Dynamic Speed Optimization and Berth Reallocation for Autonomous Vessels Under Sailing Time Disturbances

Tingting Chen, Lingxiao Wu, Shuaian Wang

***Abstract*—Autonomous vessels (AVs) have attracted growing attention due to their potential advantages in operational efficiency and navigational safety. However, their voyages may be affected by stochastic disturbances, which can lead to delayed arrivals at ports and the unavailability of pre-assigned berths. This paper first proposes a dynamic optimization approach for AV speed optimization and berth reallocation to mitigate the impacts of stochastic disturbances. Specifically, the sailing speeds of AVs are dynamically adjusted if stochastic disturbances affect their expected arrival times. Meanwhile, the real-time berth reallocation for AVs is performed when their originally allocated berths become unavailable. To meet real-time operational requirements, a rolling horizon framework is employed, which supports dynamic and adaptive adjustments to sailing speeds and berth reallocation based on the latest information on stochastic disturbances and berth occupancy. In each decision period, the problem is formulated as a mixed integer nonlinear programming model to minimize the total cost. To solve the proposed model efficiently, a tailored branch-and-cut algorithm incorporating an outer approximation method is developed. To evaluate the performance and effectiveness of the proposed model and solution method, extensive numerical experiments based on the operational data of a maritime logistics company were conducted. The results demonstrate that the proposed algorithm significantly outperforms both the Gurobi solver and a "first-come-first-served" greedy algorithm in terms of solution quality. Sensitivity analyses revealed that greater sailing time disturbances and lower penalty costs for arrival delays tend to reduce the punctuality of AVs at ports. Moreover, higher fuel prices prompt AVs to adopt lower sailing speeds to reduce energy costs.**

***Index Terms*—Maritime transportation, Autonomous vessel, Dynamic speed optimization, Berth reallocation, Branch-and-cut algorithm.**

I. Introduction

Maritime transportation is responsible for over 80% of global trade [1]. However, conventional manned vessels (MVs) have encountered significant challenges, such as the shortage of seafaring personnel, vessel emissions, and maritime accidents caused by human errors [2]. Autonomous vessels (AVs) have emerged as a promising solution to address these issues and revolutionize the global shipping industry [3]. Currently, there are over one thousand AVs operating worldwide [4]. For example, China's autonomous container vessel, "ZHI FEI", is actively engaged in commercial operations between Port of Qingdao with Port of Dongjiakou [5].

In practice, vessel speeds are typically optimized by liner shipping companies, while decisions regarding berth allocation are usually made by port operators. However, when liner shipping companies or port operators do their planning independently, higher fuel consumption costs for liner shipping companies and inefficient operations for port operators will occur [6]. To address this issue, collaboration between liner shipping companies and port operators has been recognized as a promising solution. Recent studies have investigated collaboration mechanisms in the maritime industry and analyzed their feasibility, such as [6], [7], and [8]. The collaboration can be realized in two primary scenarios: a unified entity managing both vessels and ports or through a third-party coordinator. A practical example of the first form is Maersk, which owns both the liner shipping company Maersk Line and the terminal operator APM Terminals [9], [10].

Under the collaboration situation, it is essential for planners (i.e., both liner shipping companies and port operators) to simultaneously optimize vessel speeds at legs and berth allocations at ports. In practice, at the operational stage, vessels may encounter stochastic disturbances, such as sailing time disturbance induced by variations of wave heights and wind directions [11]. These disturbances may prevent vessels from adhering to the schedule designed at the planning stage. If a vessel arrives at a port later than its scheduled arrival time due to such disturbances, its initial allocated berth may no longer be available. In such situations, berth reallocation becomes necessary to ensure vessels to successful moor at berths.

In practice, MVs are typically operated by experienced captains, who can actively respond to stochastic disturbances by adjusting sailing speeds and trim settings during voyages [12]. These adjustments help mitigate fluctuations in sailing times, enabling MVs to maintain their scheduled arrival times at ports. In contrast, due to the absence of onboard captain intervention, AVs rely heavily on the

Tingting Chen is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: tingting-tina.chen@connect.polyu.hk)

Lingxiao Wu is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: lingxiao-leo.wu@polyu.edu.hk)

Shuaian Wang is with the Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: wangshuaian@gmail.com)

pre-specified speed control system [3]. Without explicitly accounting for stochastic disturbances in the speed control system, AVs may struggle to adhere to their predetermined schedule, leading to greater variations in sailing times and a higher likelihood of late arrivals at ports compared to MVs. Therefore, it is crucial to consider the impacts of stochastic disturbances when optimizing the sailing speeds of AVs.

In light of the above, this study proposes a dynamic AV speed optimization and berth reallocation approach to mitigate the impacts of stochastic disturbances on the predetermined schedule. Specifically, when stochastic disturbances affect expected arrival times of AVs, their sailing speeds at each leg are dynamically adjusted to reduce arrival delays. Additionally, when originally allocated berths become unavailable, real-time berth reallocation is performed to ensure the successful mooring of AVs. To meet real-time operational requirements, a rolling horizon framework is employed to provide real-time and dynamic optimization decisions. In each decision period, the AVs' sailing speeds and berth reallocation are jointly optimized based on the latest information on stochastic disturbances and berth occupancy. The problem is formulated as a mixed-integer nonlinear programming (MINLP) model aimed at minimizing the total cost. A tailored branch-and-cut algorithm incorporating an outer approximation method is developed to efficiently solve the model.

The contributions of this paper are threefold. First, due to the absence of onboard captain intervention, AVs are more susceptible to stochastic disturbances during operations. These disturbances can lead to deviations in expected arrival times and allocated berths. To address this issue, we propose a dynamic AV speed optimization and berth reallocation framework that explicitly accounts for stochastic disturbances.To the best of our knowledge, this is the first study that jointly optimizes AV sailing speeds and berth reallocation to mitigate the impact of stochastic disturbances at the operational stage.

Second, to meet real-time operational requirements, the investigated problem is modeled with a rolling horizon framework. In each decision period, an MINLP model is built to jointly optimize AV sailing speeds and berth reallocation decisions, aiming to minimize the total cost, which includes penalties for arrival time delays, additional container transportation costs due to berth deviations, and fuel consumption costs. Given the nonlinearity in the arrival time and fuel consumption calculations, several linearization techniques are applied to reformulate the model as a mixed-integer linear programming (MILP) model. To solve the MILP model efficiently in each decision period, we develop a tailored branch-and-cut algorithm, where cutting planes are dynamically generated using an outer approximation method.

Third, to evaluate the effectiveness and performance of the proposed model and solution method, we conducted extensive numerical experiments based on operational data from a Chinese maritime logistics company. To further validate the performance of the proposed algorithm, we also compared it with Gurobi solver and a greedy algorithm simulating the "first-come-first-served" principle. Finally, sensitivity analyses on several key parameters are performed.

The remainder of this study is structured as follows. Section II reviews the related literature. Section III provides a detailed problem description and mathematical formulations. Section IV describes the solution method. Numerical experiments are conducted in Section V. Finally, Section VI concludes the paper.

II. Literature review

This study is first related to the vessel speed optimization problem, which has garnered significant attention in maritime transportation due to its economic and environmental implications [6]. From an economic perspective, numerous studies have examined the impact of vessel speed on fuel consumption. For example, [13] jointly optimized the sailing speeds at all legs and bunker fuel procurement at each port to minimize total fuel and operational costs. [14] determined the optimal ship fleet configuration from a set of candidate vessels and further optimized sailing speeds and routing sequences to minimize the total cost. There are also some research investigating the vessel speed from the environmental perspective. For instance, [15] assessed the economic and environmental impacts of sulfur emission control areas by optimizing fleet size and distinguishing sailing speeds inside and outside the emission control zones. Similarly, [16] differentiated sailing speeds across these zones while jointly optimizing sailing routes to minimize the total fuel consumption cost and sulfur dioxide emissions.

The sailing speeds of vessels have direct impacts on their sailing times, and thus affect their arrival times at ports and berth allocation. In recent years, some researcher have incorporated berth allocation into vessel speed optimization. The berth allocation problem (BAP) was initially proposed by [17], which determined the berthing time and berthing positions for incoming vessels. Over the past two decades, the BAP has attracted considerable attention from both researchers and practitioners. Some recent notable studies are as follows. [18] proposed a reactive berth allocation and scheduling model that incorporated various recovery strategies to handle uncertain vessel arrival times. [19] developed online algorithms to solve the BAP, with the objective of minimizing the maximum load of all berths. [20] formulated an MILP model for the dynamic discrete BAP to minimize the total vessel turnaround cost. For comprehensive reviews about the BAP and its recent developments, readers are referred to [21], and [22].

For studies that simultaneously optimize vessel sailing speeds and berth allocation, [23] proposed a bi-objective optimization model to optimize the BAP considering fuel consumption and emission, where sailing speed optimization was integrated into the fuel consumption. [24] further investigated the impact of tides on operations at a tidal container seaport, where berth allocation and sailing speeds were jointly optimized. [6] extended the traditional

BAP to cover multiple ports in a liner shipping network and optimized the vessel sailing speeds. Similar to [6], [8] also addressed the multi-port berth allocation problem, while also optimizing sailing speeds of vessels. [25] proposed a multi-objective mixed-integer programming model for the berth allocation and quay crane assignment problem, where sailing speeds were determined by vessel arrival times. Building on this, [26] introduced the vessel service differentiation and proposed a bi-level multi-objective optimization model. The upper level handled berth allocation and crane assignments, while the lower level optimized vessel speeds. [27] proposed an integrated optimization problem to determine the fleet deployment with different green technologies, sailing speeds, cargo allocation for each route, and berth allocation.

However, most existing studies on integrated berth allocation and speed optimization adopt deterministic models, without considering real-time stochastic disturbances that may impact vessel operations. In addition, previous studies do not consider the acceleration and deceleration phases. However, these two phases are particularly relevant for short-sea and inland operations, since they constitute a non-negligible portion of the total sailing time. To fill these gaps, this study investigates a dynamic AV speed optimization and berth reallocation problem to mitigate the impact of stochastic disturbances, where acceleration and deceleration phases are explicitly modeled. Besides, a rolling horizon framework is employed to enable dynamic and adaptive adjustments of sailing speeds and berth reallocation.

III. Model formulation

A. Problem description

Consider a fleet of AVs transporting containers on short-sea or inland shipping lines. At the operational stage, AVs are expected to follow a predetermined schedule designed at the planning stage. The schedule specifies the sequences of ports to be visited by AVs, expected arrival times at ports, berth allocation, the numbers of containers to be loaded and unloaded, and the corresponding dwell times at ports. For modeling convenience, the operational stage is discretized into a set of time points denoted by $\mathcal{T}$, where the interval between two consecutive time points represents a time slot.

Let $\mathcal{V}$ be the set of AVs. For each AV v, the set of indices of ports to be visited is denoted by $\mathcal{S}_v$, and the distance between its ith ($i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}$) and $(i+1)$st visiting ports is represented by h_i^v. In practice, these ports are geographically proximate, with inter-port distances typically ranging from tens to hundreds of nautical miles. Consequently, the sailing time at each leg usually spans several hours to a day, with acceleration and deceleration durations typically lasting up to half an hour each, totaling approximately an hour. Therefore, the acceleration and deceleration phases cannot be ignored and are explicitly incorporated into the model. As illustrated in Fig. 1, the sailing process of AV v between its two consecutive visiting ports consists of three distinct phases: acceleration, cruising, and deceleration phases. The acceleration and deceleration rates are denoted by g_i^v and ω_i^v, respectively. After acceleration, AV v enters the cruising phase, sailing at a constant speed, referred to as the cruising speed. Let $\overline{\eta}$ and $\underline{\eta}$ represent the upper and lower bounds of cruising speeds, respectively. Since fuel consumption grows nonlinearly with sailing speeds, a careful trade-off between sailing speeds and fuel consumption should be considered. Let f denote the unit fuel price. During navigation, AVs are exposed to sailing time disturbances, denoted by θ_i^v, arising from variations in wave heights and wind directions. These disturbances can cause deviations in actual sailing times, thereby resulting in deviations in arrival times at ports. Therefore, it is essential to dynamically adjust the sailing speeds of AVs at each leg to mitigate the impacts of stochastic disturbances and maintain schedule reliability.

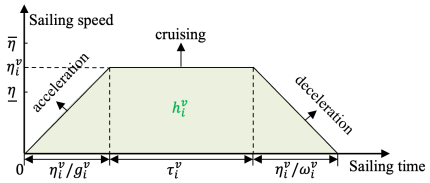

Fig. 1. Illustration of the sailing process of an AV at a leg.

Each AV v is expected to arrive at its ith visiting port at the scheduled arrival time $\bar{t}_i^v$. When the actual arrival time deviates from $\bar{t}_i^v$, a penalty cost c_i^v will incur. Note that the deviations in arrival times may affect the berth availability, as originally assigned berths could already be occupied by other AVs. Let $\mathcal{B}$ represent the set of all berths, and $\mathcal{B}_i^v \subseteq \mathcal{B}$ denote the set of available berths at the ith visiting port for AV v. A binary parameter $\overline{y}_{i,b}^v$ is introduced to indicate whether AV v is expected to moor at its initial allocated berth $b \in \mathcal{B}_i^v$ of its ith visiting port. If an AV cannot moor at its designated berth, berth reallocation is required to assign an alternative berth. However, mooring at a berth different from the originally assigned one will incur an additional container transportation cost, denoted by φ_i^v per container. The total additional container transportation cost depends on the total number of containers handled at ports. Specifically, let a_i^v and b_i^v denote the numbers of containers unloaded from and loaded onto AV v at its ith visiting port, respectively. Given the container handling efficiency ρ_i^v, the corresponding dwell time d_i^v is calculated as $d_i^v = \rho_i^v \cdot (a_i^v + b_i^v)$.

In summary, this paper investigates the dynamic sailing speed optimization and berth reallocation problem for AVs in response to stochastic disturbances at the operational stage. The objective is to minimize the total cost for the operator, including the total penalty cost for arrive delays, the total additional transportation cost of containers incurred by deviations in allocated berths, and the total fuel consumption cost.

B. Model formulation

Before presenting the mathematical formulation, we first introduce decision variables involved in the model. For each AV v, let μ_i^v and r_i^v denote the adjusted sailing time

and actual sailing time between its ith and $(i+1)$st visiting ports, respectively. The actual arrival time of AV v at its ith visiting port is represented by t_i^v. During a leg between the ith and $(i+1)$st visiting ports of AV v, the cruising speed and cruising time are given by η_i^v and τ_i^v, respectively. The fuel consumption of AV v during the same leg is captured by e_i^v. To model berth reallocation decisions, a set of binary variables is introduced. Specifically, $\delta_{i,b,t}^v$ equals 1 if AV v starts mooring at berth $b \in \mathcal{B}_i^v$ of its ith visiting port at time point t, and 0 otherwise. Similarly, $z_{i,b,t}^v$ equals 1 if AV v moors at berth $b \in \mathcal{B}_i^v$ of its ith visiting port at time point t, and 0 otherwise. Finally, $y_{i,b}^v$ equals 1 if AV v moors at berth $b \in \mathcal{B}_i^v$ of its ith visiting port, and 0 otherwise.

The mathematical formulation consists of several groups of constraints, including those related to sailing time, fuel consumption, actual arrival times, and berth reallocation.

Sailing time constraints

The actual sailing time of AV v from its ith visiting port to its $(i+1)$st visiting port depends on the adjusted sailing time μ_i^v and the sailing time disturbance θ_i^v, as follows:

$$r_i^v = \mu_i^v + \theta_i^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\} \tag{1}$$

where adjusted sailing time μ_i^v depends on the sailing speed and is applied to adjust the sailing time.

The cruising speed of AV v between its ith and $(i+1)$st visiting ports must satisfy following constraints:

$$\underline{\eta} \le \eta_i^v \le \overline{\eta}, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}. \tag{2}$$

Once cruising speed η_i^v is determined, the acceleration and deceleration times can be calculated as η_i^v/g_i^v and η_i^v/ω_i^v, respectively. Then, the adjusted sailing time μ_i^v, which includes acceleration, cruising, and deceleration times, can be expressed as:

$$\mu_i^v = \frac{\eta_i^v}{g_i^v} + \tau_i^v + \frac{\eta_i^v}{\omega_i^v}, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}. \tag{3}$$

It is evident that the adjusted sailing time μ_i^v depends only on cruising speed η_i^v and cruising time τ_i^v.

In Fig. 1, the green shaded area denote the distance between the ith and $(i+1)$st visiting ports of AV v, which can be expressed as follows:

$$h_i^v = \frac{1}{2}\cdot g_i^v \left(\frac{\eta_i^v}{g_i^v}\right)^2 + \eta_i^v \cdot \tau_i^v + \frac{1}{2}\cdot \omega_i^v \left(\frac{\eta_i^v}{\omega_i^v}\right)^2, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \backslash \{|\mathcal{S}_v|\} \tag{4}$$

Based on constraints (4), the cruising time τ_i^v can be further expressed as follows:

$$\tau_i^v = \frac{h_i^v}{\eta_i^v} - \frac{(g_i^v + \omega_i^v)\cdot \eta_i^v}{2 \cdot g_i^v \cdot \omega_i^v}, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}. \tag{5}$$

By incorporating constraints (3) and (5) into constraints (1), the actual sailing time r_i^v can be further represented as:

$$r_i^v = \frac{(g_i^v + \omega_i^v)\cdot \eta_i^v}{2 \cdot g_i^v \cdot \omega_i^v} + \frac{h_i^v}{\eta_i^v} + \theta_i^v, \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}. \tag{6}$$

Constraints (6) show that the actual sailing time r_i^v can be adjusted by cruising speed η_i^v. Note that constraints (6) are nonlinear because of the nonlinear term h_i^v/η_i^v.

Fuel consumption constraints

The fuel consumption of AV at each leg consists of three components: fuel consumption during the acceleration, cruising, and deceleration phases. According to [28], the fuel consumption per hour is $[m \cdot (\eta_i^v)^n]/24$, where m and n are coefficients. Then, the fuel consumption is modeled as follows:

$$\begin{aligned} e_i^v = &\int_0^{\frac{\eta_i^v}{g_i^v}} \frac{m \cdot (g_i^v \cdot t)^n}{24}\,\mathrm{d}t \quad + \int_0^{\frac{h_i^v}{\eta_i^v} - \frac{(g_i^v+\omega_i^v)\cdot\eta_i^v}{2\cdot g_i^v\cdot\omega_i^v}} \frac{m\cdot(\eta_i^v)^n}{24}\,\mathrm{d}t \\ &+ \int_0^{\frac{\eta_i^v}{\omega_i^v}} \frac{m\cdot(\omega_i^v\cdot t)^n}{24}\,\mathrm{d}t, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}. \end{aligned} \tag{7}$$

By calculating the integration, e_i^v can be expressed as:

$$\begin{aligned} e_i^v = &\frac{m\cdot(g_i^v+\omega_i^v)\cdot(\eta_i^v)^{n+1}}{24\cdot g_i^v\cdot\omega_i^v\cdot(n+1)} + \frac{m\cdot h_i^v\cdot(\eta_i^v)^{n-1}}{24} \\ &- \frac{m\cdot(g_i^v+\omega_i^v)\cdot(\eta_i^v)^{n+1}}{48\cdot g_i^v\cdot\omega_i^v}, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \setminus \{|\mathcal{S}_v|\}. \end{aligned} \tag{8}$$

Note that constraints (8) are nonlinear.

Actual arrival time constraints

The dynamic equations of the actual arrival time of AVs at ports can be modeled as follows:

$$t_i^v = \begin{cases} \overline{t}_1^v, & \text{if } i = 1, \\ t_{i-1}^v + r_{i-1}^v + d_{i-1}^v, & \text{otherwise.} \end{cases} \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \tag{9}$$

where $\overline{t}_1^v$ denotes actual arrival time of AV v at its first visiting port, which is known. Constraints (9) indicate the actual arrival time at the port is determined by the actual arrival and dwell times at the previous port, and the actual sailing time at this leg.

Berth reallocation constraints

For each berth $b \in \mathcal{B}$, at most one AV can occupy it at any time point t, which can be modeled as follows:

$$\sum\nolimits_{v\in\mathcal{V}} \sum\nolimits_{i\in\mathcal{S}_v : b\in\mathcal{B}_i^v} z_{i,b,t}^v \le 1, \quad \forall b \in \mathcal{B}, t \in \mathcal{T}. \tag{10}$$

Each AV must be assigned to exactly one berth at each of its visiting ports, which can be modeled by the following constraints:

$$\sum\nolimits_{b\in\mathcal{B}_i^v} y_{i,b}^v = 1, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \tag{11}$$

$$y_{i,b}^v \le \sum\nolimits_{t\in\mathcal{T}} z_{i,b,t}^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v, b \in \mathcal{B}_i^v \tag{12}$$

$$y_{i,b}^v \ge z_{i,b,t}^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v, b \in \mathcal{B}_i^v, t \in \mathcal{T} \tag{13}$$

where constraints (11) stipulate each AV must moor at one berth. Constraints (12)–(13) illustrate the relationship between $y_{i,b}^v$ and $z_{i,b,t}^v$. If $\sum_{t\in\mathcal{T}} z_{i,b,t}^v = 0$, $y_{i,b}^v = 0$. If $\sum_{t\in\mathcal{T}} z_{i,b,t}^v \ge 1$, $y_{i,b}^v$ can be 1 or 0. However, constraint (13) enforces $y_{i,b}^v \ge z_{i,b,t}^v$ for any t, ensuring $y_{i,b}^v = 1$.

Each AV can moor at each berth once and should moor at a berth within consecutive time slots. Then,

$$\sum\nolimits_{b\in\mathcal{B}_i^v} \sum\nolimits_{t\in\mathcal{T}} \delta_{i,b,t}^v = 1, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v \tag{14}$$

$$\delta_{i,b,1}^v = z_{i,b,1}^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{S}_v, b \in \mathcal{B}_i^v \tag{15}$$

$$\delta_{i,b,t}^v \ge z_{i,b,t}^v - z_{i,b,t-1}^v, \forall v \in \mathcal{V}, i \in \mathcal{S}_v, b \in \mathcal{B}_i^v, t \in \mathcal{T} \setminus \{1\} \tag{16}$$

where constraints (14) ensures that each AV moors exactly once at one berth at its each visiting port. Constraints (15)–(16) establish the relationship between $\delta_{i,b,t}^v$ and $z_{i,b,t}^v$. Specifically, they ensure that $\delta_{i,b,t}^v$ takes the value 1 only at the beginning of a mooring operation,

namely, when $z^v_{i,b,t}$ changes from 0 to 1 between two consecutive time slots.

The earliest time at which an AV can moor at the berth is equal to its arrival time at the port, following [3] and [8], which is presented as:

$$\sum\nolimits_{b\in\mathcal{B}_i^v}\sum\nolimits_{t\in\mathcal{T}} t\cdot\delta^v_{i,b,t}=t_i^v,\quad \forall v\in\mathcal{V}, i\in\mathcal{S}_v. \tag{17}$$

If an AV moors at a berth, its berthing time should equal its dwell time, which can be formulated as follows:

$$\sum\nolimits_{b\in\mathcal{B}_i^v}\sum\nolimits_{t\in\mathcal{T}} z^v_{i,b,t}=d_i^v,\quad \forall v\in\mathcal{V}, i\in\mathcal{S}_v \tag{18}$$

where $\sum_{b\in\mathcal{B}_i^v}\sum_{t\in\mathcal{T}} z^v_{i,b,t}$ denotes the total time slots that AV moors at the berth, i.e., berthing time.

Based on the above constraints, the optimal AV sailing speed optimization and berth reallocation model can be built as follows:

$$\begin{aligned}\min &\sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{S}_v} c_i^v\cdot\max\left\{t_i^v-\overline{t}_i^v,0\right\}+\sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{S}_v\setminus\{|\mathcal{S}_v|\}} f\cdot e_i^v\\ &+\sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{S}_v}\sum_{b\in\mathcal{B}_i^v}\varphi_i^v\cdot(a_i^v+b_i^v)\max\left\{y^v_{i,b}-\overline{y}^v_{i,b},0\right\}\end{aligned} \tag{19}$$

subject to constraints (2), (6)–(9), (10)–(18), and

$$\delta^v_{i,b,t}, z^v_{i,b,t}, y^v_{i,b}\in\{0,1\}\quad \forall v\in\mathcal{V}, i\in\mathcal{S}_v, b\in\mathcal{B}_i^v, t\in\mathcal{T}. \tag{20}$$

Objective function (19) minimizes the total cost, including the total penalty resulting from arrival time delays at ports, the total fuel consumption cost and the total additional transportation cost of containers incurred by deviations in allocated berths. Objective function is nonlinear, due to terms $\max\left\{t_i^v-\overline{t}_i^v,0\right\}$ and $\max\left\{y^v_{i,b}-\overline{y}^v_{i,b},0\right\}$.

C. Rolling horizon framework

In practice, decisions related to AVs during operations are typically made dynamically to cope with real-time sailing time disturbances. Prior studies have shown that short-term disturbances can be predicted with reasonable accuracy using updated environmental data [12], [29]. However, long-term forecasts of such disturbances tend to be highly unreliable, and making operational decisions (including sailing speeds and berth reallocation) based on such long-term forecasts may result in sub-optimal or even infeasible results. To address this issue and meet real-time operational requirements, we adopt a rolling horizon framework that has been widely adopted in the literature [30], [31]. This framework enables the incorporation of short-term disturbance forecasts as inputs in each decision period, and supports dynamic and adaptive adjustments to sailing speeds and berth reallocation based on the latest information of disturbances. Consequently, it enhances the robustness and practical applicability of the overall decision-making process. Different from the traditional rolling horizon framework which divides the whole optimization period into several decision periods based on the time length, this paper divides decision periods based on the number of visiting ports of AVs. In each decision period, an optimization model is formulated to optimize decisions of all AVs at their next several visiting ports, but only decisions associated with the AV with the earliest arrival time at its second visiting port are performed. Specifically, the decisions associated with the AV with the earliest arrival time include the berth reallocation at its first visiting port, and the cruising speed between its first visiting port and second visiting port. Then, in the next decision period, the optimization process is repeated based on the optimization results in the last decision period.

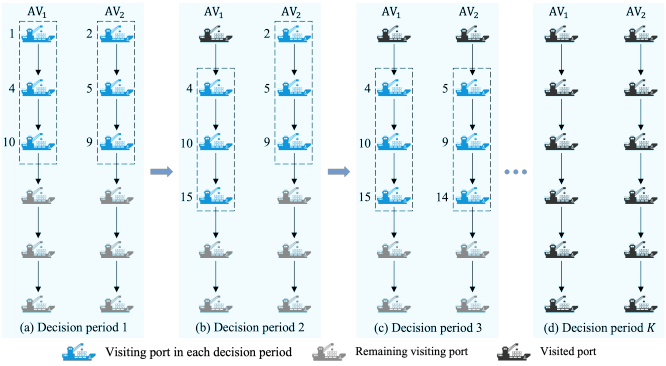

Fig. 2. An illustration example for the rolling horizon framework.

Fig. 2 illustrates our rolling horizon framework, where each AV has six visiting ports, and three ports are considered in each decision period. The blue ports represent the visiting ports in each decision period and the decisions at these ports should be optimized. The black ports depict the visited ports, where decisions have already been executed, including berth reallocation and cruising speed to the next port. The grey ports denote the remaining visiting ports not yet involved in the current decision period. The numbers besides ports indicate arrival times. In decision period 1, AV_1 arrives at its second visiting port earlier (time 4) than AV_2 (time 5). Hence, decisions for AV_1 are executed, including the berth assignment at its first port and the cruising speed between its first and second ports. The first visiting port of AV_1 then becomes a visited port and is removed from the list of visiting ports. A subsequent visiting port is then added to constitute the list of visiting ports of AV_1 in decision period 2. In decision period 2, AV_2 has the earlier arrival time at its second visiting port than AV_1. The decisions related to AV_2 are implemented. This rolling process is repeated until decisions of AVs at all visiting ports have been implemented.

Let $\tilde{\mathcal{S}}_v^k$ represent the set of the sequences of ports to be visited by AV v in decision period k. In decision period k, the available information optimized in decision period $k-1$ are inputted to optimize decisions in decision period k: Parameter $\tilde{t}_{i_1}^{v,k-1}$ represents actual arrival time of AV v at its first visiting port ($i_1\in\tilde{\mathcal{S}}_v^k$). Binary parameter $\tilde{z}_{b,t}^{k-1}$ indicates whether berth b is occupied at time point t.

With the given available information, the optimization model [**M-**k] in decision period k is formulated as below:

$$\begin{aligned}\min &\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k} c_i^v\cdot\max\left\{t_i^{v,k}-\overline{t}_i^v,0\right\}+\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|} f\cdot e_i^{v,k}\\ &+\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k}\sum_{b\in\mathcal{B}_i^v}\varphi_i^v\cdot(a_i^v+b_i^v)\max\left\{y^{v,k}_{i,b}-\overline{y}^v_{i,b},0\right\}\end{aligned} \tag{21}$$

subject to

$$\underline{\eta} \le \eta_i^{v,k} \le \overline{\eta}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \setminus \{|\tilde{\mathcal{S}}_v|\} \tag{22}$$

$$r_i^{v,k} = \frac{(g_i^v + \omega_i^v) \cdot \eta_i^{v,k}}{2 \cdot g_i^v \cdot \omega_i^v} + \frac{h_i^v}{\eta_i^{v,k}} + \theta_i^v, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \setminus \{|\tilde{\mathcal{S}}_v|\} \tag{23}$$

$$e_i^{v,k} = \frac{m \cdot (g_i^v + \omega_i^v) \cdot (\eta_i^{v,k})^{n+1}}{24 \cdot g_i^v \cdot \omega_i^v \cdot (n+1)} + \frac{m \cdot h_i^v \cdot (\eta_i^{v,k})^{n-1}}{24} - \frac{m \cdot (g_i^v + \omega_i^v) \cdot (\eta_i^{v,k})^{n+1}}{48 \cdot g_i^v \cdot \omega_i^v}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \setminus \{|\tilde{\mathcal{S}}_v|\} \tag{24}$$

$$t_{i_1}^{v,k} = \tilde{t}_{i_1}^{v,k-1}, \quad \forall v \in \mathcal{V}, i_1 \in \tilde{\mathcal{S}}_v^k \tag{25}$$

$$t_i^{v,k} = t_{i-1}^{v,k} + r_{i-1}^{v,k} + d_{i-1}^v, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \setminus \{|\tilde{\mathcal{S}}_v|\} \tag{26}$$

$$\sum\nolimits_{v \in \mathcal{V}} \sum\nolimits_{i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v} z_{i,b,t}^{v,k} + \tilde{z}_{b,t}^{k-1} \le 1, \quad \forall b \in \mathcal{B}, t \in \mathcal{T} \tag{27}$$

$$\sum\nolimits_{b \in \mathcal{B}_i^v} y_{i,b}^{v,k} = 1, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \tag{28}$$

$$y_{i,b}^{v,k} \le \sum\nolimits_{t \in \mathcal{T}} z_{i,b,t}^{v,k}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v \tag{29}$$

$$y_{i,b}^{v,k} \ge z_{i,b,t}^{v,k}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v, t \in \mathcal{T} \tag{30}$$

$$\sum\nolimits_{b \in \mathcal{B}_i^v} \sum\nolimits_{t \in \mathcal{T}} \delta_{i,b,t}^{v,k} = 1, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \tag{31}$$

$$\delta_{i,b,1}^{v,k} = z_{i,b,1}^{v,k}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v \tag{32}$$

$$\delta_{i,b,t}^{v,k} \ge z_{i,b,t}^{v,k} - z_{i,b,t-1}^{v,k}, \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v, t \in \mathcal{T} \setminus \{1\} \tag{33}$$

$$\sum\nolimits_{b \in \mathcal{B}_i^v} \sum\nolimits_{t \in \mathcal{T}} t \cdot \delta_{i,b,t}^{v,k} = t_i^{v,k}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \tag{34}$$

$$\sum\nolimits_{b \in \mathcal{B}_i^v} \sum\nolimits_{t \in \mathcal{T}} z_{i,b,t}^{v,k} \ge d_i^v, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \tag{35}$$

$$z_{i,b,t}^{v,k}, \delta_{i,b,t}^{v,k}, y_{i,b}^{v,k} \in \{0,1\}, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v, t \in \mathcal{T}. \tag{36}$$

Objective function (21) minimizes the total cost in decision period k. Constraints (22) specify ranges of cruising speeds. Constraints (23) state actual sailing times. Constraints (24) calculate fuel consumption. Constraints (25)–(26) update arrival times. Constraints (27)–(35) determine berth reallocation. Constraints (36) define variables.

After solving model [**M-**k] in decision period k, only decisions of the AV v with the earliest arrival time $t_{i_2}^{v,k}$ at its second visiting port i_2 are be implemented. The AV with the earliest arrival time at its second visiting port can be obtained by equation (37) as follows:

$$v^* = \text{argmin}_{v \in \mathcal{V}} t_{i_2}^{v,k}. \tag{37}$$

The decisions of AV v^*, including the berth reallocation at its first visiting port (i.e., $z_{i_1,b,t}^{v^*,k}$, $\delta_{i_1,b,t}^{v^*,k}$ and $y_{i_1,b}^{v^*,k}$) and its cruising speed between its first and second visiting ports (i.e., $\eta_{i_1}^{v^*,k}$) will be applied. Then, in decision period $k+1$, the set of sequences of ports to be visited by AVs will be updated, as follows:

$$\tilde{\mathcal{S}}_v^{k+1} = \begin{cases} \tilde{\mathcal{S}}_{v^*}^k \setminus \{i_1\} \cup \{i'\}, & \text{if } v = v^*, \\ \tilde{\mathcal{S}}_v^k, & \text{otherwise.} \end{cases} \quad \forall v \in \mathcal{V} \tag{38}$$

where i_1 ($i_1 \in \tilde{\mathcal{S}}_{v^*}^k$) is the first visiting port of AV v^* in decision period k. i' is the next visiting port of AV v^* after its $|\tilde{\mathcal{S}}_{v^*}^k|$th visiting port.

Additionally, the initial states of AVs in decision period $k+1$ will also be updated, as follows:

$$\tilde{t}_{i_1}^{v,k+1} = \begin{cases} t_{i_1}^{v^*,k}, & \text{if } v = v^*, \\ t_{i_1}^{v,k}, & \text{otherwise.} \end{cases} \quad \forall v \in \mathcal{V} \tag{39}$$

$$\tilde{z}_{b,t}^{k+1} = \max \left\{ \sum_{i_1 \in \mathcal{S}_{v^*} : b \in \mathcal{B}_{i_1}^{v^*}} z_{i_1,b,t}^{v^*,k}, \tilde{z}_{b,t}^{v,k} \right\}, \forall b \in \mathcal{B}, t \in \mathcal{T} \tag{40}$$

where $t_{i_1}^{v,k}$ in equations (39) indicates the arrival time of AV v at its first visiting port $i_1 \in \tilde{\mathcal{S}}_v^{k+1}$ in decision period $k+1$. $\tilde{z}_{b,t}^{k+1}$ in equations (40) denotes the berth occupancy parameter in decision period $k+1$.

IV. Solution algorithm

In this section, the MINLP model [**M-**k] is first linearized and then transformed to the MILP model. Then, to meet real-time operational requirements of AVs, a tailored branch-and-cut algorithm is devised to solve the MILP model in each decision period.

A. Model linearization

Our optimization model [**M-**k] contains nonlinear objective function (21) and constraints (23)–(24). For the objective function (21), two auxiliary variables $\gamma_i^{v,k}$ and $\beta_{i,b}^{v,k}$ are introduced to linearize it as follows:

$$\gamma_i^{v,k} \ge t_i^{v,k} - \overline{t}_i^v, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \tag{41}$$

$$\gamma_i^{v,k} \ge 0, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k \tag{42}$$

$$\beta_{i,b}^{v,k} \ge y_{i,b}^{v,k} - \overline{y}_{i,b}^v, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v \tag{43}$$

$$\beta_{i,b}^{v,k} \ge 0, \quad \forall v \in \mathcal{V}, i \in \tilde{\mathcal{S}}_v^k, b \in \mathcal{B}_i^v. \tag{44}$$

For constraints (23)–(24), outer-approximation method is employed to approximate the nonlinear constraints to a set of linear constraints. The out-approximation method has been exploited by many scholars to obtain a ε-optimal solution of an MINLP problem with nonlinear constraints [14]. The ε-optimal solution refers to the solution that the difference between its objective function value obtained by approximation and the optimal objective function value is within the given tolerance ε.

In constraints (23)–(24), we replace the nonlinear terms $h_i^v/\eta_i^{v,k}$ and $e_i^{v,k}$ with $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$, respectively. To control the approximation error caused by outer-approximation method, an absolute objective value tolerance ε is defined. Note that the first term in objective function (21) is implicitly related to $Q(\eta_i^{v,k})$, since AV's actual sailing time at each leg influences their actual arrival time at ports. Besides, the third term in objective function (21) is related to $F(\eta_i^{v,k})$. Then, we allocate the total approximation errors ε_1 and ε_2 to $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$, where $\varepsilon_1 + \varepsilon_2 = \varepsilon$. Then, we first calculate approximation error for a $Q(\eta_i^{v,k})$, i.e., $\overline{\varepsilon}_1$. The detailed derivation process is as follows:

$$\begin{aligned} t_i^{v,k} - \overline{t}_i^v &= t_{i-1}^{v,k} + r_{i-1}^{v,k} + d_{i-1}^v - \overline{t}_i^v \\ &= t_1^{v,k} - \overline{t}_i^v + \sum_{i'=1}^{i-1} \left[\frac{(g_{i'}^v + \omega_{i'}^v) \cdot \eta_{i'}^{v,k}}{2 \cdot g_{i'}^v \cdot \omega_{i'}^v} + Q(\eta_{i'}^{v,k}) + \theta_{i'}^v + d_{i'}^v \right] \end{aligned} \tag{45}$$

$$\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k} c_i^v\cdot(t_i^{v,k}-\overline{t}_i^v)=\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|}\left(\sum_{i'>i}c_{i'}^v\right)\cdot\left[Q(\eta_{i'}^{v,k})+\frac{(g_{i'}^v+\omega_{i'}^v)\cdot\eta_{i'}^{v,k}}{2\cdot g_{i'}^v\cdot\omega_{i'}^v}+\theta_{i'}^v+d_{i'}^v\right]+\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k}c_i^v\cdot(t_1^{v,k}-\overline{t}_i^v) \tag{46}$$

where equation (46) indicates the worst case of the number of $Q(\eta_i^{v,k})$ in the first term of objective function (21), i.e., $c_i^v\cdot\max\left\{t_i^{v,k}-\overline{t}_i^v,0\right\}$. In some cases, the term $\max\left\{t_i^{v,k}-\overline{t}_i^v,0\right\}$ can be zero, leading to the disappearance of some $Q(\eta_i^{v,k})$. Hence, the number of $Q(\eta_i^{v,k})$ is always not larger than $\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|}\sum_{i'>i}c_{i'}^v$. Then, the maximum approximation error for a $Q(\eta_i^{v,k})$, i.e., $\overline{\varepsilon}_1$, will be:

$$\overline{\varepsilon}_1=\frac{\varepsilon_1}{\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|}\left(\sum_{i'>i}c_{i'}^v\right)}. \tag{47}$$

From the second term in objective function (21), i.e., $\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|}f\cdot e_i^{v,k}$, it can be observed that the number of $F(\eta_i^{v,k})$ is $\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|}f$. Then, we can derive the approximation error for a $F(\eta_i^{v,k})$, i.e., $\overline{\varepsilon}_2$, which is as follows:

$$\overline{\varepsilon}_2=\frac{\varepsilon_2}{\sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|}f}. \tag{48}$$

If the total approximation errors for $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$ are not greater than ε_1 and ε_2, then the overall objective value error will be not greater than ε.

Constraints (23)–(24) can thereby be relaxed by replacing $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$ with a set of linear functions being tangent to the convex curves $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$. These linear functions are grouped into two sets denoted by $\mathcal{R}_i^{v,k}$ and $\mathcal{O}_i^{v,k}$ for $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$, respectively. Fig. 3 illustrates the linearization of $Q(\eta_i^{v,k})$ as an example. Then, the linear counterpart of constraints (23) can be expressed as follows:

$$r_i^{v,k}=\frac{(g_i^v+\omega_i^v)\cdot\eta_i^{v,k}}{2\cdot g_i^v\cdot\omega_i^v}+Q\left(\eta_i^{v,k}\right)+\theta_i^v,\quad\forall v\in\mathcal{V},i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\} \tag{49}$$

$$Q(\eta_i^{v,k})\geq p_{i,r}^{v,k}\cdot\eta_i^{v,k}+q_{i,r}^{v,k},\forall v\in\mathcal{V},i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\},r\in\mathcal{R}_i^{v,k} \tag{50}$$

where $p_{i,r}^{v,k}$ and $q_{i,r}^{v,k}$ denote the slope and intercept of the rth ($r\in\mathcal{R}_i^{v,k}$) tangent line of the curve $Q(\eta_i^{v,k})$ at tangent point $\eta_i^{v,k}$, namely, $p_{i,r}^{v,k}=-h_i^v/(\eta_i^{v,k})^2$, $q_{i,r}^{v,k}=Q(\eta_i^{v,k})-p_{i,r}^{v,k}\cdot\eta_i^{v,k}$.

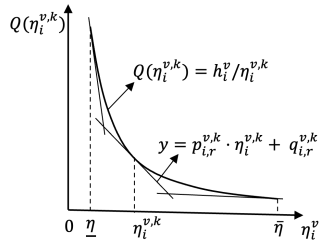

Fig. 3. Illustration of the outer-approximation method for $Q(\eta_i^{v,k})$.

Analogously, constraints (24) will be:

$$e_i^{v,k}=F(\eta_i^{v,k}),\quad\forall v\in\mathcal{V},i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\} \tag{51}$$

$$F(\eta_i^{v,k})\geq\xi_{i,o}^{v,k}\cdot\eta_i^{v,k}+\zeta_{i,o}^{v,k},\forall v\in\mathcal{V},i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\},o\in\mathcal{O}_i^{v,k} \tag{52}$$

where $\xi_{i,o}^{v,k}$ and $\zeta_{i,o}^{v,k}$ denote the slope and intercept of the oth ($o\in\mathcal{O}_i^{v,k}$) tangent line of the curve $F(\eta_i^{v,k})$ at tangent point $\eta_i^{v,k}$. The detailed values for $\xi_{i,o}^{v,k}$ and $\zeta_{i,o}^{v,k}$ are calculated by equations (53)–(54), shown below:

$$\xi_{i,o}^{v,k}=\frac{m\cdot(g_i^v+\omega_i^v)\cdot(\eta_i^{v,k})^n}{24\cdot g_i^v\cdot\omega_i^v}+\frac{m\cdot(n-1)\cdot h_i^v\cdot(\eta_i^{v,k})^{n-2}}{24}-\frac{m\cdot(n+1)\cdot(g_i^v+\omega_i^v)\cdot(\eta_i^{v,k})^n}{48\cdot g_i^v\cdot\omega_i^v},\quad\forall v\in\mathcal{V},i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\},o\in\mathcal{O}_i^{v,k} \tag{53}$$

$$\zeta_{i,o}^{v,k}=e_i^{v,k}-\xi_{i,o}^{v,k}\cdot\eta_i^{v,k},\quad\forall v\in\mathcal{V},i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\},o\in\mathcal{O}_i^{v,k}. \tag{54}$$

Algorithm 1 Tangent lines generation for $Q(\eta_i^{v,k})$

1: Initialize tangent points set $G=\left\{(\eta_i^{v,k})^L,(\eta_i^{v,k})^R\right\}$;
2: $[G]=\text{FindTangentPoint}((\eta_i^{v,k})^L,(\eta_i^{v,k})^R,\overline{\varepsilon}_1,G)$;
3: $[p_{i,r,1}^{v,k},q_{i,r,1}^{v,k}]=\text{TangentLine}((\eta_i^{v,k})^L)$;
4: $[p_{i,r,2}^{v,k},q_{i,r,2}^{v,k}]=\text{TangentLine}((\eta_i^{v,k})^R)$;
5: $[\eta_i^{v,k},\widehat{Q}(\eta_i^{v,k})]=\text{Intersection}(p_{i,r,1}^{v,k},q_{i,r,1}^{v,k},p_{i,r,2}^{v,k},q_{i,r,2}^{v,k})$;
6: $Error=|Q(\eta_i^{v,k})-\widehat{Q}(\eta_i^{v,k})|$;
7: **if** $Error\geq\overline{\varepsilon}_1$ **then**
8: $\quad G\leftarrow\eta_i^{v,k}$;
9: $\quad[G]=\text{FindTangentPoint}((\eta_i^{v,k})^L,\eta_i^{v,k},\overline{\varepsilon}_1,G)$;
10: $\quad[G]=\text{FindTangentPoint}(\eta_i^{v,k},(\eta_i^{v,k})^R,\overline{\varepsilon}_1,G)$;
11: $\quad$**return** G;
12: **else**
13: $\quad$**return** G;
14: **end if**
15: **for** $v\in\mathcal{V}$ **do**
16: $\quad$**for** $i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|$ **do**
17: $\quad\quad\mathcal{R}_i^{v,k}=\emptyset$; // tangent lines set
18: $\quad\quad A=\emptyset$; // slopes set
19: $\quad\quad B=\emptyset$; // intercepts set
20: $\quad\quad[G]=\text{FindTangentPoint}(\underline{\eta},\overline{\eta},\overline{\varepsilon}_1,G)$;
21: $\quad\quad$**for** $x\in\Omega$ **do**
22: $\quad\quad\quad r=0$;
23: $\quad\quad\quad[p_{i,r}^{v,k},q_{i,r}^{v,k}]=\text{TangentLine}(x)$;
24: $\quad\quad\quad A\leftarrow p_{i,r}^{v,k}$;
25: $\quad\quad\quad B\leftarrow q_{i,r}^{v,k}$;
26: $\quad\quad\quad\mathcal{R}_i^{v,k}\leftarrow r$;
27: $\quad\quad\quad r=r+1$;
28: $\quad\quad$**end for**
29: $\quad\quad$**output** $p_{i,r}^{v,k},q_{i,v,k}^v$;
30: $\quad$**end for**
31: **end for**

Algorithm 1 schematically illustrates the generation of tangent lines for $Q(\eta_i^{v,k})$. Lines 1–14 determine a set of tangent points for $Q(\eta_i^{v,k})$, and Lines 15–31 generate the slopes and intercepts of corresponding tangent lines. In detail, Line 1 gives the initial tangent lines set. Line 2 defines a recursive function *FindTangentPoint*. *TangentLine* function in Lines 3 and 4 finds the slopes and intercepts of tangent lines at the given points $((\eta_i^{v,k})^L,Q((\eta_i^{v,k})^L))$ and $((\eta_i^{v,k})^R,Q((\eta_i^{v,k})^R))$. *Intersection* function in Line 5 calculates the coordinates of the intersection (i.e., $(\eta_i^{v,k},\widehat{Q}(\eta_i^{v,k}))$) of the above tangent lines. Lines 7–14 check whether the intersection should be added to the tangent points set. In the following lines, for each AV at its each visiting port, *FindTangentPoint* function is applied

to find a set of tangent points. Then, for each tangent point, *TangentLine* function is used to generate the slopes and intercepts. The generation of tangent lines for $F(\eta_i^{v,k})$ is similar to that for $Q(\eta_i^{v,k})$. Hence, we omit the detailed process for generation of tangent lines for $F(\eta_i^{v,k})$.

By replacing nonlinear constraints (23)–(24) with constraints (49)–(52), the optimization model of decision period k can be formulated as an MILP model [**M-k-1**]:

$$\begin{aligned}\min \sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k} c_i^v\cdot\gamma_i^{v,k} + \sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k\setminus|\tilde{\mathcal{S}}_v|} f\cdot e_i^{v,k} \\ + \sum_{v\in\mathcal{V}}\sum_{i\in\tilde{\mathcal{S}}_v^k}\sum_{b\in\mathcal{B}_i^v}\varphi_i^v\cdot(a_i^v+b_i^v)\cdot\beta_{i,b}^{v,k}\end{aligned} \tag{55}$$

subject to constraints (22), (25)–(36), (41)–(44), and (49)–(52).

B. Branch-and-cut algorithm

Although heuristic algorithms can efficiently generate feasible solutions within an acceptable computational time, they do not guarantee global optimality. Moreover, the quality of the obtained solutions is often difficult to evaluate, as the optimality gap (i.e., the difference between the obtained objective function value and the optimal objective function value) remains unknown [32]. Therefore, we develop an exact solution method, namely, a tailored branch-and-cut method, to solve the MILP model [**M-k-1**]. The branch-and-cut method is a combinatorial optimization technique that combines the branch-and-bound framework with dynamically generated cutting planes to solve MILP models. In our tailored branch-and-cut method, all integer constraints in model [**M-k-1**] are initially relaxed to obtain a linear relaxation model. At each node of the branch-and-bound tree, the linear relaxation model is solved, and the resulting solution falls into one of the following three cases: (i) If the solution is fractional, this node will be branched to generate two child nodes. (ii) If the solution is integral and satisfies the given approximation error of outer-approximation method, this solution will be used to update the best upper bound. (iii) If the solution is integral but violates the given approximation error, new cutting planes generated by the outer-approximation method will be added to the linear relaxation model. Then, the updated linear relaxation model will be resolved until either a fractional solution is obtained or an integral solution that satisfies the given approximation error is found.

In detail, at the root node, all integer constraints (36)are relaxed. Besides, we only consider the subsets of tangent line sets $\mathcal{R}_i^{v,k}$ and $\mathcal{O}_i^{v,k}$ for $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$, instead of the whole sets denoted as constraints (50) and (52). The constraints related to the subsets are stated as follows, which are tangent lines of curves $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$ at tangent points $\underline{\eta}$ and $\overline{\eta}$:

$$Q(\eta_i^{v,k}) \geq p_{i,\underline{r}}^{v,k}\cdot\underline{\eta} + q_{i,\underline{r}}^{v,k}, \quad \forall v\in\mathcal{V}, i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\} \tag{56}$$

$$Q(\eta_i^{v,k}) \geq p_{i,\overline{r}}^{v,k}\cdot\overline{\eta} + q_{i,\overline{r}}^{v,k}, \quad \forall v\in\mathcal{V}, i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\} \tag{57}$$

$$F(\eta_i^{v,k}) \geq \xi_{i,\underline{o}}^{v,k}\cdot\underline{\eta} + \zeta_{i,\underline{o}}^{v,k}, \quad \forall v\in\mathcal{V}, i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\} \tag{58}$$

$$F(\eta_i^{v,k}) \geq \xi_{i,\overline{o}}^{v,k}\cdot\overline{\eta} + \zeta_{i,\overline{o}}^{v,k}, \quad \forall v\in\mathcal{V}, i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\}. \tag{59}$$

Then, the initial linear relaxation model will only contain objective function (55), constraints (22), (25)–(35), (49), (51), (41)–(44), (56)–(59), and continuous relaxations of constraints (36).

For any node in the branch and bound tree, if the optimal solution is integral but violates the given approximation error, new cutting planes will be added to reduce the approximation error. Specifically, under a current solution $(\boldsymbol{\eta^k})^* = ((\eta_i^{v,k})^*)$, if the difference between the curve $Q((\eta_i^{v,k})^*)$ and its tangent lines at any point $(\eta_i^{v,k})^*$ is larger than the predefined error $\bar{\varepsilon}_1$, i.e., $Q((\eta_i^{v,k})^*) - \left[p_{i,r}^{v,k}\cdot(\eta_i^{v,k})^* + q_{i,r}^{v,k}\right] > \bar{\varepsilon}_1$, we will adding the cutting plane at the point $(\eta_i^{v,k})^*$, as follows:

$$Q((\eta_i^{v,k})^*) \geq p_{i,r}^{v,k}\cdot(\eta_i^{v,k})^* + q_{i,r}^{v,k}, \forall v\in\mathcal{V}, i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\}. \tag{60}$$

Besides, if the difference between the curve $F((\eta_i^{v,k})^*)$ and its tangent lines at any point $(\eta_i^{v,k})^*$ is larger than the predefined error $\bar{\varepsilon}_2$, i.e., $F((\eta_i^{v,k})^*) - \left[\xi_{i,o}^{v,k}\cdot(\eta_i^{v,k})^* + \zeta_{i,o}^{v,k}\right] > \bar{\varepsilon}_2$, we will adding the cutting plane at the point $(\eta_i^{v,k})^*$, as follows:

$$F((\eta_i^{v,k})^*) \geq \xi_{i,o}^{v,k}\cdot(\eta_i^{v,k})^* + \zeta_{i,o}^{v,k}, \forall v\in\mathcal{V}, i\in\tilde{\mathcal{S}}_v^k\setminus\{|\tilde{\mathcal{S}}_v|\}. \tag{61}$$

When all nodes in the branch-and-bound tree are searched, the global optimal solution of optimization model [**M-k-1**] will be found. After solving optimization models in all decision periods, we can obtain the optimal solution by combining the solutions in each decision period. An optimal solution includes $(\eta_i^v)^*$, $(\gamma_i^v)^*$, $(\beta_{i,b}^v)^*$, $(t_i^v)^*$, $(e_i^v)^*$, $(z_{i,b,t}^v)^*$, $(\delta_{i,b,t}^v)^*$, and $(y_{i,b}^v)^*$. By substituting optimal solution into the objective function (19), we can acquire the objective function value.

V. Numerical experiments

In this section, we conduct extensive numerical experiments to demonstrate the effectiveness of the proposed model and solution method. The proposed algorithm is coded with PyCharm 2023.1.2, and the linear programming models are solved by Gurobi 10.0.0. In our implementation, the cuts generated by the outer-approximation method are dynamically added via the lazy constraint callback mechanism of the Gurobi solver. All numerical experiments are performed on a Windows PC with an Intel Core i7-8700, 3.90 GHz CPU, and 32 GB RAM.

A. Experimental setup

We consider a two-week operational period, which is split into 336 time slots and the length of each time slot was set to one hour. We select seven ports served by Zhonggu Logistics [33] as candidate visiting ports for AVs, including Port of Dalian, Port of Tianjin, Port of Yantai, Port of Qingdao, Port of Lianyungang, Port of Shanghai, and Port of Ningbo. The distances between these seven ports are calculated using information from the Ports.com Website [34]. The mooring of AVs at berths requires additional auxiliary sensing equipments. Hence, we assume

that there are two berths suitable for AVs to moor at each port. The capacity of all AVs is set as 1,000 TEU. The numbers of containers unloaded from and loaded onto AVs are generated based on the uniform distribution. In practice, the loading and unloading tasks for these AVs are generally handled by three quay cranes, each capable of handling 40 TEU per hour [26], resulting in a container handling efficiency ρ_i^v of 1/120 hour/TEU. The dwell time of AVs at ports is calculated based on the number of containers being loaded or unloaded and the container handling efficiency. The additional transportation cost of a container caused by the berth deviation of AVs at ports φ_i^v is set as 10 USD/TEU [35]. The fuel price f is set as 685 USD/ton [27]. The two fuel consumption coefficients m and n in the fuel consumption function (8) are set as 0.01 and three, respectively [28]. The lower and upper bounds of cruising speeds are set as $\underline{\eta} = 15$ knots and $\overline{\eta} = 30$ knots, respectively. The sailing time disturbance θ_i^v is randomly generated within a specified range, with the maximum disturbance set to 2. The penalty cost for the arrival time delay of AVs at ports, c_i^v, is set as 8,000 USD/hour. We assume that the acceleration and deceleration are equal, both fixed at 80 nm/hour^2. The number of visiting ports $|\tilde{\mathcal{S}}_v|$ in each decision period is set as three.

B. Computational results

To evaluate the performance and scalability of the proposed branch-and-cut algorithm, we generate 13 case groups with gradually increasing problem sizes. The largest case group involves up to 24 AVs, each visiting 12 ports. Each case group is labeled using a two-field notation "$|\mathcal{V}|$-$|\mathcal{S}_v|$", where $|\mathcal{V}|$ denotes the number of AVs and $|\mathcal{S}_v|$ represents the number of visiting ports per AV. For example, case group "10-12" involves 10 AVs, each having 12 visiting ports. To mitigate the impact of randomness, each case group comprises five independently generated instances, resulting in a total of 65 instances.

The computational results are shown in Table I. Columns "LB" and "UB" express the lower and upper bounds, respectively. Specially, lower bound is the objective function value calculated by our branch-and-cut algorithm. The upper bound is the objective function value calculated by the exact values of solution. Column "Gap(%)" calculates the relative gap between lower bound and upper bound, i.e., (UB−LB)×100/LB, which reflects the relative error of our model resulting from the application of piecewise-linear approximation functions. Column "TL" states the number of tangent lines generated by branch-and-cut algorithm. Column "CPU (s)" gives the computational time. From Table I, it can be observed that as the case group size increases from 5-12 to 24-12, the number of tangent lines (i.e., cuts) generated by the outer-approximation method increases substantially, accompanied by a corresponding increase in computational time. Besides, for all case groups, the optimality gaps between the lower and upper bounds are significantly less than 0.1%. This demonstrate the high reliability of the proposed branch-and-cut algorithm, which incorporates the outer-approximation method, in obtaining near-optimal solutions. For small-size case groups, where the number of AVs ranges from 5 to 10 and and each AV visits 10 or 12 ports, the branch-and-cut algorithm consistently obtains high-quality solutions within 700 seconds. As the group size increases to medium-size case groups (with up to 18 AVs and each AV having 12 visiting ports), the branch-and-cut algorithm is still able to achieve near-optimal solutions within one hour. These results highlight both the computational efficiency and effectiveness of the proposed branch-and-cut algorithm in solving small- and medium-scale instances. For large-size case groups (e.g., 21-12 and 24-12), a notable increase in computational time (up to around 8,000 seconds) is observed. This increase reflects the growth in computational complexity as the number of decision variables and constraints rises with problem size. Nevertheless, the solution quality remains consistently high, with optimality gaps still below 0.1%. These results validate both the scalability and high reliability of the proposed branch-and-cut algorithm in achieving near-optimal solutions for large-scale instances.

TABLE I
COMPUTATIONAL RESULTS OF BRANCH-AND-CUT ALGORITHM

Case	LB	UB	Gap (%)	TL	CPU (s)
5-10	3,164,710	3,164,727	0.0006	7,037	127.55
5-12	3,751,767	3,751,781	0.0004	8,944	158.12
7-10	4,611,228	4,611,245	0.0004	13,610	284.55
7-12	4,970,553	4,970,576	0.0005	17,498	347.15
10-10	7,126,407	7,126,424	0.0002	26,765	643.09
10-12	8,685,038	8,685,054	0.0002	35,067	678.80
12-10	7,364,695	7,364,712	0.0002	39,402	1,164.99
12-12	9,097,914	9,097,927	0.0001	50,801	1,355.68
15-10	10,462,682	10,462,698	0.0001	60,513	2,305.17
15-12	14,326,669	14,326,680	0.0001	77,782	2,676.29
18-12	14,561,170	14,561,179	0.0001	125,906	3,536.83
21-12	16,585,820	16,597,764	0.0720	162,000	4,633.60
24-12	17,105,652	17,105,663	0.0001	193,793	8,010.08

C. Comparison of branch-and-cut algorithm and Gurobi

We then compare the performance of our branch-and-cut algorithm with Gurobi solver. When using Gurobi solver, tangent lines for $Q(\eta_i^{v,k})$ and $F(\eta_i^{v,k})$ are generated in one time, using Algorithm 1. The detailed computational results are reported in Table II. Column "Obj" records objective function values obtained separately by branch-and-cut algorithm and Gurobi solver. Column "Gap_o" gives the relative gap between these two objective function values. Column "TL" states the numbers of tangent lines. Column "Ratio" represents the ratio of the number of tangent lines generated by Gurobi solver to that produced by branch-and-cut algorithm. Column "CPU" shows the computational time. Column "Gap_c" shows the relative gap between the computational time of branch-and-cut algorithm and Gurobi solver. In Table II, it is obvious that the branch-and-cut algorithm consistently outperforms Gurobi solver, consistently yielding superior solutions with lower objective function values across all case groups. Notably, Gurobi consistently generates a notably larger number of tangent lines compared to

TABLE II
COMPUTATIONAL RESULTS OF BRANCH-AND-CUT ALGORITHM AND GUROBI.

Case	Obj		Gap_o	TL		Ratio	CPU (s)		Gap_c
	Gurobi	BC	(%)	Gurobi	BC		Gurobi	BC	(%)
5-10	3,170,777	3,164,710	0.19	106,508	7,037	15.14	131.94	127.55	3.44
5-12	3,764,414	3,751,767	0.34	156,644	8,944	17.51	165.70	158.12	4.79
7-10	4,630,632	4,611,228	0.42	250,724	13,610	18.42	294.11	284.55	3.36
7-12	4,990,137	4,970,553	0.39	358,301	17,498	20.48	373.49	347.15	7.59
10-10	7,145,572	7,126,407	0.27	613,527	26,765	22.92	879.24	643.09	36.72
10-12	8,716,677	8,685,038	0.36	890,561	35,067	25.40	971.24	678.80	43.08
12-10	7,407,271	7,364,695	0.58	965,429	39,402	24.50	1,206.12	1,164.99	3.53
12-12	9,124,021	9,097,914	0.29	1,421,479	50,801	27.98	1,556.88	1,355.68	14.84
15-10	10,497,903	10,462,682	0.34	1,680,547	60,513	27.77	2,590.45	2,305.17	12.38
15-12	14,400,764	14,326,669	0.52	2,457,989	77,782	31.60	3,293.23	2,676.29	23.05

the branch-and-cut algorithm. Besides, the ratio of the number of tangent lines generated by Gurobi solver to that generated by branch-and-cut algorithm can exceed 25 in larger case groups, such as case groups 12-12, 15-10, and 15-12. Moreover, with the increase in the case group size, the computational times of Gurobi solver and branch-and cut algorithm all rise significantly. However, our branch-and-cut algorithm is always superior to Gurobi solver in the computational time of all cases. These results demonstrate the effectiveness and efficiency of our branch-and-cut algorithm.

D. Comparison of branch-and-cut algorithm and the greedy algorithm

TABLE III
COMPUTATIONAL RESULTS OF BRANCH-AND-CUT ALGORITHM AND GREEDY ALGORITHM.

Case	Obj		Gap	CPU (s)	
	Greedy	BC	(%)	Greedy	BC
5-10	3,949,231	3,164,710	24.79	13.95	127.55
5-12	4,854,892	3,751,767	29.40	17.27	158.12
7-10	5,912,396	4,611,228	28.22	22.38	284.55
7-12	6,251,729	4,970,553	25.78	27.67	347.15
10-10	10,171,861	7,126,407	42.73	37.98	643.09
10-12	12,778,144	8,685,038	47.13	46.38	678.80
12-10	9,134,740	7,364,695	24.03	49.68	1,164.99
12-12	11,291,717	9,097,914	24.11	62.17	1,355.68
15-10	14,158,917	10,462,682	35.33	71.41	2,305.17
15-12	22,930,158	14,326,669	60.05	88.25	2,676.29

We then compare the branch-and-cut algorithm with a heuristic method, i.e., a greedy algorithm that simulates the most commonly used "first-come-first-served" policy, where port operators typically prioritize berth allocation according to the vessels' arrival times [35], [36]. The comprehensive computational results are presented in Table III. Column "Obj" gives the objective function values. Column "Gap" calculates the relative gap between objective function values of the greedy algorithm and branch-and-cut algorithm. Column "CPU" states the computational time. Table III shows that, although the computational speed of the greedy algorithm surpasses that of the branch-and-cut algorithm across all case groups, the solutions produced by the greedy algorithm are significantly worse than those obtained by branch-and-cut algorithm. Specifically, the objective function values derived from the greedy algorithm are notably larger compared to those obtained by the branch-and-cut algorithm. Furthermore, as the size of case group increases, there is a general growing trend of relative gaps in objective function values of these two approaches. These computational results demonstrate the effectiveness of our branch-and-cut algorithm.

E. Computational performance of branch-and-cut algorithm with different numbers of ports per decision period

We further evaluate the computational performance of the proposed branch-and-cut algorithm under different numbers of visiting ports in each decision period within the rolling horizon framework. An instance from case group 10–12 is selected for this analysis, and the computational results are illustrated in Fig. 4. From Fig. 4, it can be found that increasing the number of visiting ports in each decision period leads to a reduction in the objective function value but an increase in computational time. This is because a larger number of visiting ports in each decision period enhances the likelihood of achieving a global optimal solution, thereby causing a decline in the objective function value. However, it also increases the problem size and computational complexity, resulting in longer computation times. Specifically, when the number of visiting ports increases from two to three, the objective function value drops dramatically, while the computational time increases only slightly. In contrast, as the number of visiting ports ranges from three to seven, the reduction in the objective function value is relatively moderate, whereas the computational time rises sharply.

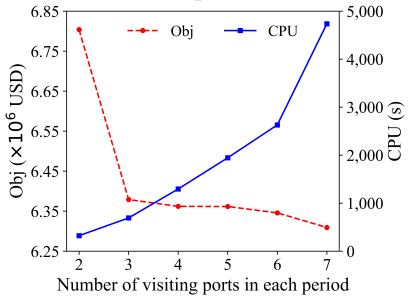

Fig. 4. Impact of the number of visiting ports in each decision period.

F. Sensitivity analysis

In this section, we conduct sensitivity analysis using the same instance from case group 10–12 to examine the

impacts of sailing time disturbances, penalty costs for arrival time delays, and fuel price.

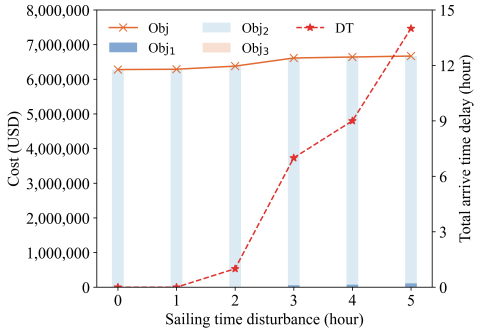

Fig. 5. Impact of the sailing time disturbance.

1) Impact of the sailing time disturbance: We investigate the impact of sailing time disturbances by varying their maximum values from 0 to 5. The detailed computational results are illustrated in Fig. 5. In Fig. 5, labels "Obj_1", "Obj_2", and "Obj_3" represent the total penalty cost for arrival time delays of AVs, total fuel consumption cost, and total additional transportation cost of containers caused by berth deviations, respectively. The label "Obj" refers to the total cost, which is the sum of the three aforementioned costs. The label "DT" gives the total arrival time delay of AVs. As shown in Fig. 5, both the total fuel consumption cost and the total cost increase as the maximum sailing time disturbance expands from 0 to 5. However, the total additional transportation cost of containers remains zero in all cases, indicating that all AVs successfully moor at their initially allocated berths without any deviations. Notably, when the maximum sailing time disturbance ranges from zero to 1, both the total arrival time delay and the total penalty cost are zero. This suggests that AVs manage to arrive punctually at ports by adopting higher sailing speeds, even at the expense of increased fuel consumption. However, as the maximum sailing time disturbance continue to expand, both the total arrival time delay and the total penalty cost grow significantly. This indicates that AVs are no longer able to avoid delays by sailing at higher speeds, as higher speeds result in disproportionately greater fuel consumption. In particular, when the maximum disturbance reaches 5, the total arrival time delay accumulates to 14 hours. These findings highlight the substantial impact of sailing time disturbances on AV operations.

2) Impact of the penalty cost for the arrival time delay: The computational results with different penalty costs for arrival time delays are illustrated in Fig. 6. When the penalty cost increases from 3,000 USD/hour to 8,000 USD/hour, the total arrival time delay decreases sharply from 437 hours to one hour. Meanwhile, the total fuel consumption cost increases accordingly. This indicates that AVs tend to sail at lower speeds to reduce fuel consumption when the penalty cost is relatively low, but this comes at the expense of significant deviations from their expected arrival times. As the penalty cost increases beyond 8,000 USD/hour, the total arrival time delay stabilizes at one hour. Notably, the total cost first decreases as the penalty cost increases from 3,000 USD/hour to 8,000 USD/hour, but begins to increase as the penalty cost continues to grow. These findings highlight the importance of appropriately setting the penalty cost to ensure the punctual arrival of AVs at ports while maintaining a relatively low total cost.

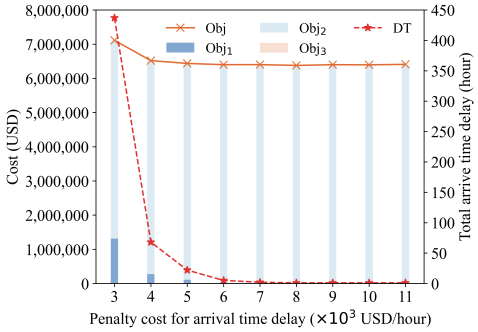

Fig. 6. Impact of the penalty cost for the arrival time delay.

3) Impact of the fuel price: Fig. 7 presents the computational results with varying fuel prices. As the fuel price increases from 550 USD/ton to 800 USD/ton, the total arrival time delay of AVs remains constant at one hour, while the total fuel consumption cost rises accordingly. This indicates that the punctuality of AVs arriving at ports can be maintained although the total fuel consumption cost increases to a certain extent. However, when the fuel price exceeds 800 USD/ton, both the total arrival time delays and the total fuel consumption cost experience an increase. This suggests that under excessively high fuel prices, AVs tend to adapt lower sailing speeds to save fuel consumption cost, which in turn compromises their punctuality at ports. These findings highlight the trade-off between the total penalty cost for arrival time delays (reflecting the punctuality) and the total fuel consumption cost.

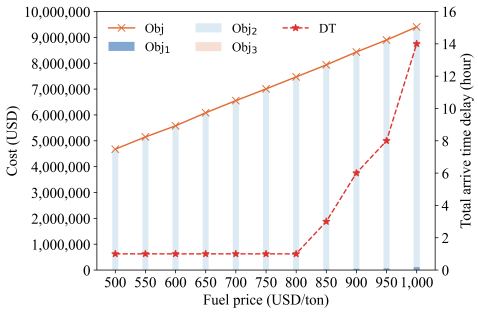

Fig. 7. Impact of the fuel price.

VI. Conclusions

In this study, we focus on a dynamic AV speed optimization and berth reallocation problem under the collaboration between liner shipping companies and port operators at the operational stage, which aims at mitigating the impact of stochastic disturbances on the designed schedule. Specifically, sailing speeds of AVs at each leg are dynamically adjusted to reduce the arrival time delays caused by stochastic disturbances. Besides, when the allocated berths are unavailable for AVs due to arrival time

delay, the berth for AVs are reallocated. To meet real-time operational requirements, a rolling horizon framework is employed. In each decision period, an MINLP model is formulated to optimize decisions about sailing speeds and berth reallocation, with the objective of minimizing the total cost. Several linearization techniques are applied to transform the MINLP model into the MILP model. Then, a tailored branch-and-cut algorithm is developed to solve the MILP model, where cutting planes are dynamically generated by an outer-approximation method.

The efficiency and effectiveness of proposed model and solution algorithm are evaluated through extensive numerical experiments. The computational results demonstrate that our model and solution algorithm can efficiently solve the problem and provide a reliable solution. We further validate the effectiveness of our proposed algorithm by comparing it with Gurobi solver and a greedy algorithm simulating the "first-come-first-served" principle. The computational results indicate that the proposed algorithm outperforms both the Gurobi solver and the greedy algorithm in terms of solution quality. Finally, we perform sensitivity analyses to examine the impacts of the sailing time disturbance, the penalty cost for the arrival time delay, and the fuel price. The experimental results provide several valuable managerial insights. First, greater sailing time disturbances significantly increase the total arrival time delay, especially when the disturbance exceeds a certain threshold. Second, lower penalty costs for the arrival time delay may encourage AVs to adopt more energy-efficient sailing speeds, which helps reduce the total fuel consumption costs but compromises their punctuality at ports. Third, higher fuel prices lead AVs to prioritize fuel efficiency by sailing at lower speeds, which in turn reduces their punctuality at ports.

There are several potential directions for future research. First, this study does not consider quay crane assignments, which may affect the cargo handling efficiency of AVs and disrupt their schedules. A natural extension of this work would be to integrate speed optimization, berth reallocation, and quay crane scheduling into a unified optimization framework. Second, the optimization problem in each decision period is deterministic, which may not fully captured the uncertainties of disturbances. To more accurately reflect such uncertainties, future research could consider robust or stochastic programming formulations. Third, although the proposed branch-and-cut algorithm is capable of producing high-quality solutions for both small- and large-scale instances, its computational efficiency becomes limited when solving large-scale instances. Future work could explore more efficient solution methods.

References

[1] UNCTAD, "Review of maritime transport 2022. United Nations Conference on Trade and Development," 2022, https://unctad.org/rmt2022.

[2] X. Yang, W. Gu, W. Wang, and S. Wang, "Optimal scheduling of autonomous vessel trains in a hub-and-spoke network," *Ocean Coast. Manage.*, vol. 231, p. 106386, Jan. 2023.

[3] W. Zhang and S. Wang, "Autonomous vessel scheduling," *J. Oper. Res. Soc. China.*, vol. 8, pp. 391–414, Oct. 2020.

[4] The Maritime Executive, "Autonomous vessels are becoming a commercial reality," 2021, https://maritime-executive.com/editorials/autonomous-vessels-are-becoming-a-commercial-reality.

[5] R. R. Negenborn, F. Goerlandt, T. A. Johansen, P. Slaets, O. A. Valdez Banda, T. Vanelslander, and N. P. Ventikos, "Autonomous ships are on the horizon: here' s what we need to know," *Nature*, vol. 615, no. 7950, pp. 30–33, Feb. 2023.

[6] G. Venturini, Ç. Iris, C. A. Kontovas, and A. Larsen, "The multi-port berth allocation problem with speed optimization and emission considerations," *Transport. Res. D-Tr. E.*, vol. 54, pp. 142–159, Jul. 2017.

[7] M. A. Dulebenets, "Minimizing the total liner shipping route service costs via application of an efficient collaborative agreement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 123–136, 2018.

[8] B. Martin-Iradi, D. Pacino, and S. Ropke, "The multiport berth allocation problem with speed optimization: Exact methods and a cooperative game analysis," *Transport. Sci.*, vol. 56, no. 4, pp. 972–999, Feb. 2022.

[9] Maersk, "Maersk, Logistics made easy through digital solutions," 2024, https://www.maersk.com/.

[10] APM, "APM terminals," 2024, https://www.apmterminals.com/.

[11] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Closed-loop scheduling and control of waterborne agvs for energy-efficient inter terminal transport," *Transport. Res. E-Log.*, vol. 105, pp. 261–278, Sep. 2017.

[12] Y. Du, Q. Meng, S. Wang, and H. Kuang, "Two-phase optimal solutions for ship speed and trim optimization over a voyage using voyage report data," *Transport. Res. B-Meth.*, vol. 122, pp. 88–114, 2019.

[13] Y. Wang, Q. Meng, and H. Kuang, "Jointly optimizing ship sailing speed and bunker purchase in liner shipping with distribution-free stochastic bunker prices," *Transport. Res. C-Emer.*, vol. 89, pp. 35–52, Apr. 2018.

[14] Y. Wang and S. Wang, "Deploying, scheduling, and sequencing heterogeneous vessels in a liner container shipping route," *Transport. Res. E-Log.*, vol. 151, p. 102365, Jul. 2021.

[15] D. Sheng, Q. Meng, and Z.-C. Li, "Optimal vessel speed and fleet size for industrial shipping services under the emission control area regulation," *Transport. Res. C-Emer.*, vol. 105, pp. 37–53, Aug. 2019.

[16] L. Zhen, Z. Hu, R. Yan, D. Zhuge, and S. Wang, "Route and speed optimization for liner ships under emission control policies," *Transport. Res. C-Emer.*, vol. 110, pp. 330–345, Jan. 2020.

[17] A. Imai, K. Nagaiwa, and C. W. Tat, "Efficient planning of berth allocation for container terminals in Asia," *J. Adv. Transport.*, vol. 31, no. 1, pp. 75–94, Jan. 1997.

[18] M. A. Dulebenets, "A diffused memetic optimizer for reactive berth allocation and scheduling at marine container terminals in response to disruptions," *Swarm Evol. Comput.*, vol. 80, p. 101334, 2023.

[19] C. Chen, F. Wang, J. Pan, L. Xu, and H. Gao, "Algorithm design for an online berth allocation problem," *J. Mar. Sci. Eng.*, vol. 12, no. 10, p. 1722, 2024.

[20] B. Li, P. Afkhami, R. Khayamim, M. Borowska-Stefańska, S. Wiśniewski, A. M. Fathollahi-Fard, S. Ozkul, and M. A. Dulebenets, "An intelligent hyperheuristic algorithm for the berth allocation and scheduling problem at marine container terminals," *Transport. Res. E-Log.*, vol. 198, p. 104104, 2025.

[21] C. Bierwirth and F. Meisel, "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals," *Eur. J. Oper. Res.*, vol. 244, no. 3, pp. 675–689, Aug. 2015.

[22] F. Rodrigues and A. Agra, "Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey," *Eur. J. Oper. Res.*, vol. 303, no. 2, pp. 501–524, Dec. 2022.

[23] Y. Du, Q. Chen, X. Quan, L. Long, and R. Y. Fung, "Berth allocation considering fuel consumption and vessel emissions," *Transport. Res. E-Log.*, vol. 47, no. 6, pp. 1021–1037, Nov. 2011.

[24] Y. Du, Q. Chen, J. S. L. Lam, Y. Xu, and J. X. Cao, "Modeling the impacts of tides and the virtual arrival policy in berth allocation," *Transport. Sci.*, vol. 49, no. 4, pp. 939–956, Feb. 2015.

[25] Q.-M. Hu, Z.-H. Hu, and Y. Du, "Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels," *Comput. Ind. Eng.*, vol. 70, pp. 1–10, Apr. 2014.
[26] J. Yu, G. Tang, and X. Song, "Collaboration of vessel speed optimization with berth allocation and quay crane assignment considering vessel service differentiation," *Transport. Res. E-Log.*, vol. 160, p. 102651, Apr. 2022.
[27] L. Zhen, Y. Wu, S. Wang, and G. Laporte, "Green technology adoption for fleet deployment in a shipping network," *Transport. Res. B-Meth.*, vol. 139, pp. 388–410, Sep. 2020.
[28] S. Wang and Q. Meng, "Sailing speed optimization for container ships in a liner shipping network," *Transport. Res. E-Log.*, vol. 48, no. 3, pp. 701–714, May 2012.
[29] R. Yan, S. Wang, and Y. Du, "Development of a two-stage ship fuel consumption prediction and reduction model for a dry bulk ship," *Transport. Res. E-Log.*, vol. 138, p. 101930, Jun. 2020.
[30] L. Zhang, W. Shan, B. Zhou, and B. Yu, "A dynamic dispatching problem for autonomous mine trucks in open-pit mines considering endogenous congestion," *Transport. Res. C-Emer.*, vol. 150, p. 104080, May 2023.
[31] L. Zhang, Q. Meng, H. Wang, and B. Yu, "Joint bus dispatching and bus bridging timetabling for mass rapid transit disruption management," *Transport. Res. B-Meth.*, vol. 196, p. 103215, 2025.
[32] L. Zhang, Z. Liu, B. Yu, and J. Long, "A ridesharing routing problem for airport riders with electric vehicles," *Transport. Res. E-Log.*, vol. 184, p. 103470, 2024.
[33] Zhonggu Logisitics, "Internal trade voyage of Zhonggu Logistics," 2023, https://www.zhonggu56.com/pages/ourService/internalTradeVoyage.jsp.
[34] Ports.com, "Sea route and distance," 2023, http://ports.com/sea-route/.
[35] L. Guo, J. Zheng, J. Liang, and S. Wang, "Column generation for the multi-port berth allocation problem with port cooperation stability," *Transport. Res. B-Meth.*, vol. 171, pp. 3–28, May 2023.
[36] L. Zhen, D. Zhuge, S. Wang, and K. Wang, "Integrated berth and yard space allocation under uncertainty," *Transpor. Res. B-Meth.*, vol. 162, pp. 1–27, Aug. 2022.

Tingting Chen is a Ph.D. student in Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. Her current research interests focus on the scheduling of autonomous vessel and electric vessel in the maritime transportation.

Lingxiao Wu is an assistant professor in Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. His research interests lie in the design and implementation of methods that combine operations research and data analytics for solving problems arising in logistics management and transportation. His research has appeared in leading international journals, including Operations Research, Transportation Research Part B/E, and Transportation Science.

Shuaian Wang is a professor in Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University. His research interests include big data in shipping, green shipping, shipping operations management, port planning and operations, and logistics and supply chain management. He has published over 300 papers in Transportation Research Part A/B/C/D/E, Transportation Science, Management Science, and Operations Research.