed based on the current traffic condition. The introduction of such a simple traffic measure, namely counting the number of AGVs within each sector, has significantly increased the performance of the traffic manager, see Digani et al. (2015b) for more details and Sabattini et al. (2015) for an industrial benchmark. The decentralized coordination within each sector is performed by means of a negotiation strat-

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-018-9730-9) contains supplementary material, which is available to authorized users.

✉ Lorenzo Sabattini
lorenzo.sabattini@unimore.it

Valerio Digani
digani.v@elettric80.it

M. Ani Hsieh
m.hsieh@seas.upenn.edu

Cristian Secchi
cristian.secchi@unimore.it

[1] Elettric80 s.p.a., Via G. Marconi 23, 42030 Viano, Italy

[2] Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, USA

[3] Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Reggio Emilia, Italy

egy among the AGVs based on a priority scheme. A shared resource, *i.e.*, a road segment, is allocated to the winner of each negotiation round, while the loser of the negotiation round must wait until the shared resource becomes free. It is worth noting that this negotiation mechanism inevitably leads the AGVs to spend most of their time waiting. Thus a drawback of these previous approaches is that vehicles crossing a sector end up spending most of their time waiting for negotiations. Thus the amount of time spent locally, namely inside each sector, dominates the overall performance of the fleet.

This paper aims at overcoming this major drawback by minimizing the number of interactions among AGVs. The idea is to minimize the amount of time vehicles spend negotiating complex traffic patterns within each sector as they avoid collisions with one another while navigating through the workspace. In this work, we propose a strategy that completely avoids negotiations by partitioning the indoor environment into several sectors. Within each sector, a local coordination strategy that maximizes the AGV throughput through the sector is formulated as a centralized optimization problem. The contribution of the paper lies in a complete formulation of the coordination problem as a quadratic optimization problem (QP), where the crossing time for the vehicles, *i.e.*, the time it takes for the vehicles to enter and leave a given sector, is minimized with respect to the vehicle velocities. Furthermore, an exhaustive experimental validation is provided, in real scenarios extracted from real automated warehouses with AGV systems. The paper extends the idea in Digani et al. (2015a) and exploits the framework presented in Digani et al. (2014b), Digani et al. (2015b). Some preliminary results in which a draft version of the method is applied to a virtual environment can be found in Digani et al. (2015a).

1.1 Related work

Coordination of groups of autonomous vehicles is a relevant topic in multi-robot systems that has been widely addressed in the literature (LaValle 2006). In general, existing approaches can be divided into two main categories: centralized versus decentralized. *Decentralized* approaches (see for instance Yang et al. 2008; Zhang and Mehrjerdi 2013) are known to scale well with the number of robots. Namely, decentralized approaches directly address the complexity of large-scale multi-robot coordination problems. In these methods, each robot autonomously determines its routes, resolving conflicts with other robots by collecting information from them (Habib et al. 2013; Jager and Nebel 2001; Zheng et al. 2013). Decentralized techniques are generally faster than centralized ones, but they present several drawbacks. While existing works, such as Fanti et al. (2015), Manca et al. (2011), provide deadlock-free strategies, decentralized approaches can fail to find valid paths for all robots and feasible solutions may be far from optimal. As such, decentralized strategies may not be suitable for many industrial applications where efficiency is paramount.

With *centralized* strategies, a single decision maker determines the entire path plan for all the robots. The main advantage of centralized approaches is their ability to find optimal solutions (LaValle and Hutchinson 1998). For this reason, fleets of AGVs in industrial applications are generally coordinated by a centralized supervisor (the control center) which manages all the information coming form the Warehouse Management System (WMS) and from the environment. The control center handles the coordination of the fleet, solving a multi-robot path planning problem (see e.g Olmi et al. 2008; Hoshino and Seki 2013). Recently, existing works have proposed dynamic strategies to coordinate a fleet of robots where the central coordinator dynamically re-computes the paths using local information (Secchi et al. 2015; Pecora et al. 2012). However, in most cases the dimension of the multi-robot configuration space may be very high and require considerable computational resources to obtain a solution.

A multi-layer structure world representation can reduce the multi-robot path planning search space. As explained in Park et al. (2012), the approach constructs a hierarchical map which can abstract the traversable areas using an adequate number of nodes and edges in a graph. The path is obtained by searching through the graphs of the various layers. Other popular solutions to reduce the search space rely on a roadmap approach (Kavraki et al. 1996; Makarem and Gillet 2012; Hui 2010), where local interaction rules or local strategies for conflict resolution are often included (Pallottino et al. 2007; Reveliotis and Roszkowska 2011). Recently, in Digani et al. (2014b), Digani et al. (2015b) a hierarchical strategy was proposed that exploits the benefits of both centralized and decentralized approaches for the coordination of an AGV fleet. Based on this approach, a dynamic traffic model was derived in Digani et al. (2016). While centralized approaches are known to scale poorly with respect to the number of agents, they are often preferred in many industrial applications because of their solution optimality guarantees which provide performance assurances. The reliance on centralized approaches is also further backed by the ever decreasing cost of fast high-end computing and the potential loss in revenue caused by deadlocks.

Lastly, path planning strategies can also be performed by employing *prioritized* schemes. Namely, the paths are sequentially computed based on the priorities of the robots in order to avoid conflicts (Cap et al. 2015; van den Berg and Overmars 2005). These approaches have been shown to be effective in practice, but they are incomplete, *i.e.*, there are solvable problem instances that the algorithms fail and thus provide no guarantees of success: completeness guarantees can be provided by appropriately structuring the operational environment (van den Berg and Overmars 2005). Other approaches include *fixed-path* coordination which aims at coordinating robots on a predefined set of paths to avoid collision (LaValle 2006). This work falls in this last category, where the vehicles are constrained to move on a predefined roadmap.

Irrespective of the approach, existing work has shown the impact of uncertainties on the overall system performance of large-scale multi-agent systems. Specifically, Mather et al. (2010), Mather and Hsieh (2010) have shown how uncertainties inherent in any systems deployed in a realistic setting can detrimentally impact the overall system performance. These uncertainties may arise from sensor/actuation noise, model parameters, and local vehicle-to-vehicle interactions can lead to ensemble effects that are difficult to diagnose without an appropriate macroscopic view of the system. In Mather et al. (2010), Mather and Hsieh (2010) showed how these detrimental impacts can be mitigated by minimizing the local vehicle-to-vehicle interactions within a given system. As such, this work focuses on maximizing traffic throughput in specific regions of the workspace and thus minimizing the local vehicle-to-vehicle interactions that can lead to detrimental ensemble effects.

Coordination along a roadmap in a specific region can be achieved using different approaches (Alami et al. 1998). In particular, complete and optimal solutions have been developed (see e.g. Peng and Akella 2005; Siméon et al. 2002; Olmi et al. 2011), but are known to be very computationally expensive. The method proposed in this work provides a solution that is only locally optimal, but is shown to have very low computational requirements.

1.2 Organization

The paper is organized as follows: the problem formulation and key assumptions to the approach are stated in Sect. 2. Section 3 describes the optimization problem and the formulation of the problem as a quadratic program. Section 4 shows the implementation of the proposed method and Sect. 5 presents simulation and experimental results. Finally Sect. 6 summarizes key results and Sect. 7 concludes with some discussions for future works.

2 Problem definition

We consider the problem of coordinating a fleet of $\bar{N}$ AGVs in an automatic warehouse. In modern AGV systems, the vehicles are constrained to move on a predefined (virtual) roadmap $\mathcal{R}$. The roadmap is partitioned into a collection of segments $\mathcal{P} = \{p_1, \ldots, p_W\}$. The tasks the AGVs have to accomplish correspond to picking some goods in a location of the roadmap and delivering it to another location. Thus, each time a task is assigned to AGV i, the path π_i the vehicle has to follow is computed. A path is simply a sequence of adjacent segments, *i.e.*, $\pi_i = \left\{p_i^1, \ldots, p_i^{M_i}\right\}$.

Each vehicle is modeled as a circle with the center on the path it is following and radius $\delta/2$, where $\delta > 0$ is chosen in order to obtain a circle sufficiently big to enclose the biggest AGV traveling along the roadmap.[1] For each segment $p \in \mathcal{P}$, we denote with $\mathcal{V}(p)$ the portion of space occupied by a vehicle moving along p, *i.e.*, the trace of the vehicle along that segment. Two AGVs traveling on two segments that are too close to each other may collide. Formally, two segments $p_i, p_j \in \mathcal{P}$ are *colliding segments* if $\mathcal{V}(p_i) \cap \mathcal{V}(p_j) \neq \emptyset$. Consequently, given a segment $p \in \mathcal{P}$, the *Collision Segment Set of p* (CSS(p)) of p is given by:

$$CSS(p) = \{\rho \in \mathcal{P}, \rho \neq p \text{ s.t. } \mathcal{V}(p) \cap \mathcal{V}(\rho) \neq \emptyset\} \tag{1}$$

As shown in Digani et al. (2014a), Beinschob et al. (2017), it is possible to automatically build a roadmap and a partition of the warehouse into a finite set of sectors $S = \{S_1, \ldots, S_Q\}$ such that each sector contains at most an intersection area, *i.e.*, a portion of the roadmpap where the AGVs need to coordinate their motion in order to avoid collisions. In this paper we will consider a constant partitioning of the environment. In Sabattini et al. (2016) an algorithm for the dynamic partitioning of the warehouse has been proposed.

Consider the sector S_σ, where $\sigma \in \{1, \ldots, Q\}$. The set of segments contained in S_σ is given by $\mathcal{P}_\sigma = \bigcup_{i=1}^{\bar{N}} (\pi_i \cap S_\sigma)$. The intersection area (that can be empty) of S_σ is given by

$$\mathcal{A}_\sigma = \bigcup_{p \in \mathcal{P}_\sigma} CSS(p) \tag{2}$$

An example of the partitioning of a real workspace is shown in Figs. 1 and 2 highlights segments and paths within a sector.

Given a set of $\bar{N}$ tasks to complete and, consequently, $\bar{N}$ paths the AGVS have to track, the following metrics are used for evaluating the performance of the AGV system.

- *Completion Time* the time needed by one AGV for tracking its assigned path
- *AGV-Time* the time necessary for tracking all the paths, *i.e.*, the sum of all the completion times

The sector partitioning can be exploited to deal with the complexity of the AGV coordination problem. This can be accomplished using hierarchical approaches where the coordination problem is solved within each sector and the planning problem is solved between sectors. For example, in Digani et al. (2015b), the traffic manager works on two layers.

[1] This condition is usually not conservative. In fact, in an automated warehouse, all the AGVs have typically the same size.

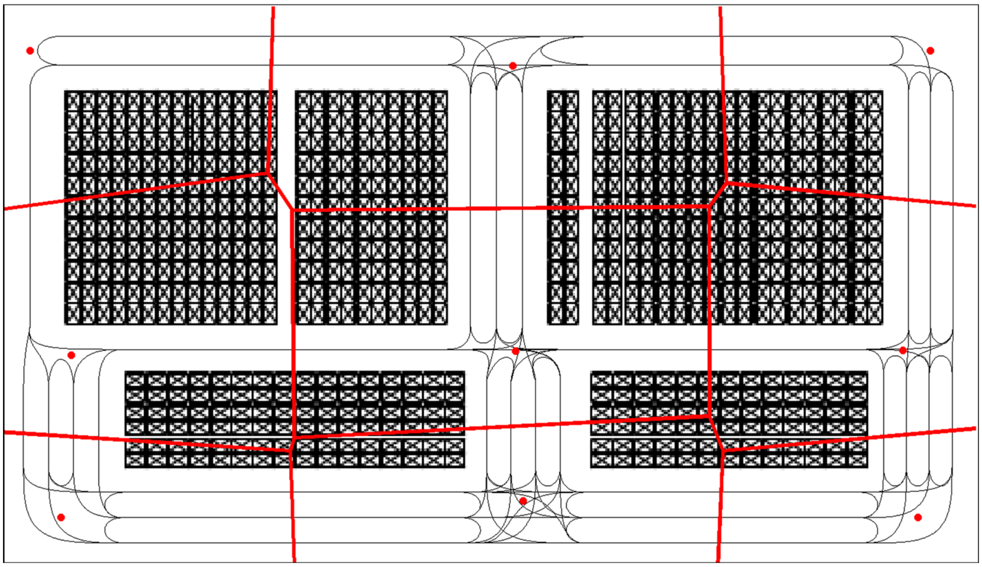


Fig. 1 Roadmap and sectors in a real environment: black areas are obstacles, red lines represents sectors' borders and red dots are the centres of the sectors (Color figure online)

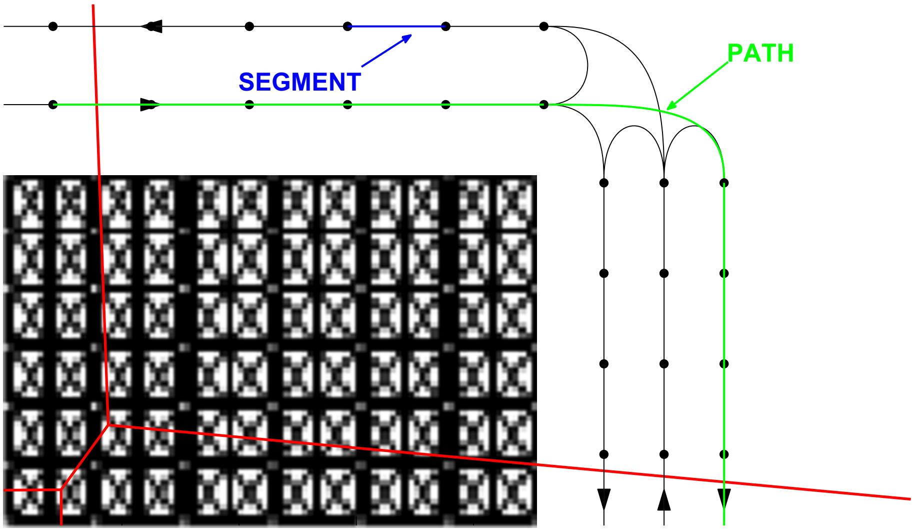


Fig. 2 Definitions of path and segment for a sector

The first layer performs a high level finite-horizon centralized planning that computes the sequence of sectors each AGV has to cross to reach its destination using the A^* algorithm on the sectors. The second layer represents a low level decentralized coordination strategy for safe navigation inside each sector. Thus, coordination is only required within each sector, and in particular within at most one intersection area, where the number of interactions among the agents is given by the number of local negotiations among them. A shared resource, *i.e.*, a segment, is then allocated to the winner of each negotiation round, while the losers of the negotiation round must wait until the shared resource becomes free. It is worth noting that the results of these negotiations are not known a priori and that they depend on the paths and on possible priorities of the AGVs. Thus, the system is not deterministic and the traffic control and management is not trivial. In particular, the negotiation process affects the total time a vehicle spends traversing a given sector.

The goal of this paper is to propose a novel coordination strategy for AGVs crossing an intersection area. The proposed methodology aims at determining for each AGV the best speed it should travel along its path in order to minimize the total crossing time while avoiding conflicts (*i.e.*, possible collisions) with other vehicles. This will allow rid the system of non deterministic negotiations or of other heuristics for shared resource allocations.

Coordination happens in sectors, which contain at most one intersection area. Thus, the problem of coordinating $\bar{N}$ AGVs moving on $\mathcal{R}$ can be recast into the problem of coordinating $N \leq \bar{N}$ AGVs moving inside a generic sector.

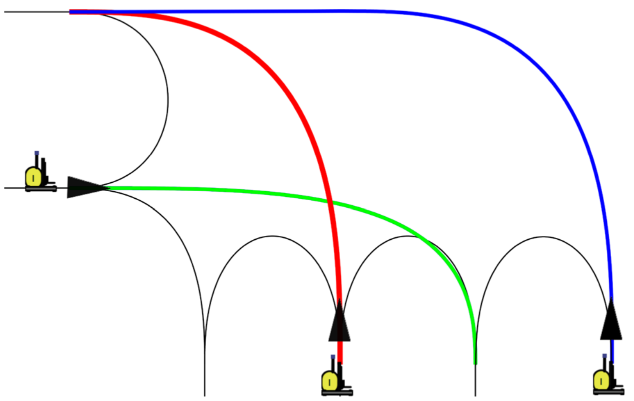


Fig. 3 Multi AGV coordination problem in an intersection area of a sector

Improving the coordination performance inside a sector will lead to an improvment of the coordination performance of the overall fleet on the roadmap.

Formally, we aim at solving the following problem

Problem 1 Given a group of $N \leq \bar{N}$ AGVs crossing a sector on paths $\{\pi_1, \ldots, \pi_N\}$, define the velocity profile of each AGV in order to minimize the AGV-Time necessary for completing the tasks while avoiding conflicts among the AGVs.

Figure 3 shows a simple example of the problem where three AGVs have to follow the assigned paths while avoiding collisions. Each AGV starts in an initial position along its path and has to reach its own final position.

We will make the following assumptions:

A1 No external obstacles (e.g. people, manual forklifts) can be found on the roadmap.

A2 The velocity along a segment is constant

A3 The velocity of each AGV is bounded:

$$0 < V_{min} < \nu < V_{max} < \infty \tag{3}$$

where ν represents the velocity of an AGV and V_{min} and V_{max} are the lower and upper velocity bounds respectively.[2]

A4 Each segment can be occupied by at most one AGV

A5 Each AGV has a different pair of initial and final positions

We would like to remark that these assumptions do not introduce significant limitations to the applicability of the proposed strategy. In particular, Assumption A1 is generally verified, during normal operations, since human operators and manual forklifts are not allowed to occupy areas traveled by AGVs: their presence on the AGVs' path is handled by safety systems.

Assumption A2 is imposed for the sake of simplicity of explanation: while it would be possible to extend the proposed approach considering acceleration and deceleration along each segment, we prefer to keep the explanation simple, and consider only the average velocity. Conversely, Assumption A3 is trivially verified for any real AGV.

Assumption A4 is generally imposed in any AGV system used for industrial logistics, for safety and coordination reasons: in fact, it helps in guaranteeing collision avoidance, and it simplifies the coordination problem.

Finally, Assumption A5 is, in general, trivially verified, thanks to the redundancy of roadmaps typically utilized in industrial environments.

3 Modeling and optimization problem

In this section we show how to model the AGV coordination problem (Problem 1) as a QP problem. The output is a set of velocities that guarantees a safe minimum distance between all agents and that minimizes the total time the AGVs need to accomplish the tasks.

Let $\pi_i = \left\{p_i^1, \ldots, p_i^{M_i}\right\}$, $i = 1, \ldots, N$, be the path the i-th AGV has to track inside the sector. Let d_i^k and v_i^k be the length of p_i^k and the velocity of the AGV i along segment p_i^k respectively. Since, according to Assumption A2, v_i^k is constant, the completion time for AGV i to track π_i is given by:

$$\Delta t_i = \sum_{k=1}^{M_i} \frac{d_i^k}{v_i^k} \tag{4}$$

and the AGV-Time is given by

$$\Delta T = \sum_{i=1}^{N} \sum_{k=1}^{M_i} \Delta t_i = \sum_{i=1}^{N} \sum_{k=1}^{M_i} \frac{d_i^k}{v_i^k}. \tag{5}$$

Let $\bar{M} = M_1 + \cdots + M_N$ be the total number of segments to be crossed by N AGVs and let $\mathbf{v} \in \mathbb{R}^{\bar{M}}$ and $\mathbf{d} \in \mathbb{R}^{\bar{M}}$ be defined as

$$\mathbf{v} = \left(v_1^1, \ldots, v_1^{M_1}, \ldots, v_N^{M_N}\right)^T \tag{6}$$

$$\mathbf{d} = \left(d_1^1, \ldots, d_1^{M_1}, \ldots, d_N^{M_N}\right)^T \tag{7}$$

Moreover, let $v_i \in \mathbb{R}$ and $d_i \in \mathbb{R}$, $i = 1, \ldots, \bar{M}$, denote the i-th component of $\mathbf{v}$ and $\mathbf{d}$, respectively.

Minimizing the AGV-Time corresponds to finding a set of velocities each AGV has to track the segments of its paths with. Formally

[2] The bounds may depend on the AGV and on the segment to be crossed. In order to keep the notation simple, we have considered constant values. All the results obtained in the paper can be easily extended to variable bounds.

$$\underset{\mathbf{v}}{\operatorname{argmin}} \Delta T = \underset{\mathbf{v}}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{k=1}^{M_i} \frac{d_i^k}{v_i^k} \tag{8}$$

We introduce now a new vector of optimization variables $\phi \in \mathbb{R}^{\bar{M}}$ defined as follows:

$$\phi = \left(\frac{1}{v_1^1}, \ldots, \frac{1}{v_1^{M_1}}, \ldots, \frac{1}{v_N^{M_N}} \right)^T \tag{9}$$

Moreover, let $\phi_i \in \mathbb{R}$, $i = 1, \ldots, \bar{M}$, denote the i-th component of ϕ. Clearly, finding optimal the value of ϕ leads to directly defining the corresponding optimal value of $\mathbf{v}$.

The functional to minimize is the weighted sum of terms depending only on one of the variables ϕ_i to optimized, and the weights are represented by the elements of vector $\mathbf{d}$. Hence, solving (8) can be written as the problem of minimizing the following linear objective function:

$$f(\phi) = -\sum_{i=1}^{\bar{M}} d_i \phi_i . \tag{10}$$

We consider two constraints for the optimization problem. The first is related to the boundedness of the velocities and the second is the maintenance of a safe distance between the AGVs. Considering (3), we have that each component of ϕ is bounded as:

$$\Phi_{min} \leq \phi_i \leq \Phi_{max}, \quad \forall i = 1, \ldots, \bar{M} \tag{11}$$

which can be represented as a linear constraint with respect to ϕ. According to (3), $\Phi_{min} = {}^{1}\!/_{V_{max}}$ and $\Phi_{max} = {}^{1}\!/_{V_{min}}$.

As remarked in Sect. 2, each AGV is modeled as a circle with radius $\delta/2$. Let the center of the circle representing the i-th AGV be denoted by $\mathcal{X}_i(t) = [x_i(t), y_i(t)]^T$, where $[x_i(t), y_i(t)]^T$ is the cartesian position of the vehicle. Collision avoidance is guaranteed by ensuring that the relative distance $\gamma_{i,j}$ between any two AGVs is greater than δ, *i.e.*:

$$\begin{aligned} \gamma_{i,j}(t) &= ||\mathcal{X}_i(t) - \mathcal{X}_j(t)|| \\ &= \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} > \delta \\ &\quad i \neq j \in \{1, \ldots, N\} \end{aligned} \tag{12}$$

This is a non convex constraint that needs to be imposed for all the pairs of AGVs all over their paths. The non convexity of (12) makes the optimization problem too complex to be solved continuously for each sector, especially if the number of AGVs is quite big. Luckily, the constraint in (12) can be reformulated as a quadratic constraint, which leads to a computationally efficient solution.

Let each path π_i be parameterized by a curvilinear abscissa $s_i \in [0, d_i]$, where $d_i = \sum_{k=1}^{M_i} d_i^k$, and indicate, with a slight abuse of notation, with $\pi_i(s_i)$ the cartesian position (x, y) on path π_i corresponding to the value of the curvilinear abscissa s_i. We denote with:

$$\Delta X_{i,j} = \left\{ s_i \mid \exists s_j \text{ such that } ||\pi_i(s_i) - \pi_j(s_j)|| \leq \delta \right\}. \tag{13}$$

the portion of the of path π_i where a collision between AGV i and AGV j can happen on path i. It is worth noting that the collision regions can be computed totally off-line since they are based solely on geometric information regarding the roadmap and the size of the vehicles.

Let s_{ij}^{min} and s_{ij}^{max} be the values of the curvilinear abscissa corresponding to the beginning of the segment that contains $min \Delta X_{ij}$ and the end of the segment that contains $max \Delta X_{ij}$. Thus, the portion $[s_{ij}^{min}, s_{ij}^{max}]$ represents the smallest subset of segments of π_i containing the collision region $\Delta X_{i,j}$.

In order to keep the notation simple, we will hereafter consider that ΔX_{ij} is a single connected sub-interval of $[0, d_i]$. While this may not be true when considering the paths along the whole roadmap, this is always true in practice when considering a single intersection area in a sector. All the results of the paper can be straightforwardly generalized to the case in which $\Delta X_{i,j}$ is a disconnected subset. In particular, this can be achieved, for instance, splitting the paths into sub-paths that contain at most a connected component of $\Delta X_{i,j}$, and then apply the results that will be derived in the sequel to each pair of intersecting sub-paths.

Furthermore, let $\omega_{i,j}^{min}$ and $\omega_{i,j}^{max}$ be the time instants at which AGV i reaches $\pi_i\left(s_{ij}^{min}\right)$ and $\pi_i\left(s_{ij}^{max}\right)$ respectively. Thus, AGV i can collide with AGV j in the time interval $\Omega_{i,j} = \left[\omega_{i,j}^{min}, \omega_{i,j}^{max}\right]$. Similarly, it is possible to define the portion of the path π_j where a collision between AGV j and AGV i can happen on π_j and, consequently, the time interval $\Omega_{j,i}$ where a collision between AGV j and AGV i can happen. Collision avoidance between the AGV i and AGV j is verified only if:

$$\Omega_{i,j} \cap \Omega_{j,i} = \emptyset. \tag{14}$$

Figure 4 pictorially depicts the described problem: with an abuse of notation, the vertical axis represents both the parameter s_i and s_j. The condition (14) implies that the set $\Omega_{i,j}$, $\Omega_{j,i}$, denoted by the red and blue line segments on the time axis in Fig. 4, are disjoint.

It is possible to reformulate (14) using the midpoints of the sets $\Omega_{i,j}$, $\Omega_{j,i}$. In particular the distance between two mid-points has to be grater than the sum of the distances between the mid-point and one extreme point of both the sets. Formally, we define

$$\begin{aligned} \alpha_{i,j} &= \omega_{i,j}^{max} + \omega_{i,j}^{min} \\ \beta_{i,j} &= \omega_{i,j}^{max} - \omega_{i,j}^{min} \end{aligned}$$

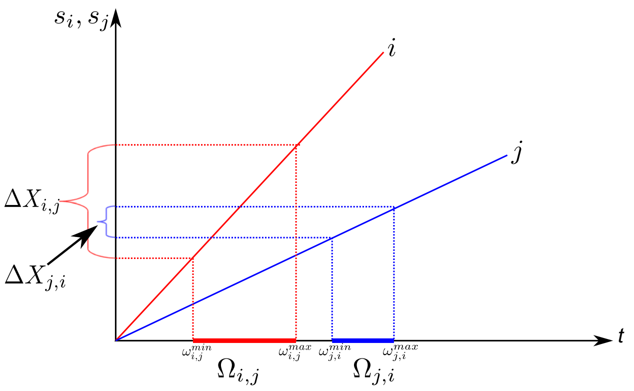


Fig. 4 Coordination space in the case of two agents: the lines represent the velocities of the AGVs

The condition described by (14) can then be formalized as follows:

$$\left|\frac{\alpha_{i,j}}{2} - \frac{\alpha_{j,i}}{2}\right| > \frac{\beta_{i,j}}{2} + \frac{\beta_{j,i}}{2} \tag{15}$$

which simplifies to

$$|\alpha_{i,j} - \alpha_{j,i}| > \beta_{i,j} + \beta_{j,i} \tag{16}$$

and squaring (16), the constraint becomes quadratic:

$$(\alpha_{i,j} - \alpha_{j,i})^2 > \left(\beta_{i,j} + \beta_{j,i}\right)^2 . \tag{17}$$

Let us now introduce the following Proposition which will show that (17) is quadratic with respect to ϕ. This will be instrumental in the formulation of the problem as a QP.

Proposition 1 *The constraint in* (17) *can be represented in the following form*

$$\left(\phi_i^T, \phi_j^T\right)^T \Lambda_{ij} \left(\phi_i^T, \phi_j^T\right) > 0$$

for an appropriately defined matrix Λ_{ij}.

Proof Let $\left\{p_i^{m_i}, \ldots p_i^{m_i+n_i}\right\}$ and $\left\{p_j^{m_j}, \ldots p_j^{m_j+n_j}\right\}$, where $m_i, m_j, n_i, n_j \geq 1$, $m_i + n_i \leq M_i$ and $m_j + n_j \leq M_j$, be the set of segments containing ΔX_{ij} and ΔX_{ji} respectively. Considering that the velocity of the AGVs is constant on each segment, we can write:

$$\alpha_{i,j} = \sum_{k=1}^{m_i+n_i} \frac{d_i^k}{v_i^k} + \sum_{k=1}^{m_i-1} \frac{d_i^k}{v_i^k} = \sum_{k=1}^{m_i+n_i} a_{ik} \frac{1}{v_i^k} \tag{18}$$

where

$$a_{ik} = \begin{cases} 2d_i^k & \text{if} \quad k < m_i \\ d_i^k & \text{if} \quad k \geq m_i \end{cases} \tag{19}$$

Similarly,

$$\alpha_{j,i} = \sum_{k=1}^{m_j+n_j} a_{jk} \frac{1}{v_j^k} \tag{20}$$

where

$$a_{jk} = \begin{cases} 2d_j^k & \text{if} \quad k < m_j \\ d_j^k & \text{if} \quad k \geq m_j \end{cases} \tag{21}$$

It is convenient to write (18) and (20) in matrix form as:

$$\begin{aligned} \alpha_{i,j} &= \mathbf{1}^T_{(m_i+n_i)} A_i \phi_i \\ \alpha_{j,i} &= \mathbf{1}^T_{(m_j+n_j)} A_j \phi_j \end{aligned} \tag{22}$$

where $\mathbf{1}_\rho$ indicates a vector of ones of dimension ρ. Furthermore,

$$A_i = diag(a_{i1}, \ldots, a_{i(m_i+n_i)})$$

and

$$\phi_i = \left(\frac{1}{v_i^1}, \ldots, \frac{1}{v_i^{(m_i+n_i)}}\right)^T$$

The quantities A_j and ϕ_j are defined analogously. Thus, we can write

$$\begin{aligned} (\alpha_{i,j} - \alpha_{j,i})^2 = &\left(1^T_{(m_i+n_i)} A_i \phi_i - 1^T_{(m_j+n_j)} A_j \phi_j\right)^T \\ &\times \left(1^T_{(m_i+n_i)} A_i \phi_i - 1^T_{(m_j+n_j)} A_j \phi_j\right) \end{aligned} \tag{23}$$

By straightforward computations we can write:

$$(\alpha_{i,j} - \alpha_{j,i})^2 = \phi_{ij}^T M_{ij} \phi_{ij} \tag{24}$$

where $\phi_{ij} = \left(\phi_i^T, \phi_j^T\right)^T$ and

$$M_{ij} = \begin{pmatrix} A_i \mathbf{1}_{m_i+n_i} \mathbf{1}^T_{m_i+n_i} A_i & -A_i \mathbf{1}_{(m_i+n_i)} \mathbf{1}^T_{(m_j+n_j)} A_j \\ -A_j \mathbf{1}_{(m_j+n_j)} \mathbf{1}^T_{(m_i+n_i)} A_i & A_j \mathbf{1}_{m_j+n_j} \mathbf{1}^T_{m_j+n_j} A_j \end{pmatrix} \tag{25}$$

Similarly, we can write:

$$\beta_{i,j} = \sum_{k=1}^{m_i+n_i} \frac{d_i^k}{v_i^k} - \sum_{k=1}^{m_i-1} \frac{d_i^k}{v_i^k} = \sum_{k=1}^{m_i+n_i} b_{ik} \frac{1}{v_i^k} \tag{26}$$

where

$$b_{ik} = \begin{cases} 0 & \text{if} \quad k < m_i \\ d_i^k & \text{if} \quad k \geq m_i \end{cases} \tag{27}$$

and

$$\beta_{j,i} = \sum_{k=1}^{m_j+n_j} b_{jk} \frac{1}{v_j^k} \tag{28}$$

where

$$b_{jk} = \begin{cases} 0 & \text{if} \quad k < m_j \\ d_j^k & \text{if} \quad k \geq m_j \end{cases} \tag{29}$$

Thus, using a vector notation as for obtaining (25), we can write:

$$(b_{i,j} + b_{j_i})^2 = \phi_{ij}^T N_{ij} \phi_{ij} \tag{30}$$

with

$$N_{ij} = \begin{pmatrix} B_i \mathbf{1}_{m_i+n_i} \mathbf{1}_{m_i+n_i}^T B_i & B_i \mathbf{1}_{(m_i+n_i)} \mathbf{1}_{(m_j+n_j)}^T B_j \\ B_j \mathbf{1}_{(m_j+n_j)} \mathbf{1}_{(m_i+n_i)}^T B_i & B_j \mathbf{1}_{m_j+n_j} \mathbf{1}_{m_j+n_j}^T B_j \end{pmatrix} \tag{31}$$

where $B_i = diag(b_{i1}, \ldots, b_{i(m_i+n_i)})$ and $B_j = diag(b_{j1}, \ldots, b_{j(m_j+n_j)})$.

Thus, we can rewrite (17) as:

$$\phi_{ij}^T (M_{ij} - N_{ij}) \phi_{ij} > 0 \tag{32}$$

Define now matrix $\Lambda_{ij} \in \mathbb{R}^{(m_i+n_i+m_j+n_j)\times(m_i+n_i+m_j+n_j)}$ as follows:

$$\Lambda_{ij} = M_{ij} - N_{ij} \tag{33}$$

The inequality in (32) can then be rewritten as follows:

$$\phi_{ij}^T \Lambda_{ij} \phi_{ij} > 0 \tag{34}$$

Since $\phi_{ij} = \left(\phi_i^T, \phi_j^T\right)^T$, then the statement is proven. □

3.1 Quadratic constraints linear programming

The coordination within a single sector is modeled as an optimization problem using a linear objective function, and a set of quadratic constraints with linear constraints on the boundary.

For this purpose, consider the inequality in (34). Matrix Λ_{ij} can be partitioned as follows:

$$\Lambda_{ij} = \begin{pmatrix} \Lambda_{ij}^{ii} & \Lambda_{ij}^{ij} \\ \Lambda_{ij}^{ji} & \Lambda_{ij}^{jj} \end{pmatrix} \tag{35}$$

where $\Lambda_{ij}^{ii} \in \mathbb{R}^{(m_i+n_i)\times(m_i+n_i)}$, $\Lambda_{ij}^{jj} \in \mathbb{R}^{(m_j+n_j)\times(m_j+n_j)}$, and $\Lambda_{ij}^{ij} = {\Lambda_{ij}^{ji}}^T \in \mathbb{R}^{(m_i+n_i)\times(m_j+n_j)}$.

The optimization problem is then given by:

$$\text{minimize} \quad f(\phi) \tag{36a}$$

$$\text{subject to} \quad A\phi \leq b \tag{36b}$$

$$\text{and} \quad \phi^T H_{ij} \phi \leq 0 \quad \forall i, j = 1, \ldots, \bar{M}, \quad \text{with} \quad j > i \tag{36c}$$

Matrix $A \in \mathbb{R}^{2\bar{M}\times\bar{M}}$ and vector $b \in \mathbb{R}^{2\bar{M}}$ in the linear constraint (36b) encode the bounds on the elements of ϕ defined in (11).

The quadratic constraint (36c) represents the collision avoidance constraint, according to Proposition 1: matrix $H_{ij} \in \mathbb{R}^{\bar{M}\times\bar{M}}$ represents the inequality (32). It is defined as a block matrix, whose block (h, k) is given by

$$H_{ij}^{hk} = \begin{cases} \Lambda_{ij}^{ii} & \text{if} \quad h = i, k = i \\ \Lambda_{ij}^{jj} & \text{if} \quad h = j, k = j \\ \Lambda_{ij}^{ij} & \text{if} \quad h = i, k = j \\ \Lambda_{ij}^{ji} & \text{if} \quad h = j, k = i \\ \mathbb{O} & \text{otherwise} \end{cases} \tag{37}$$

where $\mathbb{O}$ is a zero matrix of appropriate dimension.

4 Implementation

The proposed method aims at coordinating a fleet of agents on a map composed by several sectors. For this reason, each sector is provided with a dedicated process which manages the optimization routine in order to implement the proposed methodology. The optimization algorithm runs independently in each sector whenever a new agent enters or leaves the intersection area.

Some variations are introduced in order to implement the methodology among several sectors and thus among several intersections. The methodology proposed in the previous sections is based on the assumption that all the vehicles are at the initial position of their paths in the sector. The optimization algorithm runs whenever a new agent enters or leaves the intersection area. Thus, in general, at each computation of the algorithm a vehicle i can be at any position along its path, not necessary the initial position (where $s_i = 0$).

Algorithm 1 shows the actual implemented method where $\mathcal{Q}_{new}(\mathcal{A}_q)$ represents the current list of vehicles contained the intersection area $\mathcal{A}_q$, and $\mathcal{Q}_{old}(\mathcal{A}_q)$ represents list of vehicles at the previous iteration of the algorithm. The starting position of each vehicle along its path, namely the current position at the instant of iteration of the algorithm, is collected in the vector $init(\mathcal{A}_q)$. Finally $\mathbf{v}$ is the vector of the optimized velocities, obtained by the solution ϕ of the optimization problem. Algorithm 1 is performed at each sample time τ.

Algorithm 1: Optimized Coordination Main Routine

```
foreach $\mathcal{A}_q$ do
    if $\mathcal{Q}_{new}(\mathcal{A}_q) \neq \mathcal{Q}_{old}(\mathcal{A}_q)$ then
        $\mathcal{Q}_{old}(\mathcal{A}_q) = \mathcal{Q}_{new}(\mathcal{A}_q)$;
        get $init(\mathcal{A}_q)$ $\forall$ paths $\pi \in \mathcal{A}_q$;
        $\phi \leftarrow$ solve (36);
        $v_i = \dfrac{1}{\phi_i} \quad \forall i$;
    end
    else
        continue;
    end
end
```

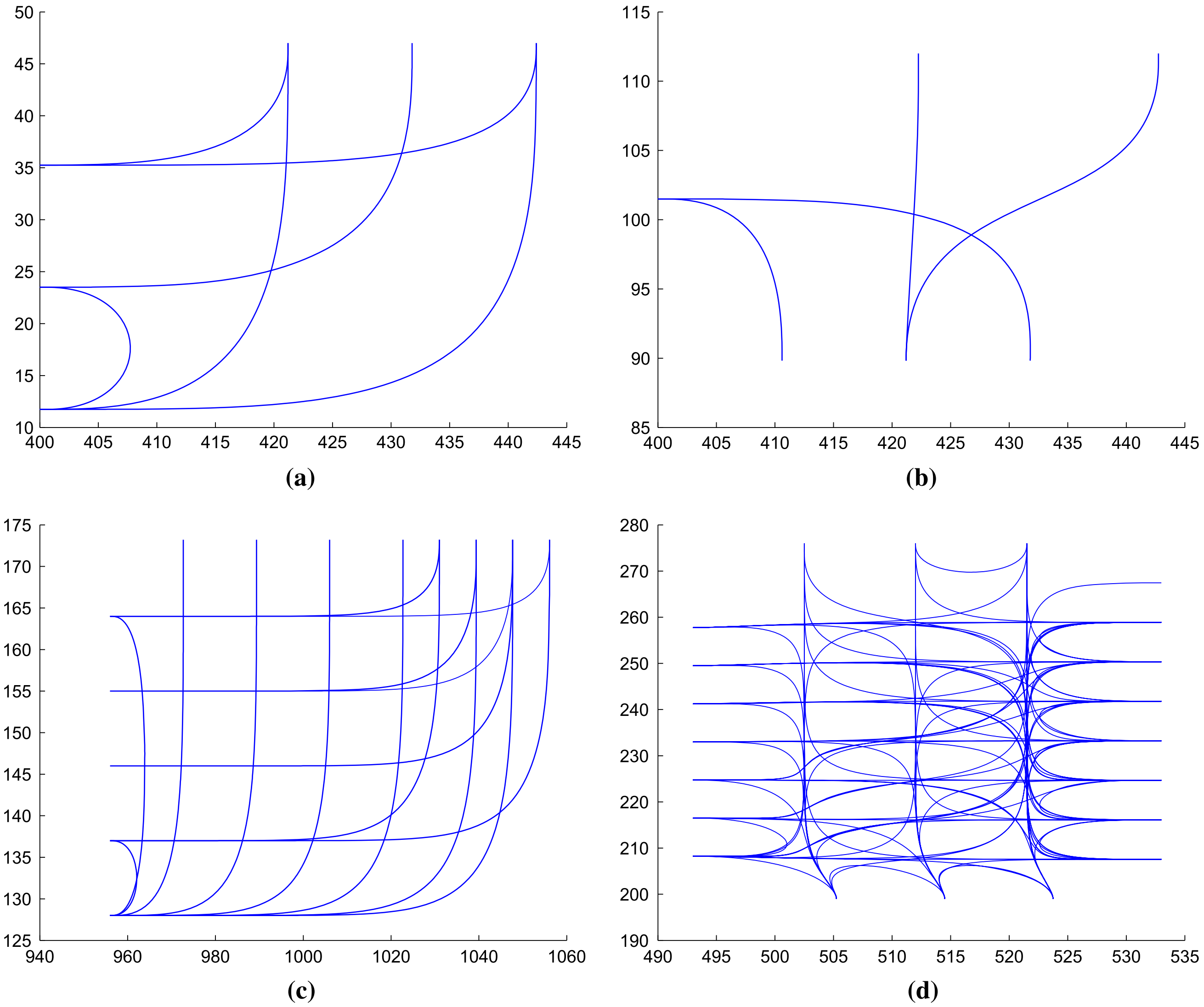
Algorithm 2 describes the single implemented controller for the i-th vehicle. It is worth noting that each AGV is controlled in a decentralized manner. Thus, each AGV entering a new intersection area $\mathcal{A}_q$ updates its position on the vector $init(\mathcal{A}_q)$ in order to get the correct and optimized velocity vector

$$\mathbf{v}(i) = \left[v_i^1, \ldots, v_i^{M_i}\right].$$

The i-th AGV stays in the queue $\mathcal{Q}_{old}(\mathcal{A}_q)$ and $\mathcal{Q}_{new}(\mathcal{A}_q)$ until it leaves the intersection area.

5 Validation

The proposed optimized coordination is validated by means of comparison with the coordination strategy first presented in Digani et al. (2014b) which relies on local negotiations. Hereafter we refer to the current proposed methodology as the *Optimized Strategy*, and to the other one as *Negotiated*



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

Fig. 5 Different intersections used during the simulations. **a** Intersection 1, **b** intersection 2, **c** intersection 3, **d** intersection 4

Algorithm 2: Agent's controller

```
Data: AGV $i$
while do
    if $i$ enters $\mathcal{A}_q$ then
        $\mathcal{Q}_{new}(\mathcal{A}_q) \leftarrow i$;
        get $\mathbf{v}(i)$;
    end
    else if $i$ is moving within $\mathcal{A}_q$ then
        update $init(\mathcal{A}_q)$ of $i$ along path $\pi_i$;
        get $\mathbf{v}(i)$;
    end
    else
        remove $i$ from $\mathcal{Q}_{old}(\mathcal{A}_q)$;
        remove $i$ from $\mathcal{Q}_{new}(\mathcal{A}_q)$;
    end
end
```

Strategy. A decoupled optimal priority scheme (Bennewitz et al. 2001) is applied to the *Negotiated Strategy* in order to obtain a better comparison.

The simulations are performed on different single intersections extracted form the roadmap of real warehouses, as shown in Fig. 5. Subsequently experimental tests are performed on a complete real environment. Figure 6a shows a simple real environment composed by four rectangular obstacles and nine sectors. A roadmap is built for that environment using the algorithm described in Digani et al. (2014a).

5.1 Simulations

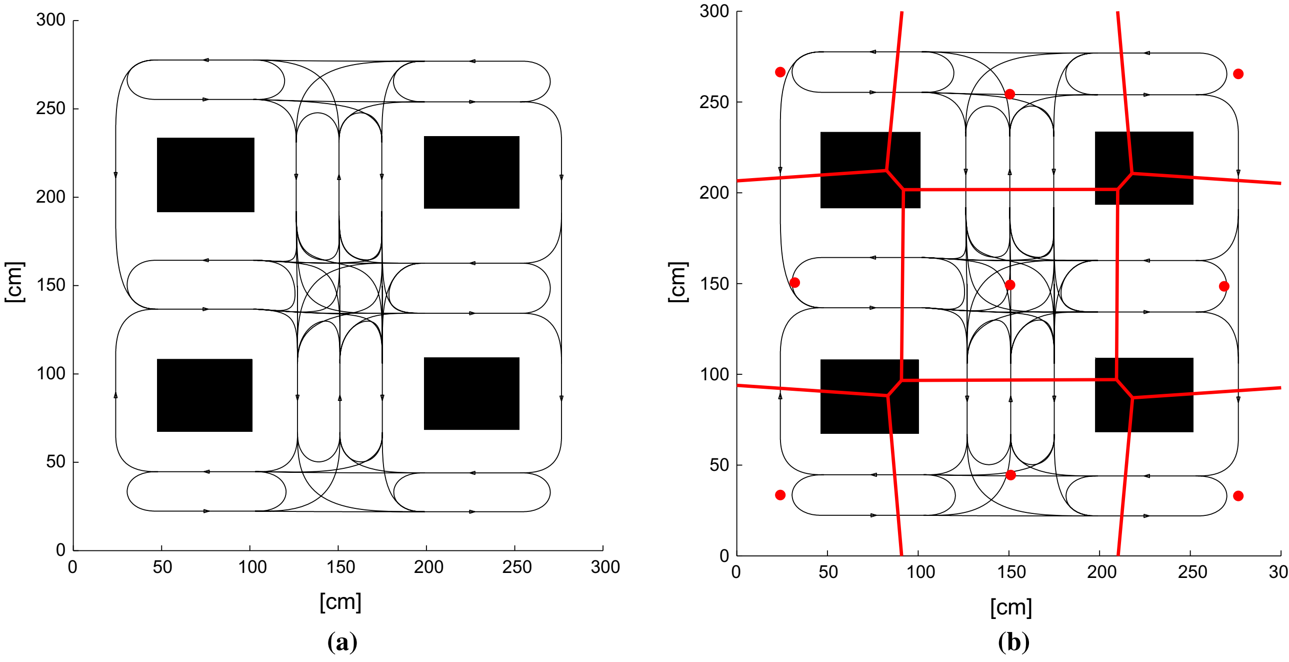
The topological complexity of the roadmaps is different in each scenario which depends on the number of possible paths, the number of possible interactions, and physical size of the environment. Repeated tests have been conducted under the following conditions:

- 4 topology of intersection
- number of AGVs $\in [2, 11]$,
- the simulation stops when the intersection area is cleared,
- the priorities (Digani et al. 2014b) for the *Negotiated Strategy* are optimized each time,
- the same paths assigned to the AGVs are considered for the comparison, and
- the paths are assigned randomly.

Ten simulation runs were performed for each configuration.

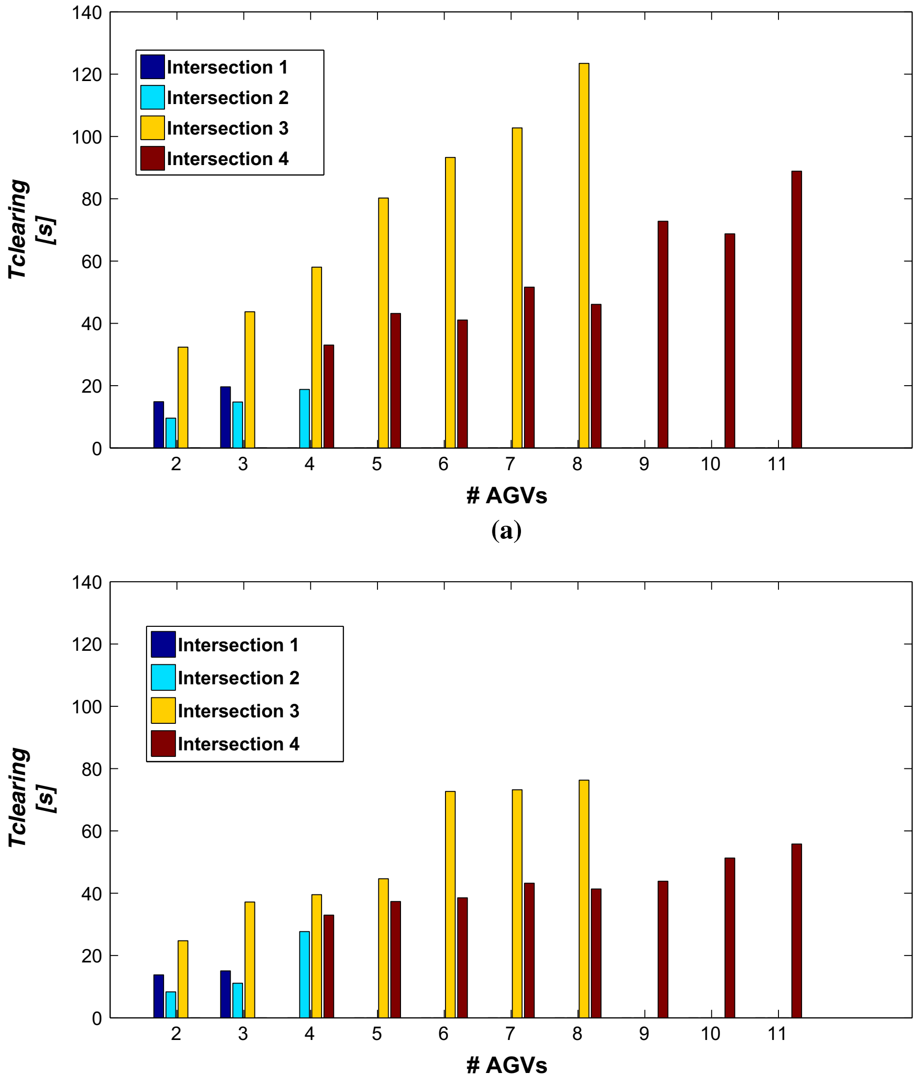
To compare, we consider the time needed for all AGVs to clear the sector or the maximum of the crossing time ($t_{clearing}$). We also considered the worst waiting time (t_{wait}), the average waiting time ($\bar{t}_{wait}$), and the computational time needed to obtain a solution (t_{calc}). The waiting time is the time that the AGVs have to wait in the same position, with zero velocity, when yielding to other robots with higher priorities at an intersection in the roadmap. The worst waiting time is the maximum waiting time for all the AGVs in the sector. The computational time is the actual time to compute a solution to the centralized optimization problem. The worst waiting time and the average waiting time are computed for the *Negotiated Strategy*, while the computational time is computed for the *Optimized Strategy*.

The simulations were performed in Matlab with the standard optimization solver. The results are summarized in Fig. 7.



(a) **(b)**

Fig. 6 Real map used in the simulations. **a** Real map, **b** sectors



(a)

(b)

Fig. 7 Average of the clearing time versus number of AGVs in the different single intersections on 10 runs. **a** Negotiated strategy, **b** Optimized strategy

5.2 Experiments

The proposed approach has been implemented on a real indoor environment. The experimental environment and the implemented roadmap are shown in Fig. 8. The roadmap is divided into several sectors as shown in Fig. 6a.

The robots are differential-drive robots equipped with an onboard processor and odometry, an RGB-D sensor, and wifi networking capabilities. Localization for the vehicles is provided by an external motion capture system. Figure 9 shows the implementation architecture of the system. The core of the system, namely the *Optimized Coordination*, is implemented in the Matlab workspace. Then each robot is controlled by means of an independent ROS (Robot Operating System Quigley et al. 2009) module which manages the tracking of the trajectories of the roadmap.

The experiments aim at validating the proposed optimized coordination method in a real scenario. As such a mixed-reality approach was employed where additional simulated robots are deployed with the real ones in order to increase the number of the fleet and thus the number of inter-robot interactions. The validation has been conducted under the following experimental set-up:

- 5 real robots
- 3 to 15 simulated robots
- tasks randomly assigned both to real and simulated robots

A representative instance of the performed experiments is shown in the accompanying video.

The main difference compared to the single intersection simulation is that the coordination of the fleet is based on the hierarchical control architecture presented in Digani et al.

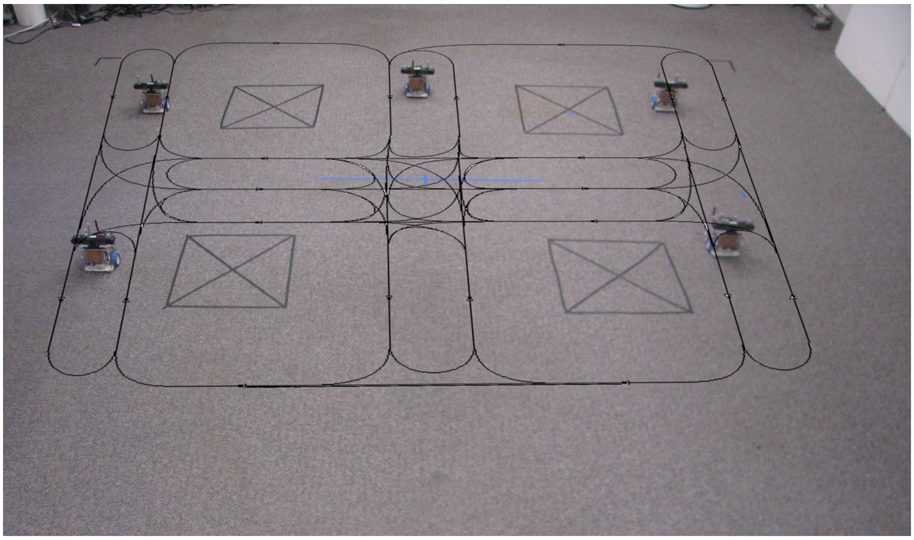


Fig. 8 Real environment used for the experimental validation

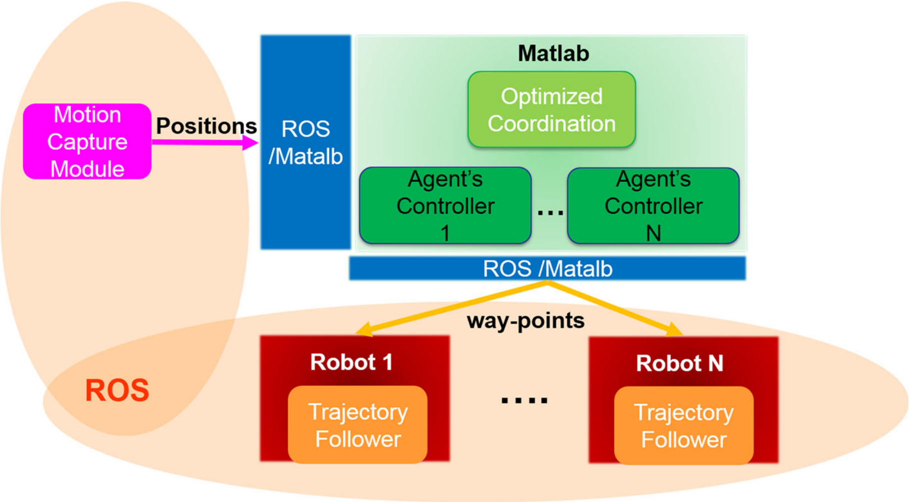


Fig. 9 Implementation and architecture of the system

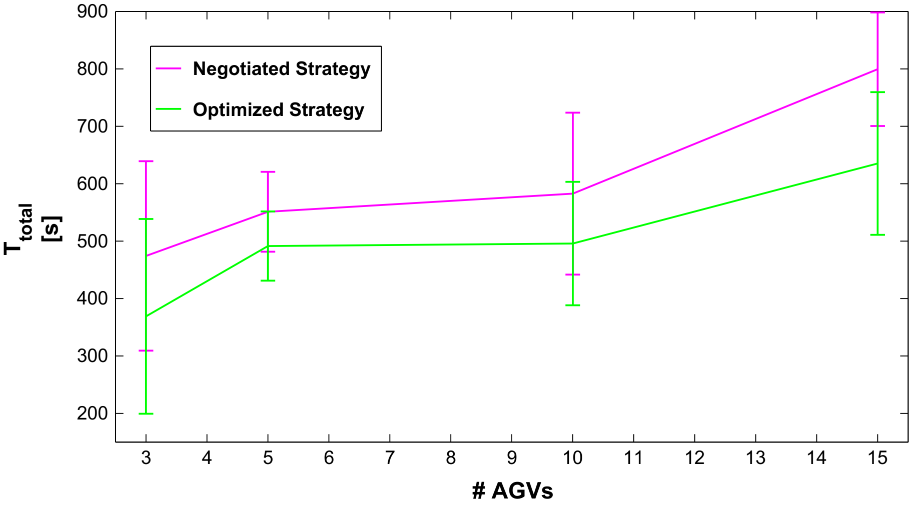


Fig. 10 Total time versus number of agents in the complete map with standard deviation

Table 1 Statistical validity of the comparison

#AGV	h	p-value
3	1	0.0346
5	1	0.0422
10	1	0.0163
15	1	0.0249

(2014b). The proposed method (*Optimized Strategy*) is still compared to the *Negotiated Strategy* with respect to the time needed for all AGVs to reach their final destination, that is t_{total}. Figure 10 shows the trend of the t_{total} with respect the number of AGVs on both of the strategies.

The data was also evaluated in terms of significance. Table 1 shows the statistical evaluation where $h = 1$ indicates a rejection of the null hypothesis at the 5% significance level and $h = 0$ indicates a failure to reject the null hypothesis at the 5% significance level.

6 Discussion

The simulations for a single intersection show that the average clearing time (or the maximum crossing time) is always smaller using the *Optimized Strategy* compared to the *Negotiated Strategy*. The results clearly show the proposed methodology performs better than the previous one. Figure 11 shows the relationship between the waiting time and the time a vehicle is actually moving (*i.e.*, the time in which its velocity is non-zero) in the sector for the *Negotiated Strategy*. Overall, the time a vehicle spends waiting or idling is considerable when the *Negotiated Strategy* is used. In particular, the results show that the average waiting time is almost 50% of the total time a vehicle spends within the sector and the worst waiting time is up to 90% of the total time. One can conclude that under the *Negotiated Strategy* the time an AGV spends at an intersection is mostly spent on waiting for its turn to cross. It is important to note that by construction the waiting time for each vehicle under the *Optimized Strategy* is always null. Here, coordination is achieved by managing the relative velocities of the agents to ensure collision avoidance while constraining the velocities to be always higher than zero.

In light of the average per vehicle waiting time resulting from the *Negotiated Strategy*, the computational time of the *Optimized Strategy* is essentially negligible (see Fig. 12). The optimization algorithm does not suffer from a high computational burden despite being implemented in a centralized manner. This claim is supported by the fact that the strategy is designed to work with specific roadmaps and thus always tuned to the worst case scenario (Fig. 5d). Furthermore, the required computation burden still stays low when the number of segments increases. The Optimized Strategy is concerned only with the number of vehicles since the collision regions are computed totally offline. Figure 13 shows a comparison between the scalability of the *Negotiated Strategy*, which is almost linear, and the *Optimized Strategy*. We note that in the *Optimized Strategy* the maximum crossing time does not linearly increase with the number of agents but rather its trend is piecewise linear. This suggests that the *Optimized Strategy* can potentially be further exploited when the system is complex. This is further supported by the fact that the waiting time for the *Negotiated Strategy* is two orders of magnitude higher than the computational time for the *Optimized Strategy*.

For the complete scenario and for the real experiments, the results were very similar to the previous simulation. In particular t_{total} in the *Optimized Strategy* is always less then in the *Negotiated Strategy*. In addition, on a complete and real map the *Optimized Strategy* works better in term of total time. It is worth noting that the coordination and the performances within a sector are, in general, independent of the

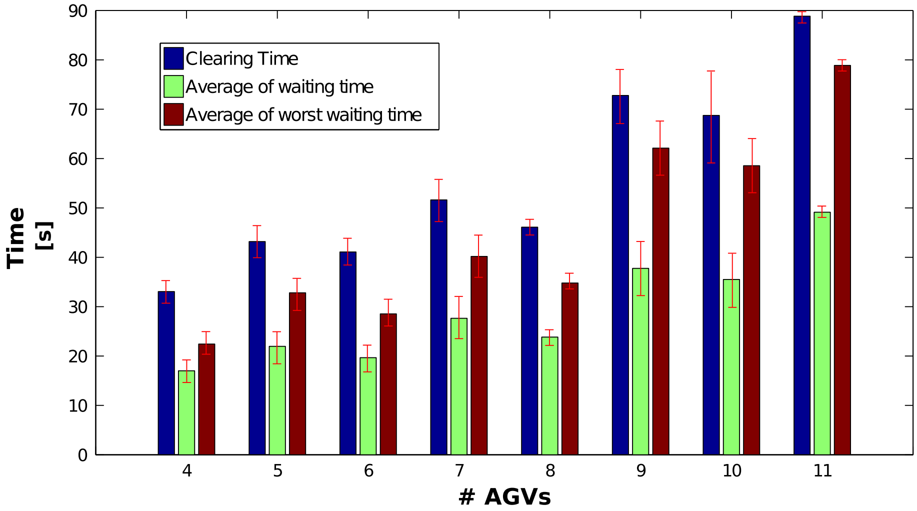


Fig. 11 Waiting time versus clearing time for the *Negotiated strategy*. Data presented for intersection 4

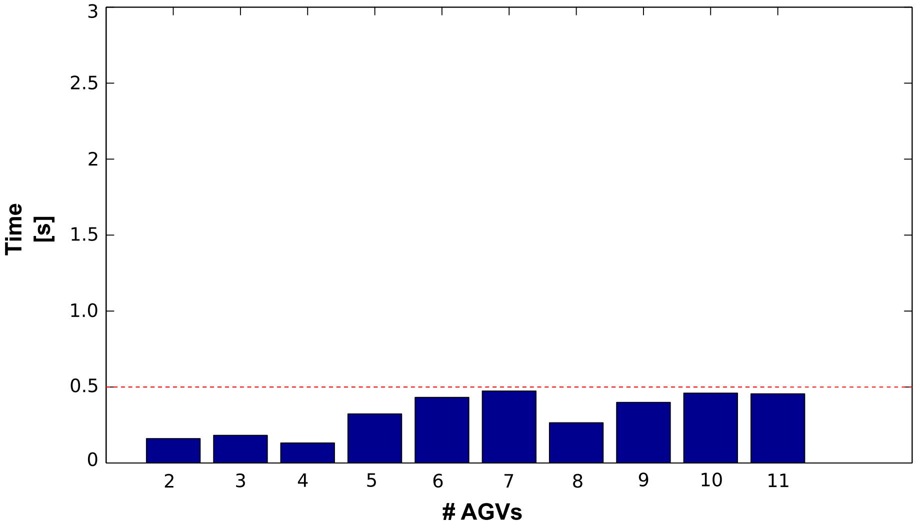


Fig. 12 Computational time for the optimized strategy with respect to the number of AGVs

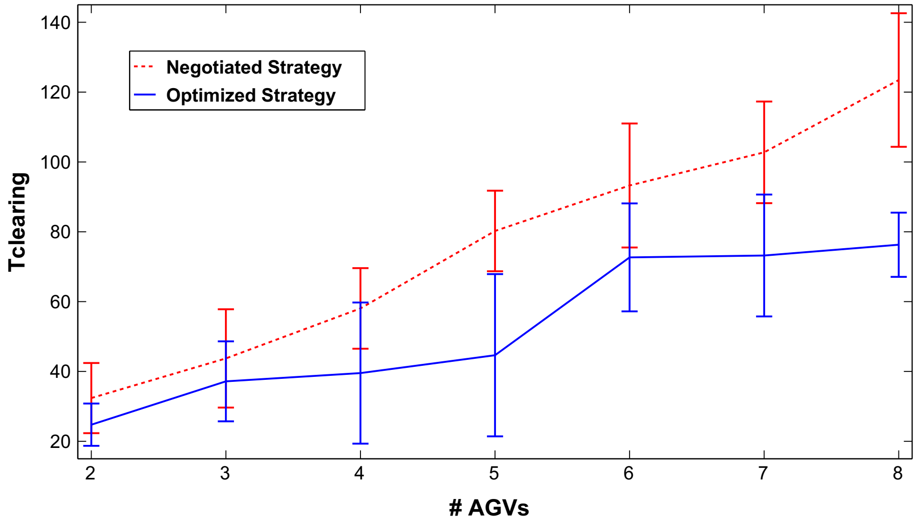


Fig. 13 Clearing time versus number of agents in the intersection 3 with standard deviation

other sectors. Thus the complete behavior of the proposed methodology on the full map is approximately a linear combination of the local behavior within each sector.

Lastly, Fig. 10 shows the difference between the total time between the two strategies increases as the number of AGVs increases. As more vehicles are added to the workspace, the bigger the difference between the total time of the two strategies. While the complexity of the scenario affects the total time, the *Optimized Strategy* consistently performs better in the presence of more agents. The experiments also show that the proposed method works well under realistic conditions, where noise and disturbances normally arise (e.g. localization, delay, etc.).

7 Conclusion

An optimized coordination strategy which aims at minimizing the time a fleet of AGVs takes to traverse different sectors has been proposed. The typical applicable scenario of such a strategy is an indoor environment for industrial applications like automated warehouses. The coordination problem among the vehicles is modeled as an optimization problem and a procedure was used to transform a non-linear and non-convex optimization problem into an equivalent quadratic program. Coordination is then achieved by efficiently solving the optimization problem in a centralized fashion within each sector of the warehouse and a set of optimal velocities is assigned to each vehicle.

Several simulations and real experiments validate the theoretical assertions and confirm the existence of solutions to the optimization problem. The method was compared to a decentralized negotiated strategy presented in a previous work. The negotiated strategy allowed AGVs to traverse through a desired sector in the environment while avoiding collisions with other AGVs in the same sector by assigning different priorities to each AGV. The experimental set-up provided a realistic validation scenario where real robots together with virtual ones move on the roadmap. Simulation and experimental results show that the proposed optimized strategy significantly outperforms the decentralized negotiated strategy.

In this paper, the problem of coordinating the motion of the AGVs on the higher layer was not considered, focusing on the coordination within each sector. While the coordination within each sector can be performed independently, the overall performance of the AGV fleet may depend on the interplay between inter-sector and intra-sector coordination and is a direction for future work. Nevertheless, the proposed strategy can be used to compliment any existing coordination strategy to further improve the overall system performance by smoothing out potential traffic congestions in high volume sectors that may arise from any system uncertainties that are difficult to model.

The method proposed in this paper aims at minimizing the sum of the travel times for all the AGVs. Future work will aim at considering different cost functions, comparing the obtained solutions in different scenarios.

References

Alami, R., Fleury, S., Herrb, M., Ingrand, F., & Robert, F. (1998). Multi-robot cooperation in the martha project. *IEEE Robotics Automation Magazine*, *5*(1), 36–47. https://doi.org/10.1109/100.667325.

Beinschob, P., Meyer, M., Reinke, C., Digani, V., Secchi, C., & Sabattini, L. (2017). Semi-automated map creation for fast deployment of AGV fleets in modern logistics. *Robotics and Autonomous Systems*, *87*, 281–295.

Bennewitz, M., Burgard, W., & Thrun, S. (2001). Constraint-based optimization of priority schemes for decoupled path planning techniques. In: *KI 2001: Advances in artificial intelligence* (pp. 78–93). Springer.

Cap, M., Novak, P., Kleiner, A., & Selecky, M. (2015). Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering*, *12*(3), 835–849. https://doi.org/10.1109/TASE.2015.2445780.

Digani, V., Hsieh, M., Sabattini, L., & Secchi, C. (2015a). A quadratic programming approach for coordinating multi-AGV systems. In *Proceedings of the IEEE international conference on automation science and engineering (CASE)*. Gothenburg.

Digani, V., Sabattini, L., & Secchi, C. (2016). A probabilistic eulerian traffic model for the coordination of multiple AGVs in automatic warehouses. *IEEE Robotics and Automation Letters*, *1*(1), 26–32.

Digani, V., Sabattini, L., Secchi, C., & Fantuzzi, C. (2014a). An automatic approach for the generation of the roadmap for multi-AGV systems in an industrial environment. In *IEEE International conference on intelligent robots and systems (IROS)*.

Digani, V., Sabattini, L., Secchi, C., & Fantuzzi, C. (2014b). Hierarchical traffic control for partially decentralized coordination of multi AGV systems in industrial environments. In *IEEE International conference on robotics and automation (ICRA)*.

Digani, V., Sabattini, L., Secchi, C., & Fantuzzi, C. (2015b). Ensemble coordination approach in multi-AGV systems applied to industrial warehouses. *IEEE Transactions on Automation Science and Engineering*, *12*(3), 922–934. https://doi.org/10.1109/TASE.2015.2446614.

Fanti, M., Mangini, A., Pedroncelli, G., & Ukovich, W. (2015). Decentralized deadlock-free control for AGV systems. In *American control conference (ACC), 2015* (pp. 2414–2419). https://doi.org/10.1109/ACC.2015.7171094.

Habib, D., Jamal, H., & Khan, S. A. (2013). Employing multiple unmanned aerial vehicles for co-operative path planning. *International Journal of Advanced Robotic Systems*, *10*, 235.

Hoshino, S., & Seki, H. (2013). Multi-robot coordination for jams in congested systems. *Robotics and Autonomous Systems*, *61*, 808–820.

Hui, N. (2010). Coordinated motion planning of multiple mobile robots using potential field method. In: *2010 International conference on industrial electronics, control robotics (IECR)* (pp. 6–11). https://doi.org/10.1109/IECR.2010.5720131.

Jager, M., & Nebel, B. (2001). Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. In *Proceedings of the IEEE/RSJ International conference on intelligent robots and systems, 2001* (Vol. 3, pp. 1213–1219). https://doi.org/10.1109/IROS.2001.977148.

Kavraki, L., Svestka, P., Latombe, J. C., & Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, *12*(4), 566–580.

LaValle, S., & Hutchinson, S. (1998). Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, *14*(6), 912–925. https://doi.org/10.1109/70.736775.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.

Makarem, L. & Gillet, D. (2012). Fluent coordination of autonomous vehicles at intersections. *IEEE International conference on systems, man, and cybernetics (SMC), 2012* (pp. 2557–2562). https://doi.org/10.1109/ICSMC.2012.6378130.

Manca, S., Fagiolini, A., & Pallottino, L. (2011). Decentralized coordination system for multiple AGVs in a structured environment. In *18th World congress of the international federation of automatic control (IFAC 2011)* (Vol. 18, pp. 6005–6010).

Mather, T. W., Braun, C., & Hsieh, M. A. (2010). Distributed filtering for time-delayed deployment to multiple sites (best paper award winner). In *10th International symposium on distributed autonomous robotics systems (DARS 2010)*. Lausanne.

Mather, T. W., & Hsieh, M. A. (2010). Ensemble modeling and control for congestion management in automated warehouses (best paper finalist). In *IEEE International conference on automation science and engineering (CASE 2012)*. Seoul.

Olmi, R., Secchi, C., & Fantuzzi, C. (2008). Coordination of multiple AGVs in an industrial application. In *IEEE International conference on robotics and automation, 2008. ICRA 2008* (pp. 1916–1921).

Olmi, R., Secchi, C., & Fantuzzi, C. (2011). Coordination of industrial AGVs. *International Journal of Vehicle Autonomous Systems*, *9*(1), 5–25.

Pallottino, L., Scordio, V. G., Bicchi, A., & Frazzoli, E. (2007). Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Transactions on Robotics*, *23*(6), 1170–1183. https://doi.org/10.1109/TRO.2007.909810.

Park, B., Choi, J., & Chung, W. K. (2012). An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment. *IEEE International conference on robotics and automation, 2012* (pp. 180–186). https://doi.org/10.1109/ICRA.2012.6225368.

Pecora, F., Cirillo, M., & Dimitrov, D. (2012). On mission-dependent coordination of multiple vehicles under spatial and temporal constraints. In *IEEE/RSJ International conference on intelligent robots and systems (IROS)* (pp. 5262–5269). IEEE.

Peng, J., & Akella, S. (2005) Coordinating multiple double integrator robots on a roadmap: Convexity and global optimality. In *Robotics and automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International conference on* (pp. 2751–2758). IEEE.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: An open-source robot operating system. *ICRA Workshop on Open Source Software*, *3*, 5.

Reveliotis, S., & Roszkowska, E. (2011). Conflict resolution in free-ranging multivehicle systems: A resource allocation paradigm. *IEEE Transactions on Robotics*, *27*(2), 283–296. https://doi.org/10.1109/TRO.2010.2098270.

Sabattini, L., Digani, V., Lucchi, M., Secchi, C., & Fantuzzi, C. (2015). Mission assignment for multi-vehicle systems in industrial environments. In *Proceedings of the IFAC symposium on robot control (SYROCO)*. Salvador.

Sabattini, L., Digani, V., Secchi, C., Cotena, G., Ronzoni, D., Foppoli, M., et al. (2013). Technological roadmap to boost the introduction of AGVs in industrial applications. In *IEEE International conference on intelligent computer communication and processing (ICCP)*.

Sabattini, L., Digani, V., Secchi, C., & Fantuzzi, C. (2016). Hierarchical coordination strategy for multi-AGV systems based on dynamic geodesic environment partitioning. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*. Daejeon.

Secchi, C., Olmi, R., Rocchi, F., & Fantuzzi, C. (2015). A dynamic routing strategy for the traffic control of AGVs in automatic warehouses. In *Robotics and automation (ICRA), 2015 IEEE International conference on* (pp. 3292–3297). https://doi.org/10.1109/ICRA.2015.7139653.

Siméon, T., Leroy, S., & Lauumond, J. P. (2002). Path coordination for multiple mobile robots: A resolution-complete algorithm. *IEEE Transactions on Robotics and Automation*, *18*(1), 42–49.

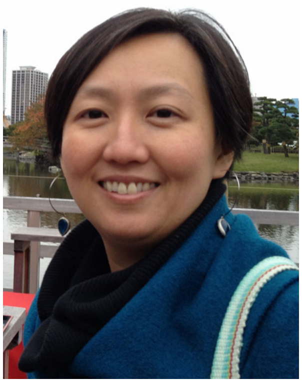
van den Berg, J., & Overmars, M. (2005). Prioritized motion planning for multiple robots. In *Intelligent robots and systems, 2005 (IROS 2005). 2005 IEEE/RSJ International Conference on* (pp. 430–435). https://doi.org/10.1109/IROS.2005.1545306.

Yang, P., Freeman, R., & Lynch, K. (2008). Multi-agent coordination by decentralized estimation and control. *IEEE Transactions on Automatic Control*, *53*(11), 2480–2496.

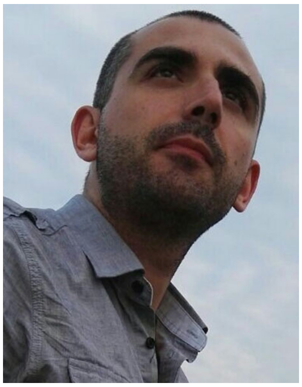
Zhang, Y., & Mehrjerdi, H. (2013). A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations. In *International conference on unmanned aircraft systems (ICUAS)* (pp. 1087–1096). IEEE.

Zheng, K., Tang, D., Gu, W., & Dai, M. (2013). Distributed control of multi-AGV system based on regional control model. *Production Engineering*, *7*, 433–441.

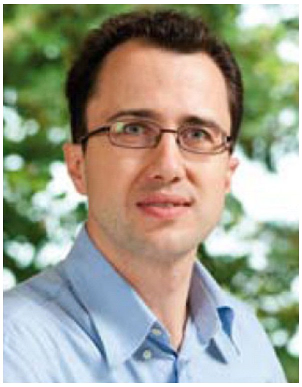
Valerio Digani is a R&D Scientist/Engineer at Elettric 80 s.p.a. He received the Bachelor and the Master degree in Mechatronic engineering from the University of Modena and Reggio Emilia in 2010 and 2012 respectively, and his Ph.D. in Innovation Industrial Engineering (Mechatronics curriculum) from the University of Modena and Reggio Emilia in 2016. Its Thesis was titled “Traffic Coordination for AGV Systems: an Ensemble Modeling Approach”. In 2014, he has been a Visiting Researcher at Drexel University, Philadelphia, Pennsylvania (USA), for 6 months. The main research interests include multi-robot system, coordination and path planning and optimization algorithms. He currently works on predictive traffic models, task assignment problems, path planning and data fusion algorithms for industrial environments (AGV Systems). He has participated to the European FP7 PAN-ROBOTS (Plug and Navigate Robots) project and to the 2011/2012 Sugar project in collaboration with Stanford University (USA), HSG University (Switzerland), University of Modena and Reggio Emilia (Italy) and AUDI



M. Ani Hsieh received the B.S. degree in engineering and the B.A. degree in economics from Swarthmore College, Swarthmore, PA, USA, in 1999, and the Ph.D. degree in mechanical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2007. She is a Research Associate Professor in the Department of Mechanical Engineering and Applied Mechanics at the University of Pennsylvania. She was an Associate Professor in the Department of Mechanical Engineering and Mechanics at Drexel University. Her research interests include control and coordination of multi-robot systems, nonlinear dynamical systems, and distributed sensing ad monitoring of geophysical flows by mobile robot teams. Dr. Hsieh is a member of the Institute of Electrical and Electronic Engineers. She has served as an Associate Editor for the IEEE Transaction on Automation Science and Engineering (2012–2016) and is currently serving as an Editorial Member of Swarm Intelligence. She received the 2012 Office of Naval Research (ONR) Young Investigator Award and the 2013 National Science Foundation (NSF) CAREER Award.



Lorenzo Sabattini is an Assistant Professor at the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy, since 2013. He received his B.Sc. and M.Sc. in Mechatronic Engineering from the University of Modena and Reggio Emilia (Italy) in 2005 and 2007 respectively, and his Ph.D. in Control Systems and Operational Research from the University of Bologna (Italy) in 2012. In 2010 he has been a Visiting Researcher at the University of Maryland, College Park, MD (USA). His main research interests include multi-robot systems, decentralized estimation and control control, and mobile robotics. He is one of the founding co-chairs of the IEEE RAS Technical Committee on Multi-Robot Systems: he has served as the corresponding co-chair since its foundation, in 2014. He has been serving as Associate Editor for the IEEE Robotics and Automation Letters since 2015, and for IEEE Robotics and Automation Magazine since 2017.



Cristian Secchi received the M.Sc. degree in Computer Science Engineering, from the University of Bologna, Italy, in 2000 and the Ph.D. in Information Engineering, from the University of Modena and Reggio Emilia, Italy, in 2004. He has been visiting student at the University of Delft (NL) and at the University of Twente (NL) in 2000 and in 2002 respectively. He is currently an Associate Professor at the University of Modena and Reggio Emilia. His research deals with human-robot physical interaction, telerobotics, mobile robotics and surgical robotics and he has published more than 100 papers on international journals and conferences. His Ph.D. thesis has been selected as one of he three finalists of the 5th Georges Giralt Award for the best Ph.D. thesis on robotics in Europe. He has been co-chair of the IEEE Robotics and Automation Society Technical Committee on Telerobotics for 2007–2012. He has been a guest Editor for the Special Issue on Design and Control Methodologies in Telerobotics, published on Elsevier Mechatronics in October 2010. He has served as Associate Editor of the IEEE Robotics and Automation Magazine from January 2005 to December 2008. Since September 2012 he is serving as an Associate Editor of the IEEE Transactions on Robotics and since June 2015 he is serving as an Associate Editor for the IEEE Robotics and Automation Letters.