Closed-loop scheduling and control of waterborne AGVs for energy-efficient Inter Terminal Transport

Zheng, Huarong; Negenborn, Rudy; Lodewijks, Gabri

DOI
10.1016/j.tre.2016.07.010

Publication date
2017

Document Version
Final published version

Published in
Transportation Research. Part E: Logistics and Transportation Review

Transportation Research Part E 105 (2017) 261–278

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Closed-loop scheduling and control of waterborne AGVs for energy-efficient Inter Terminal Transport

Huarong Zheng *, Rudy R. Negenborn, Gabriël Lodewijks

Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 18 January 2016
Received in revised form 22 July 2016
Accepted 22 July 2016
Available online 4 August 2016

Keywords:
Closed-loop scheduling and control
Coordinated berthing times
Energy efficiency
Inter terminal transport
Waterborne AGVs

ABSTRACT

We propose closed-loop energy-efficient scheduling and control of an autonomous Inter Terminal Transport (ITT) system using waterborne Autonomous Guided Vessels (waterborne AGVs). A novel pick-up and delivery problem considering safety time intervals between berthing time slots of different waterborne AGVs is proposed. Waterborne AGVs are controlled in a cooperative distributed way to carry out the assigned schedules. Real-time scheduling and control loop is closed by a partial scheduling model and an interaction model with feedback reflecting neglected lower level factors. Simulation results demonstrate the effectiveness of the proposed methodology and the potential of applying waterborne AGVs towards an autonomous ITT system.

1. Introduction

The Port of Rotterdam envisions annually more than 30 million Twenty-foot Equivalent Unit (TEU) to be handled by 2035 (Port of Rotterdam Authority, 2011). A large part of this throughput is actually going to take place internally among terminals, i.e., Inter Terminal Transport (ITT) (Tierney et al., 2014). A novel vehicle type, waterborne Autonomous Guided Vessels (waterborne AGVs) (Zheng et al., in press, 2015), has been previously proposed for ITT. Waterborne AGVs show great potential for transport in increasingly larger and busier ports since: (1) waterborne AGVs are labor cost free; (2) waterborne AGVs offer another transport mode to handle the expected large throughput instead of exploiting limited land in port areas for road traffic; (3) waterborne AGVs, comparable to land-side Automated Guided Vehicles (AGVs) (Xin et al., 2014), can be optimally operated 24/7 with reliable performance and improve port efficiency; (4) for terminals with longer distances by land than by water, waterborne AGVs save energy compared to road vehicles; and (5) waterborne AGVs are in line with the development of smart ports and are deemed as very relevant to the ITT practice in the port of Rotterdam (Erasmus Smart Port Rotterdam, 2015). Previous work (Zheng et al., in press, 2015) on waterborne AGVs has been focusing on designing efficient controllers at the control level. In Zheng et al. (2015), multiple waterborne AGVs are controlled in a cooperative distributed way given one fixed ITT request per waterborne AGV, i.e., the scheduling and control problems are solved in an open-loop way. Sharing the common aims of making economical and environmentally friendly decisions, scheduling and controlling waterborne AGVs are expected to achieve further benefits in a closed-loop way with tighter integration.

However, scheduling and control, typically as two distinct levels in a transportation decision-making hierarchy have been explored independently by researchers in the two areas, see e.g., Li et al. (2016) and Zheng et al. (in press) for ITT systems. Analogous levels of a typical ITT system using waterborne AGVs are shown in Fig. 1. Within an ITT system, the port authority

* Corresponding author.
E-mail addresses: h.zheng-1@tudelft.nl (H. Zheng), R.R.Negenborn@tudelft.nl (R.R. Negenborn), G.Lodewijks@tudelft.nl (G. Lodewijks).

http://dx.doi.org/10.1016/j.tre.2016.07.010
1366-5545/

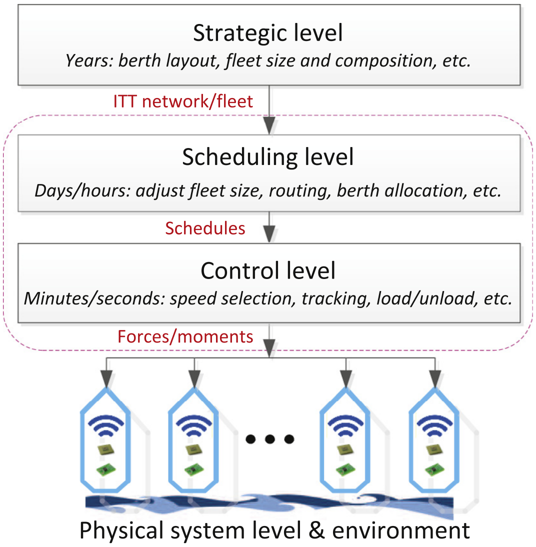

Fig. 1. Different levels of an ITT system using waterborne AGVs (adapted from Christiansen et al. (2007)).

runs a fleet of waterborne AGVs shuttling among terminals internally in the port area to transport containers in a cost-effective way. Strategic decisions regarding locations of berths (exclusively for waterborne AGVs), fleet size and composition issues, etc. are long term decisions in the order of years. Scheduling and control levels determine, for each waterborne AGV, the chronological events that occur at the hours time scale and the amount of power to input that occurs at the seconds time scale in order to assure those events are executed as scheduled, respectively. In other words, schedules, e.g., terminals and time to visit, load or unload certain amount of containers are discrete-event based while control problems usually considering fast lower-level dynamics in the presence of operational constraints and uncertainties are approximately time-continuous oriented. Although both levels largely rely on mathematical models and optimization techniques and both aim at either maximizing profit or minimizing cost, the inherently different time-scale nature brings technical challenges for an integrated and computationally tractable solution. On the one hand, discrete decisions involved in scheduling problems restrict them to nothing but low dimension models solved in low frequency and off-line; on the other hand, feedback and closed-loop operation in real-time are essential in control systems to handle disturbances and complex dynamics. Efforts have been made either from a "Top-down" perspective by considering control elements in a scheduling problem (Xin et al., 2014) or from a "Bottom-up" perspective by including scheduling-oriented economic terms in the cost function of a control problem (Angeli, 2014). In the field of process industry, the economic benefits of integrating scheduling and control have been recently recognized and emphasized (Baldea and Harjunkoski, 2014). A so-called "time scale-bridging" model is proposed in Du et al. (2015), but this model counts on an explicit, low-order representation of the input/output process dynamics which is by all means hard to derive for general systems, and moreover, operational constraints cannot be incorporated. A decent solution to integrated scheduling and control has to date not yet been proposed.

Hierarchically, nonetheless, problems of scheduling and control are both well solved. Waterborne AGV scheduling for ITT essentially is a pick-up and delivery problem (PDP) (Savelsbergh and Sol, 1995) with time windows using capacitated vehicles. PDP is a generalization of vehicle routing problems (VRP) (Toth and Vigo, 2014). Both PDPs and VRPs decide a set of optimal routes for a fleet of vehicles but differ in that PDP deals with transportation between distinct pick-up and delivery locations while in VRP, either the pick-up or the delivery location needs to be the same, i.e., the depot. Within the operational research realm in a logistical context, it is customary and sufficient to only care for setting schedules on discrete events and neglect any details on how these events really happen. From a control point of view, however, vehicles concerned in VRPs or PDPs are actually assumed as dimensionless mass points predominantly with constant speeds such that any lower level feedback become irrelevant in a scheduling problem. We observe that two variants of VRPs are exceptional. The time-dependent VRPs (Figliozzi, 2012) adopt a time-dependent speed model which, to some extend, considers lower level information, e.g., traffic congestion. But the speed model is known prior rather than a decision variable that could be manipulated. The time-dependent VRPs belong to a more generic class of dynamic VRPs (Psaraftis et al., 2016) dealing with dynamism such as online requests and dynamic travel times and update route responsively. Solutions with acceptable computational efficiency are largely of concern for dynamic VRPs. The second exception is the pollution-routing problem proposed in Bektaş and Laporte (2011), which considers factors as load and speed in producing "environmental-friendly" vehicle routes. The resulting problem is more difficult to solve but yields lower cost. Still, the combined route-speed optimization is open-loop and far from being able to consider lower level complex dynamics.

In the maritime sector, the relation between ship speed and energy consumption is highlighted even more by both practitioners and researchers. The engine of Maersk "Triple-E" (Maersk, 2013) was designed to sail relatively slowly to reduce

50% of the CO_2 emitted on the Asia and North Europe transport route. Another common practice in the shipping industry known as "slow-steaming" (Maloni et al., 2013) by cruising at a lower speed than the design speed to reduce cost has also been widely accepted and implemented (Psaraftis and Kontovas, 2014). Arrival times are optimized in Fagerholt et al. (2010) and Norstad et al. (2011) to obtain optimal speeds along shipping routes. Results of applying the method to real shipping routes shows the potential for reducing environmental emissions is substantial. Besides the emphasis on speed, coordination of arrival times of ships at terminals to avoid unnecessary waiting or conflicts is more critical than land-based vehicles. The reasons are twofold. First, it is more frequent for ships to visit the same terminal considering the limited pick-up and delivery locations. This is particularly the case in ITT. In fact, most PDPs assume distinct pick-up and delivery locations and each vehicle visits each location exactly once (Savelsbergh and Sol, 1995), which diminishes the arrival time coordination. Secondly, loading/unloading of ships could take more time than land-based vehicles, and thus cannot be neglected. Berthing time clash avoidance is modeled in Pang et al. (2011) by constraining, for pick-up and delivery visits sharing the same berth, the departure time of a visit to be not larger than the arrival times of a later visit. This is problematic when extra time intervals are imposed between departure and arrival times which is practically the case if ship dimensions and safety distances are considered. Another characteristic of maritime logistics is that environmental uncertainties are prevalent. These uncertainties include current, waves, wind and encounters with other moving objects that not only interact with waterborne AGV dynamics at the operational level but also influence the scheduling level. This calls for a closed-loop system involving both scheduling and control that makes decisions based on real-time feedback.

In this article, we propose a novel closed-loop scheduling and control approach for a fleet of waterborne AGVs towards an autonomous ITT system. Closed-loop means that both scheduling and control levels make decisions online based on system states measured with a fast sampling time. Decisions are still made hierarchically to guarantee tractability. Moreover, a new PDP scheduling model considering necessary time intervals between different waterborne AGVs visiting a particular berth is proposed. Further more, we propose a partial scheduling problem that is efficient to solve, and an interaction model that integrates the scheduling and control problems. Solving the scheduling problem generates for each waterborne AGV a sequence of terminals to visit satisfying service time windows. Cooperative distributed model predictive control based on a fast ADMM algorithm (Zheng et al., 2015) is then applied by the group of involved waterborne AGVs to execute the schedules safely and accurately. The main advantage of using a closed-loop scheme over an open-loop scheme is that real-time factors like unconsidered physical system limit, disturbances, and collision avoidance that are difficult, if not impossible, to be integrated in a scheduling problem can be reflected timely by the online updated schedules.

The remainder of this article is organized as follows. The overall problem statement for an autonomous ITT system using waterborne AGVs is first introduced in Section 2. Then in Section 3, we formulate the energy efficient scheduling problem with coordinated berthing times. The closed-loop scheduling and control based on a real-time coupling speed assignment problem and an interaction model is proposed in Section 4. In Section 5, simulation experiments and results are presented, followed by concluding remarks and future research directions in Section 6.

2. Problem statement

We consider an autonomous ITT system: a fleet of waterborne AGVs that handles a set of emerging ITT requests to transport specified amounts of containers between specified origins and destinations within specified time windows autonomously in an energy efficient way. Assumptions are made that: (1) the fleet size and composition have been decided by the strategic level (see Fig. 1) in a way that there are always sufficient number of waterborne AGVs available. Note that the available fleet size does not necessarily coincide with the real deployed fleet which is the decision to be made at the scheduling level, as to be introduced in Section 3; (2) the ITT network has also been designed by the strategic level. The network includes: berths that can accommodate waterborne AGVs by providing charging, maintenance, parking, etc., and fixed routes as shortest paths connecting berths; (3) each terminal has one waterborne AGV berth. In practice, a terminal can have multiple berths, which can then be viewed as multiple pickup/delivery locations and thus the problem formulation is in essential not changed; and (4) ITT requests are decoupled between different planning horizons so that requests arising within each planning horizon are completed within that horizon.

Waterborne AGVs are designed with specifications for the autonomous ITT system. Each waterborne AGV has a finite capacity which can accommodate mixing containers from different requests. Processing units, measurement and communication devices are on board of waterborne AGVs so that they are able to measure their own system states, communicate with other waterborne AGVs within a certain range and make their own control decisions. Besides, waterborne AGVs use "environmentally friendly" engines and perform "slow steaming" by cruising at lower speeds if possible. There is no central depot for waterborne AGVs and they stay at the park lot of the final service berth: show up at the berth when starting to working and disappear from the berth when having finished all tasks. Finally, waterborne AGVs are with certain dimensions and need to keep a certain safety distance from others to avoid collisions. Collision avoidance is achieved among moving waterborne AGVs by cooperative distributed control (Zheng et al., 2015) while waterborne AGVs visiting the same berth to perform loading or unloading operations are yet to be coordinated. Fig. 2 shows an illustration of an ITT system with a fleet of three waterborne AGVs, six waterborne AGV berths and 12 routes.[1]

[1] These routes are all with much shorter distances by water than by land while routes connecting, e.g., berth 5 and 6, with short land distances are considered by other ITT modes.

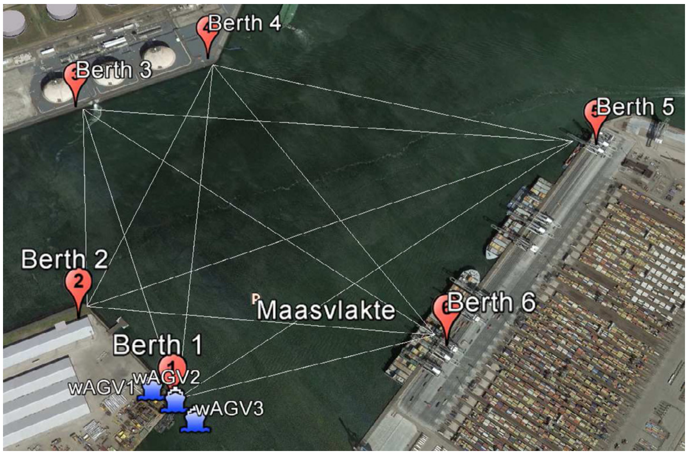

Fig. 2. Illustration of an ITT system with six berths for three waterborne AGVs.

In light of the above ITT network and available fleet of waterborne AGVs, the list of ITT requests should be available upon making decisions. In particular, each ITT request is associated with information on seven aspects: (1) request ID which is sorted by all requests' release times; (2) origin berth ID corresponding to the pick-up location; (3) destination berth ID corresponding to the delivery location; (4) release time defining when a set of containers are ready to be shipped, being the earliest time that the loading service can start; (5) due time defining when the set of containers are ready for subsequent operations, being the latest time of completing this request including the unloading time at the destination berth; and (6) volume of the set of containers to be shipped in TEUs; (7) service time for loading/unloading the set of containers. 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 be serviced within soft time windows, but customer inconvenience cost will incur if not within hard time windows. Note that trade-offs can also be made by using more waterborne AGVs to reduce delays. Splitting of request volume is not allowed, i.e., containers with the same request ID cannot be shipped by different waterborne AGVs. Finally, containers have to be transported without transhipment, i.e., loading and unloading operations happen exactly once for each request.

The autonomous ITT system runs in a closed-loop fashion, i.e., both scheduling and control level problems in Fig. 1 are solved in real-time using updated system states. Control of a fleet of waterborne AGVs is realized as in Zheng et al. (2015) while the energy efficient scheduling problem as well as the closed-loop scheduling and control design are presented in the following two sections.

3. Energy efficient scheduling of ITT using waterborne AGVs

Traditionally, the scheduling problem of a fleet of ships traveling back and forth among terminals to transport goods relies on human dispatchers necessarily with high competence and experience. Complex decisions need to be made satisfying various possibly conflicting objectives (e.g., saving energy by sailing at low speeds while meeting time windows at high speeds) considering transport requests and available vehicle lists. Allocation of waterborne AGVs to routes and timing can break such an operator. In face of real-time operational delays and uncertainties, the problem can frustrate human dispatchers even more. We next present a scheduling model based on mixed integer programming for ITT using waterborne AGVs to ease the workload of human dispatchers. We first introduce relevant notations including input parameters to the model and decision variables to be solved from this model. Then the mathematical model is presented which is further transformed into a mixed integer linear programming (MILP) problem to reduce required computation times.

3.1. Notations

The planning horizon considered within which a set of ITT requests $\mathcal{R}$ among the set of berths $\mathcal{B}$ arise is $[0, T_{\mathrm{p}}]$. For each request $i \in \mathcal{R}$, we denote a 7-element tuple $\langle i, p_i, d_i, t_{i,\min}, t_{i,\max}, q_i, s_i \rangle$ to represent the information associated with request i as described in Section 2, i.e., request ID, pick-up berth, delivery berth, 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. The set of n_v waterborne AGVs is $\mathcal{V}$ and homogenous. The set for start locations for all waterborne AGVs is defined as $\mathcal{V}_{\mathrm{o}} = \{1, \ldots, n_v\}$ and the set for end locations as $\mathcal{V}_{\mathrm{e}} = \{n_v + 2n + 1, \ldots, 2n_v + 2n\}$ with $n = |\mathcal{R}|$. All waterborne AGVs have the same capacity Q in TEUs, curb weight m and cruising speed range $[u_{\min}, u_{\max}]$. All TEU of containers are assumed to have the same weight of m_c.

As has been discussed before, ITT scenarios inevitably involve waterborne AGVs shuttling back and forth, and thus pick-up and delivery locations of different requests might actually be the same physical berths. This is one of the main differences of our scheduling problem with land-based VRPs (Toth and Vigo, 2014) or PDPs (Savelsbergh and Sol, 1995) based on assumptions of distinct pick-up and delivery locations and vehicles visit each location exactly once. We, hence, define virtual pick-up and delivery node sets $\mathcal{P}_\mathrm{n} = \{n_v+1, n_v+2, \ldots, n_v+n\}$ and $\mathcal{D}_\mathrm{n} = \{n_v+n+1, n_v+n+2, \ldots, n_v+2n\}$, respectively. Then, our scheduling problem is defined over the virtual graph $\mathcal{G}_s = (\mathcal{N}, \mathcal{A})$ with node set $\mathcal{N} = \mathcal{P}_\mathrm{n} \cup \mathcal{D}_\mathrm{n} \cup \mathcal{V}_\mathrm{o} \cup \mathcal{V}_\mathrm{e}$ and arc set $\mathcal{A} = \{(i,j)|(i,j) \in ((\mathcal{P}_\mathrm{n} \cup \mathcal{D}_\mathrm{n}) \times ((\mathcal{P}_\mathrm{n} \cup \mathcal{D}_\mathrm{n})) \cup \{(i,j)|i \in \mathcal{V}_\mathrm{o}, j \in \mathcal{P}_\mathrm{n} \cup \mathcal{D}_\mathrm{n}\} \cup \{(i,j)|i \in \mathcal{P}_\mathrm{n} \cup \mathcal{D}_\mathrm{n}, j \in \mathcal{V}_\mathrm{e}\}, i \neq j\}$. The physical locations of nodes in virtual graph $\mathcal{G}_s$ are mapped as a vector L corresponding to $\mathcal{N}$. Denote d_{ij} as the travel distance between nodes i and j for all $(i,j) \in \mathcal{A}$. Note that since waterborne AGVs stay at their final service berth, the locations for virtual end nodes $\mathcal{V}_\mathrm{o}$ vanish and distance $d_{ij} = 0$ if $i \in \mathcal{P}_\mathrm{n} \cup \mathcal{D}_\mathrm{n} \cup \mathcal{V}_\mathrm{o}, j \in \mathcal{V}_\mathrm{e}$. For duplicated elements in L (same berths), we further cluster the corresponding nodes as set $\mathcal{C}_b = \{i|L_i = b, b \in \mathcal{B}\}$. For the nodes in the same set $\mathcal{C}_b$, if they are visited by different waterborne AGVs, a time interval T is imposed to the service time slots of the waterborne AGVs to keep safety considering waterborne AGV dimensions.

The following decision variables are introduced to solve the scheduling problem:

- Binary variables: x_{ijv} for $(i,j) \in \mathcal{A}$ 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} for $i \in \mathcal{N}$ and $v \in \mathcal{V}$ equals to 1 if node i is visited by waterborne AGV v and 0 otherwise;
- Binary variables: I_{ij} for $i,j \in \mathcal{C}_b,\ b \in \mathcal{B},\ i \neq j$ equals to 1 if node i is visited before node j and 0 otherwise;
- Binary variables: S_{ij} for $i,j \in \mathcal{C}_b,\ b \in \mathcal{B}$ equals to 1 if nodes i,j are visited by different waterborne AGVs and 0 otherwise;
- Integer variables: y_i for $i \in \mathcal{N}$ denotes the load on board the waterborne AGV upon arriving node i;
- Continuous variables: a_i for $i \in \mathcal{N}$ specifies the arrival time at node i;
- Continuous variables: w_i for $i \in \mathcal{N}$ is the waiting time at node i;
- Continuous variables: d_i for $i \in \mathcal{N}$ is the delay time at node i;
- Continuous variables: u_{ij} for $(i,j) \in \mathcal{A}$ is the speed a waterborne AGV travels at on leg $i \to j$.

Additional auxiliary variables for the transformation into an MILP problem will be introduced in Section 3.3.

3.2. Mixed integer programming problem

The overall goal is to compute a set of schedules that minimize the cost of fulfilling all requests in $\mathcal{R}$ according to some cost metrics while satisfying various constraints. For our case, a mixed integer programming problem is formulated as follows:

$$\min \quad c_1 \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{P}_\mathrm{n}\cup\mathcal{D}_\mathrm{n}} \sum_{i\in\mathcal{V}_\mathrm{o}} x_{ijv} + c_2 \sum_{v\in\mathcal{V}} \sum_{(i,j)\in\mathcal{A}} x_{ijv}(m_c y_j + m) d_{ij} + c_3 \sum_{(i,j)\in\mathcal{A}} u_{ij}^2 d_{ij} + c_4 \|A_{\mathcal{V}_e} - A_{\mathcal{V}_o}\|_1 + c_5 \|\mathbf{w}\|_1 + c_6 \|\mathbf{d}\|_1 \tag{1}$$

$$\text{subject to} \quad \sum_{v\in\mathcal{V}} z_{iv} = 1 \quad \forall i \in \mathcal{N}, \tag{2}$$

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

$$z_{ii} = 1, \quad \forall i \in \mathcal{V}_\mathrm{o}, \tag{4}$$

$$\sum_{j\in\mathcal{N}} x_{ijv} = \sum_{j\in\mathcal{N}} x_{jiv} = z_{iv}, \qquad \forall i \in \mathcal{N}, \quad v \in \mathcal{V}, \tag{5}$$

$$\sum_{j\in\mathcal{N}/\mathcal{V}_\mathrm{e}} x_{v_\mathrm{o} j v} = 1, \quad \forall v \in \mathcal{V}, \tag{6}$$

$$\sum_{i\in\mathcal{N}/\mathcal{V}_\mathrm{o}} x_{i v_\mathrm{d} v} = 1, \quad \forall v \in \mathcal{V}, \tag{7}$$

$$a_i \leqslant A_{i+n_r}, \quad \forall i \in \mathcal{P}_\mathrm{n}, \tag{8}$$

$$x_{ijv} = 1 \Rightarrow \max(A_i, t_{i,\min}) + s_i + d_{ij}/u_{ij} = A_j, \qquad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \tag{9}$$

$$t_{i,\min} - w_i \leqslant a_i \leqslant t_{i,\max} - s_i + d_i, \quad \forall i \in \mathcal{N}, \tag{10}$$

$$0 \leqslant w_i \leqslant w_{\max}, \quad \forall i \in \mathcal{N}, \tag{11}$$

$$0 \leqslant d_i \leqslant d_{\max}, \quad \forall i \in \mathcal{N}, \tag{12}$$

$$I_{ij} + I_{ji} = 1, \qquad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \tag{13}$$

$$S_{ij} = 1 - \sum_{v\in\mathcal{V}} z_{iv} z_{jv}, \qquad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \tag{14}$$

$$I_{ij} S_{ij} = 1 \Rightarrow \max(a_i, t_{i,\min}) + s_i + T \leqslant A_j, \qquad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \tag{15}$$

$$y_{v_\mathrm{o}} = y_{v_\mathrm{e}} = 0, \quad \forall v \in \mathcal{V}, \tag{16}$$

$$x_{ijv} = 1 \Rightarrow y_i + q_i = y_j, \qquad \forall i \in \mathcal{N}, \quad v \in \mathcal{V}, \tag{17}$$

$$0 \leqslant y_i \leqslant Q, \quad \forall i \in \mathcal{N}, \tag{18}$$

$$u_{\min} \leqslant u_{ij} \leqslant u_{\max}, \quad \forall (i,j) \in \mathcal{A}, \tag{19}$$

$$x_{ijv}, z_{iv}, I_{ij}, S_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \tag{20}$$

where the objective (1) contains six cost terms that are related to energy efficient schedules for waterborne AGVs. The first term counts the number of waterborne AGVs deployed for the set of requests $\mathcal{R}$. The fleet of deployed waterborne AGVs is not necessarily the same with the fleet of available waterborne AGVs; we always minimize the number of deployed waterborne AGVs considering high fixed deployment cost. Both the second and third terms measure the cost of energy consumption and emissions traveling from node $i \rightarrow j$. The pollution-routing problem (Bektaş and Laporte, 2011) employed similar emission measurement terms. Cost term 2 is incurred due to the weight including waterborne AGV curb weight and the weight of containers on board of the waterborne AGV. Cost term 3 reflects the nonlinear dependence of energy consumption on cruising speed and distance. "Slow steaming" is imposed by minimizing this term if possible. The fourth term considers the total sojourn time of all waterborne AGVs. Departure times from starting locations are also optimized with this formulation. The last two terms account for customer inconvenience measured by waiting and delay times, respectively. The trade-off among these cost penalties is balanced by weight parameters $c_1, c_2, \ldots, c_6$.

Constraint (2) represents that each node is visited exactly by one waterborne AGV. By constraints (3) and (4), we ensure that pick-up and delivery nodes of a particular request are visited by the same waterborne AGV and all waterborne AGVs visit their own starting nodes, respectively. Constraint (5) restricts that a waterborne AGV only enters and leaves a node if it visits that node. Constraints (6) and (7) impose that each waterborne AGV starts and ends at the right locations, respectively. Constraints (8)–(15) together impose time constraints. Specifically, inequality (8) guarantees that pick-up nodes are visited before delivery nodes. Constraint (9) 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 (10)–(12). The coordinated berthing times taking waterborne AGV dimensions and safety distances into consideration are realized with constraints (13)–(15). The logic in (15) implies that if node i, j relate to the same physical location $(i, j \in \mathcal{C}_b)$ and are visited by different waterborne AGVs $(S_{ij} = 1)$ and node i is visited before node j $(I_{ij} = 1)$, that then the arrival time of the waterborne AGV behind should be later than the departure time of the earlier waterborne AGV at least for a time T. This is a novel feature of our waterborne AGV scheduling problem. VRPs and variants have typically assume vehicles as dimensionless mass points without consideration of safety distances. Load consistence and capacity constraints are introduced via (16)–(18). Lastly, cruising speed is bounded by (19) and (20) defines binary variables.

The above mixed integer programming problem (1)–(20) involves several nonlinearities: (a) the multiplication of binary variable x_{ijv} and integer load variable y_j in the second cost term of (1); (b) the quadratic energy function of speed variable u_{ij} in the third cost term of (1); (c) the reciprocal of speed u_{ij} in (9); (d) logic implications in constraints (9), (15) and (17); and (e) the multiplications of binary variables $z_{iv}z_{jv}$ in (14) and $I_{ij}S_{ij}$ in (15). All these nonlinearities bring about even more challenges to finding an optimal solution to the already notorious NP-hard routing problem. We next present transformations of these nonlinearities to obtain an easier to solve MILP problem.

3.3. Transformations into linearity

Linearizations of the above reported nonlinearities rely mainly on two techniques: discretization (Bektaş and Laporte, 2011; Fagerholt et al., 2010), and logic and integer formulations (Williams, 1977). We first deal with the nonlinearities caused by nonlinear functions of speed by discretization.

Generally, two discretization approaches are proposed: discrete speeds as in Bektaş and Laporte (2011) and discrete travel times as in Fagerholt et al. (2010). Essentially, these two approaches are the same since speeds and travel times are related by a constant travel distance. We apply discrete speeds as in Bektaş and Laporte (2011) due to its more intuitive formulation in modeling the "slow steaming" effect. The continuous cruising speed range $[u_{\min}, u_{\max}]$ is discretized by equal intervals $(u_{\max} - u_{\min})/n_u$ into a set of n_u speed levels $[u_{r,\min}, u_{r,\max}]$ for $r = 1, \ldots, n_u$. An average speed is then calculated as $\bar{u}_r = (u_{r,\min} + u_{r,\max})/2$ and assigned to that level. Therefore, the speed optimization in the continuous range $[u_{\min}, u_{\max}]$ becomes the optimal speed selection in the discrete speed set $\{\bar{u}_r | r = 1, \ldots, n_u\}$. Correspondingly, we introduce binary variables b_{ijrv} equal to 1 if waterborne AGV v travels from node $i \rightarrow j$ at speed $\bar{u}_r$. Then, the third cost term, which is a quadratic speed function, is rewritten as:

$$\sum_{(i,j)\in\mathcal{A}} u_{ij}^2 d_{ij} = \sum_{(i,j)\in\mathcal{A}} \left(\sum_{r=1}^{n_u} \bar{u}_r^2 b_{ijrv} \right) d_{ij}. \tag{21}$$

Similarly, the reciprocal speed term in (9) is rewritten as:

$$d_{ij}/u_{ij} = \sum_{r=1}^{n_u} (d_{ij}/\bar{u}_r) b_{ijrv}. \tag{22}$$

Note that different from the VRP in Bektaş and Laporte (2011) with distinct visiting locations, we have duplicated pick-up and delivery locations, which leads to $d_{ij} = 0$ when nodes i, j are actually the same physical berths. Therefore, the relation between b_{ijvv} and x_{ijv} is constrained as:

$$\sum_{r=1}^{n_u} b_{ijrv} = \delta_{ij} x_{ijv}, \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \tag{23}$$

where δ_{ij} is a binary constant equal to 1 if $d_{ij} > 0$ and 0 if d_{ij}. This formulation enforces a zero speed on arc (i,j) if nodes i,j are the same physical berths and a non-zero speed otherwise.

To deal with the nonlinearities due to multiplications of binary and integer variables $x_{ijv}y_j$ in (1), multiplications of binary and binary variables $z_{iv}z_{jv}$ and $I_{ij}S_{ij}$ in (14) and (15), respectively, and logic implications in (9), (15) and (17), the following linearizing approaches based on logic and integer formulations (Williams, 1977) are implemented.

Introduce auxiliary real variables $X_{ijv} = x_{ijv}y_j$, then we are able to replace the nonlinear term in $x_{ijv}y_j$ in (1) by X_{ijv} subject to the following set of linear constraints:

$$\begin{aligned} &X_{ijv} \leqslant Qx_{ijv}, \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \\ &X_{ijv} \geqslant 0, \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \\ &X_{ijv} \leqslant y_j, \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \\ &X_{ijv} \geqslant y_j - Q(1 - x_{ijv}) \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}. \end{aligned} \tag{24}$$

The equivalence is due to the bound on variables y_j as constraints (18).

Slightly different, we replace the nonlinear terms in (14) and (15) with auxiliary binary variables $Z_{ijv} = z_{iv}z_{jv}$ and $Y_{ij} = I_{ij}S_{ij}$ along with the following two sets of linear constraints, respectively:

$$\begin{aligned} &-z_{iv} + Z_{ijv} \leqslant 0, \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \quad v \in \mathcal{V}, \\ &-z_{jv} + Z_{ijv} \leqslant 0, \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \quad v \in \mathcal{V}, \\ &z_{iv} + z_{jv} - Z_{ijv} \leqslant 1, \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \quad v \in \mathcal{V}, \end{aligned} \tag{25}$$

$$\begin{aligned} &-I_{ij} + Y_{ij} \leqslant 0, \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \\ &-S_{ij} + Y_{ij} \leqslant 0, \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \\ &I_{ij} + S_{ij} - Y_{ij} \leqslant 1, \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}. \end{aligned} \tag{26}$$

Finally, logic implications in (9) are transformed as:

$$\begin{aligned} &\max(a_i, t_{i,\min}) + s_i + \sum_{r=1}^{n_u} (d_{ij}/\bar{u}_r) b_{ijrv} \leqslant A_j + M^1_{ij}(1 - x_{ijv}), \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \\ &-\left(\max(a_i, t_{i,\min}) + s_i + \sum_{r=1}^{n_u} (d_{ij}/u_r) b_{ijrv}\right) \leqslant -A_j - m^1_{ij}(1 - x_{ijv}), \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \end{aligned} \tag{27}$$

with $M^1_{ij} = t_{i,\max} + w_{\max} + s_i + d_{ij}/u_{\min} - (t_{i,\min} - w_{\max})$, $m^1_{ij} = t_{i,\min} + s_i - (t_{i,\max} + w_{\max})$, and (17) as:

$$\begin{aligned} &y_i + q_i \leqslant y_j + M^2_{ij}(1 - x_{ijv}), \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \\ &-(y_i + q_i) \leqslant -y_j - m^2_{ij}(1 - x_{ijv}), \quad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \end{aligned} \tag{28}$$

with $M^2_{ij} = Q + q_i, \ m^2_{ij} = q_i - Q$, and (15) simply as:

$$\max(a_i, t_{i,\min}) + s_i + T \leqslant A_j + M^3_{ij}(1 - Y_{ij}), \quad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \tag{29}$$

with $M^3_{ij} = t_{j,\max} + w_{\max} + s_i + T - (t_i - w_{\max})$.

So far, we have transformed the nonlinear mixed integer programming problem (1)–(20) into an MILP problem by replacing nonlinear terms in the cost function and constraints with auxiliary variables and linear constraints as formulated as (21)–(28). Schedules generated by solving the MILP problem are, for each waterborne AGV $v \in \mathcal{V}$, sequences of nodes $\mathcal{N}_v = \{i | i \in \mathcal{N}, z_{iv} = 1, v \in \mathcal{V}\}$ to visit, the corresponding arrival times $\mathcal{A}_v = \{a_i | z_{iv} = 1, v \in \mathcal{V}\}$ in ascending order, load/unloading volumes $\mathcal{Q}_v = \{q_i | z_{iv} = 1, v \in \mathcal{V}\}$, as well as traveling speeds $\mathcal{U}_v = \{u_{ij} | x_{ijv} = 1, (i,j) \in \mathcal{A}, v \in \mathcal{V}\}$ on each leg.

4. Real-time closed-loop scheduling and control

Ideally, a seamless integration of scheduling and control problem requires that all decisions are made simultaneously achieving objectives and satisfying various constraints at both levels, which still remains an open, though important, issue. We have identified in Section 1 that the main technical challenges lie in the different time horizons and different nature of decisions (continuous time and discrete events) which result in a highly complex problem to solve considering current computing power. A descriptive integrated problem **P** could be defined as:

$$\begin{aligned} (\mathbf{u}_s^*, \mathbf{u}_c^*) \;&:=\; \arg\min \; J(J_s, J_c) \\ \text{Subject to} \;\;& \text{Scheduling constraints,} \\ & \text{Waterborne AGV dynamics,} \\ & \text{Disturbance dynamics,} \\ & \text{Control constraints,} \end{aligned} \tag{30}$$

where the total cost J depends on scheduling cost J_s and control cost J_c. The optimal scheduling and control decisions $\mathbf{u}_s^*, \mathbf{u}_c^*$ are simultaneously made by solving problem **P**. Apparently, the decision frequency of problem **P** should be the same as the control problem which has faster decision frequencies. However, the already complex scheduling problem (1)–(29) coupled still with lower level motion and disturbance models will typically not be easily solved to optimality at a high frequency. Even if we decompose problem **P** hierarchically and solve scheduling and control problems sequentially every time a control decision is implemented and new system states are available, solving the scheduling problem at the control frequency could still preclude it from practical applications. We propose an interaction model and a real-time scheduling problem that enable solving scheduling and control problems both hierarchically and real-time in a closed-loop way.

4.1. Modeling interactions and real-time speed assignment

Literature dealing with scheduling or control problems independently can actually be viewed as considering a simplified problem **P** with assumptions. From a scheduling perspective, implicit assumptions are made that schedules are all executed perfectly, i.e., waterborne AGVs depart and arrive at berths following the exact scheduled order and timing irrespective of operational disturbances. This is implicitly achieved by assuming a constant speed and thus constant travel times on all routes if speed is not an optimization variable, e.g., in Pang et al. (2011), or variable speeds as functions of discrete events if a combined route-speed optimization problem is solved, e.g., in Bektaş and Laporte (2011), Fagerholt et al. (2010), and Norstad et al. (2011) as well as our scheduling problem (1)–(29). No operational disturbances and physical system limitations can further be incorporated. Therefore, waterborne AGV motions are simply modeled as a first order integrator with constant speeds on one arc in a combined routing-speed optimization problem, i.e.,

$$s_v(k+1) = s_v(k) + u_{i(i+1)v} T_s, \quad \forall i \in \mathcal{N}_v(k), \quad v \in \mathcal{V}(k), \quad kT_s \in \left[\max\left(a_i(k), t_{i,\min}\right), A_{i+1}(k)\right], \tag{31}$$

where k stands for discrete time for computational implementations relating to continuous time t by $t = kT_s$ and T_s is the sampling time of waterborne AGV systems and s_v is the distance waterborne AGV v has traveled on route i.

We take advantage of the implicit motion model in the scheduling problem and propose an interaction model based on a two-level parameterization of reference paths similar as in Zheng et al. (in press). The lower level is embedded in the online control problem which takes care of the waterborne AGV dynamics, system limitations, operational disturbances, and collision avoidance with other traffic, and modeled as:

$$s_v^{\mathrm{l}}(k+1) = s_v^{\mathrm{l}}(k) + u_v^{\mathrm{l}}(k) T_s, \quad \forall v \in \mathcal{V}(k), \tag{32}$$

where $s_v^{\mathrm{l}}(k), u_v^{\mathrm{l}}(k)$ are a lower level path parameter and its speed, respectively. Both $s_v^{\mathrm{l}}(k)$, $u_v^{\mathrm{l}}(k)$ are decision variables in online control optimization problems, as to be introduced further in Section 4.2. The path parameter determines the reference orthogonal projections of waterborne AGV v onto its current route $(i, i+1)$, $i \in \mathcal{N}_v$ by

$$\begin{aligned} x_p(k) &= s_v^{\mathrm{l}}(k)\sin(\psi_i) + x_i, \\ y_p(k) &= s_v^{\mathrm{l}}(k)\cos(\psi_i) + y_i, \end{aligned} \tag{33}$$

as shown in Fig. 3, with $(x_p(k), y_p(k)), (x_i, y_i)$ the inertial frame coordinates of the reference projection and node i, respectively.

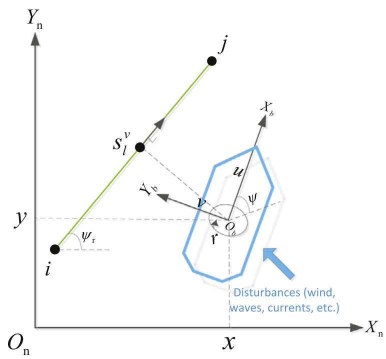

Fig. 3. Waterborne AGV v and projection in route (i,j).

The upper level of the two-level parameterization scheme is a partial scheduling problem of problem (1)–(29) and updates schedules based on the lower level states which in turn reflect waterborne AGV operational details including real-time control performances, delays caused by environmental disturbances or collision avoidance, etc. In this way, the two-level interaction model connects scheduling and control problems but make them decomposable from each other while still allowing both to be solved online using real-time feedback. The upper level problem is formulated as a real-time speed assignment problem as follows.

At each time step k, we collect feedback information for the set of waterborne AGVs $\mathcal{V}(k)$ that are scheduled with tasks but have not arrived at the scheduled last node each with load $y_v^0(k),\ v \in \mathcal{V}(k)$ on board. For each waterborne AGV $v \in \mathcal{V}(k)$, the list of yet to visit nodes are $\mathcal{N}_v(k)$, the corresponding list for time windows $\left\{\left[t_{\min,i}^v, t_{\max,i}^v\right]\right\},\ \forall i \in \mathcal{N}_v(k)$ and the list of service times $\{s_i^v\},\ \forall i \in \mathcal{N}_v(k)$. Besides time window constraints, different waterborne AGVs still need to coordinate their service time slots at a particular berth by guaranteeing a safety time interval among them. Variables related to the real-time speed assignment problem are:

- Binary variables: $I_{ij}(k)$ for $i,j \in \mathcal{C}_b, i \in \mathcal{N}_p, j \in \mathcal{N}_q, p \neq q$ equal to 1 if node i is visited by waterborne AGV p before node j by waterborne AGV q and 0 otherwise;
- Continuous variables: $a_i^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ specifies the arrival time of waterborne AGV v at node i;
- Continuous variables: $w_i^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ is the waiting time of waterborne AGV v at node i;
- Continuous variables: $d_i^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ is the delay time of waterborne AGV v at node i;
- Continuous variables: $u_{i(i+1)}^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ is the speed waterborne AGV v travels at on leg $i \to i+1$.

The overall goal is to compute schedules that still minimize the overall cost of fulfilling all remaining requests while satisfying time window and coordinated berthing constraints. The mixed integer programming problem is formulated as:

$$\min \quad c_3 \sum_{(i,j)\in\mathcal{A}} u_{ij}^2(k) d_{ij}(k) + c_4 \|A_{\mathcal{V}_e}(k) - A_{\mathcal{V}_o}(k)\|_1 + c_5 \|\mathbf{w}(k)\|_1 + c_6 \|\mathbf{d}(k)\|_1 \tag{34}$$

$$\text{subject to} \quad \max\left(a_i(k), t_{i,\min}^v\right) + s_i^v + d_{i(i+1)}(k)/u_{i(i+1)}(k) = A_{i+1}, \qquad \forall i \in \mathcal{N}_v(k), \quad v \in \mathcal{V}(k), \tag{35}$$

$$t_{i,\min}^v - w_i^v(k) \leqslant a_i(k) \leqslant t_{i,\max}^v - s_i^v + d_i^v(k), \qquad \forall i \in \mathcal{N}_v(k), \quad v \in \mathcal{V}(k), \tag{36}$$

$$0 \leqslant w_i^v(k) \leqslant w_{\max}, \qquad \forall i \in \mathcal{N}_v(k), \quad v \in \mathcal{V}(k), \tag{37}$$

$$0 \leqslant d_i^v(k) \leqslant d_{\max}, \qquad \forall i \in \mathcal{N}_v(k), \quad v \in \mathcal{V}(k), \tag{38}$$

$$I_{ij}(k) + I_{ji}(k) = 1, \qquad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \quad i \in \mathcal{N}_p, \quad j \in \mathcal{N}_q, \quad p \neq q, \tag{39}$$

$$I_{ij}(k) = 1 \Rightarrow \max\left(a_i, t_{i,\min}^v\right)(k) + s_i^v(k) + T \leqslant A_j, \qquad \forall i,j \in \mathcal{C}_b, \quad b \in \mathcal{B}, \quad i \in \mathcal{N}_p, \quad j \in \mathcal{N}_q, \quad p \neq q, \tag{40}$$

$$\underline{u} \leqslant u_{i(i+1)}^v(k) \leqslant \bar{u}, \qquad \forall i \in \mathcal{N}_v, \quad v \in \mathcal{V}(k), \tag{41}$$

$$I_{ij}(k) \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, \quad v \in \mathcal{V}, \tag{42}$$

where objective (34) contains four terms that are the same as the last four terms in (1), minimizing energy consumption due to variable speed, total sojourn times, waiting times, and delay times, respectively. Constraints (35) indicates the time consistency between two successive nodes in the node list of each waterborne AGV $v \in \mathcal{V}(k)$. Soft time windows are imposed by constraints (36) with constraints (37) and (38) specifying the maximum waiting and delay times, respectively. Constraints (39) and (40) together formulate the coordinated berthing times between different waterborne AGV visiting the same berth. Variable speeds are bounded by constraint (41) and constraint (42) define the only binary variable in this problem. By solving this problem, we obtain updated schedules $\{\mathcal{N}_v(k), \mathcal{A}_v(k), \mathcal{Q}_v(k), \mathcal{U}_v(k)\}$ for waterborne AGVs $v \in \mathcal{V}(k)$ as well as parameterized reference paths at the upper level, defined as:

$$s_v^{\mathrm{u}}(k+1) = s_v^{\mathrm{u}}(k) + u_{i(i+1)}(k) T_s, \qquad \forall v \in \mathcal{V}(k), \quad i \in \mathcal{N}_v(k), \tag{43}$$

where $s_v^{\mathrm{u}}(k), u_{i(i+1)}(k)$ are references for $s_v^{\mathrm{l}}(k), u_v^{\mathrm{l}}(k)$, respectively.

4.2. Closing the real-time loop

With the interaction model, we are now ready to decompose problem **P** hierarchically into a scheduling problem **Ps**

$$\begin{aligned} &\mathbf{u}_s^* := \arg\min \ J_s \\ &\text{Subject to} \quad \text{Scheduling constraints}, \end{aligned} \tag{44}$$

and a control problem **Pc**

$$\begin{aligned} \mathbf{u}_c^* &:= \arg\min \ J_c \\ \text{Subject to} \quad & \text{Waterborne AGV dynamics,} \\ & \text{Disturbance dynamics,} \\ & \text{Control constraints.} \end{aligned} \tag{45}$$

In particular, at each control time step k, we first solve scheduling problem **Ps** based on updated feedback information: the set of waterborne AGVs $\mathcal{V}(k)$ that are scheduled with tasks but have not arrived at the scheduled last node, each with load $y_v^0(k),\ v \in \mathcal{V}(k)$ on board, projected waterborne AGV positions $\left(x_p^v(k), y_p^v(k)\right) \forall v \in \mathcal{V}(k)$ that are determined by measured waterborne AGV positions $x_v k, y_v(k) \forall v \in \mathcal{V}(k)$. The projected positions are utilized to (1) update the Euclidean distances $d_{i(i+1)}(k) \forall i \in \mathcal{N}_v(k),\ v \in \mathcal{V}(k)$; and (2) initialize both levels in the interaction model by $s_v^{\mathrm{l}}(k) = s_v^{\mathrm{u}}(k) = \left\| \begin{matrix} x_p^v(k) - x_i \\ y_p^v(k) - y_i \end{matrix} \right\|_2$. Note that only at the beginning of the planning horizon, problem (1)–(29) needs to be solved as the scheduling problem **Ps**. Real-time scheduling is achieved by solving problem (34)–(42) which is efficient to solve to optimality.

At each time step k, the control problem **Pc** is solved sequentially after receiving references from scheduling problem **Ps**. The overall control goals are to (1) execute schedules to fulfill ITT requests in an economical way; (2) maintain safety by satisfying system physical limitations and avoiding collisions with other traffic in the presence of disturbances; and (3) maneuver in a distributed way. We present the formulation of J_c here which involves the lower level of the interaction model in the closed-loop scheduling and control:

$$J_c(k) = \sum_{v \in \mathcal{V}(k)} \left(c_7 \|\boldsymbol{\eta}(k) - \boldsymbol{\eta}_{\mathrm{r}}(k)\|_2^2 + c_8 \left\| s_v^{\mathrm{l}}(k) - s_v^{\mathrm{u}}(k) \right\|_2^2 + c_9/2 \|\boldsymbol{v}(k)\|_{\boldsymbol{M}(k)}^2 \right), \tag{46}$$

where $\eta = [x \quad y \quad \psi]^{\mathrm{T}},\ \boldsymbol{v} = [u \quad v \quad r]^{\mathrm{T}}$ are the pose (inertial frame coordinates and heading angle) and velocity (surge speed, sway speed and yaw rate) states, respectively. Reference pose η_{r} is determined by (32) and (33) as $\eta_{\mathrm{r}} = \left[x_p \quad y_p \quad \psi_i\right]^{\mathrm{T}}$. Control performance is also affected by scheduling results reflected in the mass matrix as:

$$\boldsymbol{M}(k) = \begin{bmatrix} m + m_c y_v^0(k) & 0 & 0 \\ 0 & m + m_c y_v^0(k) & m + m_c y_v^0(k) x_g \\ 0 & m + m_c y_v^0(k) x_g & I_z \end{bmatrix} + \boldsymbol{M}_{\mathrm{A}}, \tag{47}$$

where x_g is the distance between waterborne AGV's center of gravity and the origin of the body-fixed coordinate frame and I_z is the moment of inertia in the yaw rotation; $\boldsymbol{M}_{\mathrm{A}} \in \mathcal{R}^{3\times 3}$ is hydrodynamic added mass (Zheng et al., 2015). The first goal of executing schedules in an energy efficient way is then achieved by minimizing $J_c(k)$. For the detailed modeling and algorithms for a control problem achieving three goals together with $J_c(k)$, we refer to Zheng et al. (2015). We focus in this paper on the closed-loop schedule and control of waterborne AGVs as shown in Fig. 4.

5. Simulation experiments and discussion

Simulations are run to illustrate the effectiveness of the proposed closed-loop scheduling and control of waterborne AGVs for ITT. The ITT scenario is built based on Fig. 2 with six berths and a fleet of three waterborne AGVs. Assume that there are seven ITT requests arising within the scheduling horizon 0–2100 s as detailed in Table 1 of which the available request infor-

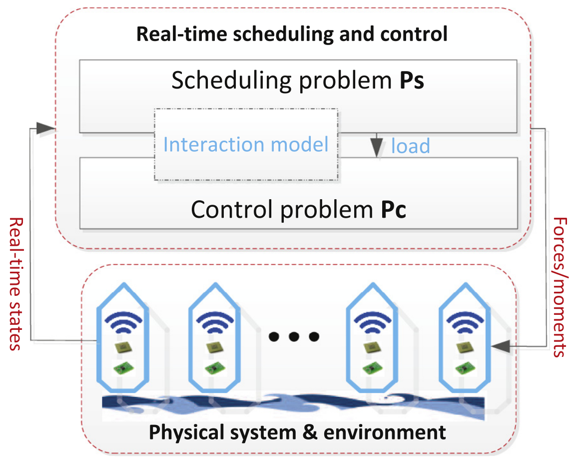

Fig. 4. Closed-loop scheduling and control of energy-efficient waterborne AGVs.

Table 1
ITT requests to be executed.

Request ID	Origin	Destination	Release time (s)	Due time (s)	Volume (TEU)
1	5	2	125	1865	2
2	1	3	690	1155	2
3	1	4	700	1485	1
4	6	2	725	1535	2
5	6	1	1230	1755	2
6	2	3	1345	1750	2
7	1	4	1640	2085	1

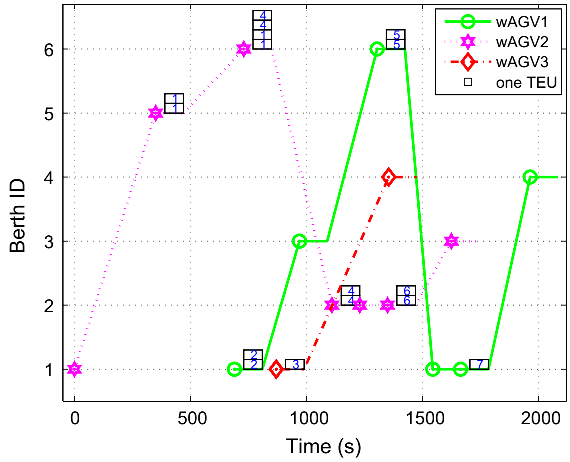

Fig. 5. Initial energy efficient schedules.

mation structure has been designed according to Schroër et al. (2014). Note that in practice, the fleet of waterborne AGVs could be larger and more ITT requests could arise in the port area. Scenarios set here are assumed as in Section 2, i.e., the number of waterborne AGVs is sufficient and requests in different planning horizons are decoupled so that small sets of requests can be considered independently. Positions of the six berths are determined in latitude/longitude and subsequently converted to coordinates in inertial frame with Berth 1 as the origin. The fleet of available waterborne AGVs for this set of ITT requests are initially all positioned at Berth 1.

Waterborne AGVs are designed to sail in the speed range of $u_{\min} = 2.57$ m/s, $u_{\max} = 6.68$ m/s which are corresponding to 5 knots and 13 knots, respectively. Each waterborne AGV can carry a maximum of four TEUs, i.e., $Q = 4$, and each TEU of container weighs $m_c = 24,000$ kg. Each move of a quay crane can load/unload one or two TEUs and requires 120 s (Tierney et al., 2014). Therefore, for all ITT requests in Table 1, service times are the same as $t_s = 120$ s. The necessary safety time interval between different waterborne AGVs visiting the same berth is set to $T = 60$ s based waterborne AGV lengths and sailing speeds. Other parameters concerned with waterborne AGV dynamics are implemented as in Zheng et al. (in press). The weight parameters in cost functions (1), (34) and (46) for trade-offs of different performance metrics are set as: $c_1 = 10^4, c_2 = 10^{-2}, c_3 = 10^2, c_4 = 10^3, c_5 = 10^8, c_6 = 10^8, c_7 = 100, c_8 = 100, c_9 = 1$. Algorithms are implemented in MATLAB 2011b (MATLAB, 2011). Optimization problems are solved by Cplex (ILOG, 2016). All the simulations are run on a platform with Intel (R) Core (TM) i5-3470 CPU @3.20 GHz.

The closed-loop schedule and control algorithm as shown in Fig. 4 needs to replace human operators to make "smart" decisions: (1) for the fleet of waterborne AGVs, energy efficient schedules as well as actuator inputs to execute these schedules should be autonomously generated satisfying waterborne AGV physical limitations, e.g., maximum capacity, rudder force range, etc. and guaranteeing safety; (2) for the list of ITT requests, certain amount of containers should be transported from specified origins to destinations after the release time while before the due time; and (3) each berth can accommodate at most one waterborne AGV, and service time slots of different waterborne AGVs should keep a buffer time interval. Simulation results from these three perspectives are presented next to demonstrate the effectiveness of the proposed algorithm.

5.1. From the waterborne AGV perspective

The set of seven ITT requests as shown in Table 1 calls for all three waterborne AGVs of which initial schedules by solving problem (1)–(28) are shown in Fig. 5 as green-circle[2] line, magenta-hexagram dotted line, and green-square dashed line,

[2] For interpretation of color in 'Figs. 5 and 11', the reader is referred to the web version of this article.

respectively. All three waterborne AGVs start from Berth 1 but at different times. The small rectangles are one TEU containers and the numbers attached identify IDs of requests that the containers belong to. We display the set and mix of containers on board waterborne AGVs departing berths. Each schedule contains information on the sequence of berths to visit, the corresponding arrival and departure times as well as the load/unload operations at each berth. Take the schedule of waterborne AGV 2 as an example, we place a hexagram upon waterborne AGV 2's arrival at a berth. There are three hexagrams at Berth 2 between 1000 s and 1500 s because waterborne AGV 2 performs three load/unload operations at Berth 2. From the set and mix of containers on board when departing, we can derive that waterborne AGV 2 first unloads the two containers from request 1 taking 120 s, then unloads the two containers from request 4 taking another 120 s and finally loads the two containers from request 6 before departing from Berth 2 to Berth 3 which is its final destination.

Travel speeds along all route segments are also explicitly optimized. In fact, with berth IDs, arrival and departure times known, travel speeds can easily be derived from 5. The travel speed profile for waterborne AGV 2 along its route is shown as Fig. 6. As can also be observed in Fig. 5, all three waterborne AGVs carry no more than four TEU containers. Fig. 7 further shows the total number of containers on board throughout the simulation which are all within the maximum capacity of four TEU containers.

Waterborne AGVs receiving schedules as shown in Fig. 5 are then controlled with a first goal of guaranteeing operational safety and a secondary goal of executing those schedules. Since complex system dynamics, physical limitations, disturbances and collision avoidance between moving waterborne AGVs are not considered in the scheduling problem, real-time waterborne AGVs do not necessarily behave safely and as scheduled: following the scheduled route at specified speed and arriving at scheduled berths at specified times. Fig. 8 shows the evolution of velocities of three scheduled waterborne AGVs, respectively. Velocities in three degrees of freedom: surge, sway and yaw are all within safe maneuvering ranges as the red lines show. Surge velocity is a function of time and sees fluctuations which is different with piecewise constant speeds determined in scheduling problems which are functions of events as shown in Fig. 6. This is due to the necessary accelerations and decelerations when operating in real environment. Sway velocity and yaw rate are not considered in scheduling problems at all.

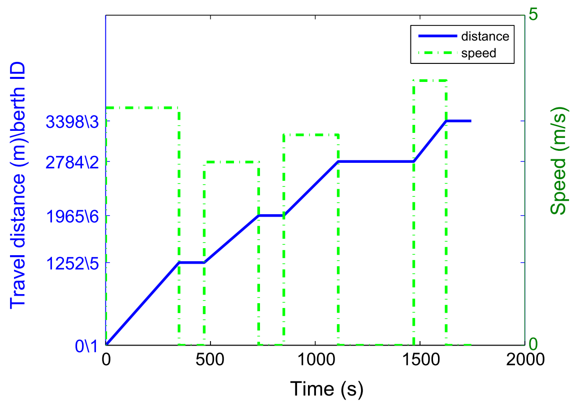

Fig. 6. Travel speed profile of waterborne AGV 2.

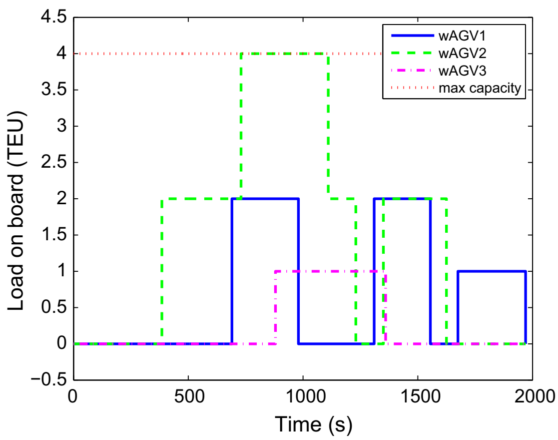

Fig. 7. Containers on board of the waterborne AGVs.

Likewise, control inputs in surge, sway and yaw interact with complex system dynamics and environment to achieve control goals and are all within the safety limitations as shown in Fig. 9.

Waterborne AGVs are treated equally as agents that are controlled in a distributed way and make control decisions parallelly using a fast distributed control method proposed in Zheng et al. (2015). During the execution of assigned requests, waterborne AGVs might encounter conflicts with each other. The distributed control algorithm guarantees ITT request fulfillment locally for each waterborne AGV while achieves an overall minimal cost and safety for all waterborne AGVs. The route following performance with small tracking errors of waterborne AGV 3 is demonstrated as in the top subplot of Fig. 10. Fluctuations are seen at the beginning and around 1200 s due to the start up and encountering with waterborne AGV 2 and 1. However, a safety distance away from them is ensured by the control level when waterborne AGVs are in close range. The bottom subplot of Fig. 10 shows the distances between waterborne AGV 2 and the other two waterborne AGVs which are all above the minimum safety distances. More details on waterborne AGV tracking performance and collision avoidance behaviors can be found in the movie attached.

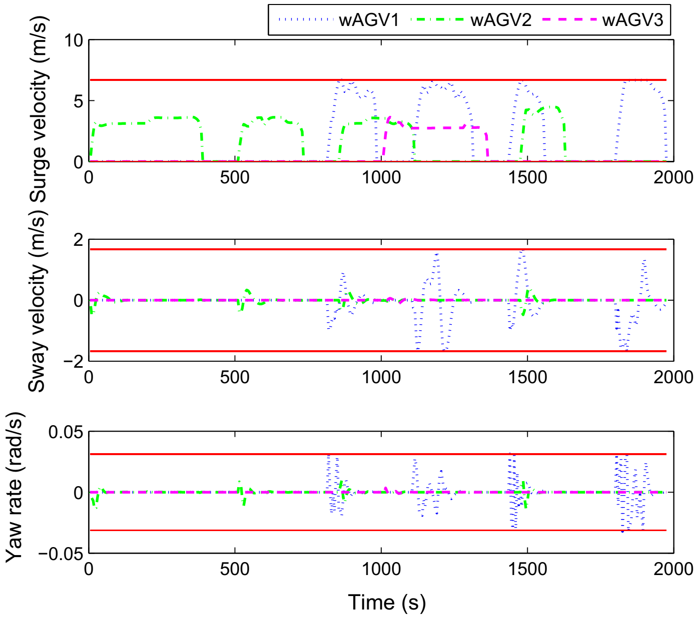

Fig. 8. Velocity evolutions in surge, sway and yaw.

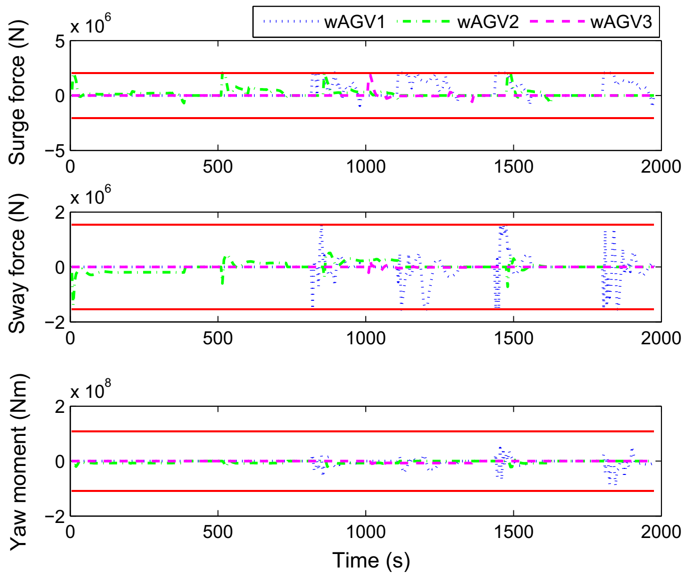

Fig. 9. Control input evolutions in surge, sway and yaw.

5.2. From the ITT request perspective

Assignment of requests in Table 1 to three waterborne AGVs can be derived from the request IDs attached with containers in Fig. 5. This assignment is produced being aware of constraints as waterborne AGV capacities and request time windows as well as travel cost specified in (1). For each request, Fig. 11 shows the time window (red bar) specified by the release and due times of the request, the initial scheduled duration time (yellow bar) solving problem (1)–(28) and the actual duration time (green bar) specified by the waterborne AGV's arrival at the origin berth and departure from the destination berth. All the scheduled and actual duration times are within required time windows, i.e., time windows of all requests are satisfied by the scheduling problem and the control problem succeeds in operating waterborne AGVs with timing window awareness except for request 3 with some delays. Some green bars, e.g., request 1, however, are not within the corresponding yellow bars. This is due to the real-time update of schedules by solving closed-loop scheduling problem (34)–(42). The updated scheduled duration times are not necessarily the same with the initial schedules, but still are guaranteed to satisfy time window constraints as (36). The satisfaction of time windows by actual duration times proves this.

For comparison, Fig. 12 shows the request completion times of schedules executed in open-loop, i.e., the initial schedule by solving (1)–(28) is not updated. Some requests with relatively tight time windows, e.g, request 6 and 7, see delays. This is

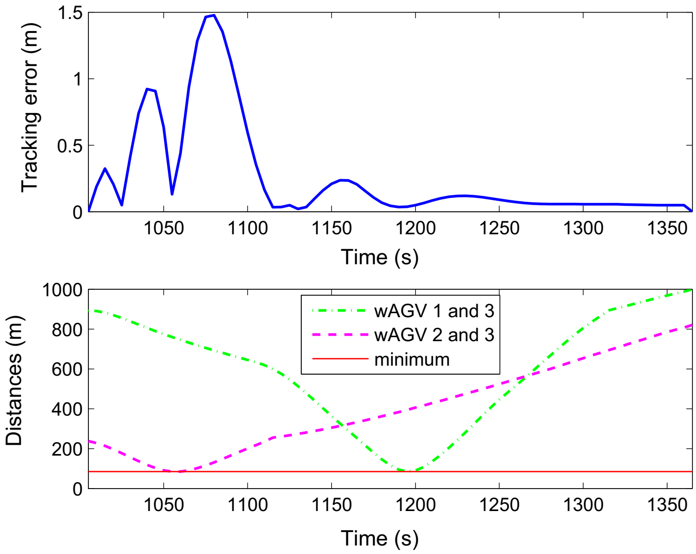

Fig. 10. Tracking performance and collision avoidance.

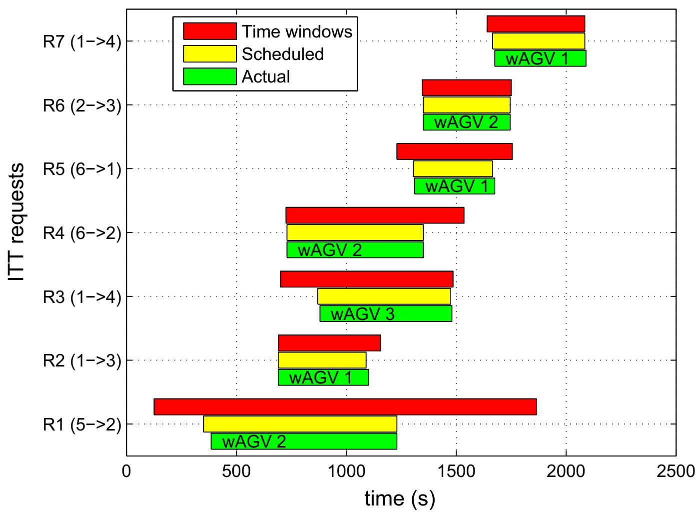

Fig. 11. Satisfaction of request time windows in closed-loop.

because lower level system details which might cause inaccurate execution of schedules are neither considered in scheduling problem (1)–(28) nor reflected in updated schedules in solving real-time problem (34)–(42). The inaccuracies accumulate along routes and lead to delays. Waterborne AGV 1, for example, serves request 2, 5, and 7 in sequence (see Fig. 5). Delays in finishing request 2 lead to later start of request 5 compared to the scheduled start time, and give rise to the violation of the tight time window of request 7 in the end. Since "non-performance", which happens when there are containers that are delivered with delays, is the most important criterion for ITT, we define the "non-performance" rate as the percentage of delayed number (in TEUs) of ITT containers with respect to the total number of ITT containers. Therefore, for the seven ITT requests with 12 TEUs in Table 1, the closed-loop approach has a "non-performance" rate of 0% while the open-loop approach is 41.67%. This illustrates the advantage of closed-loop over open-loop.

5.3. From the berth perspective

Each of the six waterborne AGV berths as shown in Fig. 2 is designed to handle maximum one waterborne AGV one time and there should be a time interval between different waterborne AGV's visits to the same berth for berth practice and guar-

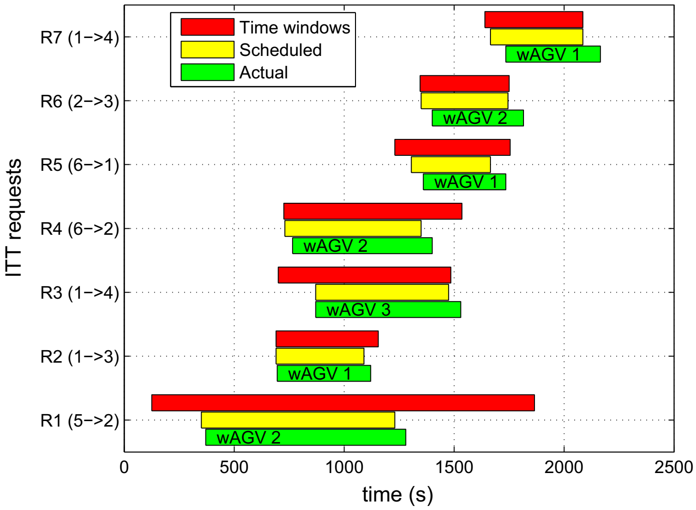

Fig. 12. Requests executed in open-loop.

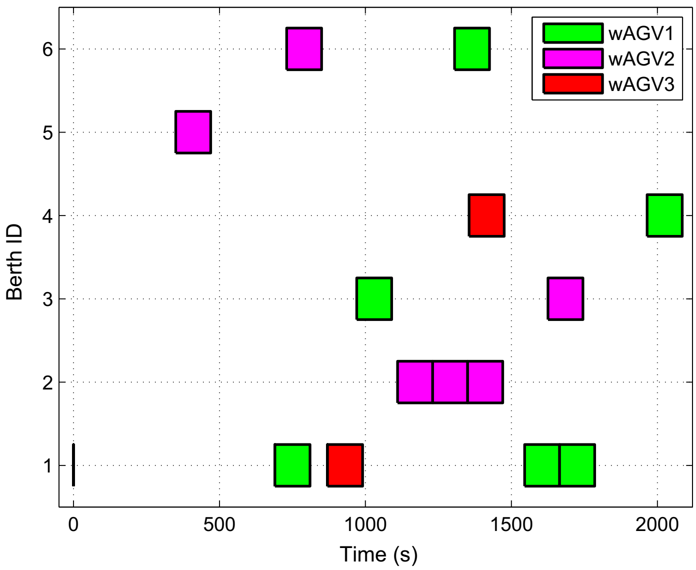

Fig. 13. Berthing time slots from initial schedule.

anteeing safety considering dimensional waterborne AGVs. In other words, the departure time of one waterborne AGV should be at least earlier the specified time interval $T = 60$ s than the arrival time of the next waterborne AGV as constrained by (13) and (14). From the berth perspective, Figs. 13–15 illustrate berth occupations over time as initially scheduled, real-time scheduled at $t = 750$ s, and the actually executed by waterborne AGVs, respectively.

For all berths, there are intervals between bars in different colors along one horizontal line representing different waterborne AGVs visiting the same berth in all three figures. Particularly in Fig. 15 the for actual berthing time slots, Berths 1, 3, 4, and 6 see visits from different waterborne AGVs. The minimum actual time interval is 60 s $\geqslant T$ (minimum safety time interval) which happens between waterborne AGVs 1 and 3 at Berth 1. For one waterborne AGV performing more than one load/unload operation at one berth, however, there is no time interval. This is shown by the bars in same colors linked together, e.g., the three magenta bars at Berth 2 in Fig. 13. Actually, waterborne AGV 2 performs three load/unload operations at Berth 2 if we recall Fig. 5 as analyzed in Section 5.1. Note that the vertical lines at Berth 1 in Figs. 13 and 15 indicates that waterborne AGV 2 travels from Berth 1 to Berth 5 directly without any load/unload operations at Berth 1. Lines instead of bars, i.e., no load/unload oper-

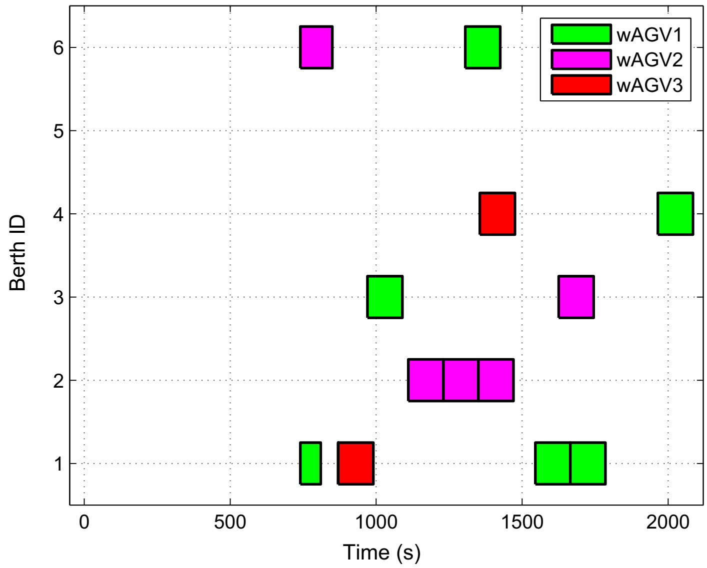

Fig. 14. Berthing time slots from schedule at $t = 750$ s.

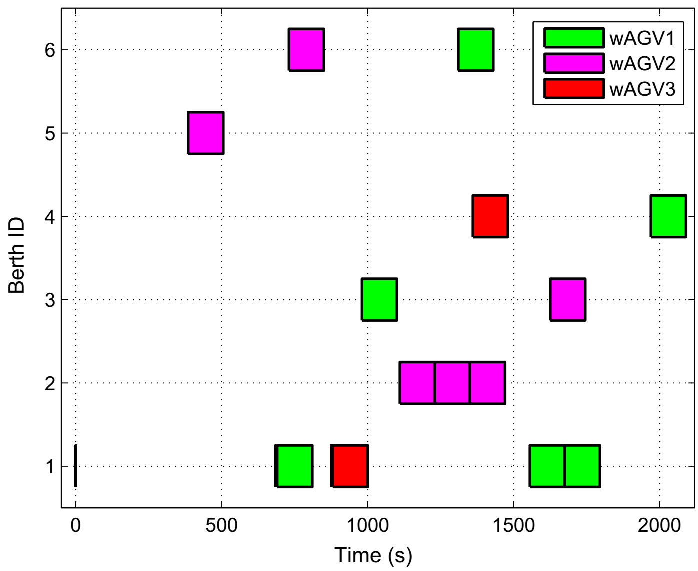

Fig. 15. Actual berthing time slots.

ations, may only arise at waterborne AGVs' initial positions (Berth 1 in our scenario) since any other berths waterborne AGVs to visit are involved an ITT request with certain amount of containers to load/unload and thus requires certain service times. Principally, all time slots (bars) should be not shorter than $t_s = 120$ s. The shorter time slots in Fig. 14, e.g., as shown by the relatively shorter green bar at Berth 1 and the first magenta bar at Berth 6, are because certain amount of service has been conducted at $t = 750$ s. A bar longer than $t_s = 120$ s simply means waterborne AGV waits for some time before the load service can start (release time).

6. Conclusions and future research

In this paper we have proposed a real-time closed-loop scheduling and control scheme for waterborne AGVs applied to Inter Terminal Transport (ITT). The contributions are twofold. Firstly, we propose a new pick-up and delivery scheduling model for ITT using waterborne AGVs by considering safety time intervals between their service time slots at one berth. For all the berths, safety time intervals are guaranteed for different waterborne AGVs. Secondly, we propose a partial scheduling problem that is efficient to solve. By integrating this partial scheduling problem with the control problem of multiple waterborne AGVs, we realize real-time closed-loop scheduling and control of an autonomous ITT system. In our simulation experiments based on a potential ITT scenario in the port of Rotterdam, time windows of all ITT requests are satisfied by the closed-loop approach with 0% "non-performance" rate compared to 41.67% from the open-loop approach. The proposed algorithm provides an effective way realizing autonomous ITT systems using waterborne AGVs.

The port of Rotterdam has seen ample pioneering applications of land-side AGVs (e.g., at ECT and APMT terminals). For waterborne AGVs, small autonomous ITT systems for busy container terminals could first be built and tested. Strategic decisions regarding cost-benefit analysis, location, infrastructure, etc. for more complex networks still need to be made. Wide applications to other ports call for more technological, methodological, and constitutional advances. Therefore, experimental tests are considered as one of our future research work. Moreover, land-side AGVs, which are mainly operated inside container terminals, if designed free-range, could also benefit from the proposed closed-loop scheduling and control algorithm. The future also sees the possibility of applying such an algorithm to a hybrid waterborne and land-side AGV or even amphibious vehicle system. Following the methodological aspects of this paper, future research will also consider ITT scheduling models that can handle dynamically arriving ITT requests within the planning horizon as well as corresponding efficient solution approaches.

Acknowledgements

This research is supported by China Scholarship Council under Grant 201206950021.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.tre.2016.07.010.

References

Port of Rotterdam Authority, December 2011. Port Vision 2030. <http://www.portofrotterdam.com/en/Port/port-in-general/port-vision-2030/Pages/default.aspx> (accessed: 2016-01-17).

Tierney, K., Voß, S., Stahlbock, R., 2014. A mathematical model of inter-terminal transportation. Eur. J. Oper. Res. 235 (2), 448–460.

Zheng, H., Negenborn, R.R., Lodewijks, G., 2015. Predictive path following with arrival time awareness for waterborne AGVs. Transport. Res. Part C: Emerg. Technol. (in press). http://dx.doi.org/10.1016/j.trc.2015.11.004.

Zheng, H., Negenborn, R.R., Lodewijks, G., 2015. Cooperative distributed collision avoidance based on ADMM for waterborne AGVs. In: Corman, F., Voß, S., Negenborn, R.R. (Eds.), Computational Logistics. Springer, pp. 181–194.

Xin, J., Negenborn, R.R., Lodewijks, G., 2014. Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. Transport. Res. Part C: Emerg. Technol. 44, 214–230.

Erasmus Smart Port Rotterdam, February 2015. SmartPort/Port Research Center Rotterdam Delft Poster Session 2014. <http://www.erim.eur.nl/erim-centres/smart-port/news/detail/3585-smartportport-research-center-rotterdam-delft-poster-session-20131-posters-were-presented-in/> (accessed: April, 2016).

Li, S., Negenborn, R.R., Lodewijks, G., 2016. Distributed constraint optimization for addressing vessel rotation planning problems. Eng. Appl. Artif. Intell. 48, 159–172.

Angeli, D., 2014. Economic model predictive control. In: Baillieul, J., Samad, T. (Eds.), Encyclopedia of Systems and Control. Springer, London, pp. 1–9.

Baldea, M., Harjunkoski, I., 2014. Integrated production scheduling and process control: a systematic review. Comput. Chem. Eng. 71, 377–390.

Du, J., Park, J., Harjunkoski, I., Baldea, M., 2015. A time scale-bridging approach for integrating production scheduling and process control. Comput. Chem. Eng. 79, 59–69.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2007. Maritime transportation. In: Barnhart, C., Laporte, G. (Eds.), Handbooks in Operations Research and Management Science: Transportation, vol. 14. Elsevier, Amsterdam, pp. 189–284.

Savelsbergh, M.W., Sol, M., 1995. The general pickup and delivery problem. Transport. Sci. 29 (1), 17–29.

Toth, P., Vigo, D., 2014, . Vehicle Routing: Problems, Methods, and Applications, second ed., vol. 18. SIAM, Philadelphia.

Figliozzi, M.A., 2012. The time dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics. Transport. Res. Part E: Logist. Transport. Rev. 48 (3), 616–636.

Psaraftis, H.N., Wen, M., Kontovas, C.A., 2016. Dynamic vehicle routing problems: three decades and counting. Networks 67 (1), 3–31.

Bektaş, T., Laporte, G., 2011. The pollution-routing problem. Transport. Res. Part B: Methodol. 45 (8), 1232–1250.

Maersk, 2013. Big Ship Means Big Pollution, Right? <http://www.maersk.com/en/test-section/wls-test/stories/big-ship-means-big-pollution-right> (Online; accessed 17-Jan-2016).
Maloni, M., Paul, J.A., Gligor, D.M., 2013. Slow steaming impacts on ocean carriers and shippers. Marit. Econ. Logist. 15 (2), 151–171.
Psaraftis, H.N., Kontovas, C.A., 2014. Ship speed optimization: concepts, models and combined speed-routing scenarios. Transport. Res. Part C: Emerg. Technol. 44, 52–69.
Fagerholt, K., Laporte, G., Norstad, I., 2010. Reducing fuel emissions by optimizing speed on shipping routes. J. Oper. Res. Soc. 61 (3), 523–529.
Norstad, I., Fagerholt, K., Laporte, G., 2011. Tramp ship routing and scheduling with speed optimization. Transport. Res. Part C: Emerg. Technol. 19 (5), 853–865.
Pang, K.-W., Xu, Z., Li, C.-L., 2011. Ship routing problem with berthing time clash avoidance constraints. Int. J. Prod. Econ. 131 (2), 752–762.
Williams, H.P., 1977. Logical problems and integer programming. Bull. Inst. Math. Appl. 13, 18–20.
Schroër, H.J., Corman, F., Duinkerken, M.B., Negenborn, R.R., Lodewijks, G., 2014. Evaluation of inter terminal transport configurations at Rotterdam Maasvlakte using discrete event simulation. In: Proceedings of Winter Simulation Conference. IEEE Press, Savannah, US, pp. 1771–1782.
MATLAB, 2011. Version 7.13 (R2011b). The MathWorks Inc., Natick, Massachusetts.
ILOG, 2016. IBM ILOG CPLEX Optimizer. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/> (Online; accessed 07-Jan-2016).