GENO2318342 VOL 00, ISS 00

Distributed predictive motion planning of automated guided vehicles: serial versus parallel schemes

Xuwen Wu, Jianbin Xin, and Andrea D’Ariano

QUERY SHEET

This page lists questions we have about your paper. The numbers displayed at left are hyperlinked to the location of the query in your paper.

The title and author names are listed on this sheet as they will be published, both on your paper and on the Table of Contents. Please review and ensure the information is correct and advise us if any changes need to be made. In addition, please review your paper as a whole for typographical and essential corrections.

Your PDF proof has been enabled so that you can comment on the proof directly using Adobe Acrobat. For further information on marking corrections using Acrobat, please visit https://authorservices.taylorandfrancis.com/how-to-correct-proofs-with-adobe/

The CrossRef database (www.crossref.org/) has been used to validate the references. Changes resulting from mismatches are tracked in red font.

AUTHOR QUERIES

QUERY NO.	QUERY DETAILS
Q1	Taylor & Francis will complete the Received/Accepted information.
Q2	Does Industry 4.0 need a reference?
Q3	Does MLD need a reference?
Q4	Please define LiDAR and SLAM. Do they need references?
Q5	Should ”flop shop scheduling” be ”flow shop scheduling”?
Q6	Does the ”improved A* algorithm” need a reference or an explanation?
Q7	The antecedent of ”their” is unclear. Please check that the sentence reads correctly or provide an alternative version.
Q8	Should ”reply” be ”relies”?
Q9	Should ”rlim and rlim” be ”rlim and slim”?
Q10	Is there a spurious opening parenthesis, or a missing closing parenthesis, in this line?

QUERY NO.	QUERY DETAILS
Q11	Authors should restrict their use of colour to situations where it is necessary on scientific, and not merely cosmetic, grounds. Colour figures will be reproduced in colour in your online article free of charge. If it is necessary for the figures to be reproduced in colour in the print version, a charge will apply. Charges for colour figures in print are 300 per figure (400$USDollars$;500 Australian Dollars; 350). For more than 4 colour figures, the fifth and above will be charged at 50 per figure (75$USDollars$;100 Australian Dollars; 65). Depending on your location, these charges may be subject to local taxes.
Q12	Authors should restrict their use of colour to situations where it is necessary on scientific, and not merely cosmetic, grounds. Colour figures will be reproduced in colour in your online article free of charge. If it is necessary for the figures to be reproduced in colour in the print version, a charge will apply. Charges for colour figures in print are 300 per figure (400$USDollars$;500 Australian Dollars; 350). For more than 4 colour figures, the fifth and above will be charged at 50 per figure (75$USDollars$;100 Australian Dollars; 65). Depending on your location, these charges may be subject to local taxes.
Q13	Authors should restrict their use of colour to situations where it is necessary on scientific, and not merely cosmetic, grounds. Colour figures will be reproduced in colour in your online article free of charge. If it is necessary for the figures to be reproduced in colour in the print version, a charge will apply. Charges for colour figures in print are 300 per figure (400$USDollars$;500 Australian Dollars; 350). For more than 4 colour figures, the fifth and above will be charged at 50 per figure (75$USDollars$;100 Australian Dollars; 65). Depending on your location, these charges may be subject to local taxes.
Q14	Authors should restrict their use of colour to situations where it is necessary on scientific, and not merely cosmetic, grounds. Colour figures will be reproduced in colour in your online article free of charge. If it is necessary for the figures to be reproduced in colour in the print version, a charge will apply. Charges for colour figures in print are 300 per figure (400$USDollars$;500 Australian Dollars; 350). For more than 4 colour figures, the fifth and above will be charged at 50 per figure (75$USDollars$;100 Australian Dollars; 65). Depending on your location, these charges may be subject to local taxes.
Q15	Authors should restrict their use of colour to situations where it is necessary on scientific, and not merely cosmetic, grounds. Colour figures will be reproduced in colour in your online article free of charge. If it is necessary for the figures to be reproduced in colour in the print version, a charge will apply. Charges for colour figures in print are 300 per figure (400$USDollars$;500 Australian Dollars; 350). For more than 4 colour figures, the fifth and above will be charged at 50 per figure (75$USDollars$;100 Australian Dollars; 65). Depending on your location, these charges may be subject to local taxes.

QUERY NO.	QUERY DETAILS
Q16	Authors should restrict their use of colour to situations where it is necessary on scientific, and not merely cosmetic, grounds. Colour figures will be reproduced in colour in your online article free of charge. If it is necessary for the figures to be reproduced in colour in the print version, a charge will apply. Charges for colour figures in print are 300 per figure (400$USDollars$;500 Australian Dollars; 350). For more than 4 colour figures, the fifth and above will be charged at 50 per figure (75$USDollars$;100 Australian Dollars; 65). Depending on your location, these charges may be subject to local taxes.
Q17	Is there a spurious opening parenthesis, or a missing closing parenthesis, in this line?
Q18	Please check ”3.0 Hz”.
Q19	Please explain why some values are in boldface type.
Q20	Please explain why some values are in boldface type.
Q21	Should ”Figure 9” be ”Figure 8”?
Q22	The funding information provided (the ”College Youth Backbone Teacher Project of Henan Province” and the ”Henan Scientific and Technological Research Project”) have been checked against the Open Funder Registry at https://doi.crossref.org/funderNames?mode=list and no matches were found. Please check and resupply the funding details.

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2024.2318342

Taylor & Francis
Taylor & Francis Group

Distributed predictive motion planning of automated guided vehicles: serial versus parallel schemes

Xuwen Wu[a], Jianbin Xin[a,b] and Andrea D'Ariano[c]

[a]School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, People's Republic of China; [b]State Key Laboratory of Intelligent Agricultural Power Equipment, Luoyang, People's Republic of China; [c]Dipartimento di Ingegneria Civile, Informatica e delle Tecnologie Aeronautiche, Universit Degli Studi Roma Tre, Rome, Italy

ABSTRACT

The article presents two distributed motion planning methods for dynamically coordinating multiple automated guide vehicles (AGVs) in an industrial setting, aiming to enhance their flexibility, robustness and scalability. A predictive motion model is utilized to describe the transport process mathematically as a dynamical system. Subsequently, alternating direction method of multipliers (ADMM)-based decomposition techniques, both serial and parallel, are developed to coordinate the AGVs and mitigate the computational burden. The efficacy of the distributed methods is verified through testing in industrial scenarios, demonstrating that they can significantly improve multi-AGV system transport productivity with minimal computational effort. Notably, the parallel scheme exhibits superior performance in coordinating the AGVs compared to the serial scheme.

ARTICLE HISTORY
Received 13 February 2023
Accepted 8 February 2024

KEYWORDS
Automated guided vehicles; motion planning; distribution optimization; alternating direction method of multipliers

1. Introduction

Automated guided vehicles (AGVs), are intelligent mobile robots that are widely used in industries such as manufacturing systems, warehouses, container terminals and more. They are primarily employed for material transportation and task execution (Kim and Hwang 2001; Lee, Cao, and Shi 2009; Xin *et al.* 2022; Zheng *et al.* 2022). With the advent of Industry 4.0 and the increasing demand for non-general and diversified products, AGVs need to become more intelligent, autonomous and efficient. This is crucial in order to enhance the resilience and agility of manufacturing systems under complex and dynamic operational conditions (Ryck, Versteyhe, and Debrouwere 2020). Q2

There are two challenges in planning the operations of an AGV fleet to meet the manufacturer's requirements. First, each AGV should be intelligent enough to respond appropriately to changes in the operational environment—*e.g.* machine breakdown, change of delivery points (Nishi *et al.* 2020)—thereby reducing the economic loss caused by these uncertainties (Fragapane *et al.* 2021). Secondly, each AGV should communicate with the other AGVs to avoid collisions as efficiently as possible (Ryck, Versteyhe, and Debrouwere 2020). The current trend in AGV operation is towards decentralization, and each AGV should decide its actions. To deal with the limitations of the computational burden and the communication band, low-cost but real-time distributed computing is recommended (Negenborn and Maestre 2014).

CONTACT Jianbin Xin ✉ j.xin@zzu.edu.cn

In this article, the focus is directed towards enhancing the flexibility, robustness and scalability of AGV motion planning in manufacturing and logistics environments, addressing the challenges previously mentioned. The AGV transport process is approached from the standpoint of discrete-event dynamical systems and distributed control. The transport process is mathematically modelled as a dynamic system using the mixed logical dynamical (MLD) representation. Additionally, both serial and parallel distributed optimization techniques are developed based on the alternating direction method of multipliers (ADMM) to coordinate multiple AGVs dynamically, providing real-time decision-making value.

1.1. Related work

One of the most fundamental issues that mobile robots must solve to perform autonomous navigation and exploration in challenging situations is motion planning. A mobile robot looking for an optimal or suboptimal path from an initial state to a goal state under specific performance criteria, given a robot and its working environment, is a typical definition of the motion planning problem. The relevant research is examined from a multi-robot perspective.

An AGV moves on a predefined roadmap, which is represented logically by a graph with nodes and arcs. This section focuses specifically on this type of robot. Motion planning is based on the results of task assignment, which optimally distributes a given set of tasks to a fleet of AGVs (Xin *et al.* 2023; Zou *et al.* 2020). The task sequences processed by each AGV are frequently involved when assigning these tasks, and the goal is typically to minimize the makespan (*i.e.* the completion time of all given tasks) or the tardiness of all defined tasks.

Following task assignment, motion planning determines the optimal path to execute these tasks in the guided roadmap from its origin and destination, and collision avoidance must be considered when planning the detailed path (Kumar and Sikander 2023). Since the introduction of new sensing technologies (*e.g.* LiDAR SLAM or visual SLAM), motion planning of multiple AGVs (or mobile robots) has received increased attention. Motion planning for multiple AGVs is more complex than for a single robot because each robot must consider the behaviour of other robots, and neighbouring robots may become dynamic obstacles.

Multiple AGV motion planning can be divided into offline and online approaches. Offline approaches typically determine the collision-free path of each AGV in advance, and the decision is not updated when each robot performs the task. The majority of offline approaches are addressed centrally. The time–space network approach, for example, is regarded as an effective modelling representation for detecting and resolving conflicts in a grid roadmap layout (Murakami 2020; Nishi, Ando, and Konishi 2005). Based on labelled Petri nets, one alternative method is to design an optimal controller to prevent AGV collisions (Luo *et al.* 2020). In addition to these centralized approaches, Fanti *et al.* (2018) propose a decentralized zone-based planning algorithm based on prioritized rules to resolve AGV conflicts.

Offline motion planning of multiple AGVs is also combined with other issues, such as task allocation and scheduling. In Nishi and Tanaka (2012), the researchers look into simultaneous dispatching and conflict-free routing for bidirectional automated guided vehicle systems. Miyamoto and Inoue (2016) combine task allocation and motion planning as a single problem using the time–space network framework. Furthermore, according to the applications required in the production workshop (Saidi-Mehrabad *et al.* 2015; Yi *et al.* 2019) or container terminals (Zhong *et al.* 2020), conflict-free AGV routing approaches are integrated with job shop scheduling and flop shop scheduling.

Online approaches, which are more capable of dealing with unexpected situations, have received little attention in comparison to the offline approaches that have been intensively investigated for the movement of AGVs. The computational load for dynamic coordination between these multiple AGVs can be significant. As a result, one effective way to address this issue is to simplify coordination by employing global congestion-aware metrics to disperse traffic flow and ensure adequate vehicle transport capacity (Fransen *et al.* 2020; Yang, Lian, and Xie 2020). The improved A* algorithm is then used to search for and avoid collisions on the idle path. Furthermore, a dynamic motion planning method for a multi-AGV system has recently been proposed in Guney and Raptis (2021), 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 the motion 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 dynamic model when replanning these AGVs. Owing to the lack of this information, the AGV performance cannot be improved further.

1.2. Contributions and outline of this article

To address the aforementioned limitations, the contributions of this article are given as follows.

- This article presents novel serial and parallel distributed motion planning methods based on ADMM-based decomposition. This coordination is performed dynamically to improve their transport efficiency. Currently, dynamic motion planning has primarily been investigated in a centralized manner and coordination among AGVs primarily reply on prioritized rules—*e.g.* Guney and Raptis (2021)—imposing potential improvements.
- The performance of serial and parallel schemes using ADMM are compared and analysed, including the convergence speed and calculating time. These distributed optimization methods fit the decentralized trend of AGV operations in the future (Ryck, Versteyhe, and Debrouwere 2020). This comparison has not been shown in the current literature.

In this article, a centralized dynamical model for motion planning of multiple AGVs is presented, and this model is prepared for distributed decomposition. Two ADMM-based distributed planning algorithms (serial and parallel) are proposed to decompose the computation. The convergence speed and calculation time of these two distributed patterns are further analysed. Specifically, the multiple motion choices for the same objective value of both the centralized and the distributed planning methods are discussed.

The remainder of the article is organized as follows: Section 2 describes the research problem and provides a dedicated centralized model predictive control (MPC) controller to determine the collision-free paths dynamically. In Section 3, an ADMM-based distributed planning method (serial iterative ADMM and parallel iterative ADMM) is proposed to coordinate the multi-AGV system. Section 4 discusses the results of the proposed methodology on the numerical tests and further analyses its performance. Section 5 concludes this article and provides future research directions.

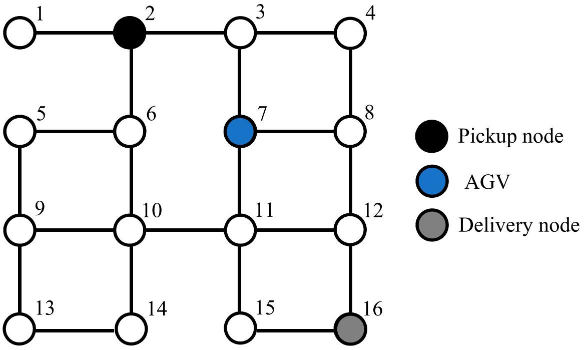
2. Problem description and modelling

This section introduces the investigated motion planning problem of multiple AGVs, as well as its mathematical formulation as a dynamical predictive model.

2.1. Problem statement

Multiple materials must be transported by multiple AGVs in a connected guided path network in an industrial environment. It is necessary to plan a collision-free vehicle path for each AGV from its starting node to its delivery node when transporting materials. Each AGV is assumed to have its starting and delivery nodes assigned in advance.

In Figure 1, a squared AGV roadmap is considered, featuring multiple nodes with equal distances between adjacent nodes. Each node of the roadmap represents a vehicle lane, and AGVs can only wait



Q11

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

or change directions at that node. Each AGV can only reach its adjacent nodes, which are connected to its location node by a single move, and it cannot reach non-adjacent connected nodes by a single move.

In this motion planning problem, the following assumptions are made, as suggested in Nishi, Ando, and Konishi (2005) and Nishi *et al.* (2020).

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

2.2. Dynamical model

In this part, a mixed logical dynamical (MLD) formulation is used to model the transport process of multiple AGVs in the guide path network. The MLD model is a powerful modelling 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) (Camacho *et al.* 2010). The MLD model has been successfully used in the domain of power systems (Tobajas *et al.* 2022), transportation systems (Cataldo and Scattolini 2016; Sirmatel and Geroliminis 2018) and other applications.

The whole planning horizon is equally discretized into a set of time slots denoted by $\{\Delta t, 2 \times \Delta t, \ldots, H \times \Delta t\}$, 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 modelling the movement process, the related state variable $x_i^k(t)$ and control variable $u_{i,j}^k(t)$ are introduced. The related symbols are given in Table 1.

Table 1. Index variables and input parameters.

Symbol	Description
Δt	Discretized time step
T_{p}	Planning time horizon
t	Discretized time instance, $t \in \{0, 1, \dots, T_{\mathrm{p}}\}$
N_i	Set of nodes adjacent 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, $\Phi = \{1, 2, \dots, N_{\mathrm{AGV}}\}$
S_k	Start node of AGV k
G_k	End node of AGV k
N_{L}	Set of local ending nodes within T_{p}
D_{j,G_k}	Approximated distance between node j and the end node G_k

- $x_i^k(t)$ 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 $t + 1$; item $\delta_j^k(t)$ is defined as the auxiliary control variable to linearize the nonlinear term $x_j^k(t) \sum_{i\in N_j, i\neq j} u_{i,j}^k(t)$;
- C_j^k is defined as the control variable to select the local ending node; $C_j^k = 1$ if AGV k arrives the local ending node j at time T_{p}; otherwise $C_j^k = 0$.

2.2.1. MPC representation

Model predictive control (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 by Rinaldi *et al.* (2020) and Zheng, Negenborn, and Lodewijks (2016).

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

$$\widetilde{J} = \sum_{k\in\Phi}\left(\sum_{j\in N_{\mathrm{L}}} C_j^k * D_{j,G_k} + T_{\mathrm{p}}\right), \tag{1}$$

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

The local objective's constraints are represented as follows:

$$\sum_j C_j^k = 1, \quad k \in \Phi, j \in N_{\mathrm{L}} \tag{2}$$

$$x_j^k(T_{\mathrm{p}}) = C_j^k, \quad k \in \Phi, j \in N_{\mathrm{L}}, \tag{3}$$

where constraint (2) ensures that each AGV chooses a local ending node in the considered local problem. Constraint (3) ensures that each AGV can only stay at one of the local ending nodes ($j \in N_{\mathrm{L}}$) by $t = T_{\mathrm{p}}$.

Furthermore, the following linear constraints are utilized to represent the motion dynamics:

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

$$\delta_j^k(t) - x_j^k(t) \le 0, \tag{5}$$

$$\delta_j^k(t) - \sum_{i \in N_j, i \neq j} u_{i,j}^k(t) \le 0, \tag{6}$$

$$x_j^k(t) + \sum_{i \in N_j, i \neq j} u_{i,j}^k(t) - \delta_j^k(t) \le 1. \tag{7}$$

Constraints (4)–(7) provide the predictive motion dynamics of the AGVs as linear constraints, indicating that $x_j^k(t+1) = 1$ if $\sum_{i \in N_j, i \neq j} u_{i,j}^k(t) = 1$, otherwise $x_j^k(t+1) = x_j^k(t)$. The notation $\delta_j^k(t)$ is an auxiliary binary variable used to linearize the nonlinear term $x_j^k(t) \sum_{i \in N_j, i \neq j} u_{i,j}^k(t)$. It is defined as $\delta_j^k(t) \triangleq x_j^k(t) \sum_{i \in N_j, i \neq j} u_{i,j}^k(t)$ and can be expressed by three linear inequalities (Constraints (5)–(7)) to facilitate the modelling.

Additionally, there are other time–space constraints concerning collision avoidance and initial positions, as follows:

$$\sum_{i \in N} x_i^k(t) \le 1, \quad \forall k \in \Phi,\ t = 0, \ldots, T_\mathrm{p} \tag{8}$$

$$\sum_{k \in \Phi} x_i^k(t) \le 1, \quad \forall i \in N,\ t = 0, \ldots, T_\mathrm{p} \tag{9}$$

$$\sum_{k \in \Phi, j \in N_i} (u_{i,j}^k(t) + u_{j,i}^k(t)) \le 1, \quad \forall\ t = 0, \ldots, T_\mathrm{p} \tag{10}$$

$$\sum_{j \in N_i} x_j^k(t+1) \ge x_i^k(t), \quad \forall i \in N,\ t = 0, \ldots, T_\mathrm{p} - 1,\ k \in \Phi, \tag{11}$$

$$x_{S_k}^k(0) = 1, \quad k \in \Phi. \tag{12}$$

Constraint (8) ensures that each AGV stays at only one node at any time. Constraint (9) guarantees that every node can be occupied by at most one AGV. Constraint (10) 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 (11) ensures that AGV k moves node i to its adjacent nodes. Constraint (12) provides the initial position of all the considered AGVs.

In this context, $\hat{\mathbf{u}} = [\mathbf{u}^\mathrm{T}(0), \ldots, \mathbf{u}^\mathrm{T}(T_\mathrm{p} - 1), \delta^\mathrm{T}(0), \ldots, \delta^\mathrm{T}(T_\mathrm{p} - 1), \mathbf{C}^\mathrm{T}]^\mathrm{T}$ is defined, where $\mathbf{C}$ represents the set of C_j^k. Consequently, the local planning problem (defined as P_1) can be formulated as follows:

$$\begin{aligned} (P_1) \quad & \min_{\hat{\mathbf{u}}} \widetilde{J} \\ \text{s.t.} \quad & (2)--3, (4)--(12). \end{aligned}$$

The planning problem P_1 is an integer linear programming (ILP) problem, which is solved in a centralized way. Centralized planning has the drawbacks of heavy computation burden, inflexible structure and technical limitations, and distributed planning is required to distribute the intelligence of the overall AGV system (Negenborn and Maestre 2014). In the next section, two distributed decompositions are proposed to achieve this goal.

3. ADMM-based distributed planning

In this section, ADMM-based distributed planners are presented, and serial and parallel decomposition schemes are proposed to coordinate the AGVs. 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 (Ryck, Versteyhe, and Debrouwere 2020). 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.

The proposed two distributed planning algorithms, as well as the separable optimization formulation for the distributed planner, are presented in the following sections.

3.1. ADMM decomposition

ADMM is a widely used decomposition method for solving distributed MPC problems (Chen *et al.* 2020; Zheng, Negenborn, and Lodewijks 2016). ADMM is recommended since this is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers (Boyd *et al.* 2011). When using the ADMM method, the augmented Lagrangian of the original optimization problem is first constructed, while the primal and dual variables are updated in an alternating or sequential fashion. The algorithm continues until a given stopping criterion is satisfied.

Next, the augmented Lagrangian function regarding the problem P_1 is explained. The capacity constraints (8) and (9) are coupled for the multiple AGVs. Two non-negative Lagrangian multipliers λ_1 and λ_2 are introduced to relax these two coupled constraints. The augmented Lagrangian function of problem P_1 is formulated as follows:

$$\begin{aligned} L_c = {} & \sum_{k\in\Phi}\left(\sum_{j\in N_{\mathrm{L}}} C_j^k * D_{j,G} + T_{\mathrm{p}}\right) \\ & + \lambda_1\left(\sum_{k\in\Phi} x_i^k(t) - 1\right) + \lambda_2\left(\sum_{k\in\Phi}(u_{i,j}^k(t) + u_{j,i}^k(t)) - 1\right) \\ & + \frac{\epsilon}{2}\left(\sum_{k\in\Phi} x_i^k(t) - 1\right)^2 + \frac{\epsilon}{2}\left(\sum_{k\in\Phi}(u_{i,j}^k(t) + u_{j,i}^k(t)) - 1\right)^2 \\ & \forall k \in \Phi,\ i \in N,\ j \in N_{\mathrm{L}},\ t = 0,\ldots,T_{\mathrm{p}}, \end{aligned} \tag{13}$$

where ϵ is the augmented Lagrangian parameter. For the sake of computation, the related notations are defined as follows:

$$\phi_k \triangleq x_i^k(t) - 1/N_{\mathrm{AGV}}, \quad \forall k \in \Phi \tag{14}$$

$$\psi_k \triangleq u_{i,j}^k(t) + u_{j,i}^k(t) - 1/N_{\mathrm{AGV}}, \quad \forall k \in \Phi. \tag{15}$$

The problem of minimizing L_c is not separable into AGV-level sub-problems owing to the incorporated quadratic penalty term. To maintain separability, a linearization technique is used for the cross-penalty part around the estimated optimal solution (Cohen and Zhu 1984). The augmented Lagrangian function L_c can be additive for AGV k by using the first-order Taylor expansion. $|\sum_{k=1}^{N_{\mathrm{AGV}}} \phi_k|^2$ can be expressed as follows:

$$\left|\sum_{k=1}^{N_{\mathrm{AGV}}} \phi_k\right|^2 = \sum_{k=1}^{N_{\mathrm{AGV}}} |\phi_k|^2 + 2\sum_{k=1}^{N_{\mathrm{AGV}}}\sum_{l=1}^{N_{\mathrm{AGV}}-1} \phi_k\phi_l. \tag{16}$$

The last (cross-product) terms of (16) are linearized by a first-order Taylor expansion around the point $(\overline{\phi_k}, \overline{\phi_l})$ for a good estimation. In this way, $|\sum_k \phi_k|^2$ is expanded into the following expressions:

$$\begin{aligned}\left|\sum_{k=1}^{N_{\mathrm{AGV}}} \phi_k\right|^2 &\approx \sum_{k=1}^{N_{\mathrm{AGV}}} \left|\phi_k + \sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\phi_l}\right|^2 - \sum_{k=1}^{N_{\mathrm{AGV}}} \left|\sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\phi_l}\right|^2 - 2\sum_{k=1}^{N_{\mathrm{AGV}}} \overline{\phi_k} \sum_{l=1}^{N_{\mathrm{AGV}}-1} \overline{\phi_l} \\ &= \sum_{k=1}^{N_{\mathrm{AGV}}} \left(\left|\phi_k + \sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\phi_l}\right|^2 - \left|\sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\phi_l}\right|^2 - 2\overline{\phi_k} \sum_{l=1}^{N_{\mathrm{AGV}}-1} \overline{\phi_l} \right), \end{aligned} \tag{17}$$

where $\overline{\phi_k} = \sum_k (\overline{x_i^k(t)} - 1/N_{\mathrm{AGV}})$, 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 sub-problems (one for each AGV). The objective function for AGV k can be written as follows:

$$\begin{aligned} L_c^k = &\left(\sum_{j\in N_{\mathrm{L}}} C_j^k * D_{j,G} + T_{\mathrm{p}}) + \lambda_1 \phi_k + \lambda_2 \psi_k \right) \\ &+ \frac{\epsilon}{2} \left(\left|\phi_k + \sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\phi_l}\right|^2 - \left|\sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\phi_l}\right|^2 - 2\overline{\phi_k} \sum_{l=1}^{N_{\mathrm{AGV}}-1} \overline{\phi_l} \right) \\ &+ \frac{\epsilon}{2} \left(\left|\psi_k + \sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\psi_l}\right|^2 - \left|\sum_{l=1,l\neq k}^{N_{\mathrm{AGV}}-1} \overline{\psi_l}\right|^2 - 2\overline{\psi_k} \sum_{l=1}^{N_{\mathrm{AGV}}-1} \overline{\psi_l} \right). \end{aligned} \tag{18}$$

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

$$q_k(\lambda_1, \lambda_2) = \min_{\tilde{u}} L_c^k(\tilde{u}, \lambda_1, \lambda_2). \tag{19}$$

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

$$\max_{\lambda_1,\lambda_2} q(\lambda_1, \lambda_2) = \sum_{k=1}^{N_{\mathrm{AGV}}} q_k(\lambda_1, \lambda_2). \tag{20}$$

To use ADMM, the following equivalent transformation of the original problem should be performed:

$$\begin{aligned} \min \quad & \sum_{k=1}^{N_{\mathrm{AGV}}} q_k(\tilde{u}_k, \lambda_1, \lambda_2) \\ \text{s.t.} \quad & \tilde{u}_k - \mathbf{x} = 0, \end{aligned} \tag{21}$$

where $\mathbf{x}$ is the system state vector.

$$\mathbf{x}(t) = \left[x_1^1(t), \ldots, x_{N_{\mathrm{n}}}^1(t), \ldots, x_1^{N_{\mathrm{AGV}}}(t), \ldots, x_{N_{\mathrm{n}}}^{N_{\mathrm{AGV}}}(t) \right]^{\mathrm{T}}.$$

The penalty parameter ϵ is updated by a self-adaptive method, as suggested in He, Yang, and Wang (2000), as follows:

$$\epsilon(p+1,t) = \begin{cases} 2\epsilon(p,t), & \text{for } r(p,t) \geq 10s(p,t) \\ \epsilon(p,t)/2, & \text{for } s(p,t) \geq 10r(p,t) \\ \epsilon(p,t), & \text{otherwise,} \end{cases} \tag{22}$$

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:

$$r(p,t) = ||x^{\mathrm{T}}(p,t) - \overline{x}(p,t)||_2, \tag{23}$$

where

$$\overline{x}(p,t) = \left(\frac{1}{N_{\mathrm{n}}} \sum_{i=1}^{N_{\mathrm{n}}} x_i^1(p,t), \ldots, \frac{1}{N_{\mathrm{n}}} \sum_{i=1}^{N_{\mathrm{n}}} x_i^{N_{\mathrm{AGV}}}(p,t) \right)$$

is the average preprocessing result for each sub-problem in the state vector at iteration p at time t.

$$s(p+1,t) = -\epsilon(p+1,t)||\overline{x}(p+1,t) - \overline{x}(p,t)||_2. \tag{24}$$

The overall algorithm continues until $r(p,t) \leq r_{\mathrm{lim}}$ or $s(p,t) \leq s_{\mathrm{lim}}$ or $p = p_{\mathrm{max}}$. r_{lim} and r_{lim} are set to 10^{-3}, as suggested in Chen *et al.* (2020). **Q9**

3.2. Serial iterative ADMM

The ADMM algorithm can adopt a serial iterative solution mechanism (S-ADMM): each time, only one sub-problem is in the computational state, and the shared variable is transferred to another sub-problem when the computation state is finished. Until all sub-problems have been computed, the Lagrange multiplier is updated and the next iteration of the optimization calculation is entered.

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

$$\hat{u}_k(p+1,t) = \operatorname{argmin}\left(q_k\left(\hat{u}_k(p+1,t), \lambda_1(p,t), \lambda_2(p,t)\right)\right), \quad \forall k \in \Phi. \tag{25}$$

The system state vector is updated as follows:

$$\mathbf{x}(p+1,t) = \frac{1}{N_{\mathrm{AGV}}} \sum_{k=1}^{N_{\mathrm{AGV}}} \left(\hat{u}_k(p+1,t) + \frac{\epsilon}{2}\left(\lambda_1(p) + \lambda_2(p)\right) \right). \tag{26}$$

The Lagrange multipliers are updated as follows:

$$\begin{aligned} \lambda_1(p+1,t) &= \lambda_1(p,t) + \epsilon(p,t)\left(\sum_k \left(x_i^k(p+1,t) - 1 \right) \right. \\ \lambda_2(p+1,t) &= \lambda_2(p,t) + \epsilon(p,t)\left(\sum_k \left(u_{i,j}^k(p+1,t) + u_{j,i}^k(p+1,t) \right) - 1 \right). \end{aligned} \tag{27}$$

Lagrangian multipliers λ_1 and λ_2 are updated using the above equation, and the penalty ϵ is updated using Equation (22). **Q10**

If the penalty ϵ is large, the objective function value will be inaccurate when searching for a solution with collisions. If it is too small, the effect of the penalty will not be significant, and the result may not satisfy the relaxed constraints. By initially setting the Lagrangian multipliers as zero, the solution of the original function under extreme conditions (without considering the relaxed constraints) is obtained. As the penalty parameter increases (within a certain range), the searched solution will be closer to the optimal solution. When the penalty parameter is too large, it may easily fall into a suboptimal solution.

3.3. Parallel iterative ADMM

The ADMM algorithm can also adopt the parallel iterative solution mechanism (P-ADMM): each time, the sub-problems are solved in parallel, and the shared variables are transferred to another sub-problem at the same time as the computation is completed. Until all sub-problems have been computed, the Lagrange multiplier is updated and the next iteration of the optimization calculation is entered.

The decision variable $\hat{u}_k$ of AGV k at iteration p is computed as follows:

$$\hat{u}_k(p+1,t) = \operatorname{argmin}\left(q_k\left(\hat{u}_k(p,t), \lambda_1(p,t), \lambda_2(p,t)\right)\right), \quad \forall k \in \Phi. \tag{28}$$

The system state vector is updated as follows:

$$\mathbf{x}(p+1,t) = \frac{1}{N_{\text{AGV}}} \sum_{k=1}^{N_{\text{AGV}}} \left(\hat{u}_k(p,t) + \frac{\epsilon}{2}\left(\lambda_1(p) + \lambda_2(p)\right)\right). \tag{29}$$

The Lagrange multipliers are updated as follows:

$$\begin{aligned} \lambda_1(p+1,t) &= \lambda_1(p,t) + \epsilon(p,t)\Bigg(\sum_k \left(x_i^k(p+1,t) - 1\right) \\ \lambda_2(p+1,t) &= \lambda_2(p,t) + \epsilon(p,t)\Bigg(\sum_k \left(u_{i,j}^k(p+1,t) + u_{j,i}^k(p+1,t)\right) - 1\Bigg). \end{aligned} \tag{30}$$

P-ADMM realizes the distributed parallel optimization of sub-problems. The advantages of P-ADMM is that, when multiple computers are used, the computing efficiency will be greatly increased.

3.4. Coordination comparison

Figures 2 and 3 show the AGV controllers' coordination scheme using the S-ADMM and P-ADMM decompositions. In Figure 2, the serial iterative solution mechanism is used by S-ADMM. When the computation is finished, the sub-controller transfers the state variable of the preceding AGVs to the next sub-controller and output control variables, and the AGVs perform the action. In Figure 3, the parallel iterative solution mechanism is used by P-ADMM. When all of the sub-problems are in the computational state at the same time, DMPC 1 is designated as the coordinator, and when the calculation is complete, the other DMPC controllers transfer the state variable to DMPC Controller 1, DMPC1 outputs the control variable to the other controllers, and the AGVs perform the action.

Algorithm 1 presents the serial and parallel pseudocode for the ADMM-based distributed motion planning algorithms. Both the serial and parallel iteration modes offer distinct advantages. In the

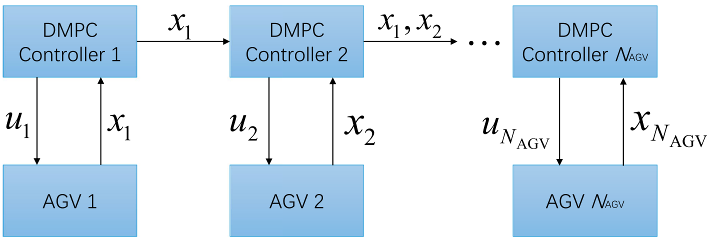


Figure 2. Coordination scheme of controllers using S-ADMM.

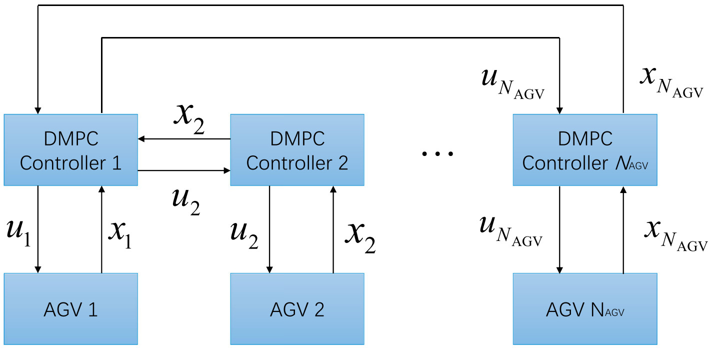


Figure 3. Coordination scheme of controllers using P-ADMM.

Colour online, B/W in print

Q13

serial mode, the sub-problems (25) are sequentially solved based on the information available in the current iteration. In the parallel mode, all of the sub-problems (28) are solved concurrently, utilizing only the information obtained from the previous iteration.

Algorithm 1 ADMM-based distributed MPC (serial and parallel)

```
Require: Task assignment of AGVs
while $t \in \{0, 1, \ldots, H\}$ do
    Initialize Lagrangian multipliers $\lambda_1 = 0$ and $\lambda_2 = 0$ and parameters $\epsilon_0 = 10$.
    for iteration $p = 1 : p_{\max}$ do
        for AGV $k = 1 : N_{\text{AGV}}$ do
            Each AGV determines $\hat{u}_k$ and $\mathbf{x}$ by solving the local sub-problem (25),(26) in serial or
            (28),(29) in parallel
            Update Lagrangian multipliers $\lambda_1(p,t)$ and $\lambda_2(p,t)$ by (27) in serial or (30) in parallel
            Update primal residual $r(p,t)$ and dual residual $s(p,t)$ by (23), (24)
            Update penalty coefficient $\epsilon(p,t)$ by (22)
        end for
        // Stopping condition
        if $r(p,t) \le r_{\lim}$ or $s(p,t) \le s_{\lim}$ or $p = p_{\max}$, then break
        $p = p + 1$
    end for
    // Update the state of each AGV and move to next step
    $t = t + 1$
end while
```

A comparison of these two modes reveals that the serial mode updates more information during a single iteration than the parallel mode, potentially leading to a better solution after a single iteration. However, the parallel mode requires less time for a single iteration compared to the serial mode. Notably, P-ADMM takes less time when the sub-problems are solved in a multi-computer distributed computing environment. In terms of computation time, the maximum computation time in the parallel mode is regarded as the computation time, while the sum of the computation time in the serial mode is considered as the computation time.

Furthermore, during the implementation of these two modes, it is important to note that the serial mode does not require full communication coverage, whereas the parallel mode necessitates full communication coverage for all the AGVs.

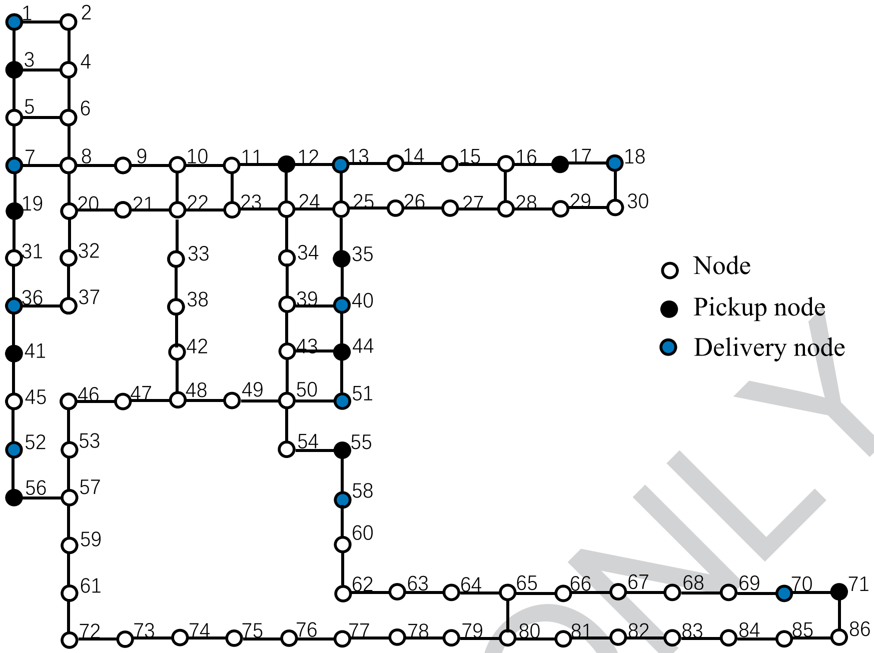


Figure 4. Roadmap layout of Benchmark 2 as suggested by Nishi *et al.* (2020).

4. Results and discussion

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 advantages of the S-ADMM and P-ADMM methods are assessed in terms of productivity and computation metrics.

4.1. Settings

To evaluate the performance of the proposed planning strategy, a benchmark related to the AGV roadmap is considered. The benchmark contains an irregular graph with unbalanced connections, which is derived from a real-world transport company, as shown in Figure 3 and in Nishi *et al.* (2020). In Figure 3, it is noted that the roadmap of Benchmark 1 is not regular because the topology is not squared. The roadmap layouts can be found in related applications for AGVs and robots (Adamo *et al.* 2018; Nishi *et al.* 2020; Yu and LaValle 2016).

To evaluate the performance of the proposed planning strategy, two benchmarks related to two types of AGV roadmaps are considered. Benchmark 1 corresponds to a squared roadmap graph, which contains $m \times m$ nodes and $2m \times (m-1)$ links (Yu and LaValle 2016), as introduced in Figure 1. Benchmark 2 contains an irregular graph with unbalanced connections, which is derived from a real-world transport company, as shown in Figure 4 and in Nishi *et al.* (2020).

The following four methods are evaluated: centralized MPC (C-MPC), serial ADMM-based distributed MPC (S-ADMM), parallel ADMM-based distributed MPC (P-ADMM), prioritized MPC (P-MPC) and dynamic prioritized planning (DMP) (Guney and Raptis 2021).

The mathematical modelling 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.

Three commonly used key performance indicators (KPIs) are considered as follows.

- The sum of completion times of all the AGVs, which is defined as cost 1.
- The completion time of the last AGV (makespan), which is defined as cost 2.
- 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, the computational results of the proposed S-ADMM and P-ADMM methods are evaluated, in comparison to the C-MPC, P-MPC and DPP methods. These experimental results are reported in Tables 2 and 3 for the industrial scenarios.

For each scenario, 20 experiments were tested for comparison in general. In each experiment, the starting point and the ending point of a particular task were given randomly in advance.

4.2. Experiential results

Tables 2 and 3 show the performance of the five tested methods concerning the industrial scenarios for Benchmarks 1 and 2. Note that the values of costs 1 and 2 were computed when all the tasks had been completed. The C-MPC, S-ADMM and P-AMDD methods outperformed the P-MPC and DPP methods in terms of productivity objectives (costs 1 and 2). Cost 1 refers to the total completion time, while cost 2 corresponds to the makespan. The C-MPC, S-ADMM and P-AMDD methods all use a predictive motion model for all AGVs, so collisions can be predicted in advance, improving transport productivity over P-MPC and DPP. The two distributed methods (S-ADMM and P-ADMM) yielded the same productivity objective values as the centralized method (C-MPC). Although using ADMM decomposition results in a small difference between the value of the augmented Lagrangian function and the value of the original objective, the values of costs 1 and 2 were collected from the determined paths, which rely on pure integer variables. Therefore, costs 1 and 2 of these methods are shown equally in these two tables.

The computation times of these five methods are also reported in Tables 2 and 3. The proposed P-ADMM method achieved the shortest computation time, while C-MPC obtained the longest computation time. The computation time of the proposed P-ADMM method was close to that of DPP. Meanwhile, owing to the efficient ADMM-based decomposition techniques, the computation time of these two distributed methods was significantly less than that of C-MPC.

The efficiency of the serial and parallel ADMM decompositions will now be examined. The convergence behaviours of the difference between the optimal costs by S-ADMM and P-ADMM are compared in Figure 5. These two methods had the same number of iterations. The convergence speed of S-ADMM was faster than that of P-ADMM. However, the computation time of P-ADMM was shorter than that of S-ADMM (as shown in Tables 2 and 3), indicating that the parallel scheme is more computationally effective than the serial scheme. If the sub-problems are solved in a multi-computer distributed computing environment, P-ADMM takes less time. Both of these distributed schemes are effective in terms of computation accuracy, achieving the same values for costs 1 and 2 as the centralized method (C-MPC), as shown in Tables 2 and 3. The features of these two ADMM decompositions are summarized in Table 4.

Table 3 shows the effectiveness of the proposed ADMM-based decomposition technique, since S-ADMM and P-ADMM achieve the same values of costs 1 and 2 as C-MPC. Note that the values of costs 1 and 2 are computed when all the tasks are completed. Since the paths of C-MPC, S-ADMM and P-ADMM are the same, the values of costs 1 and 2 of S-ADMM and P-ADMM are equal to those of C-MPC. However, some small errors exist owing to the decomposition technique introduced by ADMM. Figure 5 provides the converging trajectories.

Table 2. Comparison of the numerical results with respect to Benchmark 1. (Units: seconds.)

Settings		C-MPC			S-ADMM			P-ADMM			P-MPC			DPP		
m	N_{AGV}	cost 1	cost 2	CT	cost 1	cost 2	CT	cost 1	cost 2	CT	cost 1	cost 2	CT	cost 1	cost 2	CT
5	4	25.0	7.8	0.11	25.0	7.8	0.07	25.0	7.8	0.05	25.5	7.9	0.11	26.5	8.3	0.15
5	5	31.1	8.1	0.13	31.1	8.1	0.08	31.1	8.1	0.06	31.3	8.2	0.12	33.0	8.8	0.15
6	5	30.6	8.5	0.24	30.6	8.5	0.17	30.6	8.5	0.12	31.1	8.6	0.27	32.0	9.0	0.15
6	6	35.3	8.7	0.27	35.3	8.7	0.20	35.3	8.7	0.15	35.6	8.8	0.29	36.9	9.2	0.16
7	6	35.9	9.4	0.35	35.9	9.4	0.22	35.9	9.4	0.17	36.2	9.4	0.35	38.0	9.8	0.16
7	7	39.6	9.9	0.46	39.6	9.9	0.26	39.6	9.9	0.19	40.2	9.9	0.36	41.3	10.4	0.28
8	7	39.9	10.2	0.55	39.9	10.2	0.30	39.9	10.2	0.22	40.4	10.2	0.45	42.2	10.8	0.30
8	8	46.0	10.4	0.77	46.0	10.4	0.34	46.0	10.4	0.27	46.7	10.4	0.67	47.9	11.0	0.35
9	8	47.5	11.1	0.84	47.5	11.1	0.42	47.5	11.1	0.29	47.9	11.2	0.71	49.7	11.7	0.36
9	9	52.3	11.7	1.08	52.3	11.7	0.44	52.3	11.7	0.34	52.6	11.7	0.73	49.8	12.2	0.36
10	9	52.5	12.5	2.11	52.5	12.5	0.58	52.5	12.5	0.38	53.1	12.6	1.24	54.6	13.0	0.39
10	10	57.3	12.8	3.35	57.3	12.8	0.71	57.3	12.8	0.46	57.7	13.0	1.76	59.0	13.3	0.45
15	12	172.0	18.7	47.79	172.0	18.7	5.46	172.0	18.7	2.36	174.7	18.8	8.91	176.9	19.4	4.92
15	16	212.9	19.2	55.24	212.9	19.2	7.23	212.9	19.2	2.41	216.5	19.4	13.68	219.8	20.0	7.41
15	20	253.4	18.7	64.77	253.4	18.7	9.44	253.4	18.7	3.27	258.2	18.9	16.81	261.5	19.5	10.26
15	25	316.4	18.3	71.26	316.4	18.3	13.26	316.4	18.3	5.13	320.7	18.7	16.93	332.6	19.2	12.18
15	30	397.2	18.3	90.43	378.3	18.3	17.49	397.2	18.3	6.26	375.6	18.7	22.01	384.2	19.2	14.72
20	12	180.7	24.0	57.81	180.7	24.0	7.41	180.7	24.0	3.13	185.6	24.2	17.91	190.0	24.9	8.01
20	16	242.4	26.5	65.64	242.4	26.5	9.21	242.4	26.5	3.82	245.0	26.6	24.86	253.8	27.5	9.53
20	20	291.6	24.5	75.61	291.6	24.5	14.11	291.6	24.5	5.51	295.3	24.7	28.36	305.0	25.4	12.96
20	25	365.9	25.1	94.36	365.9	25.1	18.24	365.9	25.1	6.81	372.9	25.2	32.88	378.9	25.6	14.14
20	30	437.2	25.4	113.37	437.2	25.4	21.33	437.2	25.4	8.31	445.1	25.5	35.41	453.1	26.0	16.92
Average		152.85	15.45	33.93	152.85	15.45	5.77	152.85	15.45	**2.26**	154.00	15.57	10.22	157.58	16.14	5.19

Table 3. Comparison of numerical results with respect to Benchmark 2. (Un**Q20** seconds.)

Setting,	C-MPC			S-ADMM			P-ADMM			P-MPC			DPP		
N_{AGV}	cost 1	cost 2	CT	cost 1	cost 2	CT	cost 1	cost 2	CT	cost 1	cost 2	CT	cost 1	cost 2	CT
3	32.8	13.8	2.93	32.8	13.8	0.47	32.8	13.8	0.32	33.1	13.9	0.63	34.0	14.2	1.06
4	46.7	14.0	3.42	46.7	14.0	0.69	46.7	14.0	0.42	47.2	14.0	0.87	49.3	14.4	1.52
5	57.8	14.3	4.27	57.8	14.3	1.07	57.8	14.3	0.74	58.5	14.3	1.49	60.0	14.8	2.06
6	69.4	14.6	6.27	69.4	14.6	1.68	69.4	14.6	0.92	70.2	14.6	2.07	72.2	15.0	2.34
7	81.3	15.6	11.59	81.3	15.6	1.77	81.3	15.6	1.08	82.6	15.6	2.21	83.8	16.1	2.88
8	98.0	15.3	14.78	98.0	15.3	2.14	98.0	15.3	1.12	99.7	15.4	2.87	101.0	15.8	3.42
9	110.3	15.8	21.64	110.3	15.8	2.97	110.3	15.8	1.27	112.4	15.8	3.96	113.0	16.2	3.70
10	127.7	15.9	29.88	127.7	15.9	3.66	127.7	15.9	1.73	129.2	15.9	4.22	130.6	16.3	3.89
15	185.3	16.0	49.76	185.3	16.0	7.43	185.3	16.0	2.91	188.1	16.1	9.21	189.9	16.7	5.95
20	253.4	16.0	57.66	253.4	16.0	9.07	253.4	16.0	4.07	257.7	16.2	12.89	265.3	16.5	8.71
25	313.8	16.2	68.43	313.8	16.2	12.39	313.8	16.2	5.19	318.2	16.3	16.39	327.4	16.8	11.02
30	375.2	15.9	84.36	375.2	15.9	16.97	375.2	15.9	7.32	381.3	16.1	20.01	390.8	16.4	13.06
Average	145.98	**15.3**	29.58	145.98	**15.3**	5.03	145.98	**15.3**	**2.26**	148.2	15.4	7.46	151.4	15.8	4.97

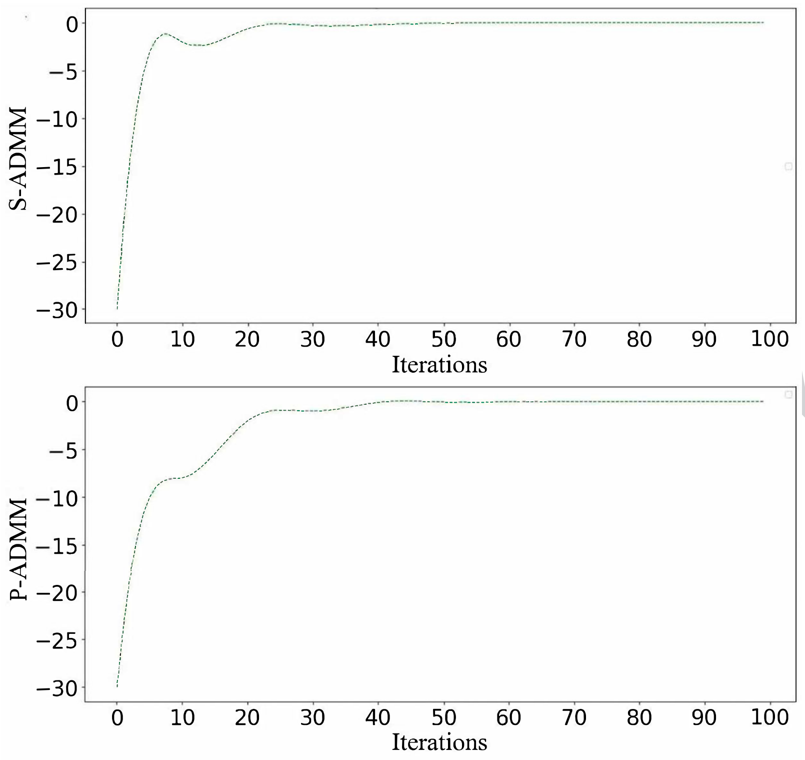


Figure 5. Convergence behaviour of the cost differences from the optimal cost by S-ADMM and P-ADMM.

Table 4. Summarized features of S-ADMM and P-ADMM.

Scheme	Type	Convergence speed	Calculation time
S-ADMM	Serial	Fast	Long
P-ADMM	Parallel	Slow	Short

Figures 6 and 7 provide the overall performance of the five methods with respect to costs 1 and 2 both on Benchmarks 1 and 2. Figures 6 and 7 show that the average values of costs 1 and 2 of C-MPC, S-ADMM and P-ADMM are both lower than those of P-MPC and DPP. C-MPC, S-ADMM and P-ADMM share the same standard deviation because S-ADMM and P-ADMM obtained the same values for costs 1 and 2 as C-MPC for all the experiments in each scenario.

Figure 8 shows the same trajectories determined by the C-MPC, S-ADMM and P-ADMM methods for a scenario of five AGVs on Benchmark 2. The trajectories of these three methods were identical. S-ADMM and P-ADMM methods produced not only the same objective values but also the same trajectories. According to Figure 8, all the motions were collision-free because no AGV stayed at the same node at the same time.

Note that there exist multiple optimal solutions when solving the same problem with identical configurations on roadmap, origins and destinations. The solutions of S-ADMM and P-ADMM can be identical to one of the multiple solutions obtained by the centralized C-MPC when these two ADMM-based methods terminate under the stopping criterion. This situation is shown in Figure 9, Q21 which has the same objective value as Figure 9 but the trajectories of AGV 1 and AGV 5 are different. The solutions of S-ADMM and P-ADMM may be different from the solution obtained by the centralized C-MPC because multiple optimal solutions exist.

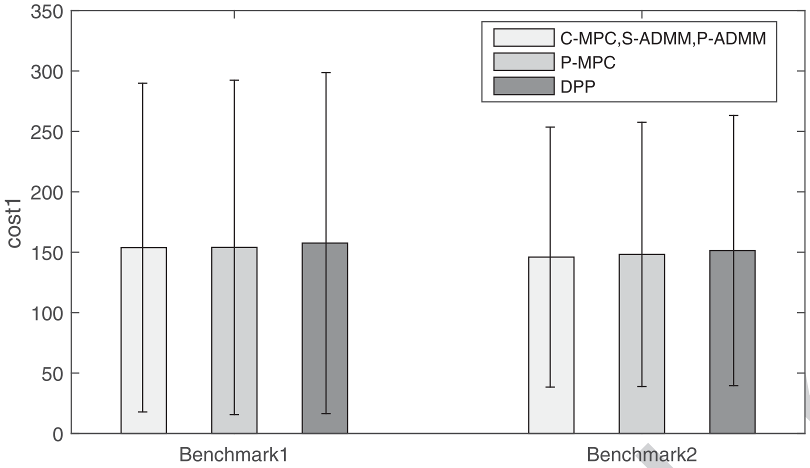


Figure 6. Overall performance for cost 1 of the five methods on Benchmarks 1 and 2.

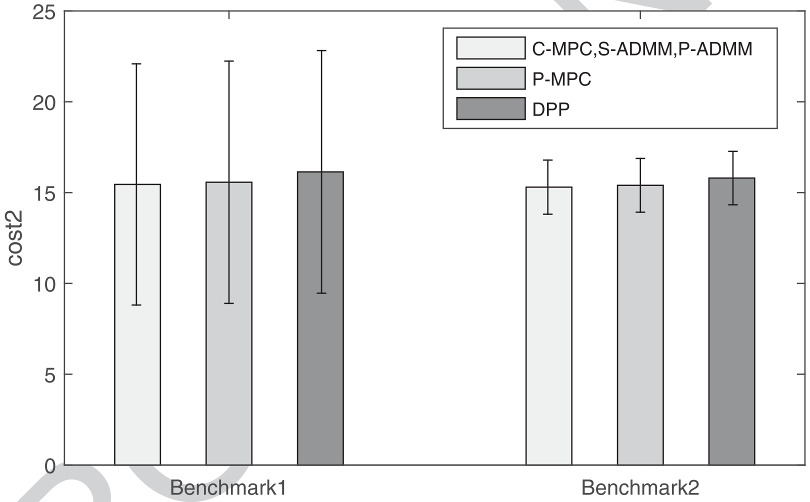


Figure 7. Overall performance for cost 2 of the five methods on Benchmarks 1 and 2.

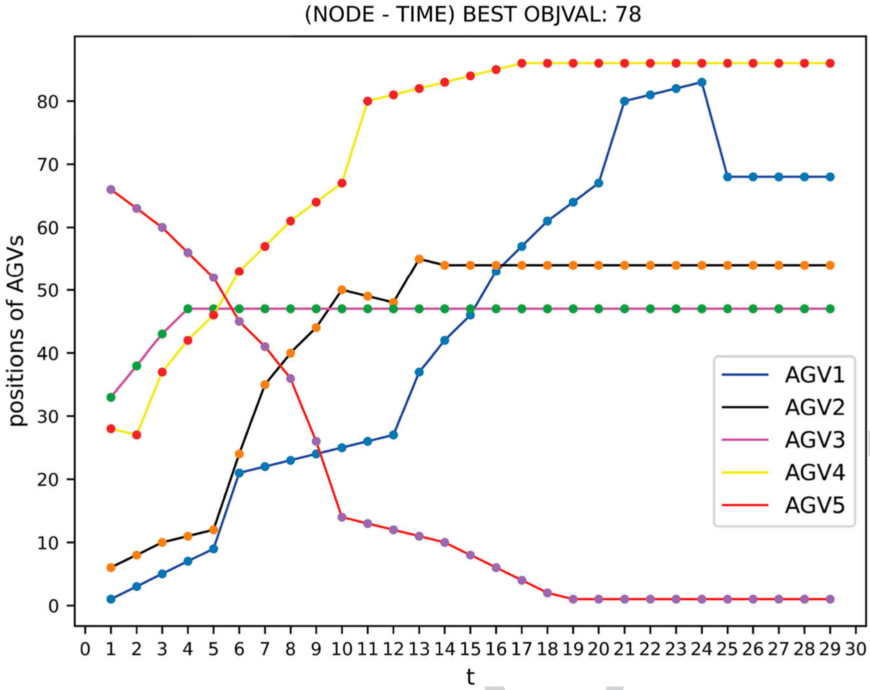
Colour online, B/W in print

5. Conclusions and future work

In the manufacturing and logistics environments, the decision-making process for AGV motion planning is expected to be flexible, robust and scalable. As a result, a novel methodological contribution to dynamic motion planning of automated guided vehicles in bidirectional guide-path layouts is proposed. A dynamical predictive representation of the material transport process is used to describe it mathematically. Additionally, serial and parallel distributed planning methods based on the alternating direction method of multipliers (S-ADMM and P-ADMM) are proposed to improve the flexibility, robustness and scalability of the AGV fleet.

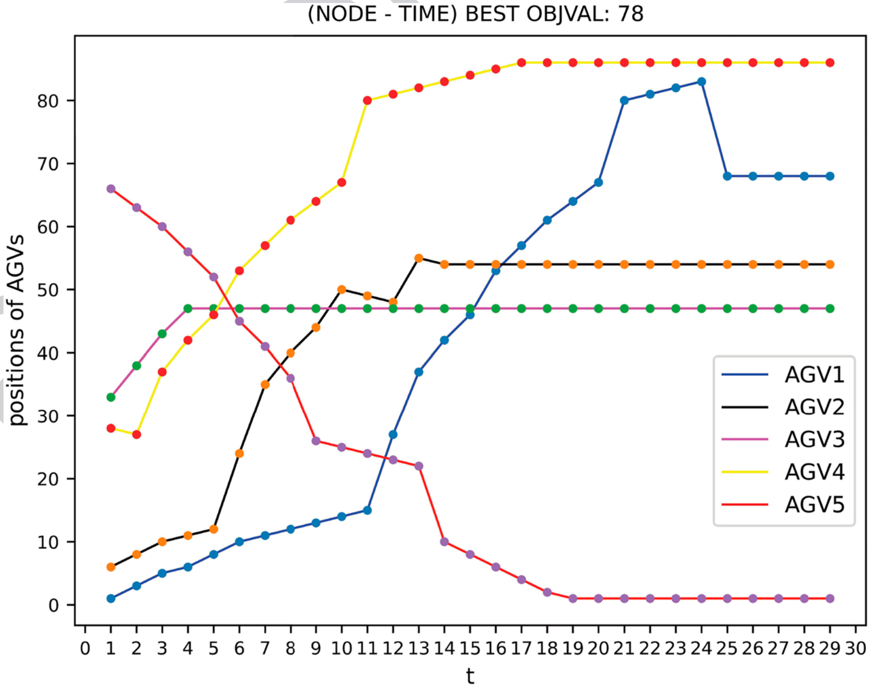
The advantages of the proposed distributed methodologies are demonstrated through case studies derived from industrial scenarios in comparison to the centralized MPC method and two cutting-edge dynamic planning methods. With a low computational burden, both S-ADMM and P-ADMM achieve high productivity metrics (*i.e.* the sum of completion times and makespan). S-ADMM and P-ADMM produce the same productivity objective values as the centralized MPC method, and both

Colour online, B/W in print



Q15 **Figure 8.** Trajectories by the C-MPC, S-ADMM and P-ADMM methods for a scenario of five AGVs on Benchmark 2.

Colour online, B/W in print



Q16 **Figure 9.** Trajectories by another S-ADMM that changes the sequence of updating the Lagrangian multipliers.

distributed methods produce the same trajectories. Meanwhile, the computation time of these two distributed methods is significantly less than that of C-MPC owing to the efficient ADMM-based decomposition techniques. P-ADMM takes less time to compute than S-ADMM, indicating that the parallel coordination scheme is more efficient than the serial coordination scheme. P-ADMM takes less time when the sub-problems are solved in a multi-computer distributed computing environment.

In conclusion, the proposed algorithms employ ADMM-based decomposition techniques to improve computational efficiency compared to the centralized method using the predictive dynamical model. Moreover, the proposed algorithms provide a dynamic planning scheme that can be more resilient and agile under complex and dynamic operation circumstances. These algorithms are well-suited for decentralized operational architecture, which is the current trend of operating AGVs into the future for manufacturing industry.

However, the proposed algorithms are not fast enough to support real-time decisions for very-large-scale cases. There is a need for the further development of more efficient methods to accelerate computation. Future research will explore the integration of model-based methods and data-driven methods in a potential planner. Furthermore, the optimal scheduling of horizontal transport vehicles in the automatic transformation of traditional container terminals will also be considered.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research is supported in part by the National Natural Science Foundation of China [Grants 62173311 and 61703372]; in part by the College Youth Backbone Teacher Project of Henan Province [Grant 2021GGJS001]; and in part by the Henan Scientific and Technological Research Project [Grant 222102220123].

Q1
Q22

Data availability statement

The data that support the findings of this study are available from the corresponding author, J. Xin, upon reasonable request.

References

Adamo, Tommaso, Tolga Bektaş, Gianpaolo Ghiani, Emanuela Guerriero, and Emanuele Manni. 2018. “Path and Speed Optimization for Conflict-Free Pickup and Delivery Under Time Windows.” *Transportation Science* 52 (4): 739–755. https://doi.org/10.1287/trsc.2017.0816.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011. “Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers.” *Foundations & Trends in Machine Learning* 3 (1): 1–122. http://dx.doi.org/10.1561/2200000016.

Camacho, Eduardo F., Daniel R. Ramírez, Daniel Limón, D. Muñoz De La Peña, and Teodoro Alamo. 2010. “Model Predictive Control Techniques for Hybrid Systems.” *Annual Reviews in Control* 34 (1): 21–31. https://doi.org/10.1016/j.arcontrol.2010.02.002.

Cataldo, Andrea, and Riccardo Scattolini. 2016. “Dynamic Pallet Routing in a Manufacturing Transport Line with Model Predictive Control.” *IEEE Transactions on Control Systems Technology* 24 (5): 1812–1819. https://doi.org/10.1109/TCST.2015.2507062.

Chen, Linying, Yamin Huang, Huarong Zheng, Hans Hopman, and Rudy Negenborn. 2020. “Cooperative Multi-Vessel Systems in Urban Waterway Networks.” *IEEE Transactions on Intelligent Transportation Systems* 21 (8): 3294–3307. https://doi.org/10.1109/TITS.6979.

Cohen, G., and D. L. Zhu. 1984. “Decomposition and Coordination Methods in Large-Scale Optimization Problems: The Nondifferentiable Case and the Use of Augmented Lagrangians.” *Advances in Large Scale Systems* 1:203–266. https://cir.nii.ac.jp/crid/1570291225344062464?lang=en.

Fanti, Maria Pia, Agostino M. Mangini, Giovanni Pedroncelli, and Walter Ukovich. 2018. “A Decentralized Control Strategy for the Coordination of AGV Systems.” *Control Engineering Practice* 70:86–97. https://doi.org/10.1016/j.conengprac.2017.10.001.

Fragapane, Giuseppe, Rene De Koster, Fabio Sgarbossa, and Jan Ola Strandhagen. 2021. “Planning and Control of Autonomous Mobile Robots for Intralogistics: Literature Review and Research Agenda.” *European Journal of Operational Research* 294 (2): 405–426. https://doi.org/10.1016/j.ejor.2021.01.019.

Fransen, K. J. C., J. A. W. M. Van Eekelen, A. Pogromsky, Marko A. A. Boon, and Ivo J. B. F. Adan. 2020. "A Dynamic Path Planning Approach for Dense, Large, Grid-Based Automated Guided Vehicle Systems." *Computers & Operations Research* 123:105046. https://doi.org/10.1016/j.cor.2020.105046.

Guney, Mehmet Ali, and Ioannis A. Raptis. 2021. "Dynamic Prioritized Motion Coordination of Multi-AGV Systems." *Robotics and Autonomous Systems* 139:103534. https://doi.org/10.1016/j.robot.2020.103534.

He, B. S., Hai Yang, and S. L. Wang. 2000. "Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities." *Journal of Optimization Theory and Applications*106 (2): 337–356. https://doi.org/10.1023/A:1004603514434.

Kim, Dae B., and Hark Hwang. 2001. "A Dispatching Algorithm for Multiple-Load AGVs Using a Fuzzy Decision-Making Method in a Job Shop Environment." *Engineering Optimization* 33 (5): 523–547. https://doi.org/10.1080/03052150108940932.

Kumar, Sunil, and Afzal Sikander. 2023. "A Modified Probabilistic Roadmap Algorithm for Efficient Mobile Robot Path Planning." *Engineering Optimization* 55 (9): 1616–1634. https://doi.org/10.1080/0305215X.2022.2104840.

Lee, D.-H., J. X. Cao, and Q. X. Shi. 2009. "Synchronization of Yard Truck Scheduling and Storage Allocation in Container Terminals." *Engineering Optimization* 41 (7): 659–672. https://doi.org/10.1080/03052150902752041.

Luo, Jiliang, Yaxin Wan, Weimin Wu, and Zhiwu Li. 2020. "Optimal Petri-Net Controller for Avoiding Collisions in a Class of Automated Guided Vehicle Systems." *IEEE Transactions on Intelligent Transportation Systems* 21 (11): 4526–4537. https://doi.org/10.1109/TITS.6979.

Miyamoto, Toshiyuki, and Kensuke Inoue. 2016. "Local and Random Searches for Dispatch and Conflict-Free Routing Problem of Capacitated AGV Systems." *Computers & Industrial Engineering*91:1–9. https://doi.org/10.1016/j.cie.2015.10.017.

Murakami, Keisuke. 2020. "Time–space Network Model and MILP Formulation of the Conflict-Free Routing Problem of a Capacitated AGV System." *Computers & Industrial Engineering* 141:106270. https://doi.org/10.1016/j.cie.2020.106270.

Negenborn, Rudy R., and Jose Maria Maestre. 2014. "Distributed Model Predictive Control: An Overview and Roadmap of Future Research Opportunities." *IEEE Control Systems Magazine* 34 (4): 87–97. https://doi.org/10.1109/MCS.2014.2320397.

Nishi, Tatsushi, Shuhei Akiyama, Toshimitsu Higashi, and Kenji Kumagai. 2020. "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* 17 (2): 966–980. https://doi.org/10.1109/TASE.8856.

Nishi, T., M. Ando, and M. Konishi. 2005. "Distributed Route Planning for Multiple Mobile Robots Using An Augmented Lagrangian Decomposition and Coordination Technique." *IEEE Transactions on Robotics* 21 (6): 1191–1200. https://doi.org/10.1109/TRO.2005.853489.

Nishi, Tatsushi, and Yuki Tanaka. 2012. "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* 42 (5): 1230–1243. https://doi.org/10.1109/TSMCA.2012.2183353.

Rinaldi, Marco, Erika Picarelli, Andrea D'Ariano, and Francesco Viti. 2020. "Mixed-Fleet Single-Terminal Bus Scheduling Problem: Modelling, Solution Scheme and Potential Applications." *Omega*96:102070. https://doi.org/10.1016/j.omega.2019.05.006.

Ryck, M. D., M. Versteyhe, and F. Debrouwere. 2020. "Automated Guided Vehicle Systems, State-of-the-Art Control Algorithms and Techniques." *Journal of Manufacturing Systems* 54 (1): 152–173. https://doi.org/10.1016/j.jmsy.2019.12.002.

Saidi-Mehrabad, Mohammad, Saeed Dehnavi-Arani, Farshid Evazabadian, and Vahid Mahmoodian. 2015. "An Ant Colony Algorithm (ACA) for Solving the New Integrated Model of Job Shop Scheduling and Conflict-Free Routing of AGVs." *Computers & Industrial Engineering* 86:2–13. https://doi.org/10.1016/j.cie.2015.01.003.

Sirmatel, Isik Ilber, and Nikolas Geroliminis. 2018. "Mixed Logical Dynamical Modeling and Hybrid Model Predictive Control of Public Transport Operations." *Transportation Research Part B: Methodological* 114:325–345. https://doi.org/10.1016/j.trb.2018.06.009.

Tobajas, Javier, Felix Garcia-Torres, Pedro Roncero-Sánchez, Javier Vázquez, Ladjel Bellatreche, and Emilio Nieto. 2022. "Resilience-Oriented Schedule Of Microgrids with Hybrid Energy Storage System Using Model Predictive Control." *Applied Energy* 306:118092. https://doi.org/10.1016/j.apenergy.2021.118092.

Xin, Jianbin, Chuang Meng, Andrea D'Ariano, Dongshu Wang, and Rudy R. Negenborn. 2022. "Mixed-Integer Nonlinear Programming for Energy-Efficient Container Handling: Formulation and Customized Genetic Algorithm." *IEEE Transactions on Intelligent Transportation Systems* 23 (8): 10542–10555. https://doi.org/10.1109/TITS.2021.3094815.

Xin, Jianbin, Liuqian Wei, Andrea D'Ariano, Fangfang Zhang, and Rudy Negenborn. 2023. "Flexible Time–Space Network Formulation and Hybrid Metaheuristic for Conflict-Free and Energy-Efficient Path Planning of Automated Guided Vehicles." *Journal of Cleaner Production* 398:136472. https://doi.org/10.1016/j.jclepro.2023.136472.

Yang, Qifan, Yindong Lian, and Wei Xie. 2020. "Hierarchical Planning for Multiple AGVs in Warehouse Based on Global Vision." *Simulation Modelling Practice and Theory* 104:102124. https://doi.org/10.1016/j.simpat.2020.102124.

Yi, Guohong, Zhili Feng, Tiancan Mei, Pushan Li, Wang Jin, and Siyuan Chen. 2019. "Multi-AGVs Path Planning Based on Improved Ant Colony Algorithm." *The Journal of Supercomputing* 75 (9): 5898–5913. https://doi.org/10.1007/s11227-019-02884-9.

Yu, Jingjin, and Steven M. LaValle. 2016. "Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics." *IEEE Transactions on Robotics* 32 (5): 1163–1177. https://doi.org/10.1109/TRO.2016.2593448.

Zheng, Huarong, Rudy R. Negenborn, and Gabriël Lodewijks. 2016. "Predictive Path Following with Arrival Time Awareness for Waterborne AGVs." *Transportation Research Part C: Emerging Technologies* 70:214–237. https://doi.org/10.1016/j.trc.2015.11.004.

Zheng, Huarong, Wen Xu, Dongfang Ma, and Fengzhong Qu. 2022. "Dynamic Rolling Horizon Scheduling of Waterborne AGVs for Inter Terminal Transportation: Mathematical Modeling and Heuristic Solution." *IEEE Transactions on Intelligent Transportation Systems* 23 (4): 3853–3865. https://doi.org/10.1109/TITS.2021.3102998.

Zhong, Meisu, Yongsheng Yang, Yasser Dessouky, and Octavian Postolache. 2020. "Multi-AGV Scheduling for Conflict-Free Path Planning in Automated Container Terminals." *Computers & Industrial Engineering* 142:106371. https://doi.org/10.1016/j.cie.2020.106371.

Zou, Wen-Qiang, Quan-Ke Pan, Tao Meng, Liang Gao, and Yu-Long Wang. 2020. "An Effective Discrete Artificial Bee Colony Algorithm for Multi-AGVs Dispatching Problem in a Matrix Manufacturing Workshop." *Expert Systems with Applications* 161:113675. https://doi.org/10.1016/j.eswa.2020.113675.