Received: 28 November 2019 | Revised: 14 May 2020 | Accepted: 14 June 2020

DOI: 10.1002/oca.2654



SPECIAL ISSUE ARTICLE

WILEY

# Receding horizon path planning of automated guided vehicles using a time-space network model

**Jianbin Xin**[1] | **Liuqian Wei**[1] | **Dongshu Wang**[1] | **Hua Xuan**[2]

[1]School of Electrical Engineering, Zhengzhou University, Zhengzhou, China

[2]School of Management Engineering, Zhengzhou University, Zhengzhou, China

**Correspondence**
Dongshu Wang, School of Electrical Engineering, Zhengzhou University, Science Road 100, Zhengzhou 450001, China.
Email: wangdongshu@zzu.edu.cn

**Funding information**
National Natural Science Foundation of China, Grant/Award Numbers: 61703372, 61973278; Outstanding Foreign Scientist Support Project in Henan Province, Grant/Award Number: GZS201908; Key Technologies R&D Programme of Henan Province, Grant/Award Number: 192102210256; Key Scientific Research Project of Henan Higher Education, Grant/Award Number: 18A413012

**Summary**

Time-space network (TSN) models have been widely used for collision-free path planning of automated guided vehicles. However, existing TSN models are planned globally. The global method suffers from computational complexity and uncertainties cannot be dealt with in the dynamic environment. To address these limitations, this article proposes a new methodology to decompose the global planning problem into smaller local planning problems, which are planned in a receding horizon way. For the local problem, new decision variables and constraints are incorporated into the TSN framework. Extensive simulation experiments are carried out to show the potential of the proposed methodology. Simulation results show that the proposed method obtains competitive performances and computational times are considerably reduced, compared with the global method.

**KEYWORDS**

automated guided vehicle, path planning, receding horizon planning, time-space network model

## 1 | INTRODUCTION

Automated guided vehicles (AGVs) are autonomous vehicles used for transporting materials or completing specific tasks in the industrial environment.[1,2] Typical application examples can be found in warehouse,[3] manufacturing,[4] container terminals,[5] or other applications.

For AGVs, path planning plays an important role in these applications. Path planning aims to find an optimal feasible path subject to certain environment constraints and answers the fundamental question ɛ*where to go*ɛ for intelligent vehicles or robots.[6,7] The related objectives include the shortest path, the shortest time, or the least cost.

Path planning of AGVs can be categorized into global planning and local planning. In global planning methods, the entire paths are planned completely from their origins to destinations and the environment information is assumed to be known globally to the planner. A well-known classical method is *Dijkstra*'*s* algorithm to obtain the shortest path of a single vehicle in the road network. Recently, time-space network (TSN) models[8-10] have been proposed for planning collision-free paths of multiple AGVs. For these TSN formulations, Lagrangian relaxation and hybrid heuristic methods were developed to address the formulated mixed integer programming problems. When multiple origins and destinations are considered for every single AGV, the planning problem becomes a routing problem[11,12] (eg, multiple vehicle pickup-and-delivery problem or vehicle routing problem). In addition to the TSN method, optimal control[13] has also been proposed.

  

Compared with global methods, local methods focus on planning regional paths, and environment information is partially known to the planner. One advantage of local planning is that it owns reasonable computational performance. However, the optimality cannot be always guaranteed, and one representative example is $A^*$ algorithm.[6] The emerging local methods, such as reinforcement learning[14] and developmental network,[15] even do not require a mathematical model when multiple vehicles are involved.

The TSN models[8,10] are regarded as a powerful tool for describing the path planning and routing of multiple AGVs in the mesh roadmap. The TSN models include location and time constraints, which enable to avoid collisions between vehicles. In the recent TSN model,[16] flexible transport times are considered and this makes possible to reduce energy consumption. However, the existing TSN models use the global planning framework, which suffers from computational complexity when the system scale increases. Meanwhile, uncertainties cannot be dealt with in a dynamic environment.

To address these drawbacks, this article proposes a new methodology for planning collision-free paths of AGVs using a TSN model. The proposed methodology decomposes the global planning problem into small local planning problems using a receding horizon way, to reduce the computational complexity and handle uncertainties throughout the whole planning process. To adapt to the local planning, new decision variables and constraints are introduced. The new methodology is then tested and analyzed by conducting extensive simulation case studies. Simulation results show that the proposed RHP strategy obtains competitive objectives and significantly reduces the computation times, in comparison with the global planning method.

The remainder of this article is organized as follows: Section 2 describes the research problem and revisit a time-space network model for path planning of AGVs globally. In Section 3, the RHP strategy is proposed and a local planning problem is formulated. Section 4 conducts simulation experiments of the proposed planning methodology and further analyzes its performances. Finally, Section 5 concludes this article and gives directions for future research.

## 2 | MODELING

This section presents the research problem of the path planning of AGVs using a TSN model and revisits the TSN model used to determine collision-free paths.

### 2.1 | Problem description

In the industrial environment, multiple materials need to be transported by multiple AGVs in a mesh roadmap. During the process of transporting materials, collision-free vehicle paths are required to be planned from their origins to destinations. In this article, we focus on the operational path planning and this corresponds to a multiple vehicle pickup-and-delivery (MVPD) problem, which is essential for detailing the execution of delivery tasks. In this problem, the task assignment is assumed to be given in advance and the assigned tasks can be determined by solving a vehicle routing pickup and delivery problem.[17]

Here we consider a uniform roadmap layout in which the distance between any two adjacent nodes is equal, as illustrated in Figure 1. This layout can be frequently found in a warehouse and a container terminal.[18] A connected path

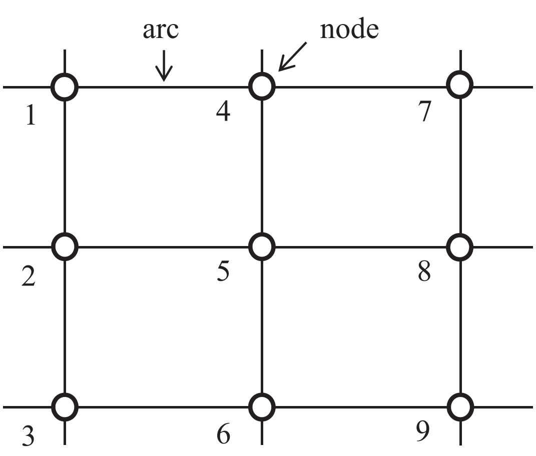


**FIGURE 1** Example of the roadmap for planning AGVs

between any two adjacent nodes is regarded as an arc. The loading and unloading locations are located at the nodes of the network. Each task for every AGV is defined to transport the material from its origin to destination.

For this MVPD problem, important assumptions adopted[8] are given as follows:

- The roadmap is strongly connected and there exist at least two paths between every two nodes. The length of AGV is approximately simplified to zero and regarded as a virtual point.
- The number of vehicles is less than the number of nodes.
- The origin point and destination point of each task are given in advance, and there are no repetitive nodes in the tasks assigned.
- The turning time is included in the transportation time.
- At any time, every arc is allowed to be occupied by only one AGV.
- No more than one AGV can occupy the same node at any time.
- Once a task is completed, the corresponding AGV does not occupy its destination node anymore.

## 2.2 | Time-space network model

In this part, a TSN model is revisited for mathematically describing the MVPD problem. This model employs the TSN framework using cumulative flow variables. First, the parameters and decision variables are defined. Afterward, the detailed mathematical model is presented.

### 2.2.1 | Time-space graph

The considered conflict-free MVPD can be modeled as a multicommodity network flow problem using a TSN formulation to avoid the conflicts among all the AGVs.[9] In the considered TSN model, a directed graph $G = (V, E)$ is considered. $V$ is the collection of nodes and $E = \{(i, j)|i \in V, j \in V\ i{\neq}j\}$ is the collection of arcs. A particular node $i$ corresponds to an intersection where the vehicle can change the direction and a particular arc $(i, j)$ maps the path from node $i$ to $j$ for two adjacent nodes. In the graph, the sequence of nodes in the arc $(i, j)$ provides a directed path and the corresponding weight is referred to as the length of the arc. For the mesh roadmap, all the arc connections represent a particular switching system.[19,20]

TSN models have been popularly in transportation and logistics to model the routing problem when time and space constraints are both considered.[10,21] For route planning of AGVs, Nishi's TSN model[8] and Miyamoto's TSN model[10] typically consider fixed transport times and therefore energy consumption cannot be further reduced. Therefore, in this article, a TSN model including flexible transport times,[16] which has the potential to reduce energy consumption, is revisited.

For the considered MVPD problem, a planning horizon $T \times\Delta t$ is considered, and the whole horizon is discretized equally into a set of time slots denoted by $\{\Delta t, 2\Delta t, \ldots, T \times\Delta t\}$. Here, $\Delta t$ is a time slot and $T$ is the total number of time slots. As a result, the time-space network can decompose the overall routing process of multiple robots into several time slots. At each time instant $t \in\{1, 2, .., T\}$, a particular AGV can visit a particular node or stay in the arc between two adjacent nodes. During each time interval, the detailed path can be considered for avoiding collisions among these AGVs.

### 2.2.2 | Mathematical model

Before detailing the TSN model to be considered, the associated index variables and decision variables are given in Tables 1 and 2. It is noted that, in Table 2 the cumulative flow variables $a(i, j, k, t)$ and $d(i, j, k, t)$ are used to describe the arrival and departure events of AGV $k$ for arc $(i, j)$. By defining the variable $y(i, j, k, t) = a(i, j, k, t) - d(i, j, k, t)$, the spatial occupation for AGV $k$ can be described explicitly and further used for spatial capacity constraints. These cumulative flow variables are useful for the multicommodity network flow formulation to model the conflict-free MVPD. The specific meaning of cumulative flow variables $a(i, j, k, t)$ and $d(i, j, k, t)$ is shown in Figure 2. As can be seen from Figure 2, AGV $k$ arrives at arc $(i, j)$ at $t = 6$ and leaves at $t = 8$.

**TABLE 1** Index variables and input parameters

| Symbol | Description |
|---|---|
| $i, j, n$ | Node indices |
| $k$ | Vehicle index |
| $t$ | Time index |
| $N$ | Set of nodes |
| $E$ | Set of arcs |
| $V$ | Set of AGVs |
| $T$ | Planning horizon |
| $H$ | Set of discrete time instance ($H$={0,1,2,...,$T$}) |
| $w(i, j)$ | Minimal traveling time of arc $(i, j)$ |
| $O_k$ | Origin node of AGV$k$ |
| $S_k$ | Destination node of AGV$k$ |
| $E^o(i)$ | Set of arcs starting from node $i$ |
| $E^s(i)$ | Set of arcs ending from node $i$ |
| $N_i$ | Set of adjacent nodes to node $i$ |

**TABLE 2** Decision variables

| Variables | Type | Description |
|---|---|---|
| $x(i, j, k)$ | Binary | If arc $(i, j)$ is selected by AGV$k$, $x_{(i, j, k)} = 1$, otherwise 0 |
| $a(i, j, k, t)$ | Binary | If AGV$k$ has reached arc $(i, j)$ by time $t$, $a(i, j, k, t) = 1$, otherwise 0 |
| $d(i, j, k, t)$ | Binary | If AGV$k$ has departed from arc $(i, j)$ by time $t$, $d_{(i, j, k, t)} = 1$, otherwise 0 |
| $y(i, j, k, t)$ | Binary | If AGV $k$ occupies arc $(i, j)$ at time $t$, $y(i, j, k, t) = 1$, otherwise 0 |
| TT$(i, j, k)$ | Integer | Transport time of AGV$k$ at arc $(i, j)$ |

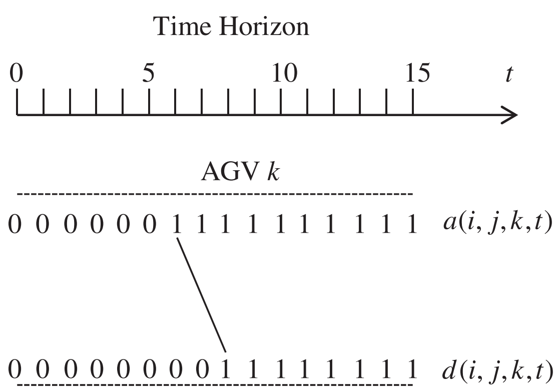


**FIGURE 2** Illustrative example of the defined variables $a(i, j, k, t)$ and $d(i, j, k, t)$

The objective is to minimize the total arrival times of all AGVs in the network, and $J$ is defined to be this objective. The arrival time of the destination for AGV $k$ can be described by $\sum_t t \times \sum_{i:(i,s_k)\in E^s(s_k)} [d(i,j,k,t) - d(i,j,k,t-1)]$ as there is only one arc to be selected from the set $E^s(s_k)$, in which $s_k$ is the ending node of vehicle $k$. Therefore, the objective function $J$ is given in detail as follows:

$$J = \sum_k \left( \sum_t t \times \sum_{i:(i,s_k)\in E^s(s_k)} [d_{(i,j,k,t)} - d_{(i,j,k,t-1)}] \right). \tag{1}$$

In addition to the objective, the constraints of the TSN model are presented in three groups:

I. The space constraints:

$$\sum_{j \in N_{o_k}} x(o_k, j, k) = 1, \quad \forall k \in V, \tag{2}$$

$$\sum_{i \in N_{s_k}} x(i, s_k, k) = 1, \quad \forall k \in V, \tag{3}$$

$$\sum_{i \in N_j} x(i, j, k) = \sum_{n \in N_j} x(j, n, k), \quad \forall k \in V, j \in N - O_k - S_k. \tag{4}$$

Constraints (2), (3), and (4) specify the starting node, ending node, and intermediate nodes for each AGV in the network, respectively.

II. The time constraints:

$$\mathrm{TT}(i, j, k) = \sum_t \{t \times [d(i, j, k, t) - d(i, j, k, t-1)]\} - \sum_t \{t \times [a(i, j, k, t) - a(i, j, k, t-1)]\}, \quad \forall k \in V, (i, j) \in E, \tag{5}$$

$$\mathrm{TT}(i, j, k) \geq w(i, j) \times x(i, j, k), \quad \forall k \in V, (i, j) \in E, \tag{6}$$

$$\sum_k \sum_{i \in N_j} \left[d(i, j, k, t) - d(i, j, k, t-1)\right] \leq 1, \quad \forall t \in H, (i, j) \in E, \tag{7}$$

$$d(i, j, k, t-1) \leq d(i, j, k, t), \quad \forall k \in V, t \in H, (i, j) \in E, \tag{8}$$

$$a(i, j, k, t-1) \leq a(i, j, k, t), \quad \forall k \in V, t \in H, (i, j) \in E. \tag{9}$$

Constraints (5) and (6) correspond to transport time constraints. Equation (5) describes the transport time of each AGV in each arc using variables $a(i, j, k, t)$ and $d(i, j, k, t)$. Inequality (6) limits the minimal transport time in each arc for each AGV. Equation (7) indicates that each arc can be occupied by one AGV at most. Equations (8) and (9) give time connectivity for the arrival and departure of AGV $k$ at arc $(i, j)$.

III. The time and space mapping constraints:

$$d(i, j, k, t + w(i, j)) \leq a(i, j, k, t), \quad \forall k \in V, t \in H, (i, j) \in E, \tag{10}$$

$$\sum_{i \in N_j} d(i, j, k, t) = \sum_{n \in N_j} a(j, n, k, t), \quad \forall k \in V, t \in H, j \in N - O_k - S_k, \tag{11}$$

$$a(i, j, k, T) = x(i, j, k), \quad \forall k \in V, (i, j) \in E, \tag{12}$$

$$y(i, j, k, t) = a(i, j, k, t) - d(i, j, k, t), \quad \forall k \in V, t \in H, (i, j) \in E, \tag{13}$$

$$\sum_k y(i, j, k, t) + \sum_k y(j, i, k, t) \leq 1, \quad \forall k \in V, t \in H, (i, j) \in E. \tag{14}$$

Equation (10) is the constraint between the cumulative flow variable $a(i, j, k, t)$ and $d(i, j, k, t)$. Equation (11) is the transformation constraint between two adjacent arcs. Equation (12) maps time and space for AGV $k$ at its destination.

Equation (13) gives the relationship among $a(i,j,k,t)$, $d(i,j,k,t)$, and $y(i,j,k,t)$. Inequality (14) enables that at any time each arc is occupied by at most one AGV.

The global MVPD problem using the considered TSN model leads to a mixed integer programming (MIP). When the system scale grows, a large number of variables and constraints are required, causing considerable computational times. Meanwhile, uncertainties cannot be handled using global planning. For addressing these problems, an effective planning strategy will be proposed in the next section.

## 3 | RECEDING HORIZON PLANNING

In this section, a receding horizon planning (RHP) methodology is proposed. This methodology can decompose the entire planning problem introduced in Section 2 into smaller planning problems for improving computational efficiency and dealing with uncertainties throughout the whole planning.

### 3.1 | Principle

For path planning of AGVs, existing TSN models[10,16] plan the routes in a global way. The global planning problem corresponds to a multicommodity network flow problem using integer variables. The related optimization problem is NP-complete,[22] indicating that the associated planning problem is computationally intractable for a large-scale road network where the dimensions of the problem grow considerably. Therefore, it is necessary to propose the RHP principle for real-time planning.

In principle, RHP utilizes a dynamical model to obtain planned routes by minimizing an objective over a finite receding horizon.[23] A similar methodology, which is referred to as receding horizon control, has been successfully used in the industrial process and transportation.[24,25] In RHP, the dynamical model is used to predict the future routes of the vehicle based on the current state and the proposed future routes. These route actions are calculated by minimizing the cost function, considering the constraints on the states and decision variables. RHP gives an on-line planning framework for routing AGVs with interacting variables, complex dynamics, and constraints.

For the receding horizon planning, both the time-driven and event-driven methods are possibly used for real-time scheduling. The replanning of time-driven methods is triggered by εtime,ε and the time-driven method has a periodic mechanism. The event-driven planning method is triggered by εeventε to capture nonperiodic changes in the system. In this article, we consider a time-driven planning method as the whole planning horizon of the time-space network model is equally discretized into several intervals, and therefore it is natural and easy to implement the proposed time-driven planning method. The structure of the proposed time-driven RHP is given in Figure 3.

Figure 4 illustrates the principle of the proposed receding horizon planning methodology. Here, the global planning problem is decomposed into smaller local planning problems, which are solved individually after a time slot. In the local planning problem, the global objective is approximated and the size of the optimization problem size is smaller than the global one. Furthermore, the local information can be detected by each AGV within its local planning horizon based on its current state, and this makes local planning possible to be implemented for each AGV.

Figure 5 illustrates the planned paths of two AGVs using the proposed RHP strategy from their origins ($O_1$ and $O_2$) to their destinations ($S_1$ and $S_2$) at $t=0$ and $t=1$. For the two instances, the solid lines represent the planned paths within

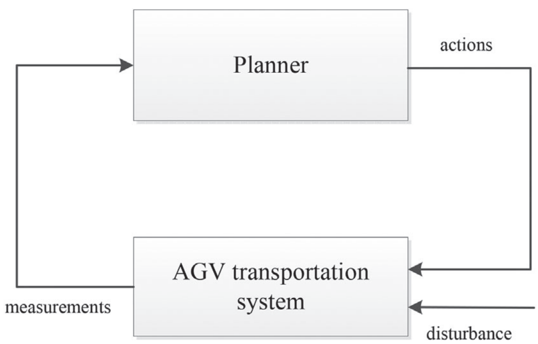


**FIGURE 3** Structure of the receding horizon planning

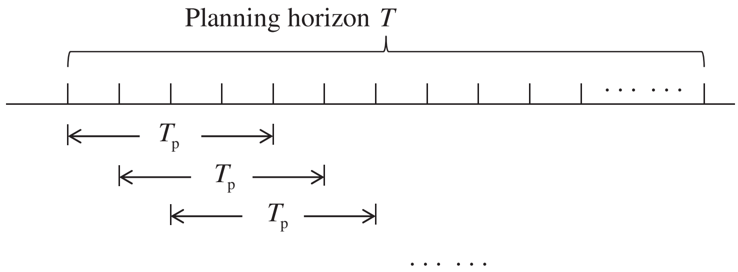


**FIGURE 4** Illustration of the receding horizon planning strategy (local planning horizon $T_p$)

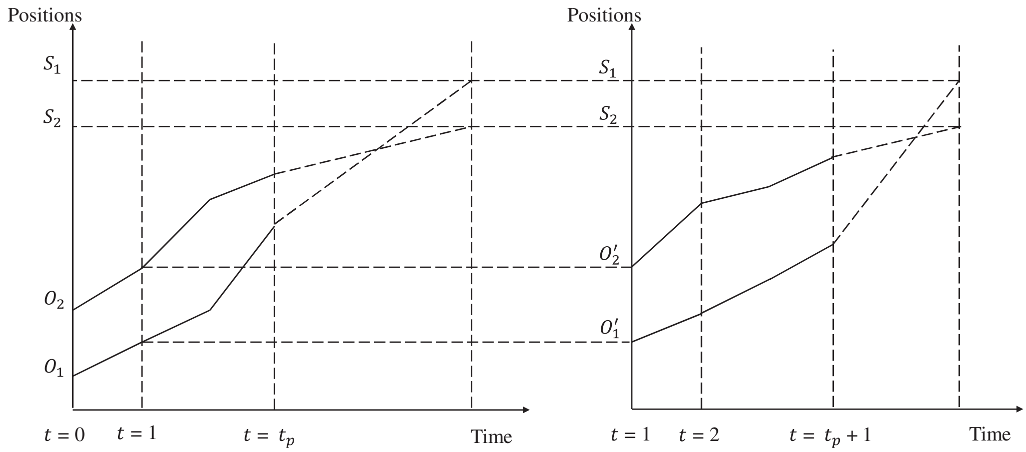


**FIGURE 5** Illustrative planned paths using the RHP approach (2 AGVs)

the local planning horizon $T_p$; the dashed lines denote the ignored paths out of the horizon $T_p$, only considering the approximated objectives. Based on the RHP strategy, the paths of these two AGVs can be planned at each time instant $t$.

In the following part, the local planning problem is formulated and later used for receding horizon planning.

## 3.2 | Local planning

In this part, a local planning problem using the TSN model is formulated in detail. For this local problem, a new model needs to be developed by introducing additional input parameters, decision variables, and constraints. These new parameters and decision variables are given in Tables 3 and 4.

Using the defined parameters and decision variables, the objective of the local planning problem, defined as $\widetilde{J}$, is given as follows:

$$\widetilde{J} = \sum_{k} \left( \sum_{j \in V_1} C(j,k) \times D(j,k) + T_p \right). \tag{15}$$

**TABLE 3** New parameters for local planning

| Symbol | Description |
|---|---|
| $R$ | A large integer number |
| $T_p$ | Local planning horizon |
| $V_1$ | Set of local endpoints within the local planning horizon $T_p$ |
| $E_1$ | Set of arcs where the AGVs can travel within the local planning horizon $T_p$ |
| $D(j,k)$ | Approximated distance between point $j$ and the destination for AGV $k$ |

**TABLE 4** New decision variables for local planning

| Variables | type | Description |
|---|---|---|
| $C(j,k)$ | Binary | If AGV$k$ choose the point $j$ as the ending point, $C(j,k) = 1$; otherwise, $C(j,k) = 0$ |

Besides the objective $\widetilde{J}$, to adapt to the local planning, several constraints need to be included or modified. Here we add the following constraints:

$$\sum_{j} C(j,k) = 1, \quad \forall k \in V, j \in V_1 - O_k, \tag{16}$$

$$\sum_{k} C(j,k) \leq 1, \quad \forall j \in V_1, \tag{17}$$

$$-R(1 - C(j,k)) \leq \sum_{t} t \times \sum_{(i,j)\in E_1} [d(i,j,k,t) - d(i,j,k,t-1)] - T_{\mathrm{p}}, \quad \forall k \in V, j \in V_1, \tag{18}$$

$$\sum_{t} t \times \sum_{(i,j)\in E_1} [d(i,j,k,t) - d(i,j,k,t-1)] - T_{\mathrm{p}} \leq R(1 - C(j,k)), \quad \forall k \in V, j \in V_1. \tag{19}$$

Constraints (16) and (17) guarantee that each AGV can only choose one local ending point and every local ending point can be selected by at most one AGV. Inequalities (18) and (19) ensure that each AGV $k$ reaches its local endpoint at $t = T_{\mathrm{p}}$.

Furthermore, constraints (3), (4), and (11) need to be modified respectively as follows:

$$-R(1 - C(j,k)) \leq \sum_{i\in N_j} x(i,j,k) - 1, \quad \forall k \in V, j \in V_1 - O_k, \tag{20}$$

$$\sum_{i\in N_j} x(i,j,k) - 1 \leq R(1 - C(j,k)), \quad \forall k \in V, j \in V_1 - O_k, \tag{21}$$

$$\sum_{i\in N_j} x(i,j,k) - \sum_{n\in N_j} x(j,n,k) = C(j,k), \quad \forall k \in V, j \in V_1 - O_k, \tag{22}$$

$$\sum_{t}\sum_{i\in N_j} d(i,j,k,t) - \sum_{t}\sum_{n\in N_j} a(j,n,k,t) = C(j,k), \quad \forall k \in V, j \in V_1 - O_k - S_k. \tag{23}$$

Constraints (20)-(22) guarantee the path consistency and specify that for each AGV there is only one path to reach the local endpoint. Constraints (20)-(21) correspond to constraint (3) to restrict the endpoint selected by each vehicle, while constraint (22) modifies (4) to provide the path continuity of intermediate points when the local endpoint is selected. Equation (23) ensures the time continuity for each AGV from its starting point to ending point in the local roadmap.

In summary, the local planning problem can be formulated as the following:

$$\begin{aligned} &\min \widetilde{J} \\ &\text{subject to (2), (5)-(12), (14), (16)-(22).} \end{aligned} \tag{24}$$

This optimization problem is a smaller scale MIP, which can be solved efficiently by commercial solvers.

## 4 | CASES STUDIES

In this section, a series of simulation experiments are carried out to analyze the performance of the proposed RHP strategy, in comparison with the global planning method. In the first part, a benchmark of a material transport system and experiment settings are introduced. Then, the performances of these two methods are compared and examples of case studies are demonstrated.

## 4.1 | Experiment settings

### 4.1.1 | Benchmark

To quantify the performance of the proposed RHP method, a benchmark system is considered. The benchmark system consists of a uniform roadmap layout used in the warehouse application suggested by the work.[18] The roadmap contains $m \times m$ nodes and $2m \times (m-1)$ arcs. In the considered benchmark, physical features of AGVs that own dynamical behavior with parameters of velocity and weight[26] can be included. Figure 6 illustrates an example case of the benchmark system.

In this article, extensive case studies will be tested to evaluate the proposed RHP approach. The detailed settings are presented in Table 5. Each case study tests 10 experiments. In each experiment, the pickup point and the delivery point of a particular task are given randomly in advance. The distance $S(i, j)$ is set to 10 m for an indoor inventory system.[18] The unit time $\Delta T$ is considered to 10 seconds.

The mathematical model is implemented in Python, due to its high productivity and user-friendly interface with the efficient commercial solver Gurobi. The benchmark system is mathematically built by Python 3.7 on Windows 10 (64-bit) and the optimization problems are solved by Gurobi 9.0. The computer hardware is Intel Core i7-9700 (3.0Hz) with 16 GB of memory.

### 4.1.2 | Choice of local planning horizon

In the proposed RHP approach, the planning horizon $T_p$ is a crucial parameter that needs to be decided properly. On one hand, a large $T_p$ leads to a computationally expensive optimization problem. On the other hand, a smaller $T_p$ shortens

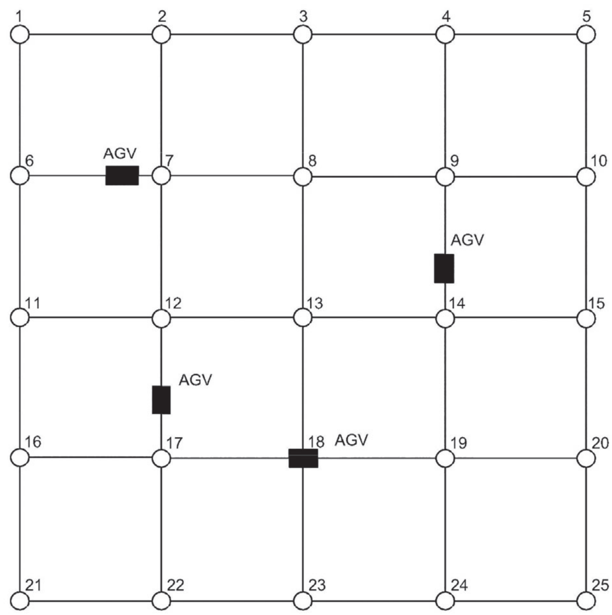


**FIGURE 6** Illustrative example of the benchmark system including a $5 \times 5$ map with four AGVs

**TABLE 5** Cases studied

| Cases | $m$ | $N$ |
|---|---|---|
| 1 | 5 | 4, 5, 6 |
| 2 | 6 | 5, 6, 7 |
| 3 | 7 | 6, 7, 8 |
| 4 | 8 | 7, 8, 9 |
| 5 | 9 | 8, 9, 10 |
| 6 | 10 | 9, 10, 11 |

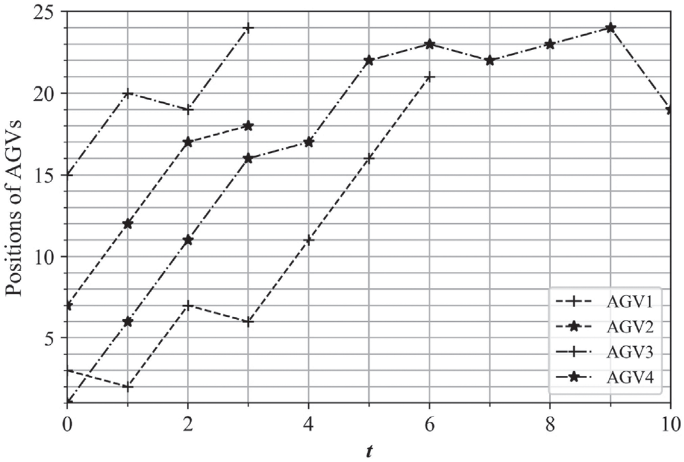


**FIGURE 7** Planned paths when $T_p$=2 ($m = 5, N = 4$)

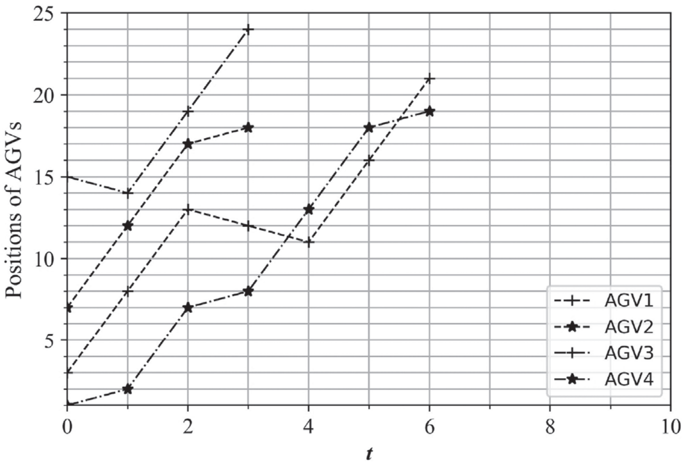


**FIGURE 8** Planned paths when $T_p$=3 ($m = 5, N = 4$)

the computation time. In that case, the planning of each AGV is overlocalized and the collision probability will increase correspondingly. Therefore, the planned paths may far beyond the optimal one.

In the case studies, since each arc $(i, j)$ owns at least one time slot, we consider $T_p > 1$. As we search for a relatively small $T_p$, we conduct several small scale simulations regarding $T_p = 2$ and $T_p = 3$. The simulation results show that the setting of $T_p = 2$ will produce considerable delays while all the vehicles almost complete their tasks on time when $T_p$ is selected to 3. This comparison is illustrated in Figures 7 and 8. As a result, in the following parts, we choose $T_p = 3$ for the following simulation case studies.

## 4.2 | Results and discussion

This part first presents and discusses performances of the proposed RHP approach, compared with the global planning method. Then, examples of routes planned are illustrated. The last part demonstrates how the proposed RHP approach deals with uncertainties.

### 4.2.1 | Performance

Tables 6 and 7 record the operation performances and the computational performances of the proposed RHP method and the global planning method.

Table 6 presents the averaged sum of all vehicle completion times of all considered cases using these two methods. It can be seen from Table 6 that, for the same roadmap layout, the total completion times increase when the number of

**TABLE 6** Sum of all vehicle completion times on average

| Setting | Global planning (unit time) | The proposed RHP (unit time) | Gap (%) |
|---|---|---|---|
| $m=5, N=4$ | 18.4 | 18.4 | 0 |
| $m=5, N=5$ | 24.3 | 24.9 | 2.5 |
| $m=5, N=6$ | 32.0 | 33.5 | 4.7 |
| $m=6, N=5$ | 24.1 | 24.8 | 2.9 |
| $m=6, N=6$ | 29.0 | 30.9 | 6.6 |
| $m=6, N=7$ | 40.7 | 42.8 | 5.2 |
| $m=7, N=6$ | 41.8 | 42.6 | 1.9 |
| $m=7, N=7$ | 45.5 | 47.5 | 4.4 |
| $m=7, N=8$ | 49.7 | 51.8 | 4.2 |
| $m=8, N=7$ | 51.2 | 52.5 | 2.5 |
| $m=8, N=8$ | 54.7 | 56.5 | 3.3 |
| $m=8, N=9$ | 64.5 | 66.9 | 3.7 |
| $m=9, N=8$ | 61.5 | 62.9 | 2.3 |
| $m=9, N=9$ | 76.8 | 79.6 | 3.6 |
| $m=9, N=10$ | 79.2 | 82.6 | 4.3 |
| $m=10, N=9$ | 74.4 | 79.6 | 6.9 |
| $m=10, N=10$ | 86.0 | 90.7 | 5.5 |
| $m=10, N=11$ | 92.4 | 97.5 | 5.5 |

**TABLE 7** Compared averaged computational times (unit: seconds)

| Setting | Global planning | The proposed RHP |
|---|---|---|
| $m=5, N=4$ | 0.65 | 0.90 |
| $m=5, N=5$ | 0.79 | 1.51 |
| $m=5, N=6$ | 1.02 | 2.16 |
| $m=6, N=5$ | 4.25 | 0.44 |
| $m=6, N=6$ | 6.09 | 0.51 |
| $m=6, N=7$ | 7.80 | 0.88 |
| $m=7, N=6$ | 18.94 | 1.03 |
| $m=7, N=7$ | 29.64 | 1.14 |
| $m=7, N=8$ | 39.39 | 1.49 |
| $m=8, N=7$ | 35.45 | 1.64 |
| $m=8, N=8$ | 58.68 | 1.89 |
| $m=8, N=9$ | 109.53 | 2.36 |
| $m=9, N=8$ | 218.66 | 1.86 |
| $m=9, N=9$ | 309.16 | 3.41 |
| $m=9, N=10$ | 438.85 | 8.95 |
| $m=10, N=9$ | 616.45 | 9.23 |
| $m=10, N=10$ | 1362.26 | 18.08 |
| $m=10, N=11$ | 1963.02 | 20.79 |

AGVs increases. The global planning method obtains the optimal paths of the MVPD problem. Compared with the global method, the proposed RHP approach achieves competitive operation times, and just causes up to 6.9% accumulated delays for all the vehicles when completing their tasks. In general, the gap becomes larger when more vehicles are involved in the same roadmap layout.

Table 7 lists the averaged computation time of these two methods. It can be easily found that, in general, the computation times of both the global method and the RHP method grow as the road network and the number of AGVs increases. For the small-scale problems, the difference regarding computation times of these two methods is not obvious. However, for large-scale problems, the global planning method owns considerable computational times, while the proposed RHP approach significantly improves computational efficiency using much shorter computational times. The low computational burden fits well with the real-time planning requirement in practice.

### 4.2.2 | Example of planned paths

This part shows the routes planned using the proposed RHP approach from one illustrative experiment of the case $m = 5$ and $N = 4$, comparing with the result from the global planning method. The detailed time-space routes are given in Figures 9 and 10.

Figure 9 shows the routes planned of the AGVs using the global TSN model. This figure shows that all the four AGVs move from different pickup points to different delivery points, and collisions between these AGVs are avoided when their routes are determined. The visited node sequences of these AGVs are given as follows: AGV1: 5-4-3-2-7-6-11-16-21; AGV2: 7-8-13-14-19-20-25; AGV3: 15-20-25-24; and AGV4: 2-7-8-13-14-19. It can be seen from this figure that no collision takes place during the transportation process.

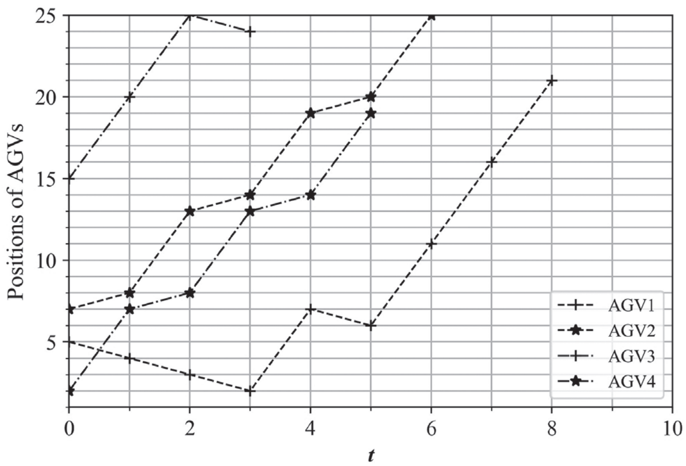


**FIGURE 9** Planned paths using the global planning method

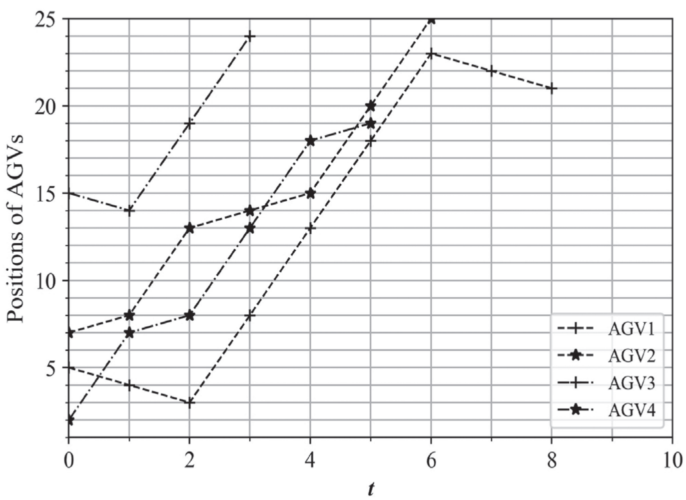


**FIGURE 10** Planned paths using the proposed RHP method

Figure 10 reports the paths planned using the proposed RHP approach with the same configuration in Figure 9. Figure 10 shows that the completion time of each AGV is the same as the global planning method. It can be seen from this figure that all the AGVs have the same pickup and delivery points as considered in Figure 9. And there is no collision happening at each node and each arc. However, for each AGV, the planned path is different from the result obtained using the global planning method. From this example, the effectiveness of the proposed RHP approach is demonstrated.

### 4.2.3 | Scenario for uncertainties

The previous parts show the computational efficiency of the proposed RHP strategy. Besides the computational benefit, another advantage of the receding horizon planning is to deal with uncertainties. In the manufacturing and warehouse environment, different types of uncertainties such as the change of orders could take place. Here we focus on dealing with the uncertainties resulting from the operation process when all the AGVs receive transportation requests. The capability is demonstrated when a particular vehicle has a breakdown during the transportation process. A case of configuration ($m = 5$ and $N = 4$) is considered and it is assumed that AGV 2 breaks down at $t = 2$.

Figures 11 and 12 present the paths of AGVs using the global planning method and the proposed RHP approach. Figure 11 shows that AGV2 and AGV3 collide at $t = 3$ as the paths are planned at time $t = 0$ and the planning is not updated at each time instant. The RHP approach uses the receding horizon optimization and the paths are replanned at each time instant. As a result, the collision is resolved and this is illustrated in Figure 12.

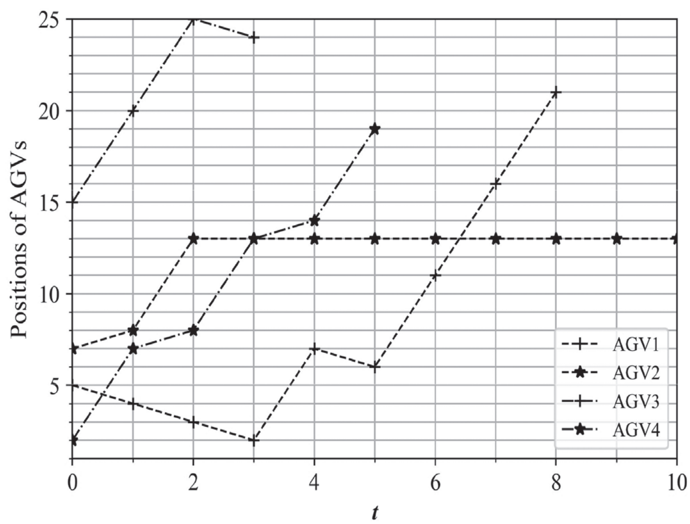


**FIGURE 11** Planned paths using the global method in the case of breakdown

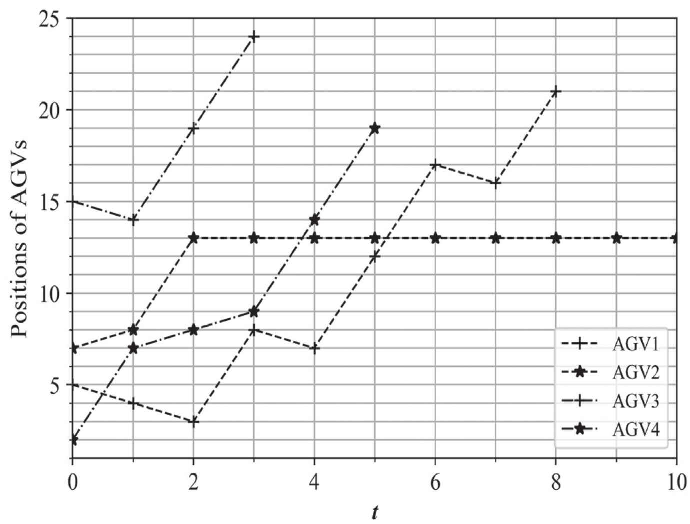


**FIGURE 12** Planned paths using the proposed RHP approach in the case of breakdown

In addition to the breakdown of the AGV, other types of uncertainties such as the operational delay and the change of the destination also can be incorporated in the framework of the proposed receding horizon planning.

## 5 | CONCLUSIONS AND FUTURE RESEARCH

This article proposes a receding horizon planning approach for path planning of AGVs using a time-space network (TSN) model. The proposed approach decomposes the existing approach that is global planning into smaller planning problems to improve computational efficiency and deal with uncertainties. The newly defined decision variables and constraints can be incorporated in the TSN framework for local planning. The simulation results conclude that the proposed RHP approach provides competitive operational performances using reasonable computational times.

Future research will consider incorporating detailed energy consumption into the objective function and an energy-efficient routing can be obtained by properly optimizing both the completion time and the energy consumption. Besides this, advanced control techniques should be developed to follow the planned route for the vehicle.[27,28]

### ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China under Grant 61703372 and U1804151, the Key Scientific Research Project of Henan Higher Education under Grant 18A413012, the Key Technologies R&D Programme of Henan Province under Grant 192102210256, and the Outstanding Foreign Scientist Support Project in Henan Province under Grant GZS201908.

### ORCID

*Jianbin Xin* https://orcid.org/0000-0002-1024-4135
*Dongshu Wang* https://orcid.org/0000-0003-3145-4956

### REFERENCES

1. Fazlollahtabar H, Saidimehrabad M. *Autonomous Guided Vehicles*. New York, NY: Springer; 2015.
2. Andreasson H, Bouguerra A, Cirillo M, et al. Autonomous transport vehicles: where we are and what is missing. *IEEE Robot Automat Mag*. 2015;22(1):64-75.
3. Yoshitake H, Kamoshida R, Nagashima Y. New automated guided vehicle system using real-time holonic scheduling for warehouse picking. *IEEE Robot Automat Lett*. 2019;4(2):1045-1052.
4. Mohammadi EK, Shirazi B. Toward high degree flexible routing in collision-free FMSs through automated guided vehicles' dynamic strategy: a simulation metamodel. *ISA Trans*. 2020;96:228–244.
5. Xin J, Negenborn R, Corman F, Lodewijks G. Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. *Transp Res Part C Emerg Technol*. 2015;60:377-396.
6. Choset HM, Hutchinson S, Lynch KM, et al. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA: MIT press; 2005.
7. Wang Y, Wang D, Zhu S. Cooperative moving path following for multiple fixed-wing unmanned aerial vehicles with speed constraints. *Automatica*. 2019;100:82-89.
8. Nishi T, Ando M, Konishi M. Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition and coordination technique. *IEEE Trans Robot*. 2005;21(6):1191-1200.
9. Corréa A, Langevin A, Rousseau L. Scheduling and routing of automated guided vehicles: a hybrid approach. *Comput Oper Res*. 2007;34(6):1688-1707.
10. Miyamoto T, Inoue K. Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Comput Ind Eng*. 2016;91:1-9.
11. Nishi T, Hiranaka Y, Grossmann IE. A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Comput Oper Res*. 2011;38(5):876-888.
12. Fazlollahtabar H, Hassanli S. Hybrid cost and time path planning for multiple autonomous guided vehicles. *Appl Intell*. 2018;48(2):482-498.
13. Demesure G, Defoort M, Bekrar A, Trentesaux D, Djemaï M. Decentralized motion planning and scheduling of AGVs in an FMS. *IEEE Trans Ind Inform*. 2018;14(4):1744-1752.
14. Long P, Fanl T, Liao X, Liu W, Zhang H, Pan J. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. Paper presented at: Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA); 2018:6252-6259; IEEE.
15. Wang D, Duan Y, Weng J. Motivated optimal developmental learning for sequential tasks without using rigid time-discounts. *IEEE Trans Neural Netw Learn Syst*. 2018;29(10):4917-4931.

16. Yin S, Xin J. Path planning of multiple AGVs using a time-space network model. Paper presented at: Proceedings of the 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC); 2019:73-78; IEEE.
17. Berbeglia G, Cordeau JF, Gribkovskaia I, Laporte G. Static pickup and delivery problems: a classification scheme and survey. *TOP*. 2007;15(1):1-31.
18. Adamo T, Bektaş T, Ghiani G, Guerriero E, Manni E. Path and speed optimization for conflict-free pickup and delivery under time windows. *Transp Sci*. 2018;52(4):739-755.
19. Yuan S, Zhang L, De Schutter B, Baldi S. A novel Lyapunov function for a non-weighted L2 gain of asynchronously switched linear systems. *Automatica*. 2018;87:310-317.
20. Yuan S, Zhang L, Baldi S. Adaptive stabilization of impulsive switched linear time-delay systems: a piecewise dynamic gain approach. *Automatica*. 2019;103:322-329.
21. Meng L, Zhou X. Simultaneous train rerouting and rescheduling on an N-track network: a model reformulation with network-based cumulative flow variables. *Transp Res B Methodol*. 2014;67:208-234.
22. Karp RM. On the computational complexity of combinatorial problems. *Networks*. 1975;5(1):45-68.
23. Pietrabissa A, Di Giorgio A, Oddi G, Chini G, Chiang ML, Poli C. Cooperative receding Horizon strategies for the multivehicle routing problem. *Opt Control Appl Methods*. 2018;39(1):248-262.
24. Zheng H, Negenborn RR, Lodewijks G. Fast ADMM for distributed model predictive control of cooperative waterborne AGVs. *IEEE Trans Control Syst Technol*. 2016;25(4):1406-1413.
25. Xin J, Negenborn RR, Lin X. Piecewise affine approximations for quality modeling and control of perishable foods. *Opt Control Appl Methods*. 2018;39(2):860-872.
26. Saidi-Mehrabad M, Dehnavi-Arani S, Evazabadian F, Mahmoodian V. An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Comput Ind Eng*. 2015;86:2-13.
27. Peng J, Dubay R. Adaptive fuzzy backstepping control for a class of uncertain nonlinear strict-feedback systems based on dynamic surface control approach. *Expert Syst Appl*. 2019;120:239-252.
28. Shi K, Liu C, Biggs JD, Sun Z, Yue X. Observer-based control for spacecraft electromagnetic docking. *Aerosp Sci Technol*. 2020;99:105759.

**How to cite this article:** Xin J, Wei L, Wang D, Xuan H. Receding horizon path planning of automated guided vehicles using a time-space network model. *Optim Control Appl Meth*. 2020;41:1889–1903. https://doi.org/10.1002/oca.2654