Hierarchical Traffic Control for Partially Decentralized Coordination of Multi AGV Systems in Industrial Environments

Valerio Digani, Lorenzo Sabattini, Cristian Secchi and Cesare Fantuzzi

***Abstract*— This paper deals with decentralized coordination of Automated Guided Vehicles (AGVs). We propose a hierarchical traffic control algorithm, that implements path planning on a two layer architecture. The high-level layer describes the topological relationships among different areas of the environment. In the low-level layer, each area includes a set of fixed routes, along which the AGVs have to move. An algorithm is also introduced for the automatic definition of the route map itself. The coordination among the AGVs is obtained exploiting shared resources (i.e. centralized information) and local negotiation (i.e. decentralized coordination). The proposed strategy is validated by means of simulations using real plant.**

I. INTRODUCTION

This paper deals with the path planning and coordination of multiple Automated Guided Vehicles (AGVs) in an automated warehouse.Coordinated motion of groups of autonomous vehicles is a relevant topic in the field of mobile multi-robot systems, and has thus been widely addressed in the literature (see, for instance, the recent papers [1], [2] and references therein). Generally speaking, main approaches can be divided into two categories: centralized and decentralized.

The main advantage of centralized approaches is that they can theoretically find optimal solutions [3]. The main issue is instead represented by the complexity of the problem, which grows exponentially with the number of robots [4].

Decentralized techniques are generally faster than centralized ones, but they present several drawbacks, such as failing in finding valid paths for all robots due to deadlocks [5], [6].

In general, both centralized and decentralized methods exploit a route map to reduce the search space [3], [7]. As explained in [10], a hierarchical route map can also abstract the traversable areas using the adequate number of nodes and edges of a graph. It is worth noting that the performance of the coordination algorithm is strongly related to the characteristics of the route map where the coordination itself is performed. An evidence of this claim can be found, for instance, in the common necessity of defining specific traffic rules to overcome local issues [9]. Hence, even though the design of the route map and the subsequent coordination are strongly related problems, in the literature they are generally treated in a separate manner, to the best of the authors' knowledge. In fact, while several strategies can be found in the literature for the definition of a route map [11], [12], none of them consider the subsequent coordination that will be performed along the route map itself, during the daily industrial operations.

The contribution of this paper is on the definition of a methodology to solve the coordination problem in a holistic manner, providing a solution that simultaneously optimizes the creation of the route map and the coordination of the AGVs. In particular:

1) We define an algorithm for the automatic definition of the optimal route map, given the geometry of the plant, in such a way that the subsequent coordination of AGVs on the route map itself can be dealt with in an automatic and optimal manner
2) We introduce a hierarchical control architecture [5] for the coordination of a fleet based on two layers. The first layer is a topological graph of the plant where each node is a macro-area of the environment. The second layer is the real route map on which the AGVs move
3) We present a partially decentralized control strategy for the coordination of multi AGV systems. Each AGV autonomously computes its path, on both layers, exploiting shared information regarding the state of the fleet

Preliminary results on this topic were introduced in [13].

II. PROBLEM STATEMENT

In this paper we will present a combined strategy for the path planning and route map creation in order to coordinate multiple AGVs. The coordination is achieved considering both aspects.

A route map is a set of routes as a highway, and it is composed by distinguished elements called segments. The AGVs are constrained to follow the route map and its segments. We first introduce the definition of a *path* on a route map.

Definition 1 Paths *Given a route map, a* path *is a set of consecutive segments.*

Geometrically a path is a spline, that is, a piecewise polynomial function. A path can be assigned to an AGV, that is then allowed to move along the segments in the path.

We introduce the following definition of *admissible set of paths*.

Definition 2 Admissible set of paths *Given a fleet of $\mathcal{N}$ AGVs, a set of $\mathcal{N}$ paths is* admissible *if, assigning each path to a different AGV, a velocity profile can be defined in such a way that collisions are avoided.*

The problem can be formally stated as follows:

Authors are with the Department of Science and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy {valerio.digani,lorenzo.sabattini, cristian.secchi, cesare.fantuzzi}@unimore.it
This paper is written within PAN-Robots project. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 314193.

Problem 1 Multi AGV path planning *Consider:*

- *a fleet of $\mathcal{N}$ AGVs*
- *a route map*
- *the initial and final positions for all the AGVs*

Define an admissible set of paths such that each AGV is able to move from its initial position to its assigned final position.

Therefore, the problem consists of planning a path for a fleet of AGVs in an 2D static environment, so that conflicts and deadlocks are avoided. Each AGV starts its path in an initial position, and has to reach its own pick/drop position. Each AGV can communicate with the others in its neighborhood and has a prior knowledge of the environment.

The following **Assumptions** are made on the system:

A1 No unforeseen events, such as the presence dynamic obstacles (manual forklift, people, etc.) are considered.

A2 Each AGV has prior knowledge of the geometry of the sectors and of the route map.

It is worth noting that removing Assumption 1 would lead to the solution of the dynamic version of the *multi AGV path planning* problem, in which the set of paths initially defined has to be modified in case of unforeseen events.

The task assignment (that is, the goal given to each AGV) is out of the scope of this paper, and will therefore not be considered.

III. Automatic Route Map

The design of the route map is an optimization problem. In detail, the problem is faced in two sequential steps: the first step is based on a topological representation of the environment and the second step on a geometrical one.

A. Topological Optimization

The topological optimization process coincides with the all-pair shortest path problem, which is formulated as:

Definition 3 All-Pair Shortest Path Problem *Given a weighted* graph *(V, E) with* weight *function $w : E \to \mathbb{R}$, determine the set of paths that connect all pairs of vertices in G, with the minimum overall length.*

The objective is then to find the shortest path between each pair of nodes. In our case, the nodes are the working stations (pick/drop locations) of the warehouse.

The process starts with a complete undirected graph in which the nodes are the stations and the edges are the links between them. A weight is assigned to each edge. The objective of this step is to reduce the complete graph removing the less favorable edges. In this way we obtain a reduced graph whose edges define the shortest paths between each pair of nodes.

The all-pair shortest path problem is resolved by using the Floyd-Warshall algorithm (complexity O($|V|^3$), where V is the number of nodes) [14]. The core of the topological optimization regards the calculation of the weight to assign to each edge. The outcome is to find the shortest path among each pair of nodes, in such a way that the distances and the obstacles are taken in account.

We introduce a weight w_{ij} for each edge between the nodes i and j, computed in a heuristic manner as follows:

$$w_{ij} = (D_{ij} + \sum_{k=1}^{O} \psi_{ijk} P(k)) \cdot \omega_\Phi \ , \ w_{ij} \in \mathbb{R} \tag{1}$$

Where:

- $D_{ij} \in \mathbb{R}$ is the Euclidean distance between the node i and the node j. The Euclidean distance represents the ideal (straight path) length of the path.
- $P(k)$ represents the perimeter of the convex hull around the k-th obstacle, while O is the total number of obstacles in the environment. The term $\psi_{ijk} \in \{0, 1\}$ is equal to 1 if the straight path between node i and node j intersects obstacle k, zero otherwise. Therefore, the term $\sum_{k=1}^{O} \psi_{ijk} P(k)$ takes in account the number and the size of the obstacles crossed by the straight path between i and j. In fact, due to the obstacles, the real path will be different from the straight one, and the distance will be higher. Therefore a penalty is used to give a heuristic approximation of the real distance. The heuristic is needed in order to gather the real possibility to cross an obstacle during the geometric path process.
- $\omega_\Phi \in \{1, 1.5, 2\}$ is the penalty factor which is computed in an empirical manner to take into account the direction of the preferred **material flow** in the warehouse. In any warehouse, in fact, this preferred flow can be defined: for instance, material is often transferred from the production area to the storage area, while the opposite flow is less utilized. Hence, this factor leads to breaking symmetries: namely, paths between the same pair of nodes with opposite directions are possible and different. A path is then penalized if its direction is opposite with respect to the material flow. In particular, we let the value ω_Φ assume three empirically defined values, based on the angle α between the vector of the material flow and the direction of the path:

$$\omega_\Phi = \begin{cases} 2 & \text{if } \pi \le \alpha < \frac{3}{2}\pi \quad \text{(high penalty)} \\ 1.5 & \text{if } \frac{\pi}{2} \le \alpha \le \pi \text{ or} \\ & \frac{3}{2}\pi \le \alpha < 2\pi \quad \text{(low penalty)} \\ 1 & \text{if } 0 \le \alpha < \frac{\pi}{2} \quad \text{(no penalty)} \end{cases} \tag{2}$$

B. Geometrical Optimization

The second step of the route map process concerns the real path design.

Without loss of generality, we define each segment of the route map as a spline, that is a piecewise polynomial function. Assume hereafter that N is the number of the splines and M_j is the number of pieces of the j-th spline with degree ε.

Given a reference system $u-v$, a piece of the spline is expressed as:

$$v=\sum_{k=1}^{\varepsilon} a_{ij}^{k} u^{\varepsilon-k} \tag{3}$$

The parameters can be rewritten in compact form as:

$$a_{ij}=\left[\begin{array}{lll} a_{ij}^{1} \cdots a_{ij}^{\varepsilon} \end{array}\right]^{T} \in \mathbb{R}^{\varepsilon} \tag{4}$$

Where a_{ij} are the set of parameters of the i-th piece of the j-th curve.

The problem consists of finding the shortest splines between the pair of nodes obtained from the topological optimization. Therefore the parameters a_{ij} are the variables to be chosen in order to optimize the process. The optimization variables can then be collected in the vector x, defined as follows:

$$x=\left[a_{11}^{T} \cdots a_{M_{1}1}^{T}\, a_{12}^{T} \cdots a_{M_{2}2}^{T} \cdots a_{1N}^{T} \cdots a_{M_{N}N}^{T}\right]^{T} \in \mathbb{R}^{n} \tag{5}$$

where $n=\varepsilon \cdot \sum_{j=1}^{N} M_{j}$.

The optimization problem can then be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & f(x)+K \cdot\|p\| && \text{(6a)} \\ \text{subject to} \quad & Ax=b && \text{(6b)} \\ & c(x) \leq 0 && \text{(6c)} \end{aligned}$$

The problem is a nonlinear constrained optimization.

The linear constraints in Eq. (6b) are the boundary conditions of the curves where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}$. A is a block diagonal matrix, whose j-th block $A_{j} \in \mathbb{R}^{m_{j} \times \varepsilon}$ is referred to the j-th curve, and its elements concern:

- Start and Goal position
- Initial and final orientation
- Continuity and derivative among the pieces the spline

In order to define the nonlinear constraints, we first introduce the concept of distance between two objects. In particular, when considering the distance between two objects in a two-dimensional space, we will refer to the *Fréchet distance* [15]: loosely speaking, considering two curves, the Fréchet distance is the length of minimum-length segment connecting two points of these curves.

The nonlinear constraints in Eq. (6c) collect inequality distance constraints. Specifically, let $\lambda>0$ be the minimum allowed distance between two curves, or between a curve and an obstacle, defined in order to avoid collisions. Moreover, let

- $d_{j,o}(x)$ be the distance between the spline i and the obstacle o. This value is required to be greater than λ for all the pairs j-o, to avoid collisions between an AGV traveling on a path and an obstacle. Hence, constraints on the distance between a spline and an obstacle are *hard constraints*.
- $l_{j,z}(x)$ be the distance between the spline j and the spline z. Preferable solutions define splines for which this value is greater than λ for all the pairs j-z. If such an admissible solution is not available, violation of this constraint is admissible: the presence of curves whose distance is less than λ can be managed creating an intersection, as described in Section III-C. Hence, constraints on the distance between two splines are *soft constraints*.

On these lines, for each pair of splines j-z we introduce the variable $p_{j,z} \geq 0$, and we define $\gamma_{j,z}(x)=l_{j,z}(x)+p_{j,z}$: as will be subsequently clarified, the presence of the term $p_{j,z}$ defines the constraint as soft. The distances can then be collected into the vector $\Gamma(x) \in \mathbb{R}^{\frac{O \cdot N^{2} \cdot(N-1)}{2}}$, defined as follows:

$$\Gamma(x)=\left[d_{1,1} \cdots d_{N,O}\ \gamma_{1,2} \cdots \gamma_{1,N} \cdots \gamma_{N-1,N}\right]^{T} \tag{7}$$

where N is the number of splines and O is the number of obstacles.

The constraints are then collected in the vector $c(x)$ defined as follows:

$$c(x)=-\Gamma(x)+\lambda \mathbf{1} \tag{8}$$

where $\mathbf{1}$ is a vector of all ones of opportune dimension.

The objective function in Eq. (6a) is composed of two terms. The term $f(x)$ represents the total length of geometric paths, and is defined as follows:

$$f(x)=\sum_{j=1}^{N} \sum_{i=1}^{M_{j}} \int_{X_{s}^{ij}}^{X_{f}^{ij}} \sqrt{1+|v'|^{2}}\, du \tag{9}$$

where N is the number of curves, X_{s}^{ij} and X_{f}^{ij} are the start and final point of the i-th piece of the j-th curve and v' is the derivative of the i-th piece of the j-th curve with respect to u.

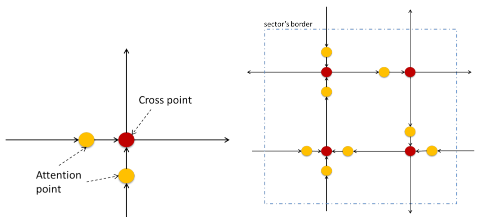
Let the vector $p \in \mathbb{R}^{\delta}$, where $\delta=\dfrac{N!}{2!\cdot(N!-2!)}$, collect all the variables $p_{j,z}$. Therefore, the term $K \cdot\|p\|$ in the objective function defines the soft constraints regarding the distances among the curves. In fact, minimizing $\|p\|$ leads to the minimum violation of the constraint on the distance between two curves. The parameter $K>0$ is a tuning gain, to be determined empirically.

C. Route map properties

Once the geometrical optimization has been performed, the route map is defined as a set of *unidirectional* segments: this property is useful for coordination purposes, since it is not possible to have pathological situations, such as two (or more) AGVs moving along the same segment with opposite directions.

In the case that a segment intersects one or more other segments, an *intersection* has to be defined. As will be clarified in the following Sections, intersections are considered as shared resources, to be allocated to a single AGV when necessary. As shown in Fig. 1a, each intersection is characterized by the presence of a *cross point* and a certain number of *attention points*. These points are defined as follows:

- Cross points: it is the point in common among two or more segments
- Attention points: for each segment entering the intersection, an attention point is defined in such a way that, if an AGV stops on the attention point itself, it does not collide with AGVs in the cross point, or in other attention points. It is worth noting that the definition of the attention points is based on the size and shape of the AGVs used in the particular application.



(a) A Route map intersection (b) General route map in a sector

Fig. 1: Route map properties

IV. Two Layer Control Architecture

The problem of coordinating an elevated number of AGVs is faced in this Section splitting the control through a multi-layer architecture. In our approach, two layers are used. The top-layer, or *Topological Layer*, is a topological map representing the global map, with different macro-cells called sectors. The layer below, or second layer, is the geometric map of each sector of the first layer, and will be hereafter referred to as *Route Map Layer*.

Therefore the path planning is done on two levels. Topology path planning searches for the best path to the final goal (actually to the final sector where the real goal is) from the current sector. Route map planning computes the path on the route map and handles the coordination inside the sector.

A. Topological layer

The first layer is the most abstract layer, it is generated by a subdivision of the geometric layout in several sectors.

1) Sector division: The layer gives a topological representation of the real map by means of an unidirected graph. Each node of the graph is a sector. A **sector** is an area, or a region, which can be distinguished from the other ones based on the following properties:

- topological aspects
- logistical aspects
- geometrical aspects
- constraints

The topological information regards the connection among the sectors. In detail, a sector is defined in such a way that a connection with neighboring sector exists. The constraints are defined based on the characteristics of the operational environment. For instance, constraints can be defined in terms of maximum number of AGVs contained in a single sector, or in terms of maximum number of operations of loading/unloading. This kind of information is owned by the sectors and it is stored in a *centralized manner*. In this way, the information is visible to all the AGVs and is shared among them from the centralized storage.

2) Path planning on the topological layer: The information owned by the sectors are used to plan the sub-optimal route for an AGV. Each vehicle has to reach its destination minimizing the length of the path.

It is worth remarking that each AGV autonomously plans its path, as soon as it is assigned a destination. As task assignment is generally asynchronous, path planning is performed in an asynchronous manner as well. Therefore, the future position of other AGVs is unknown in advance.

Hence, each AGV computes is path from the start sector to the goal one exploiting the D* algorithm [16], combined with a MPC (Model Predictive Control [17]) mechanism. Namely, at each step, the AGV checks if the previously assigned path is still admissible. According to the MPC approach, at each step a prediction horizon is defined, along which the corresponding portion of the path is computed. This combined approach is needed in order to re-plan in dynamic manner the path. In particular the re-planning event is performed based on the traffic congestion (i.e. number of AGVs) of the planned sectors in the prediction horizon. This approach provides an optimal local solution but a sub-optimal global one, because only the part of the path inside the horizon is interested by the optimization.

The procedure is summarized in Algorithm 1: in this algorithm, the vector path contains the list of sectors in the planned path, the term H is the prediction horizon, and the index i identifies the current step.

Algorithm 1: Path planning on the topological layer

```
while path[i] ≠ goal do
    path[i : i + H] = D*;
    go to path[i + 1];
end
```

B. Route Map layer

Inside each sector the coordination among AGVs is needed. The second layer manages the real path following on the route map (created according to the procedure describes in Section III) and the avoidance of deadlocks and conflicts among AGVs or among AGVs and obstacles. The coordination is managed locally (in each sector) in a decentralized manner. With this hierarchical architecture it is possible to simplify the whole control in order to focus the coordination of the AGVs only inside each sector in a local way.

Each AGV has to compute a path to reach the next planned sector on the topological layer. The path planning inside a sector consists of assigning a set of segments to each AGV. The algorithm used to find the path is the simple A*. The choice is due to the fact that the route map is fixed, and local dynamic changes are not considered.

The path planning strategy is based on the assumption that each AGV has a prior knowledge of the geometry of the sectors and of the route map, according to Assumption A2.

Conflicts among AGVs are managed by means of a hybrid approach, combining a *negotiation* mechanism with a *resource allocation* strategy In particular, the resource (intersection) is allocated only to a single AGV in order to avoid conflicts, and the negotiation permits to avoid deadlocks. This process is managed *locally*, because it takes place exclusively inside the sector: the AGVs share information among each other, without the participation of a centralized supervisor. The data exchange among AGVs concerns:

- *AGV priority*: each AGV is supposed to have a priority, related to the task it is carrying on. If this priority is not assigned a priori, several strategies can be found in the literature for decentralized priority allocation (see e.g. [18]).
- *Intersection request*: an AGV which is approaching an intersection has to communicate this intention to the others.
- *Intersection allocation*: an AGV that is allowed to go through an intersection has to communicate this to the others.

The coordination procedure is described in details in Algorithms 2 and 3. In this algorithm, the term $\mathsf{pos}\,(AGV\,[i])$ represent the position of the i-th AGV, while $\mathsf{attention\_point}\,[j]$ is the attention point related to the j-th intersection, identified as $\mathsf{intersection}\,[j]$. The term $\mathsf{request}\,[i,j]$ represents the request of the i-th AGV for the allocation of the j-th intersection. The term $AGV\,[i]_p$ is the value of the priority of the i-th AGV and $\mathsf{intersection}\,[j]_p$ is the value of the priority of the intersection j.

Algorithm 2: Coordination on the route map layer

1 **if** $\mathsf{pos}\,(AGV\,[i]) = \mathsf{attention\_point}\,[j]$ **then**
2     $\mathsf{request}\,[i,j] := \mathbf{true}$;
3   **if** $\exists k \neq i$ *such that* $\mathsf{request}\,[k,j] = \mathbf{true}$ **then**
4   Negotiation;
5   **else**
6         $AGV\,[i] := \mathbf{winner}$;
7   **end**
8   **if** $AGV\,[i] = \mathbf{winner}$ **and** $\mathsf{intersection}\,[j] = \mathbf{free}$ **then**
9   move;
10         $\mathsf{request}\,[i,j] := \mathbf{false}$;
11   **else**
12   stop;
13   go to **line 2**;
14   **end**
15 **end**

V. Simulations

In order to evaluate the proposed methodology, several simulations have been implemented in Matlab. A real plant of a warehouse has been used to build the automatic route map and to test the coordination algorithm.

Algorithm 3: Negotiation

1 **if** $AGV\,[i]_p < \mathsf{intersection}\,[j]_p$ **then**
2     $\mathsf{intersection}\,[j]_p := AGV\,[i]_p$;
3 **end**
4 **if** $AGV\,[i]_p > \mathsf{intersection}\,[j]_p$ **then**
5   return;
6 **end**
7 **if** $AGV\,[i]_p = \mathsf{intersection}\,[j]_p$ **then**
8     $AGV\,[i] := \mathbf{winner}$;
9   return;
10 **end**

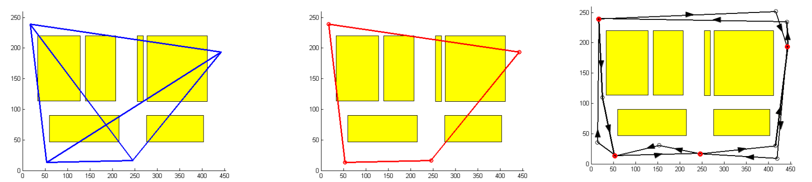
An example of route map generation is shown in Fig. 2. For the sake of clarity, we provide a simple example, with only four operation points: in this manner, the effect of the topological and geometrical optimization can be effectively interpreted by the reader. Hence, Fig. 2a shows the first step of the topological optimization, where a complete graph among the operation points is drawn. The next step reduces the graph as shown in Fig. 2b according to the all-pair shortest path algorithm. The geometric route map is then built as a composition of linear splines, as shown in Fig. 2c. It is worth noting that a feasible solution can always be found.

Subsequently the coordination algorithm has been tested on a simulated environment. In order to evaluate the performance of the proposed methodologies, repeated test have been conducted under the following conditions:

- the fleet is composed by a variable number of AGVs: 5, 10, 15, 20. For each configuration, 10 runs of simulation have been performed
- layout divided into 25 sectors
- maximum limit of 4 AGVs allowed in each sector
- a queue of 25 tasks is generated randomly
- the simulation stops when all the AGVs reach their goals and the queue of tasks is empty
- the priority is generated randomly for each AGV

In order to simulate a fleet of decentralized AGVs, the algorithm is executed in a parallel manner by implementing one single dedicated process per AGV. A statistical analysis has been carried out in order to experimentally evaluate the computational complexity of the algorithm. The results (see Fig. 3) show that the elapsed time increases almost linearly with the number of AGVs. It is worth noting that, as the the number of AGVs increased, also the variance of results increases. This is due to the high number of negotiations which, depending on the random priority of the AGVs, can provide different results on tests performed in similar conditions.

The result of a typical simulation run is shown in the accompanying video. The video shows the coordination of a fleet of 15 AGVs. The video shows a Matlab simulation where the AGVs are presented as colored triangles moving



(a) Toplogical optimization: complete graph (b) Toplogical optimization: reduced graph (c) Geometrical optimization

Fig. 2: Route map generation

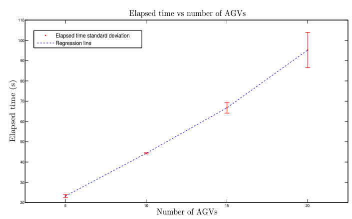


Fig. 3: Elapsed time versus number of AGVs

on a simulated real plant. The AGVs disappear after they have arrived in the final positions.

VI. Conclusion

The paper describes a coordination strategy for a fleet of AGVs, through an architecture based on a two-layer approach. The presented idea tries to treat the planning and the path optimization as a common entity. On these lines, an automatic route map generation process is proposed that autonomously defines a set of optimal paths among several operation points.

The coordination and the traffic management are treated as global functions. In order to achieve this, a hybrid path planning and coordination is achieved. The path planning is split on the two layers in order to simplify the problem. The path planning executed by each AGV is totally decentralized, but the information about the occupation of the sectors is managed in a centralized way. The local coordination is also totally decentralized. In this case, the AGVs share the information among them in order to negotiate the access to the shared resources (i.e. the intersections).

The simulations have shown that it is easily possible to manage a high number of AGVs with this approach avoiding conflicts among them. Current work aims at formally analyzing the complexity of the proposed algorithm in an analytic manner. Several strategies will be also studied in order to optimize the coordination and to include a task allocation mechanism as part of the coordination itself. Moreover, future works will aim at implementing the proposed approach on a real system working in an industrial plant, in order to compare the performances of the proposes approach with the current centralized system used in a real automatic warehouse and with the current manual built route map.

References

[1] Y. Zhang and H. Mehrjerdi, "A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2013, pp. 1087–1096.

[2] S. Hoshino and H. Seki, "Multi-robot coordination for jams in congested systems," *Robotics and Autonomous Systems*, 2013.

[3] S. LaValle and S. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.

[4] T. Simeon, S. Leroy, and J.-P. Lauumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.

[5] W. Zhang, M. Kamgarpour, D. Sun, and C. Tomlin, "A hierarchical flight planning framework for air traffic management," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 179–194, 2012.

[6] M. Jager and B. Nebel, "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, 2001, pp. 1213–1219 vol.3.

[7] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2557–2562, Oct. 2012.

[8] R. Olmi, C. Secchi, and C. Fantuzzi, "Coordination of industrial AGVs," *International Journal of Vehicle Autonomous Systems*, vol. 9, no. 1, pp. 5–25, 2011.

[9] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, Dec. 2007.

[10] B. Park, J. Choi, and W. K. Chung, "An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment," *2012 IEEE International Conference on Robotics and Automation*, pp. 180–186, May 2012.

[11] R. Geraerts and M. H. Overmars, "Creating high-quality roadmaps for motion planning in virtual environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4355–4361.

[12] R. Z. Farahani, E. Miandoabchi, W. Szeto, and H. Rashidi, "A review of urban transportation network design problems," *European Journal of Operational Research*, 2013.

[13] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Towards decentralized coordination of multi robot systems in industrial environments: a hierarchical traffic control strategy," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2013.

[14] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, pp. 345–, Jun. 1962.

[15] M. Godau and H. Alt, "Computing the Fréchet distance between two polygonal curves," *International Journal of Computational Geometry and Applications*, vol. 05, no. 01n02, pp. 75–91, 1995.

[16] A. Stentz, "Optimal and efficient path planning for partially-known environments," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, no. May, pp. 3310–3317, 1994.

[17] C. E. Garca, D. M. Prett, and M. Morari, "Model predictive control: Theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335 – 348, 1989.

[18] N. A. Lynch, *Distributed algorithms*. Morgan Kaufmann, 1996.