Dynamic rolling horizon scheduling of waterborne AGVs for inter terminal transport

Zheng, Huarong; Jin, Chen; Luo, Xiling; Negenborn, Rudy R.; Wang, Yuexuan

DOI
10.1109/ICCA51439.2020.9264557

Publication date
2020

Document Version
Accepted author manuscript

Published in
Proceedings of the IEEE 16th International Conference on Control and Automation, ICCA 2020

Citation (APA)
Zheng, H., Jin, C., Luo, X., Negenborn, R. R., & Wang, Y. (2020). Dynamic rolling horizon scheduling of waterborne AGVs for inter terminal transport. In *Proceedings of the IEEE 16th International Conference on Control and Automation, ICCA 2020* (pp. 761-766). IEEE. https://doi.org/10.1109/ICCA51439.2020.9264557

Dynamic rolling horizon scheduling of waterborne AGVs for inter terminal transport*

Huarong Zheng[1,*], Chen Jin[2], Xiling Luo[2], Rudy R. Negenborn[3], and Yuexuan Wang[4]

***Abstract*— The demand for transport between terminals within port areas, known as inter terminal transport (ITT), is increasing. This paper proposes a dynamic rolling horizon scheduling strategy for ITT using waterborne Autonomous Guided Vessels (waterborne AGVs). The strategy is dynamic in that it can handle the dynamically arriving ITT requests and adapt transport schedules accordingly in real time. Specifically, every certain period of time, we formulate and solve a pick-up and delivery problem considering the dynamic vessel states, waterway network topology, and ITT requests over a future time horizon. In the dynamic setting, waterborne AGVs are allowed to divert from the previously scheduled destination. Moreover, the distances between terminals are not calculated simply as the Euclidean metric but based on the complex port waterway network, which complicates the dynamic problem even more. Time windows of ITT requests, capacity constraints of waterborne AGVs and load/unload service times at terminals are also taken into account. A waterborne ITT transport network in the Port of Rotterdam is constructed. Simulation results demonstrate the effectiveness of the proposed dynamic scheduling strategy.**

I. Introduction

The container throughput will increase to more than 30 million Twenty-foot Equivalent Unit (TEU) per year by 2035 in the Port of Rotterdam [1]. This large throughput is handled partly by building automated container terminals where a 40% increase in productivity is foreseen due to automation [2]. The container movement between terminals via road, rail, or sea, i.e., inter terminal transport (ITT) [3], is currently handled mainly by manned multi-trailer systems. Waterborne Autonomous Guided Vessels (waterborne AGVs) [4] have been previously proposed for ITT over sea considering that the distances between some terminals are much shorter by water than by land. In [4], a fully autonomous ITT system using waterborne AGVs is achieved with transport requests known a prior. However, in practice, transport requests mostly arrive gradually. Dynamic ITT with waterborne AGVs still needs to be addressed.

Essentially, the scheduling of waterborne AGVs for ITT is a vehicle routing problem (VRP) with pick-up and delivery [5]. When all the ITT requests are received before the decision is made and when the decision does not change thereafter, a static VRP problem for waterborne AGVs is considered [4]. Along with the advances in information and communication technologies, it is possible to collect customer requests and system states in real time, which facilitates the development of dynamic VRPs [6]. Dynamic VRPs have been commonly applied in, e.g., the food or meal delivery business [7] [8], taxi dispatching [9], and also the deployment of emergency response vehicles [10]. Most dynamic VRPs require re-optimizations of the routes one way or another due to the concurrently updated inputs. According to [6], 11 criteria can be used to classify the existing literature on dynamic VRPs. As for when to perform the update of routes, it is customary to employ an event-based mechanism. This event can be the arrival of a new request [11] or the end of a predefined period of time [10]. Waiting [12] and buffering [13] strategies have been proposed by holding the request for a while before assigning it to a vehicle. On the contrary, [14] explores the possibility to further reduce the system cost by proposing an approximated non-myopic dynamic pricing over a future infinite horizon for the dynamic dial-a-ride problem. Similar with the waiting strategy, the rolling horizon approach [8] [15] [16] [9] that applies a myopic part of the planned routes is widely adopted in dynamic VRPs. The repetitively solved online optimization problems are largely an adaptation of the static version of the problem and required to be solved quickly. Therefore, heuristic approaches such as neighborhood search [17], insertion with branch & bound check for feasibility [18], branch-and-price [19] and approximate dynamic programming [12] have been used in solving dynamic VRPs. However, most of the mentioned dynamic VRPs do not allow a vehicle to divert from its current destination, and are based on simplified Euclidean metric networks.

The waterway networks in port areas are characterized by the geographical layouts of terminals which are in general non-convex. The distances between terminals and the distances from a arbitrary vessel position to a terminal cannot be simply calculated in the Euclidean space. The literature on maritime VRPs sees more static scheduling problems [6] [4] [20]. Due to the large magnitude of energy consumption in the shipping industry, the so-called “slow-steaming” by cruising at a low speed is usually considered in ship routing [20]. The static “slow-steaming” problem in [20] is extended to a dynamic sustainable ship routing problem in [21], where nonlinear mixed integer programming problems are solved online using particle algorithms. Capacity constraints are also commonplace in ship routing problems [22] [23] due to the relatively large volumes of cargoes. An exception is [15],

*This research is supported by National Natural Science Foundation of China under Grants 61903329 and 61872318.

[1]Key Laboratory of Ocean Observation-Imaging Testbed of Zhejiang Province, Zhejiang University, Zhoushan, China. [2]Transportation & Big-Data Research Center, Hangzhou Innovation Institute of Beihang University, Hangzhou, China. [3]Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands. [4]College of Computer Science and Technology, Zhejiang University, Hangzhou, China, and Department of Computer Science, University of Hong Kong, Hong Kong, China. *Corresponding author: hrzheng@zju.edu.cn

where the fleet of vessels is scheduled for the maintenance tasks of offshore wind farms. The dynamic scheduling problem is also solved in a rolling horizon framework in [15]. However, vessels are generally not able to divert if they have already started to sail towards a destination wind farm associated with relevant maintenance facilities.

In this paper, for the ITT using waterborne AGVs, we propose a dynamic rolling horizon scheduling strategy considering the complex geographical layouts of terminals in large ports. ITT requests with container pick-up and delivery tasks within specified time windows arrive dynamically. To adapt waterborne AGV routes to the dynamic requests, every certain period of time, an online mixed integer linear programming problem is solved at a decision epoch. The most recent vessel states (positioning and loads on board), dynamic topology of the waterway transport network, and ITT requests over a future time horizon are incorporated in the online problem. Vehicle diversion is possible as long as the traveling distance is minimized, and the capacity as well as timing constraints allow. A transport network based on the waterways that connect the relevant terminals in the Port of Rotterdam is constructed. To explore the influences of different route updating interval and horizon lengths on the scheduling performance, simulations of different settings are carried out and compared. Results demonstrate the effectiveness of the proposed dynamic scheduling strategy.

The remainder of this paper is organized as follows. The overall problem statement for the dynamic ITT system using waterborne AGVs is presented in Section II. Then in Section III, the rolling horizon scheduling problem is mathematically formulated. In Section IV, simulations and results are presented, followed by the concluding remarks and future research directions in Section V.

II. Problem statement

We consider an autonomous ITT system using waterborne AGVs. The port authority runs a fleet of waterborne AGVs shuttling between terminals to transport containers within the port area. The scheduling problem determines a route for each waterborne AGV, i.e., the sequence of terminals and times to visit and to load or unload certain amount of containers. For large ports like the Port of Rotterdam, terminals with different functions, e.g., empty container depot, transferring stations to other modalities (rail, road), etc., are dispersed. We are particularly interested in those terminals with shorter distances by water than by road. Figure 1 shows the ITT network for waterborne AGVs in the Maasvlakte area, Port of Rotterdam. The yellow lines represent waterways that connect the terminals by water. Apparently, due to the complex geographical layout in the Maasvlakte area, the inter terminal waterway distances cannot be simply calculated as the Euclidean distance in the Euclidean metric space as most existing literature does. Rather, the geodetic distance that is defined as the shortest path of two nodes in a connected graph [24] should be utilized. Therefore, along with solving the scheduling problem, the shortest path that connects the planned sequence of terminals is also determined via the

Fig. 1: ITT waterway network for waterborne AGVs in the Maasvlakte area, Port of Rotterdam from Google Earth [25].

Dijkstra algorithm [24]. The path consists of the terminals and waterway intersections as illustrated in Figure 1. Note that since the problem is dynamic, the current waterborne AGV position is considered as a virtual terminal that is the departure node of the waterborne AGV at the next scheduling epoch. The main challenge in a dynamic ITT system is that the nodes here refer not only to the static locations of terminals, but also to the locations of waterborne AGVs which could be evolving with time. The dynamic locations of waterborne AGVs make the transport network topology also dynamic, as to be modeled in Section III. The real-time update of the geodetic distance complicates the scheduling problem even more.

Another dynamism of the system comes from the dynamically revealed ITT requests to transport containers between the terminals shown in Figure 1 by water. Each request is characterized by the release and due times, the container pick-up and delivery terminals, and the amount of containers in TEUs. All the requests are sorted by their release times and associated with an ID according to their order. Note that since we also consider loading and unloading service times, the release time is the earliest time that the loading service can start and the due time is the latest time that the unloading service should complete. The request information is not known until their earliest starting time. The rolling horizon scheduling scheme applies periodic updates of waterborne AGV routes. This means that any new requests that arrive after the current decision step will be kept until the next decision step. In addition, since delays or waiting times do occur in reality and meeting hard time windows may fail in finding a feasible solution, requests are allowed to service within soft time windows, but customer inconvenience cost will incur if not within hard time windows. Splitting of request volume is not allowed.

Waterborne AGVs are designed for autonomous ITT systems. Each waterborne AGV has a finite capacity that can accommodate mixed containers from different requests. Containers on board are kept track of, and cannot be transported with transhipment, i.e., containers on board a waterborne AGV can only be delivered to the destination terminal by the same waterborne AGV. Localization and communication devices are on board of waterborne AGVs so that real-time positions can be measured and sent to the scheduling center. Waterborne AGVs have three working modes: *1*) waiting at a terminal being idle; *2*) providing loading or unloading service at a terminal; *3*) moving towards a terminal to pick-up or deliver containers. Waterborne AGVs move at a designed constant speed. Note that they can stay at the park lot of any terminal so that a central depot is not necessary.

III. Dynamic rolling horizon scheduling of ITT using waterborne AGVs

In the dynamic setting, the ITT requests are revealed online concurrently with the execution of previously planned routes. The routes are not necessarily completed when a new decision epoch has reached. Together with the newly arrived requests and the uncompleted requests, new decisions are made satisfying various possibly conflicting objectives considering transport tasks and system constraints. Next, we present a dynamic rolling horizon scheduling approach based on mixed integer programming for ITT using waterborne AGVs. Following the problem scenario presented in the previous section, we first introduce the system dynamic states and relevant notations. Then, decision variables and the mathematical model are presented. Furthermore, to reduce required computation times, the scheduling problem is transformed into a mixed integer linear programming (MILP) problem via mixed logic dynamic modeling.

A. Notations

We deal with a dynamic system over the time $t \in [0, \infty]$ with a fleet of n_v waterborne AGVs denoted by $\mathcal{V}$. Waterborne AGVs that are in the process of executing the previously assigned routes are in $\mathcal{V}_{\text{w}}$. Since it is impractical to optimize over the infinite time horizon and future ITT requests are not known beforehand, we carry out a re-scheduling with the interval of T_{s}. Define time step k with $k = 0, 1, 2, \ldots$, and then $t = kT_{\text{s}}$. At each time step k, a re-scheduling problem is formulated over a future time horizon $[kT_{\text{s}}, (k+N_{\text{p}})T_{\text{s}}]$ based on the current system states and the ITT requests whose release times are earlier than $(k+N_{\text{p}})T_{\text{s}}$ and have not been finished by $(k+N_{\text{p}})T_{\text{s}}$. N_{p} is the scheduling horizon. At the next time step $k+1$, new system states are measured and new ITT requests over $[(k+N_{\text{p}})T_{\text{s}}, (k+N_{\text{p}}+1)T_{\text{s}}]$ are incorporated. The re-scheduling problem is then formulated over $[(k+1)T_{\text{s}}, (k+N_{\text{p}}+1)T_{\text{s}}]$, and thus accounting for the "rolling horizon".

Consider at time step k, for each waterborne AGV $v \in \mathcal{V}$, the set of uncompleted ITT requests from the previous scheduling step is $\mathcal{R}_v(k) = \mathcal{R}_v^p(k) \cup \mathcal{R}_v^d(k)$. $\mathcal{R}_v^p(k)$ denotes the set of uncompleted requests that have not been started yet by kT_{s}. $\mathcal{R}_v^d(k)$ denotes the set of requests whose containers are currently on board of waterborne AGV v and just have not been delivered yet. The total containers on board waterborne AGV v can then be identified as $l_v(k)$ which should be less than the capacity Q. The position of waterborne AGV v is $(x_v(k), y_v(k))$ along the waterways. Since we utilize the geodetic distances over the transport network, it is also required to record the waterway segment waterborne AGVs are currently in, denoted by $g_v(k)$. Due to the loading/unloading service times at terminals, it is also possible that a waterborne AGV is in the middle of loading/unloading operations. In these cases, waterborne AGVs are tagged with a left service time $s_v(k)$ at the current position. Therefore, the dynamic waterborne AGV states are characterized by $((x_v(k), y_v(k)), g_v(k), s_v(k), l_v(k), \mathcal{R}_v(k))$, $\forall v \in \mathcal{V}$. The cruising speed of waterborne AGVs is u. It is assumed that there is always a sufficient number of waterborne AGVs available so that no ITT requests need to be rejected. Actually, since we impose soft constraints for time windows to be introduced in the following, it would always be possible to serve all the requests on the price of delays whenever necessary.

In terms of requests, the concerned request set at time step k is $\bigcup_{v\in\mathcal{V}} \mathcal{R}_v(k) \cup \mathcal{R}_{\text{new}}(k)$, where $\mathcal{R}_{\text{new}}(k)$ is the set of newly revealed requests over $[(k+N_{\text{p}}-1)T_{\text{s}}, (k+N_{\text{p}})T_{\text{s}}]$. For each request $i \in \bigcup_{v\in\mathcal{V}} \mathcal{R}_v(k) \cup \mathcal{R}_{\text{new}}(k)$, denote a 7-element tuple $\langle i, p_i, d_i, t_{i,\text{min}}, t_{i,\text{max}}, q_i, s_i \rangle$ to represent the associated information as described in Section II, i.e., request ID, pick-up terminal, delivery terminal, release time, due time, volume, and service time. For each pick-up location p_i, a positive load $+q_i$ is attached, and each delivery location d_i, a negative load $-q_i$ attached.

For modeling the ITT pick-up and delivery network, define the set of starting nodes of all waterborne AGVs as $\mathcal{V}_{\text{o}}(k) = \{1, ..., n_v\}$ and the set of virtual end nodes as $\mathcal{V}_{\text{e}}(k) = \{n_v + 2n(k) + n_d(k) + 1, ..., 2n_v + 2n + n_d\}$, with $n(k) = \sum_{v\in\mathcal{V}} |\mathcal{R}_v^p(k)| + |\mathcal{R}_{\text{new}}(k)|$ being the number of pick-up and delivery pair requests and $n_d(k) = |\mathcal{R}_v^d(k)|$ being the number of delivery-only requests. The pick-up node set is $\mathcal{P}(k) = \{n_v+1, n_v+2, ..., n_v+n(k)\}$ and the delivery node set is $\mathcal{D}(k) = \{n_v+n(k)+1, n_v+n(k)+2, ..., n_v+2n(k)\}$. For the delivery-only requests $\mathcal{R}_v^d(k)$, an additional delivery-only node set is defined as $\mathcal{D}'(k) = \{n_v+2n(k), ..., n_v+2n(k)+n_d(k)\}$. Since the containers of the delivery-only requests have already been on a waterborne AGV and since transshipment is not allowed, each delivery-only node is sure to be visited by the same waterborne AGV. Define the corresponding visiting waterborne AGVs for $\mathcal{D}'(k)$ as $\mathcal{V}_d(k)$. Following the previous definition, the service time at node $i \in \mathcal{V}_{\text{o}}(k)$ is $s_v(k)$ and the service time at node $i \in \mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k)$ is $s_i(k)$. For convenience, $s_i(k)$ is used in the following for $i \in \mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k) \cup \mathcal{V}_{\text{o}}(k)$. Then, our scheduling problem is defined over the virtual graph $\mathcal{G}(k) = (\mathcal{N}(k), \mathcal{A}(k))$ with node set $\mathcal{N}(k) = \mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k) \cup \mathcal{V}_{\text{o}}(k) \cup \mathcal{V}_{\text{e}}(k)$ and arc set

$$\begin{aligned}
&\mathcal{A}(k) = \{(i,j)| \\
&(i,j) \in ((\mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k)) \times (\mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k))) \\
&\cup \{(i,j)\,| i \in \mathcal{V}_{\text{o}}(k), j \in \mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k)\} \\
&\cup \{(i,j)\,| i \in \mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k), j \in \mathcal{V}_{\text{e}}(k)\}, i \neq j\}.
\end{aligned}$$

Actually, for each $(i,j) \in \mathcal{A}(k)$, since nodes i and j are not connected by straight lines but a sequence of waterway segments, as shown in Figure 1, a path attribute is further defined for $\mathcal{G}(k)$ as $\mathcal{S}(k)$ and $\mathcal{S}(k) = \{p(i,j)\}$. Here, $\{p(i,j)\}$ is the detailed waterway path connecting node i and node j. Note that since the waterborne AGV states are evolving with time, $\mathcal{G}(k)$ is dynamic. Denote d_{ij} as the travel distance between nodes i and j for all $(i,j) \in \mathcal{A}(k)$. Since d_{ij} is the geodesic distance over the graph, when calculating d_{ij}, it is essential also to know which waterway segment the waterborne AGV is currently in. Furthermore, since waterborne AGVs stay at their final service terminals, the locations for virtual end nodes $\mathcal{V}_\text{o}$ vanish and distance $d_{ij} = 0$ if $i \in \mathcal{P}(k) \cup \mathcal{D}(k) \cup \mathcal{D}'(k) \cup \mathcal{V}_\text{o}(k), j \in \mathcal{V}_\text{e}(k)$.

B. Rolling horizon scheduling problem

The overall schedule goal is to fulfill all the ITT requests minimizing the travel distance, the waiting and delays times of all waterborne AGVs. The following decision variables are involved in the dynamic rolling horizon scheduling problem:

- Binary variables: $x_{ijv}(k)$ for $(i,j) \in \mathcal{A}(k)$ and $v \in \mathcal{V}$ equals to 1 if waterborne AGV v travels from node $i \to j$ and 0 otherwise;
- Binary variables: $z_{iv}(k)$ for $i \in \mathcal{N}(k)$ and $v \in \mathcal{V}$ equals to 1 if node i is visited by waterborne AGV v and 0 otherwise;
- Integer variables: $y_i(k)$ for $i \in \mathcal{N}(k)$ denotes the load in TEUs on board the waterborne AGV upon arriving node i;
- Continuous variables: $A_i(k)$ for $i \in \mathcal{N}(k)$ specifies the arrival time at node i;
- Continuous variables: $w_i(k)$ for $i \in \mathcal{N}(k)$ is the waiting time at node i;
- Continuous variables: $d_i(k)$ for $i \in \mathcal{N}(k)$ is the delay time at node i.

In the following, the time dependence of the variables, i.e., $\cdot(k)$, will be omitted for notational simplicity. It should, however, be kept in mind that the scheduling problem is dynamic. At each time step k, a mixed integer programming problem is formulated as follows:

$$\min\ c_1 \sum_{i\in\mathcal{V}_\text{o}(k)} \|A_i\|_1 + c_2 \sum_{v\in\mathcal{V}} \sum_{(i,j)\in\mathcal{A}} x_{ijv} d_{ij} + c_3 \|w\|_1 + c_4 \|d\|_1 \tag{1}$$

subject to

$$\sum_{v\in\mathcal{V}} z_{iv} = 1, \qquad \forall i \in \mathcal{N}(k), \tag{2}$$

$$z_{iv} = z_{(i+n)v}, \qquad \forall i \in \mathcal{P}(k), v \in \mathcal{V}, \tag{3}$$

$$z_{ii} = 1, \qquad \forall i \in \mathcal{V}_\text{o}(k), \tag{4}$$

$$z_{iv} = 1, \qquad \forall i \in \mathcal{D}'(k), v \in \mathcal{V}_d \tag{5}$$

$$\sum_{j\in\mathcal{N}} x_{ijv} = \sum_{j\in\mathcal{N}} x_{jiv} = z_{iv}, \qquad \forall i \in \mathcal{N}(k), v \in \mathcal{V}, \tag{6}$$

$$\sum_{j\in\mathcal{N}/\mathcal{V}_\text{e}} x_{v_\text{o}jv} = 1, \qquad \forall v \in \mathcal{V}, \tag{7}$$

$$\sum_{i\in\mathcal{N}/\mathcal{V}_\text{o}} x_{iv_\text{d}v} = 1, \qquad \forall v \in \mathcal{V}, \tag{8}$$

$$A_v = kT_\text{s}, \qquad \forall v \in \mathcal{V}_\text{w}(k), \tag{9}$$

$$A_i \leqslant A_{i+n}, \qquad \forall i \in \mathcal{P}_\text{n}(k), \tag{10}$$

$$x_{ijv} = 1 \Rightarrow \tag{11}$$

$$\max(A_i, t_{i,\min}) + s_i + \frac{d_{ij}}{u} = A_j, \qquad \forall (i,j) \in \mathcal{A}(k), v \in \mathcal{V},$$

$$t_{i,\min} - w_i \leqslant a_i \leqslant t_{i,\max} - s_i + d_i, \qquad \forall i \in \mathcal{N}(k), \tag{12}$$

$$0 \leqslant w_i \leqslant w_{\max}, \qquad \forall i \in \mathcal{N}(k), \tag{13}$$

$$0 \leqslant d_i \leqslant d_{\max}, \qquad \forall i \in \mathcal{N}(k), \tag{14}$$

$$y_{v_\text{o}} = l_v, \qquad \forall v \in \mathcal{V}, \tag{15}$$

$$x_{ijv} = 1 \Rightarrow y_i + q_i = y_j, \qquad \forall i \in \mathcal{N}(k), v \in \mathcal{V}, \tag{16}$$

$$0 \leqslant y_i \leqslant Q, \qquad \forall i \in \mathcal{N}(k), \tag{17}$$

$$x_{ijv}, z_{iv} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}(k), v \in \mathcal{V}. \tag{18}$$

In (1), there are four cost terms to be minimized. The first term minimizes the starting times of waterborne AGVs so that the known set of requests could be finished as soon as possible. The second term minimizes the total travel distance of all waterborne AGVs which relates to the energy consumption. The third and the fourth terms account for customer inconvenience incurred by waiting and delay times, respectively. The trade-off among these cost penalties is balanced by weight parameters $c_1, \cdots, c_4$.

Compatibility, time consistence, capacity and binary variable constraints are imposed by (2) – (18). Specifically, constraint (2) ensures that each node is visited exactly by one waterborne AGV. Constraint (3) defines that, for a paired request $r \in \mathcal{R}_v^p(k) \cup \mathcal{R}_\text{new}(k)$, the pick-up and delivery nodes are visited by the same waterborne AGV. By constraints (4) and (5), waterborne AGVs are guaranteed to visit their own starting nodes and to visit the delivery nodes if the corresponding containers are on board, respectively. Constraint (6) restricts that a waterborne AGV only enters and leaves a node if it visits that node. Constraints (7) and (8) impose that each waterborne AGV starts and ends at the right locations, respectively. Constraints (9) – (16) together impose time constraints. Specifically, equality constraint (9) ensures the time continuity for waterborne AGVs that are carrying out tasks at time step k. Inequality (10) guarantees that pick-up nodes are visited before delivery nodes. Constraint (11) enforces time consistency where the *max* operation indicates that loading/unloading services cannot start earlier than the release times of requests. Time window constraints are specified by (12) - (14). Load consistence and capacity constraints are introduced via (15) – (17). Binary variables are defined in (18). By solving (1)–(18), we have, for each waterborne AGV $v \in \mathcal{V}$, the sequences of terminals to visit, the corresponding arrival times as well as the stopping times.

IV. Simulations and discussions

To demonstrate the effectiveness of the proposed dynamic scheduling strategy, simulations are carried out. Overall, the

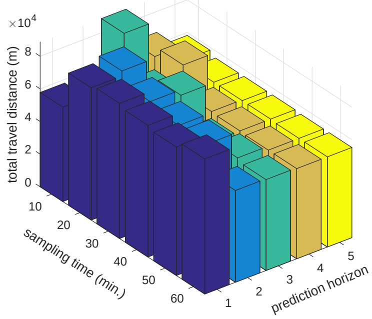

Fig. 2: Travel distances with different T_s and N_p.

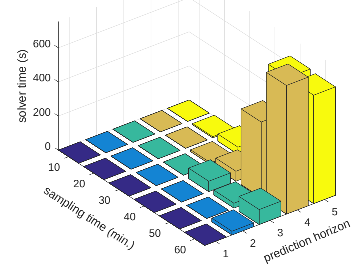

Fig. 3: Solver times with different T_s and N_p.

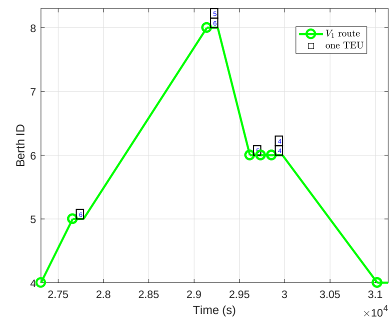

Fig. 4: Scheduled routes of waterborne AGV 1 at $k = 10$.

ITT scenario is based on the waterway transport network in the Port of Rotterdam, as shown in Figure 1. There are eight terminals and a fleet of three waterborne AGVs initially staying at one terminal being idle. We consider a set of 10 ITT requests generated from a Port of Rotterdam ITT simulator [26]. Furthermore, we run the simulations with different settings to explore the influences of different lengths of scheduling horizons and different re-scheduling intervals on the system performance.

For other simulation settings, the maximum capacity of waterborne AGVs is four TEUs, i.e., $Q = 4$. The loading/unloading service time of each move of a quay crane of two TEUs is requires 120s. Therefore, for all the considered ITT requests, service times are the same as $t_s = 120$s. The weight parameters in cost functions (1) are set as: $c_1 = 10^2, c_2 = 10^4, c_3 = 10^6, c_4 = 10^6$. A waterway ITT simulation environment is also built. All the simulations are run on a platform with Intel (R) Core (TM) i7-10710U CPU @1.10 GHz.

A. Comparisons of different scenarios

Different combinations of re-scheduling interval T_{s} and scheduling horizon N_{p} are set for the dynamic rolling horizon scheduling algorithm. For the set of ITT requests, the time windows are all satisfied and no waiting or delay times are incurred, as to be further analyzed in Figure 5. Therefore, we mainly compare the index of total travel distance in different settings, as shown in Figure 2. Specifically, the re-scheduling interval T_{s} is set as 10 min, 20 min, $\cdots$, 60 min. The scheduling horizon N_{p} is set as $1, \cdots, 5$. Therefore, we run 30 simulations with time horizon $T_{\mathrm{s}}N_{\mathrm{p}}$ ranging from 10 min to 5 hours. The travel distance costs is affected by both the time horizon and the re-scheduling interval. Overall, the trend is that longer time horizons have lower travel distances, since more ITT requests could be planned as a whole with longer time horizons. The total travel distance is defined as the sum of the distances traveled by all waterborne AGVs completing all ITT requests. Shorter re-scheduling intervals could also improve optimality.

Figure 3 further compares the mean solver times per time step of different T_{s} and N_{p} combinations. It can seen that longer time horizons result in unacceptably long computational times which are not suitable for real-time applications. In view of the scheduling performance and computational time, we next analyze the scheduling results from the simulation with $T_{\mathrm{s}} = 40$ min and $N_{\mathrm{p}} = 3$ in more details.

B. Scheduling results

In a dynamic scenario, the set of ten ITT requests are assigned to only one waterborne AGV, waterborne AGV 1. Figure 4 plots the scheduled routes of waterborne AGV 1 at time step $k = 10$. The small rectangles are one TEU containers and the numbers attached identify IDs of requests that the containers belong to. The shown containers are those on board the waterborne AGV when it departs from a terminal. The schedule contains the information on the sequence of terminals to visit, the corresponding arrival and departure times as well as the load/unload operations at each terminal. For the schedule in Figure 4, the waterborne AGV first starts from Terminal 4. Then, it goes to Terminal 5 to pick up one container from Request 6 and then go to Terminal 8 to pick up one container from Request 5. At Terminal 6, the waterborne AGV performs multiple unload and load operations, first unloading the container from Request 6, unloading the container from Request 5, and then loading two containers from Request 4 which are then delivered to Terminal 4. The waterborne AGV finishes the assigned tasks at Terminal 4 and will stay there until the next task is assigned.

Each ITT request should be completed within the release and due times. Figure 5 shows the specified time windows (red bars) and the actual duration time (green bars) specified by the waterborne AGV's arrival at the origin terminal and departure from the destination terminal. All actual duration

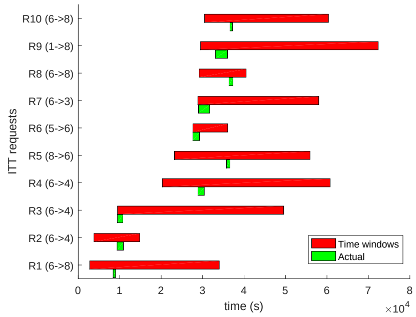

Fig. 5: Satisfaction of request time windows.

times are within required time windows, i.e., time windows of all requests are satisfied by the scheduling problem.

V. Conclusions and Future Research

For the novel type of container transport vehicles, waterborne AGVs, this paper proposes a dynamic rolling horizon scheduling strategy in view that ITT requests are actually revealed in real-time. By adopting the rolling horizon strategy, a pick-up and delivery re-scheduling problem is solved every certain period of time. System states of the fleet of waterborne AGVs including positions, waterway segments and containers on board, dynamic waterway transport network, and newly arrived ITT requests are incorporated to update the previously assigned routes. Time windows, capacity constraints and container load/unload service times are also considered. Simulation results based on the waterway transport network in the Port of Rotterdam demonstrate that with proper scheduling horizon and intervals, the proposed strategy can be applied in real-time to address the scheduling problem an autonomous ITT system. Future research will develop fast solution, probably heuristic, approaches to solve the online re-scheduling problems more efficiently.

References

[1] Port of Rotterdam Authority, "Port Vision 2030," http://www.portofrotterdam.com/en/Port/port-in-general/port-vision-2030/Pages/default.aspx, December 2011, Accessed: 2016-01-17.

[2] J. Xin, R. R. Negenborn, and T. van Vianen, "A hybrid dynamical approach for allocating materials in a dry bulk terminal," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1326–1336, 2018.

[3] L. Heilig and S. Voß, "Inter-terminal transportation: an annotated bibliography and research agenda," *Flexible Services and Manufacturing Journal*, vol. 29, no. 1, pp. 35–63, 2017.

[4] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Closed-loop scheduling and control of waterborne AGVs for energy-efficient inter terminal transport," *Transportation Research Part E: Logistics and Transportation Review*, vol. 105, pp. 261–278, 2017.

[5] M. W. Savelsbergh and M. Sol, "The general pickup and delivery problem," *Transportation science*, vol. 29, no. 1, pp. 17–29, 1995.

[6] H. N. Psaraftis, M. Wen, and C. A. Kontovas, "Dynamic vehicle routing problems: Three decades and counting," *Networks*, vol. 67, no. 1, pp. 3–31, 2016.

[7] Z. Steever, M. Karwan, and C. Murray, "Dynamic courier routing for a food delivery service," *Computers & Operations Research*, vol. 107, pp. 173–188, 2019.

[8] D. Reyes, A. Erera, M. Savelsbergh, S. Sahasrabudhe, and R. ONeil, "The meal delivery routing problem," *Optimizations*, vol. onine, 2018.

[9] F. Miao, S. Han, S. Lin, J. A. Stankovic, D. Zhang, S. Munir, H. Huang, T. He, and G. J. Pappas, "Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 463–478, 2016.

[10] A. Haghani and S. Yang, "Real-time emergency response fleet deployment: Concepts, systems, simulation & case studies," in *Dynamic Fleet Management*, V. Zeimpekis, C. D. Tarantilis, G. M. Giaglis, and I. Minis, Eds. Springer US, 2007, pp. 133–161.

[11] M. Gendreau, F. Guertin, J. Y. Potvin, and R. Seguin, "Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 3, pp. 157–174, 2006.

[12] S. Vonolfen and M. Affenzeller, "Distribution of waiting time for dynamic pickup and delivery problems," *Annals of Operations Research*, vol. 236, no. 2, pp. 359–382, 2016.

[13] V. Pureza and G. Laporte, "Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows," *Infor: Information Systems & Operational Research*, vol. 46, no. 3, pp. 165–176, 2008.

[14] H. R. Sayarshad and J. Y. Chow, "A scalable non-myopic dynamic dial-a-ride and pricing problem," *Transportation Research Part B: Methodological*, vol. 81, pp. 539–554, 2015.

[15] N. Raknes, K. Ødeskaug, M. Stålhane, and L. Hvattum, "Scheduling of maintenance tasks and routing of a joint vessel fleet for multiple offshore wind farms," *Journal of Marine Science and Engineering*, vol. 5, no. 1, p. 11, 2017.

[16] X. Wang and H. Kopfer, "Rolling horizon planning for a dynamic collaborative routing problem with full-truckload pickup and delivery requests," *Flexible Services and Manufacturing Journal*, vol. 27, no. 4, pp. 509–533, 2015.

[17] M. Wen, J.-F. Cordeau, G. Laporte, and J. Larsen, "The dynamic multi-period vehicle routing problem," *Computers & Operations Research*, vol. 37, no. 9, pp. 1615–1623, 2010.

[18] J.-F. Cordeau, M. Dell?Amico, S. Falavigna, and M. Iori, "A rolling horizon algorithm for auto-carrier transportation," *Transportation Research Part B: Methodological*, vol. 76, pp. 68–80, 2015.

[19] G. Ozbaygin and M. Savelsbergh, "An iterative re-optimization framework for the dynamic vehicle routing problem with roaming delivery locations," *Transportation Research Part B: Methodological*, vol. 128, pp. 207–235, 2019.

[20] H. N. Psaraftis and C. A. Kontovas, "Ship speed optimization: Concepts, models and combined speed-routing scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 52–69, 2014.

[21] A. De, V. K. R. Mamanduru, A. Gunasekaran, N. Subramanian, and M. K. Tiwari, "Composite particle algorithm for sustainable integrated dynamic ship routing and scheduling optimization," *Computers & Industrial Engineering*, vol. 96, pp. 201–215, 2016.

[22] G. Tirado, L. M. Hvattum, K. Fagerholt, and J.-F. Cordeau, "Heuristics for dynamic and stochastic routing in industrial shipping," *Computers & Operations Research*, vol. 40, no. 1, pp. 253–263, 2013.

[23] G. Tirado and L. M. Hvattum, "Improved solutions to dynamic and stochastic maritime pick-up and delivery problems using local search," *Annals of Operations Research*, vol. 253, no. 2, pp. 825–843, 2017.

[24] X. Wang, E. Pournaras, R. E. Kooij, and P. Van Mieghem, "Improving robustness of complex networks via the effective graph resistance," *The European Physical Journal B*, vol. 87, no. 9, p. 221, 2014.

[25] "Maasvlakte Rotterdam, 51.962398° N and 4.056800° E, Google Earth," August 7, 2013, Accessed: 2020-1-9.

[26] E. Gerritse, "Analysis for inter terminal transportation demand scenarios for the maasvlakte I and II in 2030," Delft University of Technology, Tech. Rep., 2014.