Model Predictive Path Planning of AGVs: Mixed Logical Dynamical Formulation and Distributed Coordination

Jianbin Xin, Xuwen Wu, Andrea D'Ariano, Rudy Negenborn, Fangfang Zhang

Abstract—Most of the existing path planning methods of automated guided vehicles (AGVs) are static. This paper proposes a new methodology for the path planning of a fleet of AGVs to improve the flexibility, robustness, and scalability of the AGV system. We mathematically describe the transport process as a dynamical system using an ad hoc mixed logical dynamical (MLD) model. Based on our MLD model, model predictive control is proposed to determine the collision paths dynamically, and the corresponding optimization problem is formulated as 0-1 integer linear programming. An alternating direction method of multipliers (ADMM)-based decomposition technique is then developed to coordinate the AGVs and reduce the computational burden, aiming for real-time decisions. The proposed methodology is tested on industrial scenarios, and results from numerical experiments show that the proposed method can obtain high transport productivity of the multi-AGV system at a low computational burden and deal with uncertainties resulting from the industrial environment.

Index Terms—Automated guided vehicles, Path planning, Model predictive control, Mixed logical dynamical model.

I. INTRODUCTION

AUOMTAED guided vehicles (AGVs), which are intelligent mobile robots, are commonly used in the industrial environment for transporting materials or performing specific tasks in manufacturing systems, warehouses, container terminals, and other applications [1], [2], [3]. Due to the requirements of *Industry 4.0*, as more nongeneral and diversified products are increasingly ordered, the AGVs must be more intelligent, autonomous, and efficient so that the manufacturing system can be more resilient and agile under complex and dynamic operation circumstances [4].

To meet the requirements of the manufacturer, there are two challenges for planning the operations of a fleet of AGVs. First, each AGV should be smart enough to properly

Jianbin Xin, Xuwen Wu and Fangfang Zhang are with the School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: j.xin@zzu.edu.cn, fangfangzhang@zzu.edu.cn)

Andrea D'Ariano is with Department of Engineering, Roma Tre University, via della Vasca Navale, 79 - 00146 Roma, Italy (e-mail: andrea.dariano@uniroma3.it).

R.R. Negenborn is with the Department of Marine and Transport Technology, Delft University of Technology, Mekelweg 2, 2628CD, Delft, The Netherlands (e-mail: r.r.negenborn@tudelft.nl).

respond to the change in the operational environment (e.g., breakdown of the machine, change of delivery points [5]), and therefore, reduce the economic loss resulting from these uncertainties [6]. Second, each AGV should coordinate with the other AGVs to avoid collision more efficiently [4]. The current trend of operating AGVs is toward decentralization, and each AGV should decide its actions individually. Low-cost but real-time distributed computing is recommended to handle the limitations of the computational burden and the communication band [7].

Motivated by the above challenges, in this paper, we focus on improving the flexibility, robustness, and scalability of path planning of AGVs in manufacturing and logistics environments. We investigate the transport process of the AGVs from the perspectives of discrete-event dynamical systems and distributed control. We regard the transport process as a dynamical system via the MLD representation to model the transport process mathematically. We also propose model predictive control to address uncertainties resulting from the transport process and develop an efficient distributed optimization technique based on the alternating direction method of multipliers (ADMM) to reduce the computational burden, which is considerably valuable for real-time decisions.

A. Related Work

Path planning is one of the most fundamental problems that mobile robots need to manage for autonomous navigation and exploration in complex environments. The path planning problem is typically defined as a mobile robot searching for an optimal or suboptimal path from an initial state to a target state according to certain performance criteria, given a robot and its working environment. We next review the related work from single-robot and multirobot perspectives.

1) Single robot/AGV: The path planning methods of a single robot can be mainly categorized as graph-search approaches, sampling-based approaches, and intelligent approaches. Graph-based approaches, including classical methods, such as *Dijkstra*, A* and D*, represent the environment as a graph with nodes and edges and use a search-based algorithm to compute the shortest path. Sampling-based approaches use randomization to construct a set of points sampled from the (continuous) obstacle-free space to build a roadmap of feasible trajectories. The most influential examples are probabilistic road maps (PRMs) and rapidly exploring random trees (RRTs). For intelligent approaches, artificial intelligence techniques

have been proposed for planning the path of single robots in a complex environment. Reference [8] provides a comprehensive review for readers. In addition to these three main categories, other methods exist that are based on robot kinetics, e.g., artificial potential field and dynamical window approaches [9]. These kinetics-based methods are also used for continuous path planning of autonomous vehicles in complex road environments [10].

2) Multiple AGVs: AGV moves on a predefined roadmap, which is logically represented by a graph with nodes and arcs. Here, we focus on this particular type of robot in detail. Path planning follows the results of task assignment, which distributes a given set of tasks to a fleet of AGVs optimally [11]. When assigning these tasks, the task sequences processed by each AGV are often involved, and the objective is typically to minimize the makespan (i.e., the completion time of all given tasks) or the tardiness of all the defined tasks.

Following the determined task assignment, path planning determines the optimal path to execute these tasks from its origin and destination in the guided roadmap, and collision avoidance must be considered when planning the detailed path. The path planning of multiple AGVs (or mobile robots) has received increasing attention since new sensing technologies (e.g., LiDAR SLAM or visual SLAM) have been introduced. Compared to a single robot, the path planning of multiple AGVs is more complex because each robot needs to consider the behavior of other robots, and the neighboring robots might become dynamic obstacles.

The path planning of multiple AGVs can be divided into offline and online approaches. The offline approaches typically determine the collision-free path of each AGV a priori, and the decision is not updated when performing the task for each robot. Most offline approaches are addressed in a centralized manner. For instance, the time-space network approach is regarded as an effective modeling representation to detect and resolve the conflicts in a grid roadmap layout [12], [13]. One alternative method is designing an optimal controller to prevent AGVs from any collision based on labeled Petri nets [14]. In addition to these centralized approaches, reference [3] proposed a decentralized zone-based planning algorithm relying on prioritized rules to resolve the conflicts between various AGVs.

Offline path planning of multiple AGVs is also integrated with other problems, e.g., task allocation and scheduling. A simultaneous dispatching and conflict-free routing for bidirectional automated guided vehicle (AGV) systems is investigated in [15]. Miyamoto el. al [16] integrated task allocation with path planning as an overall problem based on the time-space network framework. Furthermore, conflict-free AGV routing approaches are integrated with job shop scheduling and flop shop scheduling, according to the applications required into the production workshop [17], [18] or container terminals [19].

Compared to the off-line approaches that have been investigated intensively for the movement of AGVs, online approaches, which are more capable of dealing with unexpected situations, have received little attention. The computational burden for dynamic coordination between these multiple AGVs can be expensive. Therefore, one effective way to resolve this issue is to simplify the coordination by using global congestion-aware metrics to disperse traffic flow and ensure sufficient transport capacity of the vehicles [20], [21]. The improved A* algorithm is then used to search the idle path and avoid collisions. Moreover, a dynamic motion planning method has been recently proposed for a multi-AGV system [22], and prioritized update logic is implemented in a centralized controller to resolve motion conflicts.

However, it can be observed that the current online approaches are addressed in a centralized way for path planning of multiple AGVs. These centralized approaches simplify the coordination among these AGVs (e.g., by using prioritized rules). The position of each AGV cannot be predicted without a dynamical model when replanning these AGVs. Due to the lack of this information, the AGV performance cannot be further improved.

B. Contributions

To address the above limitations, in this paper, we make the following contributions:

- We develop a new model for describing the transport process of the AGV fleet as a dynamical system in an industrial environment. The transport process is represented by a mixed logical dynamical (MLD) model, which is able to predict the vehicle positions in advance. Advanced real-time control strategies can be further considered based on the MLD model.
- Model Predictive Control (MPC) is proposed for dynamically deciding the path of the AGVs. An ADMM-based decomposition technique is then developed to coordinate the AGVs. The computational burden is reduced considerably by the proposed methodology on tested industrial case studies.

In this paper, we use the knowledge of discrete-event dynamical systems and model the path evolution of the AGVs as a mixed logical dynamical system. We further analyze the feasibility of the global path planning problem, which can be regarded as an optimal control problem, in which an integer linear programming (ILP) problem must be solved. Based on the developed MLD model, we develop a dedicated MPC controller to determine the paths dynamically by receiving the measured vehicle position in the roadmap. The global planning problem is transformed into a local planning problem that is focused on planning regional paths, and an ADMM-based distributed planning algorithm is designed to improve the performance of online planning, which is promising for real-time decisions capable of dealing with the uncertainties resulting from the industrial environment. Experimental results show the effectiveness of the proposed planning strategy under different roadmaps and scenarios.

The remainder of the paper is organized as follows: Section II describes the research problem and provides the global planning representation based on the developed MLD model. Section III proposes a dedicated centralized MPC controller to dynamically determine the collision-free paths. In Section IV, an ADMM-based distributed planning method is proposed to coordinate the multi-AGV system. Section V discusses the

results of the proposed methodology on the numerical tests and further analyzes its performance. Section VI concludes this paper and provides future research directions.

II. PROBLEM DESCRIPTION AND FORMULATION

This section introduces the studied path planning problem of multiple AGVs and the mathematical formulation of this planning problem as a dynamical model based on the developed MLD formulation. Afterward, we present the global planning problem formulation and analyze its solution existence of the roadmap configuration.

A. Problem description

In an industrial environment, multiple materials need to be transported by multiple AGVs in a connected guided path network. In the process of transporting materials, it is required to plan a collision-free vehicle path for each AGV from its starting node to its delivery node. The starting and delivery nodes are assumed to be assigned to each AGV in advance.
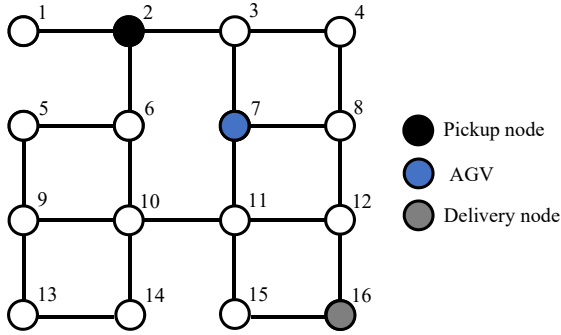

Fig. 1. Example of a 4×4 squared roadmap of the guide path network.

Here, we consider a squared AGV roadmap that contains several nodes, and the distance between any adjacent nodes is equal, as illustrated in Fig. 1. AGVs can only wait or change directions at each node of the roadmap, and every side represents the vehicle lane. Each AGV can only reach its adjacent nodes, which are connected to its location node by one move, and the AGV cannot arrive at the nonadjacent connected nodes by one move.

For this path planning problem, we make the following assumptions, as suggested in [12], [5]:

- The geometrical size of an AGV is sufficiently small, and the AGV is regarded as a point occupying a particular node.
- Each AGV is assigned a task from its pickup node to its delivery node, and these nodes are different.
- The velocity of the AGV is constant, and the turning time can be included in the corresponding traveling time.
- Each AGV can wait or change direction at the node of the path network, and each node can be occupied by at most one AGV at any time.
- The lanes are bidirectional, and a lane can be occupied by at most one AGV at any time.
- When an AGV completes a particular task, this AGV stays at the end node.

TABLE I
INDEX VARIABLES AND INPUT PARAMETERS

Symbol:	Description
Δt	Discretized time step
H	Time horizon
t	Discretized time instance, $t \in \{0, 1, ..., H\}$
N_i	Set of adjacent nodes to node i
N	Set of nodes
E	Set of arcs
N_n	Total number of nodes in the graph
N_AGV	Total number of AGVs
Φ	Set of AGVs, $\Phi = \{1, 2, ..., N_\mathrm{AGV}\}$
S_k	Start node of AGV k
G_k	End node of AGV k

- Every pickup-delivery node pair is connected, , and the number of AGVs is less than the number of nodes.

B. Mathematical formulation

In this part, we model the transport process of multiple AGVs in the guide path network via a mixed logical dynamical formulation. The MLD model is a powerful modeling approach, and this MLD model can describe the motion changes of the AGVs in a computationally friendly manner that is well-suited for the formulation of the system and control design (e.g., optimal control and model predictive control) [23]. The MLD model has been successfully used in the domain of power systems [24], transportation systems [25], [26] and other applications.

The whole planning horizon is equally discretized into a set of time slots denoted by $\{\Delta t, 2 \times \Delta t, ..., H \times \Delta t\}$, where Δt is the time step and H is the total number of time steps. The roadmap is regarded as a directed graph $G = (N, E)$. N is the set of nodes, while $E = \{(i, j) | i \in N, j \in N\}$ is the set of directed arcs (i, j). Since the roadmap can be visited by every AGV, the graph G is shared by all the AGVs.

Before providing the MLD formulation for modeling the movement process, we first introduce the related state variable $x_i^k(t)$ and control variable $u_{i,j}^k(t)$. The related symbols are given in Table I.

- $x_i^k(t)$ is defined as the state variable. $x_i^k(t) = 1$ means that AGV k arrives at node i at time t; otherwise, $x_i^k(t) = 0$;
- $u_{i,j}^k(t)$ is defined as the control variable; $u_{i,j}^k(t) = 1$ indicates that AGV k moves from node i at time t to node j at time $t + 1$. Note that nodes i and j can be identical if AGV k stays in the same node from t to $t + 1$.

Based on the defined $x_i^k(t)$ and $u_{i,j}^k(t)$, we observe that $\forall$ node j ($j \in N$), $k \in \Phi$,

$$x_j^k(t+1) = \begin{cases} 1, & \text{if } \sum\limits_{i \in N_j, i \neq j} u_{i,j}^k(t) = 1 \\ x_j^k(t), & \text{else} \end{cases} \qquad (1)$$

where N_j is the set of adjacent nodes for node j.

Remark 1: Equation (1) provides the update of state $x_j^k(t)$ from time t to time $t + 1$ in two situations. The first situation

is given as follows:

$$\sum_{i \in N_j, i \neq j} u_{i,j}^k(t) = 1, \tag{2}$$

, which means that AGV k moves from one of its adjacent nodes, and AGV k arrives at node j at time $t+1$ ($x_j^k(t+1) = 1$). The second situation, which is the complement of the first situation, is described as follows:

$$\sum_{i \in N_j, i \neq j} u_{i,j}^k(t) = 0, \tag{3}$$

which suggests that AGV k does not move from node i to its adjacent node j ($i \neq j$). For node i at time t, one connection at most can be made from nodes i to j (including node i itself), and we have the following operational constraint:

$$\sum_{i \in N_j, i \neq j} u_{i,j}^k(t) + \sum_{i \in N_j, i=j} u_{i,j}^k(t) \leq 1. \tag{4}$$

which indicates, following Equation (3), $\sum_{i \in N_j, i=j} u_{i,j}^k(t) = 1$ or $\sum_{i \in N_j, i=j} u_{i,j}^k(t) = 0$. The former means that AGV k stays at the same node j, while the latter indicates that AGV k does not occupy node j from t to $t+1$. In both conditions, the value of x_j^k does not change, and we can conclude that $x_j^k(t+1) = x_j^k(t)$.

As Equation (1) cannot be directly implemented in a standard optimization formulation, we rewrite Equation (1) as follows:

$$\begin{aligned} x_j^k(t+1) &= \sum_{i \in N_j, i \neq j} u_{i,j}^k(t) + x_j^k(t)(1 - \sum_{i \in N_j, i \neq j} u_{i,j}^k(t)) \\ &= x_j^k(t) - x_j^k(t)\sum_{i \in N_j, i \neq j} u_{i,j}^k(t) + \sum_{i \in N_j, i \neq j} u_{i,j}^k(t). \end{aligned} \tag{5}$$

The product $x_j^k(t) \sum_{i \in N_j, i \neq j} u_{i,j}^k(t)$ is a nonlinear term, and we define an auxiliary binary variable $\delta_j^k(t) \triangleq x_j^k(t) \sum_{i \in N_j, i \neq j} u_{i,j}^k(t)$. Based on this replacement, Equation (5) can be rewritten as follows:

$$x_j^k(t+1) = x_j^k(t) - \delta_j^k(t) + \sum_{i \in N_j, i \neq j} u_{i,j}^k(t). \tag{6}$$

The nonlinear equation $\delta_j^k(t) = x_j^k(t) \sum_{i \in N_j, i \neq j} u_{i,j}^k(t)$ can be expressed by a number of linear equalities [27], as follows:

$$\delta_j^k(t) - x_j^k(t) \leq 0, \tag{7}$$

$$\delta_j^k(t) - \sum_{i \in N_j, i \neq j} u_{i,j}^k(t) \leq 0, \tag{8}$$

$$x_j^k(t) + \sum_{i \in N_j, i \neq j} u_{i,j}^k(t) - \delta_j^k(t) \leq 1. \tag{9}$$

In addition to the above motion dynamics, other constraints are considered for the capacity of the roadmap of collision avoidance both for the nodes and the vehicle lanes and for initial and terminal conditions.

For the roadmap capacity and connectivity, we have the following constraints:

$$\sum_{i \in N} x_i^k(t) \leq 1, \forall k \in \Phi, t = 0,...H \tag{10}$$

$$\sum_{k \in \Phi} x_i^k(t) \leq 1, \forall i \in N, t = 0,...H \tag{11}$$

$$\sum_{k \in \Phi, j \in N_i} (u_{i,j}^k(t) + u_{j,i}^k(t)) \leq 1, \forall t = 0,...H \tag{12}$$

$$\sum_{j \in N_i} x_j^k(t+1) \geq x_i^k(t). \tag{13}$$

Constraint (10) ensures that each AGV stays at only one node at any time. Constraint (11) guarantees that every node can be occupied by at most one AGV. Constraint (12) requires that the connection between nodes i and j is unidirectional and that every connection can only be occupied by one AGV at most at any time. Constraint (13) ensures that AGV k moves node i to its adjacent nodes.

Furthermore, we have the constraints for the initial and terminal conditions as follows:

$$x_{S_k}^k(0) = 1, k \in \Phi \tag{14}$$

$$\sum_t x_{G_k}^k(t) \geq 1, \forall k \in \Phi, t = 0, ..., H \tag{15}$$

$$x_{G_k}^k(t) \geq x_{G_k}^k(t+1). \forall k \in \Phi, t = 0, ..., H-1 \tag{16}$$

Constraint (14) provides the initial position of all the considered AGVs. Constraints (15) and (16) indicate the destination of AGVs, thus requiring the AGVs to reach the destination within the total planning horizon H.

We let $\mathbf{x}(t) = \left[x_1^1(t), ..., x_{N_n}^1(t), ..., x_1^{N_{AGV}}(t), ..., x_{N_n}^{N_{AGV}}(t)\right]^T$, $\delta(t) = \left[\delta_1^1(t), ..., \delta_{N_n}^1(t), ..., \delta_1^{N_{AGV}}(t), ..., \delta_{N_n}^{N_{AGV}}(t)\right]^T$ $\mathbf{u}(t) = \left[u_{1,1}^1(t), ..., u_{N_n,N_n}^1(t), ..., u_{1,1}^{N_{AGV}}(t), ..., u_{N_n,N_n}^{N_{AGV}}(t)\right]^T$, where $\mathbf{x}(t)$ is the system state vector, $\mathbf{u}(t)$ is the control logic variable vector, and $\delta(t)$ is the auxiliary logic variable vector. Based on the abovementioned variable vectors, the motion dynamics, including (6)–(16), can be rewritten into a compact MLD model representation derived from [27] as follows:

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B_1\mathbf{u}(t) + B_2\delta(t) \tag{17}$$

$$E_2\delta(t) \leq E_3\mathbf{x}(t) + E_1\mathbf{u}(t) + E_4 \tag{18}$$

where matrices A, B_1, B_2, E_1, E_2, E_3, and E_4 are obtained from the transport process described by (6)–(16) when this is transformed into the compact form (17) and (18). This MLD model can describe the motion dynamics in a computationally friendly manner that is well-suited for the formulation of the system and control design [27].

We next provide the objective function, such that the overall optimization problem of the global planning can be given. Typically, the objective of multiple-vehicle path planning is to minimize the sum of the transport times of the AGV fleet. In this paper, we define the objective as J, which is given as follows:

$$J = -\sum_{k \in \Phi} \sum_{t=0}^{H} x_{G_k}^k(t). \tag{19}$$

Here, we define $\tilde{\mathbf{u}} = [\mathbf{u}^{\mathrm{T}}(0), ..., \mathbf{u}^{\mathrm{T}}(H-1), \delta^{\mathrm{T}}(0), ..., \delta^{\mathrm{T}}(H-1)]^{\mathrm{T}}$. Then, the global planning problem (defined as P_1) can be formulated as follows:

$$(P_1) \quad \min_{\tilde{\mathbf{u}}} J$$
$$\text{s.t. } (6) - (16),$$

, where minimizing J allows all the AGVs to arrive at their destinations as soon as possible. Problem P_1 is regarded as an optimal control problem based on motion dynamics with initial and terminal conditions. Problem P_1 is treated as a standard 0-1 ILP problem that can be solved by commercial solvers (e.g., CPLEX or Gurobi).

However, the global optimization problem P_1 is static, and uncertainties cannot be dealt with during the planning process. Furthermore, although the problem P_1 may be effectively solved by a commercial solver, its computational burden can be unsatisfactory for the real-time decision requirement as the system scale increases considerably. To address these limitations, we propose the MPC controller and discuss its centralized and distributed control strategies in the next section.

C. Path Feasibility

It can be observed that, for problem P_1, the setting of the roadmap is typically assumed that the origin (pickup) and the destination (delivery) nodes are connected [28], [5]. Especially in [28], the configuration of up to n robots and an n-vertex connected graph is investigated. For the studied problem P_1, an interesting question would be about which configurations of the roadmap planning problem P_1 is solvable, i.e., a feasible solution exists. Here, we discuss its solvable conditions.

As mentioned in Section II. A, we assume that the number of AGVs is less than the number of nodes in the graph. This condition is typically considered for discrete multirobot path planning because empty nodes are required as swap spaces [28]. The case in which the number of AGVs is equal to the number of nodes is beyond the scope of this paper.

Here, we are interested in the condition that the graph contains at least one cycle and one additional connected node, which is commonly seen in the industrial environment [5], [29]. The situation of a single cycle is discussed in [30]. The cycle is defined as the number of nodes connected in a closed chain. The free node refers to the node for which there is no AGV occupied. The bridge is defined for a particular edge if the edge connects two subgraphs, while removing the edge disconnects the graph. For the roadmap used for the AGVs, we make the following remark:

Remark 2: Assuming the graph contains at least one cycle and one additional connected node, the problem P_1 can be considered solvable only if a cycle exists in the graph and the number of free nodes is greater than or equal to the maximum bridge length.

The details of the proof for *Remark 2* can be found in Section 4 of [31], which discusses *Remark 2* under two conditions (the current position of AGV k and its goal position are in a single cycle or in two different cycles).
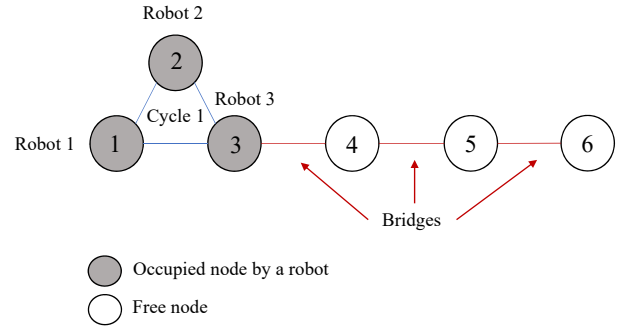

Fig. 2. Illustrative example of the solution feasibility for P_1.

Fig. 2 provides an example to illustrate the condition of solution feasibility for the global planning problem P_1. In this example, a cycle and three connected nodes are included in the graph. Cycle 1 is composed of Nodes 1, 2, and 3. The bridges are (3,4), (4,5) and (5,6). The maximum bridge length is 3, equal to the number of free nodes. Following *Remark 2*, this example is solvable, and every robot can reach any node in the graph. For example, if Robot 1 wants to move from Node 1 to Node 6, Robots 1, 2 and 3 synchronize their moves in Cycle 1 (clockwise or counterclockwise), and then Robot 1 reaches Node 3. Afterward, Robot 1 arrives at Node 6 throughout Nodes 4 and 5 (free nodes) in the bridge.

In summary, *Remark 2* provides the solvable condition for our path planning problem, and it is assumed that the number of AGVs is less than the number of nodes in the roadmap graph, which is consistent with the settings in industrial scenarios. The solution feasibility can be guaranteed when testing the planning algorithms for industrial scenarios in later sections under such conditions provided by *Remark 2*.

III. MODEL PREDICTIVE CONTROL

The MLD model obtained in Section II is indeed dynamic. Based on the developed MLD model, this section further proposes an MPC controller for planning the collision-free paths of the AGVs dynamically. The first part improves the original MLD model used for local planning in a centralized way, while the second part discusses the selection of the planning horizon.

A. MPC strategy

MPC is a control strategy that explicitly uses a dynamical model to determine control actions by minimizing the desired objective over a finite receding horizon. This control strategy has been successfully implemented in transportation and robotics [32], [33]. The MPC provides an online planning framework to control complex systems that contain interacting variables, complex dynamics, and constraints. This online planning method is suitable for handling flexible manufacturing, customized products, and complex product specifications in a dynamic manufacturing environment [34].

We consider a centralized MPC controller for planning the collision-free paths of the fleet of AGVs dynamically. The objective of this controller is to minimize the total transport

TABLE II
NEW DEFINED NOTATIONS FOR LOCAL PLANNING

Symbols	Description
T_{p}	Local planning horizon
N_{L}	Set of local ending nodes within T_{p}
D_{j,G_k}	Approximated distance between nodes j and G_k for AGV k
Variable	**Description**
C_j^k (binary)	$C_j^k = 1$ if AGV k arrives at node j at time T_{p}, else 0

times of the AGV fleet. The MPC controller plans the path according to the position state of the AGVs at a time instant and updates the path plan in real-time considering the motion change of the AGVs.

The proposed MPC planner uses a local planning strategy, which is similar to the A^* algorithm for a single robot. The global planning problem P_1 is decomposed into smaller local planning problems, which are solved repeatedly with a rolling horizon T_{p}(T_{p} is much smaller than H). In local planning, the size of the optimization problem is smaller than the global problem. Furthermore, the localization of the AGV can be limited by its sensor, which makes local planning (by using the MPC strategy) suitable for implementation in the industrial environment.

B. Local planning formulation

In this part, a local MLD model is formulated, and the corresponding MPC formulation is presented. For local planning, new decision variables and parameters need to be introduced, while several constraints of the MLD model also need to be adjusted, as the initial constraints were made for global planning to fit the framework of the MPC controller design. These new parameters and decision variables are given in Table II.

For the local MLD model, we modify constraints (15) and (16) to ensure that each AGV arrives at a local ending node within the local planning horizon T_{p}. The modified constraints are given as follows:

$$\sum_j C_j^k = 1, k \in \Phi, j \in N_{\mathrm{L}} \tag{20}$$

$$x_j^k(T_{\mathrm{p}}) = C_j^k, k \in \Phi, j \in N_{\mathrm{L}} \tag{21}$$

where constraint (20) ensures that each AGV chooses a local ending node in the considered local problem. Constraint (21) ensures that each AGV can only stay at one of the local ending nodes ($j \in N_{\mathrm{L}}$) by $t = T_{\mathrm{p}}$.

For constraints (6)-(13) of the original MLD model, when local planning is considered, the domain of t needs to be changed from $t \in \{0, 1, .., H\}$ to $t \in \{0, 1, .., T_{\mathrm{p}}\}$.

Regarding the objective function, J also needs a local version $\widetilde{J}$, which is defined as follows:

$$\widetilde{J} = \sum_{k \in \Phi}(\sum_{j \in N_{\mathrm{L}}} C_j^k * D_{j,G_k} + T_{\mathrm{p}}), \tag{22}$$

, where the parameter D_{j,G_k} is obtained by approximating the distance between node j and destination node G_k for AGV k.

Here, we define $\hat{\mathbf{u}} = [\mathbf{u}^{\mathrm{T}}(0), ..., \mathbf{u}^{\mathrm{T}}(T_{\mathrm{p}} - 1), \delta^{\mathrm{T}}(0), ..., \delta^{\mathrm{T}}(T_{\mathrm{p}} - 1), \mathbf{C}^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathbf{C}$ is the set of C_j^k. Then, the local planning problem (defined as P_2) can be formulated as follows:

$$(P_2) \quad \min_{\hat{\mathbf{u}}} \widetilde{J}$$
$$\text{s.t. } (6) - (14), (20) - (21).$$

Using the proposed MPC strategy, when carrying out the path planning of the AGVs, the local model only considers the path planning within the local range T_{p} for AGVs, finding the local ending nodes, and approximating the distance between the local ending node and the global ending node obtained in an off-line way. We use precomputed distance data to estimate the target value of this model from the global roadmap configuration. Although this method cannot guarantee the optimality of the obtained solution, the accuracy can be reasonably high with a faster computation speed than global planning if a proper T_{p} is selected.

Note that the multicommodity can be incorporated into our model by extending the node capacities of the origin and destination pairs. Additionally, the multitrip aspect can be considered based on the proposed dynamic planner by updating the corresponding destination for the new task when the current task is completed. The sequences of these multiple trips are determined by solving an allocation problem at a different level and received by the dynamic planner.

IV. DISTRIBUTED MPC PLANNING

The previous section presents the design of our MPC controller based on the developed MLD model, in which the paths of the AGVs are determined dynamically in a centralized way. However, because of the requirements of the Industry 4.0 plan (including flexibility, robustness, and scalability issues, due to memory limitations, communication, and computation), decentralized planning is required to distribute the intelligence of the overall AGV system [4]. Distributed computing decomposes the centralized problem into smaller problems, and each AGV makes its own decisions based on the interactions with the other AGVs. This can reduce the overall computational burden and greatly improve computational efficiency.

In this section, we propose an ADMM-based distributed MPC controller, in which a serial iterative control method based on the ADMM is developed. The distributed control architecture is illustrated in Fig. 3. In the following sections, the separable optimization formulation for the distributed MPC is provided, and the proposed distributed planning algorithm is presented.

A. ADMM decomposition

ADMM is a widely used decomposition method to solve distributed MPC problems [33], [35]. ADMM is recommended since this is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers [36]. When using the ADMM method, the augmented Lagrangian of the original optimization problem is first constructed, while the primal and dual variables are
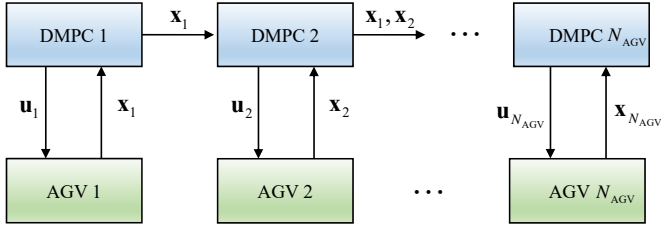
Fig. 3. Architecture of the proposed D-MPC method.

updated in an alternating or sequential fashion. The algorithm continues until a given stopping criterion is satisfied.

We next explain the augmented Lagrangian function regarding the problem P_2. The capacity constraints (8) and (9) are coupled for the multiple AGVs. We introduce two nonnegative Lagrangian multipliers λ_1 and λ_2 to relax these two coupled constraints. The augmented Lagrangian function of problem P_2 is formulated as follows:

$$
\begin{aligned}
L_c &= \sum_{k \in \Phi}(\sum_{j \in N_L} C_j^k * D_{j,G} + T_p) + \\
&\lambda_1(\sum_{k \in \Phi} x_i^k(t) - 1) + \lambda_2(\sum_{k \in \Phi}(u_{i,j}^k(t) + u_{j,i}^k(t)) - 1) \\
&+ (\epsilon/2)(\sum_{k \in \Phi} x_i^k(t) - 1)^2 + (\epsilon/2)(\sum_{k \in \Phi}(u_{i,j}^k(t) + u_{j,i}^k(t)) - 1)^2, \\
&\forall k \in \Phi, i \in N, j \in N_L, t = 0, ..., T_p
\end{aligned}
\tag{23}
$$

where ϵ is the augmented Lagrangian parameter. For the sake of computation, we let

$$
\phi_k \triangleq x_i^k(t) - 1/N_{AGV}, \forall k \in \Phi \tag{24}
$$
$$
\psi_k \triangleq u_{i,j}^k(t) + u_{j,i}^k(t) - 1/N_{AGV}. \forall k \in \Phi \tag{25}
$$

The problem of minimizing L_c is not separable into AGV-level subproblems due to the incorporated quadratic penalty term. To maintain separability, a linearization technique is used for the cross-penalty part around the estimated optimal solution [37]. The augmented Lagrangian function L_c can be additive for AGV k by using the first-order Taylor expansion. $\left| \sum_{k=1}^{N_{AGV}} \phi_k \right|^2$ can be expressed as follows:

$$
\left| \sum_{k=1}^{N_{AGV}} \phi_k \right|^2 = \sum_{k=1}^{N_{AGV}} |\phi_k|^2 + 2 \sum_{k=1}^{N_{AGV}} \sum_{l=1}^{N_{AGV}-1} \phi_k \phi_l. \tag{26}
$$

The last (cross-product) terms of (26) are linearized by a first-order Taylor expansion around the point $(\overline{\phi_k}, \overline{\phi_l})$ for a good estimation. In this way, $\left| \sum_k \phi_k \right|^2$ is expanded into the following expressions:

$$
\begin{aligned}
\left| \sum_{k=1}^{N_{AGV}} \phi_k \right|^2 &\approx \sum_{k=1}^{N_{AGV}} \left| \phi_k + \sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\phi_l} \right|^2 - \sum_{k=1}^{N_{AGV}} \left| \sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\phi_l} \right|^2 \\
&- 2 \sum_{k=1}^{N_{AGV}} \overline{\phi_k} \sum_{l=1}^{N_{AGV}-1} \overline{\phi_l} \\
&= \sum_{k=1}^{N_{AGV}} \left(\left| \phi_k + \sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\phi_l} \right|^2 - \left| \sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\phi_l} \right|^2 - 2\overline{\phi_k} \sum_{l=1}^{N_{AGV}-1} \overline{\phi_l} \right)
\end{aligned}
\tag{27}
$$

where $\overline{\phi_k} = \sum_k (\overline{x_i^k(t)} - 1/N_{AGV})$, and $\overline{\phi_k}$ is regarded as a constant. For each k, the original $|\sum_k \phi_k|^2$ is separable.

$\sum_k \psi_k$ can be linearized in a similar way. Therefore, the original centralized problem P_2 can be decomposed into a series of subproblems (one for each AGV). The objective function for AGV k can be written as follows:

$$
\begin{aligned}
L_c^k &= (\sum_{j \in N_L} C_j^k * D_{j,G} + T_p) + \lambda_1 \phi_k + \lambda_2 \psi_k) \\
&+ (\epsilon/2)(|\phi_k + \sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\phi_l}|^2 - |\sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\phi_l}|^2 - 2\overline{\phi_k} \sum_{l=1}^{N_{AGV}-1} \overline{\phi_l}) \\
&+ (\epsilon/2)(|\psi_k + \sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\psi_l}|^2 - |\sum_{l=1,l \neq k}^{N_{AGV}-1} \overline{\psi_l}|^2 - 2\overline{\psi_k} \sum_{l=1}^{N_{AGV}-1} \overline{\psi_l})
\end{aligned}
\tag{28}
$$

The dual function is defined as the minimum value obtained by the objective function L_c^k for $\hat{u}$. For each set of (λ_1, λ_2), we need to find $\hat{u}$ that minimizes L_c^k, while different (λ_1, λ_2) correspond to different values of the dual function. The dual function of each subproblem for AGV k is defined as follows:

$$
q_k(\lambda_1, \lambda_2) = \min_{\tilde{u}} L_c^k(\tilde{u}, \lambda_1, \lambda_2). \tag{29}
$$

Thus, the related dual problem can be written as follows:

$$
\max_{\lambda_1, \lambda_2} q(\lambda_1, \lambda_2) = \sum_{k=1}^{N_{AGV}} q_k(\lambda_1, \lambda_2). \tag{30}
$$

B. ADMM-based DMPC

After providing the decomposed dual problem, this part details the procedures of the proposed serial iterative ADMM-based distributed algorithm. The serial scheme is effectively scalable to coordinate subproblems for the distributed MPC[7], and this scheme can be easily implemented based on the above decomposed augmented Lagrangian function.

Algorithm 1 provides the pseudocode of the ADMM-based distributed algorithm. In the proposed distributed algorithm, the original centralized problem P_2 is decomposed into multiple blocks, and each block is updated alternately by following the vehicle index sequence. In each iteration, when the solution of each subproblem for a particular AGV is computed, the predictive path related to the subproblem is sent to the other AGVs in serial communication. As a result, possible collisions

---

Algorithm 1 Serial iterative ADMM-based distributed MPC

Require: Task assignment of AGVs

1: **while** $t \in \{0, 1, ..., H\}$ **do**
2:     Initialize Lagrangian multipliers $\lambda_1 = 0$ and $\lambda_2 = 0$ and parameters $\epsilon_0 = 10$.
3:     **for** iteration $p = 1 : p_{\max}$ **do**
4:         **for** AGV $k = 1 : N_{\mathrm{AGV}}$ **do**
5:             Each AGV determines $\hat{u}_k$ by solving the local subproblem (31)
6:             Update Lagrangian multipliers $\lambda_1(p, t)$ and $\lambda_2(p, t)$ by (32)
7:             Update primal residual $r(p, t)$ and dual residual $s(p, t)$ by (34), (35)
8:             Update penalty coefficient $\epsilon(p, t)$ by (33)
9:         **end for**
10:         // Stopping condition
11:         **if** $r(p, t) \le r_{\lim}$ or $s(p, t) \le s_{\lim}$ or $p = p_{\max}$, then break
12:         $p = p + 1$
13:     **end for**
14:     // Update the state of each AGV and move to next step
15:     $t = t + 1$
16: **end while**

---

are avoided regarding the predictive paths via the predictive position information available. The algorithm stops when the primal and dual residuals are satisfied or the maximum iteration number is reached.

For the decomposed subproblem, the decision variable $\hat{u}_k$ of AGV k at iteration p is computed as follows:

$$\hat{u}_k(p, t) = argmin(q_k(\lambda_1(p, t), \lambda_2(p, t))), \forall k \in \Phi. \quad (31)$$

The Lagrange multipliers are updated as follows:

$$\lambda_1(p+1, t) = \lambda_1(p, t) + \epsilon(p, t)(\sum_k (x_i^k(p+1, t) - 1)$$
$$\lambda_2(p+1, t) = \lambda_2(p, t) + \epsilon(p, t)(\sum_k (u_{i,j}^k(p+1, t) \quad (32)$$
$$+ u_{j,i}^k(p+1, t)) - 1).$$

Penalty parameter ϵ is updated by a self-adaptive method, as suggested in [38], as follows:

$$\epsilon(p+1, t) = \begin{cases} 2\epsilon(p, t), & \text{for } r(p, t) \ge 10s(p, t) \\ \epsilon(p, t)/2, & \text{for } s(p, t) \ge 10r(p, t) \\ \epsilon(p, t), & \text{otherwise} \end{cases} \quad (33)$$

where $r(p, t)$ and $s(p, t)$ are the primal and dual residuals, respectively. $r(p, t)$ and $s(p, t)$ are defined as follows:

$$r(p, t) = ||x^T(p, t) - \overline{x}(p, t)||_2, \quad (34)$$

where $\overline{x}(p, t) = (\frac{1}{N_n} \sum_{i=1}^{N_n} x_i^1(p, t), ..., \frac{1}{N_n} \sum_{i=1}^{N_n} x_i^{N_{\mathrm{AGV}}}(p, t))$ is the average preprocessing result for each subproblem in the state vector at iteration p at time t.

$$s(p+1, t) = -\epsilon(p+1, t)||\overline{x}(p+1, t) - \overline{x}(p, t)||_2 \quad (35)$$

Algorithm 1 continues until $r(p, t) \le r_{\lim}$ or $s(p, t) \le s_{\lim}$ or $p = p_{\max}$. $r_{\lim}$ and $r_{\lim}$ are set to 10^{-3}, as suggested in [35].

V. COMPUTATIONAL EXPERIMENT RESULTS

This section evaluates and discusses the results of the proposed methodology via the MLD model and the developed solution approaches. Numerical case studies are carried out to assess the performance. First, the involved case study setting is introduced. Then, the results on the proposed control method for dealing with uncertainty are discussed, and the advantage of our distributed MPC method is assessed in terms of productivity and computation metrics.

A. Experiment settings

To evaluate the performance of the proposed planning strategy, we consider two benchmarks related to two types of AGV roadmaps. Benchmark 1 corresponds to a squared roadmap graph, which contains $m \times m$ nodes and $2m \times (m - 1)$ links [28], [39], as introduced in Fig. 1. Benchmark 2 contains an irregular graph with unbalanced connections, which is derived from a real world transport company, as shown in Fig. 4 and in [5]. In Fig. 4, we can observe that the roadmap of Benchmark 2 is not regular because the topology is not squared. Its connectivity is not as strong as the one of Benchmark 1 because multiple nodes of Benchmark 1 have more than 2 adjacent nodes, while most nodes of Benchmark 2 have only two adjacent nodes. These two roadmap layouts can be found in related applications for AGVs and robots [28], [39], [5].

We evaluate the following five methods: centralized MPC (C-MPC), ADMM-based distributed MPC (D-MPC), prioritized MPC (P-MPC), static planning (SP), and dynamic prioritized planning (DPP). P-MPC plans the path of each AGV following a fixed priority based on the MPC framework and regards the planned paths as collision constraints. SP is a mainstream planning method implemented via the time-space model proposed in [5]. DPP is a state-of-the-art dynamic approach that uses a dynamically updated priority list to resolve AGV conflicts based on offline paths [22]. The three MPC methods are based on the proposed MLD model, which predicts the feasible AGV positions.

The mathematical modeling is implemented in Python on Windows 10, while solver Gurobi 9.0.3 is used for solving the formulated planning problems. The computer hardware is Intel Core i7-8750 (3.0 Hz) with 16 GB of memory.

We consider three commonly used key performance indicators (KPIs) as follows:
- The sum of completion times of all the AGVs, which is defined as cost1.
- The completion time of the last AGV (makespan), which is defined as cost2.
- Computational time, (CT for short), which is the time it takes to compute its optimal solution and certify its optimality, which is guaranteed by the solver, Gurobi.

The first KPI is stricter than the second since the makespan focuses on the last completed task. Next, we present the computational results with normal operations and uncertainties.
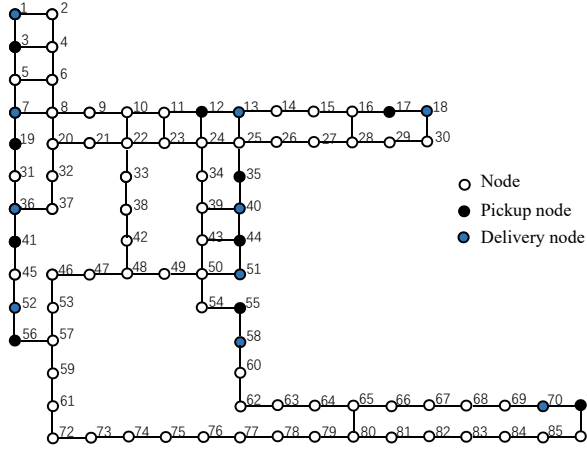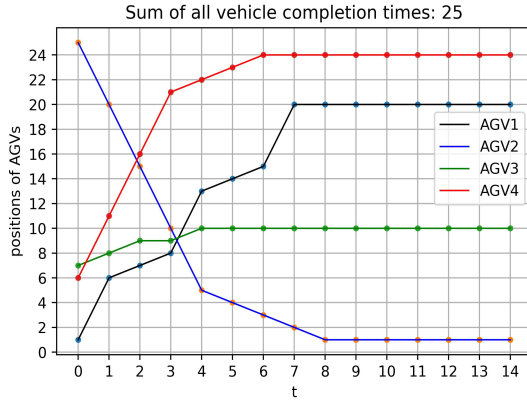
Fig. 4. Illustration of the second roadmap layout as considered in [5].

TABLE III
PERFORMANCE COMPARISONS UNDER DIFFERENT T_p VALUES (UNIT SECONDS).

| Settings		$T_\mathrm{p} = 2$		$T_\mathrm{p} = 3$		$T_\mathrm{p} = 4$		$T_\mathrm{p} = 5$	
m	N_AGV	cost1	CT	cost1	CT	cost1	CT	cost1	CT
5	4	25	0.08	24	0.10	24	0.18	24	0.26
5	5	28	0.10	28	0.13	28	0.19	28	0.28
6	6	35	0.31	34	0.27	32	0.35	32	0.54
7	7	43	0.41	41	0.48	40	0.73	40	0.91

Fig. 5. Planned AGV paths when $T_\mathrm{p} = 2$ (4 AGVs))

B. Choice of planning horizon

In the proposed MPC strategy for local planning, the planning horizon T_p is a key parameter. For a very small T_p, unnecessary waiting time increases as the path prediction is not efficient, so the planning is not productive. Each arc (i, j) has a unit time slot, and thus, we consider $T_\mathrm{p} > 1$.

Table III compares the experimental results, searching for a proper T_p. It was found from Table III that $T_\mathrm{p} = 4$ reaches the best cost1 with the shortest T_p, and we choose $T_\mathrm{p} = 4$ in the following experiments.

Figs. 5 and 6 provide the paths of four AGVs in a 5×5 grid roadmap under the conditions $T_\mathrm{p} = 2$ and $T_\mathrm{p} = 3$. Both figures show that all possible collisions are avoided since there is no more than one AGV in the same node for any t. When comparing these two figures, case $T_\mathrm{p} = 3$ has better transport productivity than case $T_\mathrm{p} = 2$, as more roadmap information
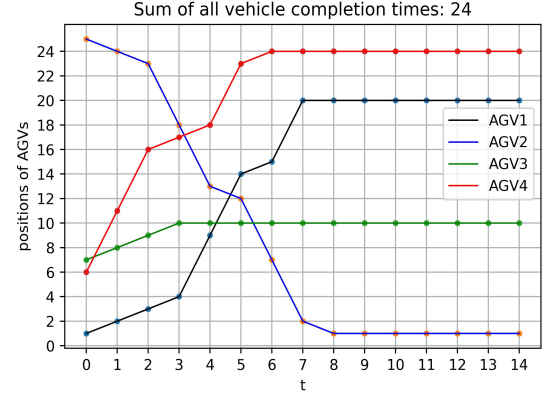
Fig. 6. Planned AGV paths when $T_\mathrm{p} = 3$ (4 AGVs))

can be considered for local planning.

C. Computational Results

We first evaluate the computational results of the proposed C-MPC and D-MPC methods in comparison to the SP, P-MPC, and DPP methods. These experimental results are reported in Table IV and Table V for Benchmarks 1 and 2, respectively.

Table IV provides the performance of the five tested methods concerning Benchmark 1, which uses the squared roadmap (m changing from 5 to 20). Table IV shows that the proposed D-MPC method presents a good balance between the objective function value and the computational burden. There are small gaps (approximately 2%) in the productivity objectives (costs 1 and 2) between the proposed D-MPC and the static planner SP. These gaps result from approximating the global cost when using the MPC planner. However, SP struggles with computational complexity as all the roadmap information is considered, and the computation times of large-scale scenarios can be more than 1000 seconds. Moreover, SP assumes that there are no disruptions during the transport process. Our MPC planner has a great advantage in computational efficiency, and the proposed D-MPC method computes a solution in approximately 10 seconds for large-scale scenarios (against approximately 1000 seconds using SP), which has the potential for real-time applications in industrial environments. Compared to the large gap in the computation time, the gaps in the productivity metrics are relatively small with respect to costs 1 and 2.

Table IV shows the effectiveness of the proposed ADMM-based decomposition technique, since D-MPC achieves the same values of costs 1 and 2 as C-MPC. Note that the values of cost 1 and cost 2 are computed when all the tasks are completed. Since the paths of C-MPC and D-MPC are the same, the values of cost 1 and cost 2 of D-MPC are equal to those of C-MPC. However, some small errors exist due to the decomposition technique introduced by ADMM. Fig. 7 provides the converging trajectories of the primal residual $r(p)$ and the dual residual $s(p)$ in addition to the cost difference between the proposed D-MPC and C-MPC when applying the MPC of the instance with $m = 10$. These converging behaviors show that there is a small difference between the

TABLE IV
COMPARISON OF NUMERICAL RESULTS WITH RESPECT TO BENCHMARK 1. (UNITS: SECONDS)

Settings		SP			C-MPC			D-MPC			P-MPC			DPP		
m	N_{AGV}	cost1	cost2	CT	cost1	cost2	CT	cost1	cost2	CT	cost1	cost2	CT	cost1	cost2	CT
5	4	24.5	7.6	0.76	25.0	7.8	0.11	25.0	7.8	0.07	25.5	7.9	0.11	26.5	8.3	0.15
5	5	30.3	7.9	0.94	31.1	8.1	0.13	31.1	8.1	0.08	31.3	8.2	0.12	33.0	8.8	0.15
6	5	29.8	8.3	1.87	30.6	8.5	0.24	30.6	8.5	0.17	31.1	8.6	0.27	32.0	9.0	0.15
6	6	34.3	8.5	2.21	35.3	8.7	0.27	35.3	8.7	0.20	35.6	8.8	0.29	36.9	9.2	0.16
7	6	34.9	9.1	3.41	35.9	9.4	0.35	35.9	9.4	0.22	36.2	9.4	0.35	38.0	9.8	0.16
7	7	38.7	9.6	3.73	39.6	9.9	0.46	39.6	9.9	0.26	40.2	9.9	0.36	41.3	10.4	0.28
8	7	38.9	9.9	5.71	39.9	10.2	0.55	39.9	10.2	0.30	40.4	10.2	0.45	42.2	10.8	0.30
8	8	44.7	10.1	6.37	46.0	10.4	0.77	46.0	10.4	0.34	46.7	10.4	0.67	47.9	11.0	0.35
9	8	46.4	10.8	8.71	47.5	11.1	0.84	47.5	11.1	0.42	47.9	11.2	0.71	49.7	11.7	0.36
9	9	50.7	11.3	9.88	52.3	11.7	1.08	52.3	11.7	0.44	52.6	11.7	0.73	49.8	12.2	0.36
10	9	51.2	12.1	13.78	52.5	12.5	2.11	52.5	12.5	0.58	53.1	12.6	1.24	54.6	13.0	0.39
10	10	55.3	12.4	15.61	57.3	12.8	3.35	57.3	12.8	0.71	57.7	13.0	1.76	59.0	13.3	0.45
15	12	166.9	18.1	476.19	172.0	18.7	47.79	172.0	18.7	5.46	174.7	18.8	8.91	176.9	19.4	4.92
15	16	206.6	18.6	624.95	212.9	19.2	55.24	212.9	19.2	7.23	216.5	19.4	13.68	219.8	20.0	7.41
15	20	245.6	18.1	897.39	253.4	18.7	64.77	253.4	18.7	9.44	258.2	18.9	16.81	261.5	19.5	10.26
15	25	307.3	18.1	1391.07	316.4	18.3	71.26	316.4	18.3	13.26	320.7	18.7	16.93	332.6	19.2	12.18
15	30	368.4	18.2	2109.18	397.2	18.3	90.43	378.3	18.3	17.49	375.6	18.7	22.01	384.2	19.2	14.72
20	12	178.4	23.2	896.81	180.7	24.0	57.81	180.7	24.0	7.41	185.6	24.2	17.91	190.0	24.9	8.01
20	16	235.2	25.6	1128.98	242.4	26.5	65.64	242.4	26.5	9.21	245.0	26.6	24.86	253.8	27.5	9.53
20	20	282.9	23.7	1496.21	291.6	24.5	75.61	291.6	24.5	14.11	295.3	24.7	28.36	305.0	25.4	12.96
20	25	357.9	24.7	2091.32	365.9	25.1	94.36	365.9	25.1	18.24	372.9	25.2	32.88	378.9	25.6	14.14
20	30	428.6	25.1	3126.48	437.2	25.4	113.37	437.2	25.4	21.33	445.1	25.5	35.41	453.1	26.0	16.92
Average		148.07	15.05	650.53	152.85	15.45	33.93	152.85	15.45	5.77	154.00	15.57	10.22	157.58	16.14	5.19

TABLE V
COMPARISON OF NUMERICAL RESULTS WITH RESPECT TO BENCHMARK 2. (UNITS: SECONDS)

Settings	SP			C-MPC			D-MPC			P-MPC			DPP		
N_{AGV}	cost1	cost2	CT	cost1	cost2	CT	cost1	cost2	CT	cost1	cost2	CT	cost1	cost2	CT
3	32.1	13.4	45.94	32.8	13.8	2.93	32.8	13.8	0.47	33.1	13.9	0.63	34.0	14.2	1.06
4	45.7	13.5	76.87	46.7	14.0	3.42	46.7	14.0	0.69	47.2	14.0	0.87	49.3	14.4	1.52
5	56.4	13.8	85.87	57.8	14.3	4.27	57.8	14.3	1.07	58.5	14.3	1.49	60.0	14.8	2.06
6	67.6	14.1	114.71	69.4	14.6	6.27	69.4	14.6	1.68	70.2	14.6	2.07	72.2	15.0	2.34
7	79.2	15.1	192.75	81.3	15.6	11.59	81.3	15.6	1.77	82.6	15.6	2.21	83.8	16.1	2.88
8	95.4	14.8	207.65	98.0	15.3	14.78	98.0	15.3	2.14	99.7	15.4	2.87	101.0	15.8	3.42
9	107.2	15.2	221.36	110.3	15.8	21.64	110.3	15.8	2.97	112.4	15.8	3.96	113.0	16.2	3.70
10	123.8	15.3	269.89	127.7	15.9	29.88	127.7	15.9	3.66	129.2	15.9	4.22	130.6	16.3	3.89
15	181.6	15.4	589.42	185.3	16.0	49.76	185.3	16.0	7.43	188.1	16.1	9.21	189.9	16.7	5.95
20	248.7	15.6	840.61	253.4	16.0	57.66	253.4	16.0	9.07	257.7	16.2	12.89	265.3	16.5	8.71
25	307.9	15.7	1447.23	313.8	16.2	68.43	313.8	16.2	12.39	318.2	16.3	16.39	327.4	16.8	11.02
30	369.2	15.4	2037.56	375.2	15.9	84.36	375.2	15.9	16.97	381.3	16.1	20.01	390.8	16.4	13.06
Average	142.9	14.78	510.82	145.98	15.3	29.58	145.98	15.3	5.03	148.2	15.4	7.46	151.4	15.8	4.97

value of the augmented Lagrangian function (Equation (23)) and the value of the original objective function (Equation (19)). This difference comes from inequality constraints (11) and (12) because $x_i^k(t) = 0$ and $u_{i,j}^k(t) + u_{j,i}^k(t) = 0$ for these nodes that are not visited by any AGV. The small errors are reported with more decimals in Fig. 7.

Table V reports the performance of the five methods for Benchmark 2, and the computational results on Benchmark 2 are consistent with those on Benchmark 1. The D-MPC achieves high productivity with a much lower computation time among the five methods, which can also be seen in the irregular roadmap of Fig. 4.

Tables IV and V show that the proposed D-MPC gives high competitive performance in comparison to C-MPC, P-MPC and DPP. Costs 1 and 2 of D-MPC are the closest to those of SP, with almost the shortest computation time. In the next subsection, we show the advantage of the proposed MPC methods when dealing with the uncertainties during the transport process.

Fig. 7. Convergence behavior of primal residuals, dual residuals and their difference with the optimal cost by the D-MPC.

D. Addressing uncertainties

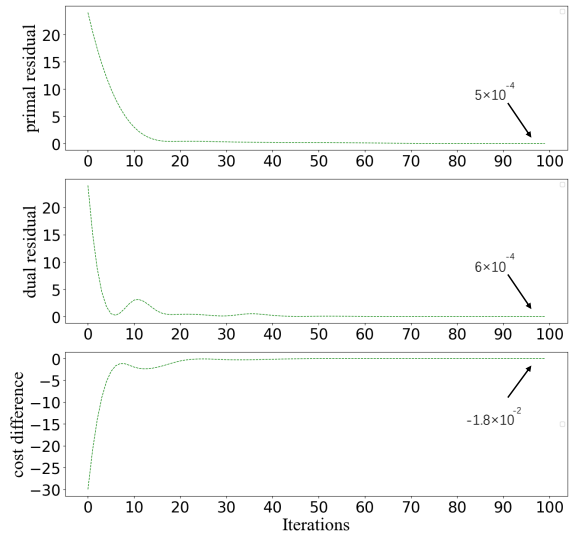We next present the advantages of the proposed MPC strategy when dealing with uncertainties in the production environment during operations. We consider the second instance of Benchmark 2 as an illustrative example, and the corresponding roadmap topology is given in Fig. 4. In this

instance, four AGVs move from their origin to their destination (AGV 1:Node 18→Node 42, AGV 2:Node 57→Node 11, AGV 3:Node 22→Node 40, AGV 4:Node 58→Node 20). To show the effectiveness of the proposed real-time decision, the centralized MPC method is compared with the static planning method, in which the AGV paths are predetermined offline.

Figs. 8 and 9 depict the predetermined paths when using static planning and the adjusted paths when using the proposed C-MPC method. The planned path is marked with a dashed line, while the actual path is labeled with a solid line. It is assumed that AGV 3 breaks down at $t = 1$ in node 33. Fig. 8 shows that the other three AGVs collide with AGV 3. AGVs 1, 2 and 4 collide with AGV 3 at $t = 11$, $t = 8$, and $t = 12$, respectively. Because these paths are scheduled off-line, they cannot be adjusted when the breakdown of AGV 3 takes place.

Fig. 9 shows that the paths of AGVs 1, 2, and 4 are adjusted after the breakdown of AGV 3 via the proposed MPC method. After this uncertainty has happened ($t = 1$), these three AGVs do not follow their predefined paths and need to search for alternative collision-free paths. The three unbroken AGVs complete their transport tasks successfully, although the total transport time of these three AGVs increases slightly because some connections become impossible.

In addition to the AGV breakdown of AGVs, other types of uncertainties, such as operation delay for picking up the materials and changes in the delivery destinations, can also be taken into consideration by the proposed MPC method.

VI. CONCLUSION AND FUTURE RESEARCH

The decision-making process for path planning of AGVs in the manufacturing and logistics environment is expected to be able to deal with uncertainty in the transport process with a low computational burden. For this reason, we propose a new methodological contribution to the dynamic path planning of automated guided vehicles in the bidirectional guide-path layout. The material transport process is mathematically described as a dynamical system via an MLD representation. Based on the MLD model, a customized MPC planner is proposed to dynamically determine collision-free AGV paths, and an ADMM-based decomposition technique is introduced to improve the computational efficiency.

The advantages of the proposed methodology are verified on case studies derived from industrial scenarios in comparison to the commonly used static planning and state-of-the-art online methods. First, the ADMM-based D-MPC achieves high productivity metrics (i.e., the sum of completion times and makespan) at a low computational burden compared to state-of-the-art online approaches (P-MPC and DPP). This superiority is valuable when implementing planning decisions in manufacturing and logistics. It was found that the computation time of the distributed MPC method is only 1% of the static planner with the increase of the objective values by less than 3% on average. Second, the uncertainties (e.g., vehicle breakdown and changes in the delivery destination) during the transport process have been better addressed by the proposed MPC planner. The related case studies show that the transport completion rate is considerably improved
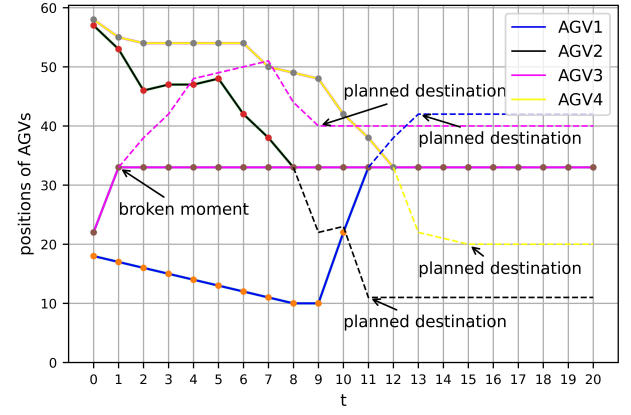

Fig. 8. Example of the static planning method in the case of a breakdown (dashed line–planned path, solid line–actual path).
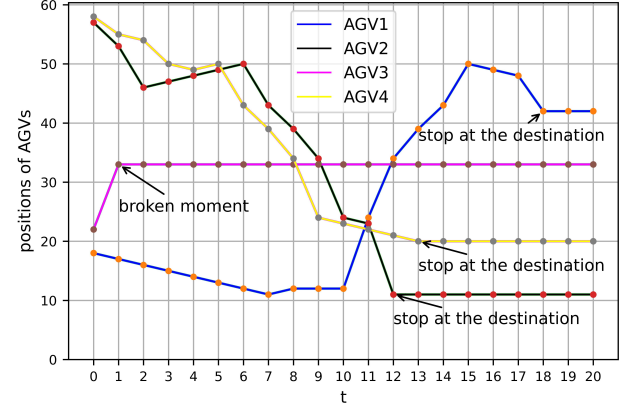

Fig. 9. Example of the proposed MPC method in the case of a breakdown

by the proposed MPC planner compared to the static planner. These improvements result from the predictable MLD model and the ADMM-based distribution coordination, and both are integrated under the MPC framework to achieve these benefits.

Regarding future research work, model-free approaches, such as deep Q-learning, can be investigated as an alternative approach to model-based solving methods. Moreover, the proposed distributed method can also be incorporated into the guide-path design problem of the AGVs, reducing the computational burden when designing the AGV layout.

REFERENCES

[1] W. Małopolski, "A sustainable and conflict-free operation of agvs in a square topology," *Computers & Industrial Engineering*, vol. 126, pp. 472–481, 2018.

[2] J. Xin, C. Meng, A. D'Ariano, D. Wang, and R. R. Negenborn, "Mixed-integer nonlinear programming for energy-efficient container handling: formulation and customized genetic algorithm," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[3] M. P. Fanti, A. M. Mangini, G. Pedroncelli, and W. Ukovich, "A decentralized control strategy for the coordination of agv systems," *Control Engineering Practice*, vol. 70, pp. 86–97, 2018.

[4] M. Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *Journal of Manufacturing Systems*, vol. 54, no. 1, pp. 152–173, 2020.

[5] T. Nishi, S. Akiyama, T. Higashi, and K. Kumagai, "Cell-based local search heuristics for guide path design of automated guided vehicle systems with dynamic multicommodity flow," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 966–980, 2020.

[6] G. Fragapane, R. De Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021.

[7] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 87–97, 2014.

[8] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.

[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[10] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2017.

[11] W.-Q. Zou, Q.-K. Pan, T. Meng, L. Gao, and Y.-L. Wang, "An effective discrete artificial bee colony algorithm for multi-agvs dispatching problem in a matrix manufacturing workshop," *Expert Systems with Applications*, vol. 161, p. 113675, 2020.

[12] T. Nishi, M. Ando, and M. Konishi, "Distributed route planning for multiple mobile robots using an augmented lagrangian decomposition and coordination technique," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1191–1200, 2005.

[13] K. Murakami, "Time-space network model and milp formulation of the conflict-free routing problem of a capacitated agv system," *Computers & Industrial Engineering*, vol. 141, p. 106270, 2020.

[14] J. Luo, Y. Wan, W. Wu, and Z. Li, "Optimal petri-net controller for avoiding collisions in a class of automated guided vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4526–4537, 2020.

[15] T. Nishi and Y. Tanaka, "Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 5, pp. 1230–1243, 2012.

[16] T. Miyamoto and K. Inoue, "Local and random searches for dispatch and conflict-free routing problem of capacitated agv systems," *Computers & Industrial Engineering*, vol. 91, pp. 1–9, 2016.

[17] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian, "An ant colony algorithm (aca) for solving the new integrated model of job shop scheduling and conflict-free routing of agvs," *Computers & Industrial Engineering*, vol. 86, pp. 2–13, 2015.

[18] G. Yi, Z. Feng, T. Mei, P. Li, W. Jin, and S. Chen, "Multi-agvs path planning based on improved ant colony algorithm," *The Journal of Supercomputing*, vol. 75, no. 9, pp. 5898–5913, 2019.

[19] M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache, "Multi-agv scheduling for conflict-free path planning in automated container terminals," *Computers & Industrial Engineering*, vol. 142, p. 106371, 2020.

[20] K. Fransen, J. Van Eekelen, A. Pogromsky, M. A. Boon, and I. J. Adan, "A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems," *Computers & Operations Research*, vol. 123, p. 105046, 2020.

[21] Q. Yang, Y. Lian, and W. Xie, "Hierarchical planning for multiple agvs in warehouse based on global vision," *Simulation Modelling Practice and Theory*, vol. 104, p. 102124, 2020.

[22] M. A. Guney and I. A. Raptis, "Dynamic prioritized motion coordination of multi-agv systems," *Robotics and Autonomous Systems*, vol. 139, p. 103534, 2021.

[23] E. F. Camacho, D. R. Ramírez, D. Limón, D. M. De La Peña, and T. Alamo, "Model predictive control techniques for hybrid systems," *Annual reviews in control*, vol. 34, no. 1, pp. 21–31, 2010.

[24] J. Tobajas, F. Garcia-Torres, P. Roncero-Sánchez, J. Vázquez, L. Bellatreche, and E. Nieto, "Resilience-oriented schedule of microgrids with hybrid energy storage system using model predictive control," *Applied Energy*, vol. 306, p. 118092, 2022.

[25] A. Cataldo and R. Scattolini, "Dynamic pallet routing in a manufacturing transport line with model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1812–1819, 2016.

[26] I. I. Sirmatel and N. Geroliminis, "Mixed logical dynamical modeling and hybrid model predictive control of public transport operations," *Transportation Research Part B: Methodological*, vol. 114, pp. 325–345, 2018.

[27] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[28] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1163–1177, 2016.

[29] S. Riazi, K. Bengtsson, and B. Lennartson, "Energy optimization of large-scale agv systems," *IEEE Transactions on automation science and engineering*, vol. 18, no. 2, pp. 638–649, 2020.

[30] D. Kornhauser, G. Miller, and P. Spirakis, "Coordinating pebble motion on graphs, the diameter of permutation groups, and applications," in *25th Annual Symposium onFoundations of Computer Science, 1984.* IEEE Computer Society, 1984, pp. 241–250.

[31] E. T. S. Alotaibi and H. Al-Rawi, "A complete multi-robot path-planning algorithm," *Autonomous Agents and Multi-Agent Systems*, vol. 32, no. 5, pp. 693–740, 2018.

[32] M. Rinaldi, E. Picarelli, A. D'Ariano, and F. Viti, "Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications," *Omega*, vol. 96, p. 102070, 2020.

[33] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Predictive path following with arrival time awareness for waterborne agvs," *Transportation Research Part C: Emerging Technologies*, vol. 70, pp. 214–237, 2016.

[34] E. K. Mohammadi and B. Shirazi, "Toward high degree flexible routing in collision-free fmss through automated guided vehicles' dynamic strategy: A simulation metamodel," *ISA Transactions*, vol. 96, pp. 228–244, 2020.

[35] L. Chen, Y. Huang, H. Zheng, H. Hopman, and R. Negenborn, "Co-operative multi-vessel systems in urban waterway networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3294–3307, 2020.

[36] S. Boyd, N. Parikh, E. C. hu, B. Peleato, and J. Ec Kstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations & Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[37] G. Cohen and D. L. Zhu, "Decomposition-coordination methods in large-scale optimization problems. the non-differentiable case and the use of augmented lagrangians," *Advances in Large Scale Systems*, vol. 1, 1983.

[38] B. He, H. Yang, and S. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, no. 2, pp. 337–356, 2000.

[39] T. Adamo, T. Bektaş, G. Ghiani, E. Guerriero, and E. Manni, "Path and speed optimization for conflict-free pickup and delivery under time windows," *Transportation Science*, vol. 52, no. 4, pp. 739–755, 2018.
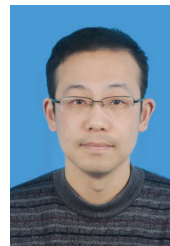
Jianbin Xin obtained BSc. degree in Electrical Engineering from Xidian University in 2007 and MSc degree in Control Science and Engineering from Xi'an Jiaotong University in 2010 both in China. In 2015, he received Ph.D. degree from Department of Maritime and Transport Technology from Delft University of Technology, the Netherlands.

Currently, he is an Associate Professor in Department of Automation at Zhengzhou University, China. His research interests include planning and control of smart logistics systems and cooperative robots.

Xuwen Wu received the Bachelor degree in Electrical Engineering from Zhengzhou University in 2020.

Currently, he is a postgraduate student in School of Electrical Engineering, Zhengzhou University. His research interest is in planning of multiple robots for manufacturing and logistics.

Andrea D'Ariano received the B.S. and M.S. degrees in Computer Science, Automation and Management Engineering at Roma Tre University. In April 2008, he successfully concluded his Ph.D. studies under the supervision of Prof. I.A. Hansen, from Department of Transport and Planning, Delft University of Technology.

Currently, he is an Associate Professor at Department of Engineering, Roma TRE University. He is also Associate Editor of well-known international journals (e.g. Transportation Research Part B, C, E) and conferences (e.g. IEEE ITSC). His main research interest is the study of scheduling problems with application to public transportation and logistics.

Rudy Negenborn received the MSc degree in computer science from Utrecht University in 2003, and the PhD degree from Delft Center for Systems and Control of Delft University of Technology in 2007.

Rudy Negenborn is now Full Professor "Multi-Machine Operations & Logistics" and head of section "Transport Engineering & Logistics" at Delft University of Technology. His fundamental research interests are in the areas of distributed control, multi-agent systems, model predictive control, and optimization.

Fangfang Zhang received the B.E. degree in applied mathematics, the M.E. degree in applied mathematics, and the Ph.D. degree in Control Science and Engineering from Shandong University, Jinan, China, in 2008, 2011, and 2015, respectively.

He is currently an Associate Professor with Zhengzhou University, Zhengzhou, China. His research interests include optimal control of multi-agent systems, multirobot formation, and machine vision.