Dynamic Tugboat Deployment and Scheduling with Stochastic and Time-Varying Service Demands

Xiaoyang Wei [a], Shuai Jia [b], Qiang Meng [c], Jimmy Koh [d]*

[a] *Institute of High Performance Computing, Agency for Science, Technology and Research (A*STAR), Singapore 138632*

[b] *Intelligent Transportation Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China 511400*

[c] *Department of Civil and Environmental Engineering, National University of Singapore, Singapore 117576*

[d] *Tugboat Mission Command Center, PSA Marine, Singapore 126800*

Abstract

Container ports serve as crucial logistics hubs in global supply chains, but navigating ships within such ports is complex due to restricted waterways. Tugboats play a critical role in ensuring safety and efficiency by escorting and towing ships under these conditions. However, the tugboat deployment and scheduling problem has received little attention. To fill the research gap, we propose a new research problem - *the dynamic tugboat deployment and scheduling problem*, in which not all requests are confirmed initially but dynamically confirmed over time and future tugging demands need to be anticipated when managing the utilization of tugboats. To formulate the problem, we propose an extended Markov decision process (MDP) that incorporates both reactive task assignment decisions and proactive tugboat waiting decisions, creating a reactive and proactive MDP. To solve the advanced MDP model efficiently for real-time decisions, we develop an anticipatory approximate dynamic programming method that incorporates appropriate task assignment and waiting strategies for deploying and scheduling a heterogeneous tugboat fleet and embed the method into an improved rollout algorithm to anticipate future scenarios. The effectiveness, efficiency, and performance sensitivity of the developed modeling and solution methods are demonstrated via extensive numerical experiments for the Singapore container port.

Keywords: Dynamic and stochastic programming; Markov decision process; Anticipation algorithm; Tugboat deployment and scheduling; Port transportation.

---

* Corresponding author, E-mail: ceemq@nus.edu.sg, Tel.: +65 6516 5494.

1. Introduction

Container ports play a pivotal role in global logistics, managing the transit of millions of twenty-foot equivalent units annually (Lee and Song, 2017). In bustling ports, the ability of ships to maneuver is often constrained by the narrow and shallow waterways, complicating the management of ship course and speed within port waters. This restricted maneuverability necessitates the use of tugboats, which perform several crucial operations such as towing, pushing, pulling, and escorting ships through these confined passages to ensure the safety and efficiency of ship movements during their arrival and departure. Given the high operational costs associated with tugboats, optimizing their deployment and scheduling is crucial for enhancing the cost-effectiveness of port operations. Despite its significant industrial importance, the tugboat deployment and scheduling problem (TDSP) has not been well studied.

Tugboat companies face two primary challenges. First, service demands exhibit dynamicity. The service requests throughout the service time horizon may be not confirmed initially when a tugboat deployment and scheduling plan is being devised. Thus, the plan needs to be updated as new service demands arise over time. Second, future request arrivals and the associated attributes (e.g., the requested service start time, service origin-destination, type and number of tugboats for each ship) pose stochasticity. Though the ship's estimated time of arrival (ETA) and estimated time of departure (ETD) are sent to the port in advance according to contractual obligations, stochasticity in request attributes is incurred by a variety of unexpected factors (e.g., weather, congestion in ship traffic and terminal quay/yard operations, mechanical issues in ships and quay/yard machines). For example, in 2019, the global container ship on-time performance was only 47%, where on-time means the delay is less than eight hours (Zhang et al., 2022). Thus, future service demand occurs stochastically in time and space, with heterogeneous demand on the type and number of tugboats for each ship. Anticipating potential demand scenarios is essential for effective tugboat fleet operations.

In this paper, we propose and study a dynamic tugboat deployment and scheduling problem (DTDSP), which aims to minimize total expected tugboat operation costs while satisfying all service demands. The DTDSP entails deploying tugboats dynamically to meet the needs of ships entering or leaving the port and devising detailed work plans for tugboats, including service start and end times for each ship. These decisions must navigate operational constraints, such as time-varying service demands that reflect the fluctuating need for tugboat services throughout a ship's service duration.

To formulate this complex problem, we extend the basic Markov decision process (MDP) framework to develop a reactive and proactive MDP (RPMDP). Motivated by the need to manage dynamic decisions under stochastic service demands, the RPMDP integrates routing and waiting decisions to reduce operational costs. This model enhances responsiveness with reactive task assignments for immediate tugboat deployment and incorporates proactive waiting strategies that position tugboats advantageously, anticipating imminent service demands to minimize unnecessary movement and enhance efficiency. The RPMDP significantly advances the route-based MDP model proposed by Ulmer et al. (2020) by effectively integrating both routing and waiting strategies.

To solve the problem, we employ approximate dynamic programming (ADP) combined with strategic heuristics such as Tabu search to optimize the deployment and scheduling of a heterogeneous tugboat fleet. We thus develop an anticipatory ADP (AADP) method that evaluates post-decision states with an estimation of expected future values, enabling efficient real-time decision-making. This method is further enhanced by integrating an advanced rollout algorithm that anticipates future service scenarios, offering an effective and efficient method for addressing the dynamic and stochastic challenges inherent in tugboat operations.

Through innovation in both modeling and solution methodology, we contribute a novel and practical approach to the field of maritime transportation, addressing a critical gap in existing research on tugboat deployment and scheduling. This study not only advances theoretical understanding but also provides actionable strategies for improving the efficiency and cost-effectiveness of tugboat operations in busy container ports worldwide. To the best of our knowledge, this work is the first to address tugboat deployment and scheduling in a dynamic and stochastic setting.

1.1. Literature review

The DTDSP represents a nascent research area. Its distinctive feature, the pattern of time-varying service demand (where a ship's need for tugging service changes throughout its service duration) differs the DTDSP from existing dynamic and stochastic deployment and scheduling problems in the literature.

Intuitively, the DTDSP may resemble the dial-a-ride problem (DARP) (Liu et al., 2015), as both involve deploying a fleet of vehicles (or tugboats) to transport (or assist) customers (or ships) from origins to destinations. However, they differ in two main aspects: (i) In the DARP, each customer is served by one vehicle, while in the DTDSP, a ship may require multiple compatible tugboats. (ii) The DTDSP demands route coordination among multiple tugboats serving the same ship, while the DARP does not require such coordination for a single customer. The DTDSP also bears similarity to the split delivery problem (SDP) (Lee et al., 2006), where an order can be fulfilled by multiple vehicles. The key differences are: (i) Vehicles that serve the same order in the SDP can deliver the cargo independently, without coordination among the vehicles in routes and timing, while the DTDSP requires route and timing coordination for tugboats assisting the same ship. (ii) In the SDP, each customer has a single and fixed location and available vehicles are free to visit customer locations; in the DTDSP, the tugboat should sail along a sequence of waypoints when serving a ship. These comparisons highlight the DTDSP's unique characteristics, making it a distinct research problem, and necessitating appropriate modeling and solution methods for satisfactory solutions. In a broader context, the literature on dynamic and stochastic routing, recently comprehensively reviewed by Soeffker et al. (2022), provides valuable reference. Readers seeking a thorough review can consult this work. For the sake of conciseness, we will not delve into a redundant review. Instead, our subsequent discussion emphasizes the research progress of tugboat deployment and scheduling.

We review the relevant studies on tugboat deployment and scheduling and identify research gaps. Liu (2009) formulates the TDSP using a multiprocessor scheduling model, in which tugboats and tugging services are treated as processors and tasks, respectively. An evolutionary heuristic method is developed to generate feasible solutions. Wang et al. (2010) study the TDSP aiming to provide tugging service for a set of ships to minimize the total ship waiting time. A particle swarm optimization (PSO) method and a chaos search heuristic are applied to calculate feasible solutions. Wang et al. (2012) build a mixed-integer linear programming (MILP) model for the TDSP, minimizing the latest finish time of tugging services. An enumeration method is used to solve toy-size instances with no more than five ships. Xu et al. (2012) treat the TDSP as a multiprocessor scheduling problem, minimizing the total tugging service time. A heuristic method integrating ant colony mechanism and simulated annealing is developed to solve instances of up to thirty ships. Ilati et al. (2014) investigate a port resource allocation problem that integrates berth allocation, tugboat deployment, and quay crane assignment. A heuristic method using path-relinking is applied to solve the problem. Wang et al. (2014) determine the sequence of ships to be served by tugboats using two rule-based strategies: first-come-first-served rule and first-fit rule. A particle swarm heuristic is applied to determine tugboat work plans.

Tugboat deployment and scheduling have recently been a notable research topic. Zhen et al. (2018) develop a column generation-based heuristic for scheduling tugboats to serve hinterland barges. Their problem is different from the TDSP in container ports, since their problem operates tugboats in inland waterways and the tugboat operation for barges is different from that for container ships. Wei et al. (2020) investigate a deterministic TDSP, assuming all service demands are known exactly at the beginning of service time horizon. In their model, each ship trajectory is modeled using a set of waypoints along a waterway, and each waypoint has a designated time window for tugboats to provide service. Kang et al. (2020) study a TDSP considering time uncertainty in ship arrivals and tugging services, assuming all service demands are known exactly in advance and only one tugboat is needed for each ship. Abou Kasm et al. (2021) schedule the ship traffic with pilotage and tugging constraints. In their problem, homogeneous tugboats are assigned to ships aiming to minimize the maximum ship waiting time. Jia et al. (2021) schedule tugboats for pushing and towing ships at berths. Their problem aims to optimize the allocation of tugboat horsepower to ships near berths, while the tugging service in waterways is not considered. Wei et al. (2021) investigate the dynamicity of service demands by assuming some service demands are not known when the service time horizon begins, however, anticipation of future service demands is not considered and tugging service for ship unberthing operations is simplified. Petris et al. (2022) introduce models and heuristics for the in-port ship scheduling and tugboat assignment problem. They propose two mathematical programming models and derive four heuristics, showing that receding horizon-based heuristics offer good-quality solutions within reasonable computational times. Zhong et al. (2022) develop a bi-objective tugboat scheduling model to minimize completion times and fuel consumption, using a non-dominated sorting genetic algorithm. Their model illustrates the balance between operational efficiency and environmental sustainability. Hao et al. (2023) tackle a barge and tugboat joint scheduling problem at a river-sea intermodal transport system. Their integer programming model optimizes the efficiency of bulk cargo transshipment between inland river ports and sea platforms, demonstrating significant improvements in transfer efficiency through joint scheduling. Wang et al. (2023) study the tugboat scheduling problem at seaports, emphasizing the importance of scheduling tugboats efficiently due to their role in ensuring safety and reducing operational costs. They develop an adaptive large neighborhood search algorithm to handle large-scale instances effectively, providing valuable managerial insights through sensitivity analysis. Yang et al. (2024) propose a novel berth-tugboat-quay crane joint scheduling method to reduce the economic costs at container ports. Their model integrates chaotic mapping and quantum computing within a new optimizer algorithm, demonstrating the model's efficacy through simulation tests in ports.

Table 1 summarizes existing works on the TDSP. These works typically assign tugboats to ships without specifying tugboat routes and timing (e.g., a particular time of departure from each location). Moreover, no work to date has investigated the anticipation of future service demands. Furthermore, existing works depend on one or more of the following simplifying assumptions: (a) the tugboat transfer time between two consecutive tasks is ignored; (b) ships are served following the first-come-first-served rule; (c) only berthing ships need tugging service, or the tugging service for unberthing ships is largely simplified; (d) tugboats are homogeneous in terms of tugging power or sail speed; (e) berth locations are identical, thus the tugboat transfer times among berths are negligible; (f) the service time for each ship simplistically depends on the ship length; and (g) only one tugboat is needed (or, a same number of tugboats are needed) for each ship, regardless of the ship size. Notably, the DTDSP has not been studied, which differs from existing works in four aspects: (1) relaxing all the above seven assumptions to make the problem realistic; (2) anticipating future service demands by exploiting the stochastic information to improve decision-making; (3) in addition to deploying tugboats for ships, optimizing tugboat routing and waiting plans with explicit tugboat departure time from each location to provide service; (4) incorporating practical tugboat operation features such as the pattern of time-varying service demand.

Table 1. Comparison between this work and existing works on the TDSP.

Work	Objective	Assumptions used	Model	Solution method	Stochastic service demands and anticipation	Routes with timing
Liu (2009)	Min. the latest service end time	(a), (b), (c)	MSM*	MH*	No	No
Wang et al. (2010)	Min. the total ship waiting time	(a), (b), (c)	MILP	MH	No	No

Wang et al. (2012)	Min. the latest service end time	(a), (b), (c)	MILP	Enumeration	No	No
Xu et al. (2012)	Min. the total service time	(a), (d)	MSM	MH	No	No
Ilati et al. (2014)	Min. the total ship waiting time	(b), (c), (e)	Simulation	MH	No	No
Wang et al. (2014)	Min. the latest service end time	(a), (b), (c), (f)	MILP	MH	No	No
Zhen et al. (2018)	Min. tugboat sail cost and lateness	(d), (f)	MILP	Branch-price	No	No
Kang et al. (2020)	Min. service time and recovery cost	(c), (g)	MILP	CPLEX solver	No	No
Wei et al. (2020)	Min. tugboat sail cost	(c)	MILP	Branch-cut	No	Yes
Abou Kasm et al. (2021)	Min. the maximum tugboat lateness	(d), (e)	MILP	CS*	No	No
Jia et al. (2021)	Min. ship tardiness and tugboat operation cost	(f)	MILP	LR*	No	No
Petris et al. (2022)	Min. the total deviation from the planned movement schedules	(d)	MILP	MH	No	Yes
Zhong et al. (2022)	Min. maximum completion time and total fuel consumption of tugboats	(a)	MILP	MH	No	No
Hao et al. (2023)	Min. time taken for bulk cargo transshipment	(d), (g)	MILP	MH	No	Yes
Wang et al. (2023)	Min. the total cost (including travel and fixed costs)	(d)	MILP	MH	No	Yes
Wei et al. (2023)	Min. tugboat operation cost	(c)	MILP	MH	No	No
Yang et al. (2024)	Min. fuel costs of tugboat operations	(c), (d), (g)	MILP	MH	No	No
This work	**Min. total tugboat operation cost in expectation**	**Relax all the above (a) - (g)**	**MDP**	**MH and improved rollout**	**Yes**	**Yes**

*Notes: MSM - multiprocessor scheduling model; MH - meta-heuristic; CS - constraint separation; LR - Lagrangian relaxation.

1.2. Contributions

The main contributions of this study are four-fold.

(i) **New Problem with Challenges:** This is the first work to propose and study the DTDSP, arising from daily port transportation and holding industrial importance. This work considers the dynamicity and stochasticity of service demands, as well as anticipation of future demands to enhance the current decision-making on tugboat work plans. Furthermore, the pattern of time-varying service demand distinguishes the DTDSP from existing dynamic and stochastic problems, marking it as a novel problem with unique challenges in solution methodology.

(ii) **Advancements in Modelling and Solution Methodology:** We significantly advance the traditional MDP by developing an RPMDP modeling framework, which incorporates proactive waiting decisions alongside routing decisions. This enhancement allows tugboats to reposition to promising locations in anticipation of near-future demands, thereby reducing non-towage sailing and enhancing operational efficiency. The motivation for integrating routing and waiting decisions into the RPMDP stems from the necessity to manage dynamic decisions under stochastic service demands efficiently, reduce operational costs, and ensure alignment with industry practices and regulatory standards. These enhancements are crucial for optimizing task assignments within a heterogeneous tugboat fleet, enabling more effective management of potential demand scenarios. To solve the RPMDP, we develop a solution methodology that integrates AADP with strategic heuristics, optimizing both the cost and efficiency of tugboat fleet operations.

(iii) **Optimization of Solution Algorithms:** To enhance the efficiency of solution process for dynamic decisions in real-time with anticipations, motivated by the indifference zone selection (IZS) idea, we developed the IZS-based two-phase rollout (IZS-2-R) algorithm. The IZS-2-R algorithm efficiently terminates the evaluation of dominated decisions - those significantly underperforming compared to others - facilitating a more computationally tractable approach to solving the DTDSP. This capability is crucial for managing future service demands within the stringent computation time limits required for real-time decision-making.

(iv) **Competitive Benchmarking and Practical Importance:** Our modeling and solution approaches are rigorously compared with three benchmark methods (see Section 5.2), including the current method employed by the tugboat industry. This comparison not only highlights the superior cost-efficiency of our methods but also offers practical importance for tugboat operations, as evidenced by extensive numerical experiments with real-world data from the representative Singapore container port. This comparative analysis underscores the effectiveness of our methodology in reducing operational costs and enhancing decision-making in dynamic and stochastic environments.

The remainder of this paper is organized as follows. Section 2 details the DTDSP. Section 3 presents the RPMDP modeling method. Section 4 elaborates on the solution method. Section 5 discusses numerical experiments, results, and managerial insights. Section 6 concludes the work and suggests future research directions. For ease of presentation, notations, the benchmark method, stochasticity statement, and solution examples are presented in Appendices.

2. Problem Description

This section provides a comprehensive overview of the research problem. Initially, we introduce a representative case from the Singapore container port to contextualize the problem, followed by essential assumptions. Finally, we present a formal problem definition with notations.

2.1. Tugboat deployment and scheduling

Tugging service is mandatory in most container ports, including the Singapore container port, which serves as our case study for the DTDSP. In the port, ships navigate through restricted water areas at low speeds, limiting their maneuverability. Consequently, tugboats are essential for escorting and towing ships along designated paths. Ships are categorized into three types based on their movements: (i) berthing ships moving from the port boundary to designated berths, (ii) unberthing ships moving from berths to the port boundary, and (iii) shifting ships moving between berths within the port. These movements, captured from the ship's automatic identification system (AIS) data, are depicted in Figure 1. Additionally, tugboats are categorized by their tugging power (see Table 2), and the ship's length and side-thruster condition dictate the required type and number of tugboats (see Table 3). A tugboat must have at least the tugging power required by the ship, although using more powerful tugboats increases operational costs.

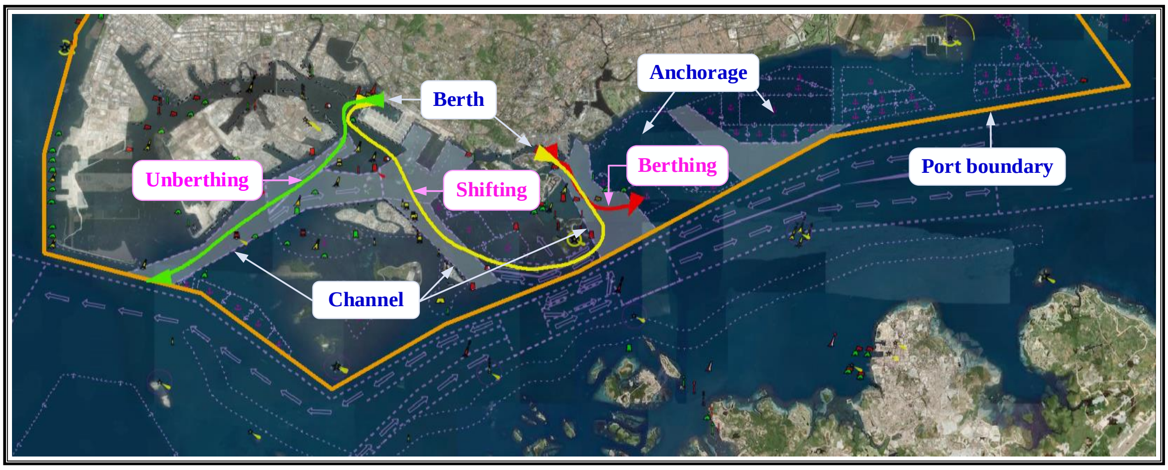


Figure 1. The Singapore container port areas and ship movements in port waters.

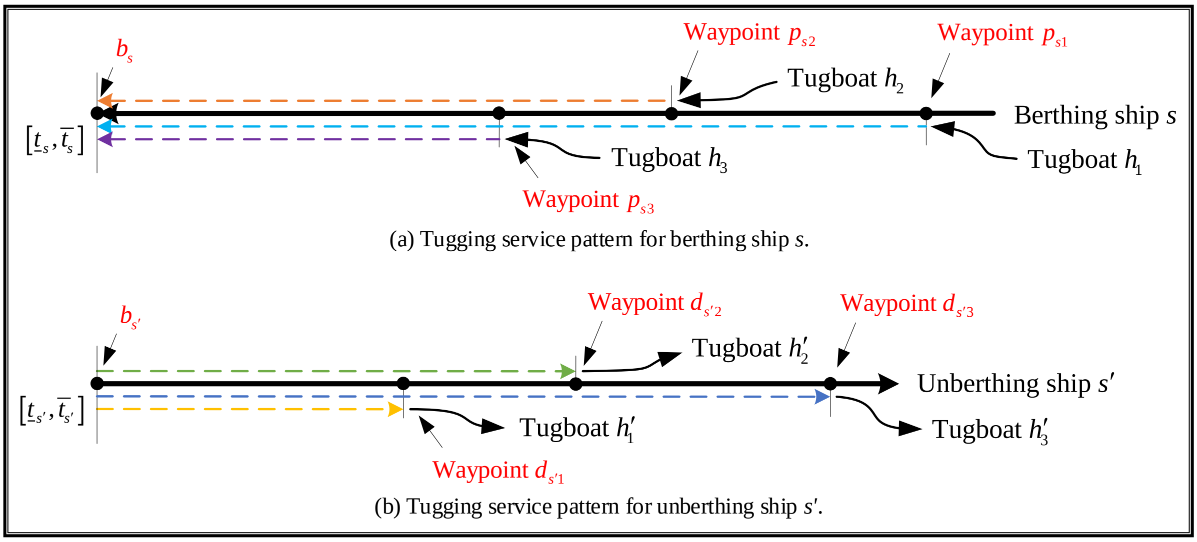
Table 2. Four tugboat types used in the Singapore container port.

Tugboat type	Tugging power	Tugboat type	Tugging power
Small (S)	10 to 16 tonnes	Big (B)	25 to 45 tonnes
Medium (M)	17 to 24 tonnes	Extra big (X)	Above 45 tonnes

Table 3. Tugboat usage guideline adopted in the Singapore container port.

Ship size in length	Type and number of tugboats needed	Adjustment
Up to 100 meters	1 type-S tugboat	One less tugboat can be used if the ship has a side-thruster in good working condition
101 to 180 meters	2 type-M tugboats	
181 to 299 meters	2 type-B tugboats	
300 meters and above	2 type-B tugboats	One more tugboat is needed if the ship has no side-thruster in good working condition

The tugging service for different ship movement types includes berthing, unberthing, and shifting, each requiring specific tugboat arrangements. During berthing, a ship sails from the port boundary to its designated berth, gradually reducing speed and maneuverability as it navigates restricted waters, necessitating an increased number of tugboats as it nears the berth (see Figure 2(a)). The tugboats must match the types and numbers outlined in Table 3. For unberthing, a ship moves from its berth to the port boundary, escorted by tugboats as specified in Table 3. As the ship exits the restricted areas and increases speed, its maneuverability improves, reducing the number of required tugboats (see Figure 2(b)). Shifting involves a ship moving from one berth to another, essentially combining an unberthing and a subsequent berthing sequence, requiring sequential adjustments in tugboat support.



(a) Tugging service pattern for berthing ship s.

(b) Tugging service pattern for unberthing ship s'.

Figure 2. The pattern of time-varying service demand.

Time-varying service demand is characterized using waypoints that divide a ship's path into segments (see Figure 2). The number of waypoints equals the number of tugboats needed at the ship's berth. Each berthing ship has a specific time window to reach the berth, and each unberthing ship has a departure window. This demand pattern is driven by operational and safety considerations: (i) As a ship approaches its berth, the number of assisting tugboats increases for safety; as a ship departs, the number of assisting tugboats decreases for optimizing costs. (ii) Operational protocols dictate that only one labor team manages the tying (or untying) of the rope between the ship and the tugboat, leading to staggered arrival (or departure) times for tugboats, typically with intervals of ten minutes or more as needed. Therefore, the required tugboats do not arrive at a berthing ship (or leave from an unberthing ship) at the same time. The pattern of time-varying service demand is common in container ports and has been evidenced in other container ports (Chen et al., 2020). This pattern requires coordination among tugboats serving a single ship, and each tugboat's operation (e.g., timing) is influenced by its earlier operations and, in turn, affects its later operations. Moreover, each tugboat operates in coordination with other tugboats to serve additional ships, alterations in one tugboat's work plan (e.g., timing) will trigger cascading effects across the fleet. This complex coordination highlights the unique challenges of the DTDSP, setting it apart from other dynamic and stochastic problems, as it demands meticulous optimization of work plans to handle these intricacies effectively.

Tugging service demands exhibit significant dynamicity and stochasticity. Requests may be confirmed less than two hours prior to service, as indicated by tugboat company PSA Marine (2023), contrasting sharply with the typical 14-hour service time horizon. This necessitates the dynamic incorporation of requests into the ongoing tugboat work plan, often accommodating emergencies and special services at short notice. Furthermore, attributes of future service requests such as start times, origin-destination, and the type and number of required tugboats exhibit stochasticity due to factors like weather, congestion, and mechanical issues, despite advanced notifications of ships' ETA and ETD (Yu et al., 2018; Xiang et al., 2021; Rodrigues et al., 2022; Zhang et al., 2022). For instance, in 2019, the global container ship on-time performance was only 47%, where on-time means the delay is less than eight hours (Zhang et al., 2022). Table 4 illustrates the stochasticity of ship arrival times at the Singapore container port, highlighting the unpredictable nature of service demand attributes (e.g., the required service start time). The stochasticity is further explained in Appendix C.

Table 4. The stochasticity of ship arrival times in the Singapore container port in June 2019.

Difference between the actual ship arrival time and the latest ETA (minute)	**[-15, 15]**	**[-30, 30]**	**[-45, 45]**	**[-60, 60]**
Ratio to the total ship arrivals	46.2%	65.1%	75.8%	80.7%

2.2. Assumptions

The following assumptions reflect real-world conditions with facilitating mathematical modeling:

(i) *Commitment of tugboats to ships:* Once a tugboat engages with a ship for berthing or unberthing, it remains committed until the ship reaches its destination or designated waypoint, respectively. An unberthing movement starts only when all required tugboats are present at the berth. This reflects the continuity and safety requirements in tugboat operations based on practices.

(ii) *Treatment of shifting service requests:* Shifting service, which involves consecutive unberthing and berthing movements, is treated as sequential combinations of these two service types (see Figure

1). Given that shifting movements constitute about 3% of all movements at the Singapore container port, this assumption facilitates the model without significantly affecting operational accuracy. This treatment aligns with practical approaches in tugboat operational practices.

(iii) *Capability of tugboat fleet:* It is assumed that the tugboat fleet during the service time horizon is sufficient to meet all service demands. This assumption aligns with the port's strategic approach to maintaining an ample fleet to ensure uninterrupted service, crucial for sustaining smooth traffic flow and safety standards within the port. This reflects the proactive management of resources typical in port settings.

2.3. Formal statement with notations

The DTDSP is defined as follows, utilizing notations summarized in Appendix A.

Ships and service demands. Denote S_{ber} and S_{unb} as the sets of berthing ships and unberthing ships, respectively, which request tugging services during a time period $[0,T]$. Define $S = S_{\text{ber}} \cup S_{\text{unb}}$. For ease of presentation, let $s \in S$ denote the service request made by ship $s \in S$. For berthing ship $s \in S_{\text{ber}}$, Γ_s tugboats are required to escort it along a sequence of given waypoints $\mathcal{P}_s = \{p_{s1}, p_{s2}, ..., p_{s\Gamma_s}\}$ to its designated berth b_s (see Figure 2 (a)). Define $\mathcal{P} = \cup_{s \in S_{\text{ber}}} \mathcal{P}_s$. The berthing ship s must arrive at berth b_s within the predefined time window $[\underline{t}_s, \overline{t}_s]$. For unberthing ship $s \in S_{\text{unb}}$, Γ_s tugboats are needed to serve it from berth b_s along a sequence of waypoints $\mathcal{D}_s = \{d_{s1}, d_{s2}, ..., d_{s\Gamma_s}\}$ (see Figure 2 (b)). Define $\mathcal{D} = \cup_{s \in S_{\text{unb}}} \mathcal{D}_s$. Unberthing ship s should leave berth b_s within its specified time window $[\underline{t}_s, \overline{t}_s]$. Let $\mathcal{B}_{\text{ber}} = \{b_s \mid s \in S_{\text{ber}}\}$, $\mathcal{B}_{\text{unb}} = \{b_s \mid s \in S_{\text{unb}}\}$, and $\mathcal{B} = \mathcal{B}_{\text{ber}} \cup \mathcal{B}_{\text{unb}}$. The sail time of ship $s \in S$ from location i to location j is denoted as $t_{i,j}^s$.

Tugging tasks. Each service request $s \in S$ encompasses multiple tugging tasks, specifically, Γ_s tasks for ship s, with each task performed by one adequately specified tugboat. Particularly, when berthing ship s arrives at one designated waypoints $p_{s\gamma}$ ($1 \le \gamma \le \Gamma_s$ and γ is an integer), a new tugboat joins to assist the ship. Thus, the task starting from the γ th waypoint of berthing ship s is associated with start location $p_{s\gamma}$, destination location b_s, and time window $[\underline{t}_s, \overline{t}_s]$ of arrival at the destination location. For example, in Figure 2 (a), the request from berthing ship s generates three tasks as indicated in dashed lines. The attribute of each task $e=\{k_e, o_e, d_e, [\underline{t}_s, \overline{t}_s]\}$ includes: (i) required tugboat type k_e (see Table 3), (ii) task starting location o_e, (iii) task destination d_e, and (iv) time window $[\underline{t}_s, \overline{t}_s]$ at the associated berth b_s, limiting the berthing ship arrival time or the unberthing ship departure time. For clarity, we refer to the task at the γ th waypoint as the γ th task of the ship. Moreover, coordination constraints are imposed on the execution of two consecutive tasks for the same ship. For example, if the start time of the γ th task for berthing ship s is t, then the $(\gamma+1)$ th task, if it exists, must commence at time $t+t^s_{p_{s\gamma}, p_{s\gamma+1}}$.

Tugboats. Denote $\mathcal{H}$ as the set of tugboats available during service time horizon $[0,T]$. For each ship $s \in S$, $\mathcal{H}_s$ ($\mathcal{H}_s \subseteq \mathcal{H}$) denotes the subset of tugboats with adequate tugging power to serve ship s (see Table 3 for power requirements). There is a set of tugboat bases $O=\{o_h \mid h \in \mathcal{H}\}$, and each tugboat h is deployed at base o_h at time 0. Tugboat $h \in \mathcal{H}$ departs from base o_h to execute assigned tasks and return to base o_h after completing all the assigned tasks. The sail time of tugboat h between location i and location j is $t^h_{i,j}$. When tugboat h is deployed for the first time within service time horizon $[0,T]$, a fixed cost λ_h is incurred, covering human and maintenance expenses for tugboat h during the service time horizon. Moreover, c_h denotes the unit-time sail cost of tugboat h.

Dynamicity and stochasticity of service requests. Service requests are inherently dynamic and stochastic. They are not fully confirmed at the beginning of the service horizon but are instead confirmed dynamically as the current tugboat work plan is executed. This necessitates continual updates to the plan to accommodate newly confirmed requests. The details of these requests, such as the service start time, origin-destination, and the type and number of required tugboats, are only revealed upon confirmation, introducing elements of stochasticity. To effectively manage tugboat operations, it is crucial to anticipate future demands. This anticipation allows tugboats to reposition to promising locations where new demands are likely to emerge, thereby minimizing non-towage sailing and enhancing cost-efficiency. Particularly, service demands unfold across three temporal levels as follows, while tugboat deployment and scheduling rely heavily on the second and third levels, rendering decision-making a challenge due to the dynamic and stochastic nature of service demands:

(i) ETA and ETD: They are approximate timings for ship arrivals and departures (see Table 4), typically announced around 12 hours or more in advance. These announcements are communicated to the port authority, not to the tugboat company.

(ii) Request confirmation: Ships formalize their need for tugging services via a booking system typically within four hours or less - often as late as two hours - before the service start time. Thus, at the beginning of service time horizon (spanning 14 hours for example), a lot of requests remain unconfirmed, presenting dynamicity and stochasticity of service requests.

(iii) Actual service demands at ship arrival or departure: These are defined by the exact timing, location, and conditions of the ship's arrival or departure, including side-thruster functionality (see Table 3) and prevailing weather conditions. Such details remain undisclosed until the actual moment of the ship's arrival or departure, posing significant stochasticity (see Appendix C) and challenges for advance planning.

Anticipation of future service demands. Anticipation leverages stochastic information to prepare for potential demand scenarios, enhancing the cost-efficiency of tugboat operations. Figure 3 demonstrates this concept, showing the anticipation of service requests. For simplicity, we assume each request generates one task. In Figure 3, the red triangle marks the tugboats' current location, the solid line represents the immediate task, and the dashed line indicates a feasible route for future tasks. The left panel shows one new confirmed request (in dark grey) and three anticipated future requests (in light grey). The middle panel displays a tugboat plan devised without anticipating future demands, contrasting with the right panel that presents a more cost-efficient plan incorporating such anticipation. This comparison highlights the benefits of proactive planning, where anticipating nearby future requests allows a tugboat to wait at a strategic location after completing a task, reducing non-towage sailing costs and enhancing overall operational efficiency. Without anticipation, fulfilling these requests often incurs higher costs due to increased non-towage sailing.

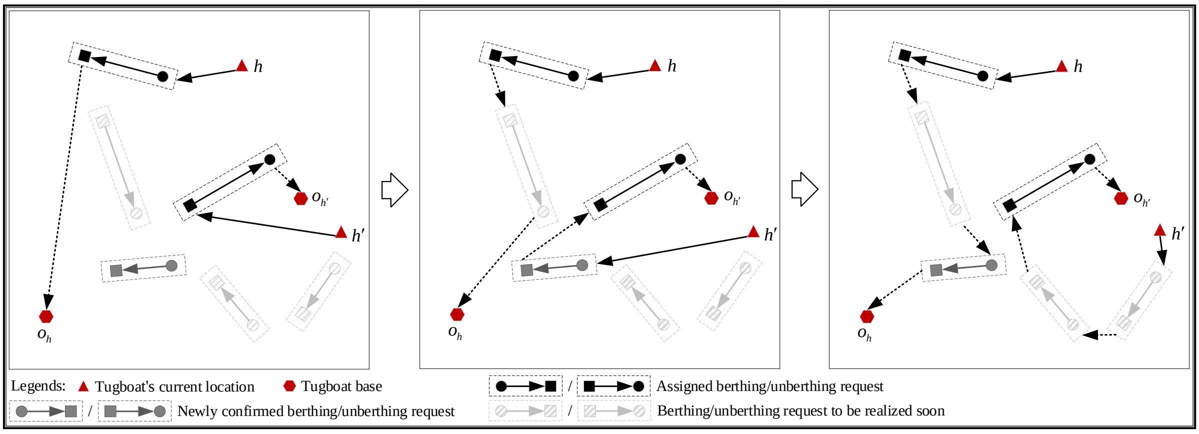


Figure 3. Illustration of anticipating service requests and the benefits.

Operational constraints and decisions. As aforementioned, each request is modeled as a sequence of tasks. Meanwhile, reposition tasks may be assigned to tugboats. Hence, the work plan of each tugboat corresponds to a sequence of tasks (including reposition tasks), each with explicit start and end times. Figure 4 exemplifies such a work plan, showing four tugboats performing tasks delineated by dashed lines, with loops indicating waiting actions. Routes are carefully planned for each tugboat, including departure times and potential waiting if a tugboat arrives early at a task, allowing for route updates to accommodate new tasks and enhance efficiency. Decisions should satisfy the following operational constraints:

(i) *Capability constraints*: Tasks must be performed by tugboats that meet or exceed the required capability. For example, a task requiring a type-B tugboat may be assigned to a type-X tugboat (see Table 2), which has higher capability, but this incurs higher costs.

(ii) *Time constraints*: One tugboat can perform one tugging task with respecting the time window at the berth associated with the task.

(iii) *Coordination constraints*: For tasks associated with the same berthing ship, the γ th $(\gamma > 1)$ task starts when the ship arrives at the γ th waypoint; similarly, for tasks associated with the same unberthing ship, the γ th task ends when the ship arrives at the γ th waypoint.

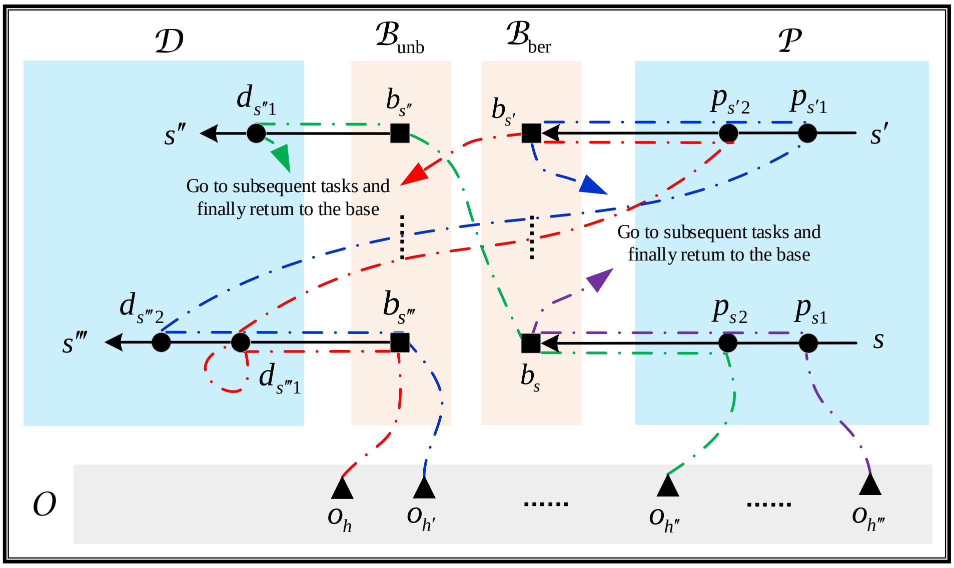


Figure 4. Example of partial work plan for tugboats ($s, s' \in S_{\text{ber}}$; $s'', s''' \in S_{\text{unb}}$).

Objective. In the tugboat industry, the primary goal is to minimize the total expected operation costs - including fixed and sail costs - to satisfy all service requests within the service time horizon. The key performance metric is adhering to specified time windows at berths: berthing ship s must arrive at berth b_s within designated time window $[\underline{t}_s, \overline{t}_s]$, and unberthing ship s' must depart from berth $b_{s'}$ within time window $[\underline{t}_{s'}, \overline{t}_{s'}]$. Although the focus is on cost optimization, this does not necessarily compromise the timeliness of services. Efficient cost management requires to enhance tugboat utilization, encouraging tugboats to serve more ships efficiently, which, in turn, ensures ships arrive or depart timely within their scheduled windows.

In summary, the DTDSP is to dynamically update the current work plan to accommodate newly confirmed requests, and optimize the work plan by anticipating future service demands to enhance decision-making, aiming to provide satisfactory tugging service at the lowest possible system-wide operation cost.

3. RPMDP Modeling Method

The DTDSP could theoretically use a rolling-horizon MILP approach for resolution. In this approach, upon confirmation of a new service request, all pending tasks would be rescheduled across available tugboats by solving the MILP model. However, given the NP-hard nature of the fleet routing problem combined with the necessity for coordination among multiple tugboats per ship, this method proves time-intensive. For instance, solving a practical-sized MILP model (see Appendix B) with 30 berthing and unberthing requests over a 14-hour service period using the Gurobi solver can exceed 35 minutes, making it impractical for real-time decisions. This duration excludes the additional time required for anticipating future service demands and incorporating NP-hard waiting decisions (Branke et al., 2005), despite efforts to streamline the model with valid inequalities (Wei et al., 2020). The computational intensity is exacerbated by the need for intricate coordination among tugboats, where alterations in one tugboat's plan can have cascading effects on others due to the interconnected nature of their operations.

Facing these challenges, motivated by the need to effectively manage dynamic decisions under stochastic service demands, reduce operational costs, and align with industry practices, the DTDSP lends itself to formulation as a sequential decision-making process. The traditional MDP approach, which typically makes one-step decisions at each state, falls short in handling the DTDSP's complexity, particularly due to its limited foresight into future actions. To overcome these limitations, we enhance the basic MDP framework into a refined RPMDP. This enriched framework incorporates both reactive task assignments for confirmed requests and proactive strategies like routing and waiting in anticipation of future demands. The RPMDP introduces significant methodological advancements by integrating waiting decisions, thus broadening its applicability to efficiently manage dynamic and stochastic service demands. This adaptation not only addresses immediate operational needs but also strategically positions tugboats to respond to emerging demands, highlighting the RPMDP's enhanced capability to optimize both immediate and future tugboat operations within the DTDSP.

3.1. Modeling framework

The reactive and proactive RPMDP model includes six elements:

Epoch	The moment at which a work plan is decided.
State	A tuple of information for deciding work plan at an epoch.
Decision	A work plan on immediate and subsequent actions at one state.
Marginal cost	The cost difference between the new decision and the last decision.
Transition	System evolution given a decision and a stochastic event (i.e., a new service request).

Objective Minimization of expected system-wide operation cost over all decision epochs.

(i) Epoch, state, and decision. Initial epoch φ_0 emerges at the beginning of service time horizon $[0,T]$. Initial state $u_{\varphi_0} := \left(\mathcal{E}_{\varphi_0}, \mathcal{H}_{\varphi_0}\right)$, where $\mathcal{E}_{\varphi_0}$ and $\mathcal{H}_{\varphi_0}$ denote all the tasks generated by the currently confirmed requests and all the available tugboats, respectively, at epoch φ_0. Decision $x_{\varphi_0} := \left(x_{\varphi_0,h}\right)_{h\in\mathcal{H}_{\varphi_0}}$, where $x_{\varphi_0,h}$ is the individual plan for tugboat $h \in \mathcal{H}_{\varphi_0}$. Essentially, decision x_φ comprises a set of routes, each for a deployed tugboat to perform the assigned tasks with a specified departure time from each location; meanwhile, if a route leads to an early arrival at a task, a waiting action is planned before the task's commencement. This pause allows for potential updates to the work plan, integrating new tasks as they arise to enhance operational cost-efficiency. Decision $x := \left(x_h\right)_{h\in\mathcal{H}}$ is feasible if there is one route for each deployed tugboat h starting from the current location of tugboat h, satisfying the assigned tasks for tugboat h, and ending at base o_h. When a new request s is confirmed at time t_s ($t_s > 0$), decision epoch $\varphi_s = t_s$ occurs, and the tugging service system occupies state $u_{\varphi_s} := \left(\mathcal{E}_{\varphi_s}, x_{\varphi_s-1} - a_{\varphi_s-1}\right)$, where $\mathcal{E}_{\varphi_s}$ denotes the currently confirmed but unfulfilled tasks, including the tasks generated by the newly confirmed request s, and $x_{\varphi_s-1} - a_{\varphi_s-1}$ is the unfinished part of the current work plan. Note that x_{φ_s-1} is the work plan made at the last state u_{φ_s-1}, a_{φ_s-1} is the action performed between the last state u_{φ_s-1} and the current state u_{φ_s}, and $x_{\varphi_s-1} - a_{\varphi_s-1}$ is the work plan x_{φ_s-1} minus the action that has been performed before state u_{φ_s}. Note that each task $e := \left(k_e, o_e, d_e, [\underline{t}_s, \overline{t}_s]\right) \in \mathcal{E}_{\varphi_s}$, having four attributes k_e, o_e, d_e, and $[\underline{t}_s, \overline{t}_s]$ as explained before. Eventually, the decision process ends at the final state u_Φ, when all the requests during service time horizon $[0,T]$ have been confirmed. After that, the tugboat work plan will not be updated, and each deployed tugboat $h \in \mathcal{H}$ will return to base o_h after completing the assigned tasks.

Each tugboat operates under an individual work plan, that allows to insert or remove tasks as required. Consequently, task assignment no longer requires that the tugboat is idle, and all unfulfilled tasks are assigned to tugboats. Moreover, the RPMDP incorporates *waiting tasks* based on anticipation of future demand scenarios, positioning tugboats at strategic locations to reduce non-towage sails and enhance responsiveness to near-future demands. Specifically, waiting task $e_h := \left(null, o_{e_h}, d_{e_h}, [0,T]\right)$, where o_{e_h} is the destination location of the last task and $d_{e_h} \in \mathcal{F}$ is the waiting location allocated to tugboat h. The calculation of candidate waiting points $\mathcal{F}$ and the determination of waiting location d_{e_h} to accommodate tugboat h are detailed in Section 4.2. An example of RPMDP solution for the DTDSP is presented in Appendix D.

(ii) Marginal cost. Unlike the basic MDP's one-step decision-making process, the RPMDP creates a work plan encompassing immediate and subsequent actions at one state. We therefore use a *marginal-cost function* for RPMDP at state u_φ to evaluate the cost-effectiveness between a new plan x_φ and the unfinished part of the previous plan $x_{\varphi-1}$, defined as:

$$F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right) = F_{\text{cost}}\left(u_\varphi, x_\varphi\right) - F_{\text{cost}}\left(u_\varphi, x_{\varphi-1} - a_{\varphi-1}\right)$$

where $F_{\text{cost}}\left(u_\varphi, x_\varphi\right)$ is the cost of plan x_φ at state u_φ, $F_{\text{cost}}\left(u_\varphi, x_{\varphi-1} - a_{\varphi-1}\right)$ is the cost of the unfinished part of plan $x_{\varphi-1}$ at current state u_φ; to be particular, $a_{\varphi-1}$ denotes the action performed between state $u_{\varphi-1}$ and state u_φ, and $x_{\varphi-1} - a_{\varphi-1}$ is the unfinished part of plan $x_{\varphi-1}$ which is made at the last state $u_{\varphi-1}$; in addition, $F_{\text{cost}}^{\Delta}\left(u_0, x_0\right) = F_{\text{cost}}\left(u_0, x_0\right)$. To calculate the cost of plan x, we need to accumulate the tugboat sail costs for each action in plan x and, if deploying a new tugboat, its fixed operation cost. Notice that work plan x extends to the end of service time horizon, thus marginal cost indicates the impact of changes in work plan on tugboat operation cost.

(iii) Transition. A transition from state u_φ to state $u_{\varphi+1}$ is induced by decision x_φ and stochastic event $\omega_{\varphi+1}$ (i.e., a new service request). Accordingly, a transition can be represented as realization $\omega_\varphi := \left(u_\varphi^{x_\varphi}, u_{\varphi+1}\right) \in \Omega_\varphi$. A transition can be divided into two processes: a deterministic transfer from decision state u_φ to post-decision state $u_\varphi^{x_\varphi}$, which is incurred by decision x_φ; and a stochastic transfer from post-decision state $u_\varphi^{x_\varphi}$ to the next decision state $u_{\varphi+1} := \left(u_\varphi^{x_\varphi}, \omega_{\varphi+1}\right)$, triggered by a stochastic event $\omega_{\varphi+1}$. Thus, implementing decision x_φ at state u_φ leads to post-decision state $u_\varphi^{x_\varphi}$, and upon the occurrence of event $\omega_{\varphi+1}$, the system advances to a subsequent state $u_{\varphi+1}$.

(iv) Objective. The objective is to identify an optimal decision policy $\pi^* \in \Pi$ to minimize the expected system-wide tugboat operation cost over service time horizon $\left[0,T\right]$, from an initial state u_0 to a final state u_Φ. Policy $\pi := \left(X_0^\pi,...,X_\varphi^\pi,...,X_\Phi^\pi\right) \in \Pi$ is a sequence of decision rules, where X_φ^π is a function driven by policy π for state u_φ. In other words, decision rule X_φ^π is the function to generate tugboat work plan x_φ at state u_φ. The optimal policy π^* that can be subsequently applied at each state to minimize the expected system-wide operation cost can be defined as:

$$\pi^* = \underset{\pi\in\Pi}{\arg\min}\, \mathbb{E}\left[\sum_{\varphi=0}^{\Phi} F_{\text{cost}}^{\Delta}\left(u_\varphi, X_\varphi^\pi\left(u_\varphi\right)\right) | \, u_0\right]$$

For each state u_φ, policy π^* holds the optimality equation as shown in Equation (1): the first term is the marginal cost at state u_φ, and the second term is the expected marginal cost from post-decision state $u_\varphi^{x_\varphi}$ to the end of service time horizon (i.e., cost-to-go). At state u_φ, a set of feasible decisions (i.e., $X_\varphi^\pi\left(u_\varphi\right)$) is searched to calculate marginal cost $F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right)$ for each decision $x_\varphi \in X_\varphi^\pi\left(u_\varphi\right)$, and evaluate cost-to-go $\mathbb{E}\left[\sum_{j=\varphi+1}^{\Phi} F_{\text{cost}}^{\Delta}\left(u_j, X_j^{\pi^*}\left(u_j\right)\right) | u_\varphi^{x_\varphi}\right]$ from post-decision state $u_\varphi^{x_\varphi}$, then policy π^* selects the *optimal decision* that minimizes the sum of these two terms.

$$X_\varphi^{\pi^*}\left(u_\varphi\right) = \underset{x_\varphi\in X_\varphi^\pi\left(u_\varphi\right)}{\arg\min}\left\{F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right) + \mathbb{E}\left[\sum_{j=\varphi+1}^{\Phi} F_{\text{cost}}^{\Delta}\left(u_j, X_j^{\pi^*}\left(u_j\right)\right) | u_\varphi^{x_\varphi}\right]\right\} \tag{1}$$

Correspondingly, the marginal-cost based Bellman equation is presented as:

$$V\left(u_\varphi\right) = \min_{x_\varphi\in X_\varphi^\pi\left(u_\varphi\right)}\left\{F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right) + \mathbb{E}\left[V\left(u_{\varphi+1}\right) | u_\varphi^{x_\varphi}\right]\right\} \tag{2}$$

where $V\left(u_\varphi\right)$ is the optimal value at state u_φ. Unlike the basic MDP, which considers only immediate actions, the RPMDP encompasses both reactive decisions (e.g., task assignments for confirmed requests) and proactive decisions (e.g., routing and waiting in anticipation of future demands), enhancing the complexity and strategic depth of the decision-making process.

3.2. Merits of the RPMDP

To demonstrate the superiority of the RPMDP, we conduct comparative analysis and present examples of both RPMDP and basic MDP models for the DTDSP, detailed in Appendix D and Appendix E. For daily tugboat operations, a work plan is required to guide the current and subsequent actions, in this way, one tugboat has visibility on future actions and only needs to follow the individual plan for it until the individual plan is updated. Note that the system-wide work plan is updated to incorporate each newly confirmed request, but the individual work plan for one tugboat is not changed very often. With comparative comparison, as exemplified in Appendix D and Appendix E, the RPMDP offers several advantages over the basic MDP.

(i) **Superior Solution Quality:** The RPMDP excels over the basic MDP by transforming infeasible solutions into feasible and cost-efficient ones through proactive actions, such as waiting. By anticipating future service demands, the RPMDP makes feasible what the basic MDP cannot, and optimizes operational costs by reducing unnecessary sailing via waiting decisions. This advantage is highlighted where the RPMDP adapts infeasible solutions from the basic MDP (see Appendix E) into feasible outcomes (see Appendix D), underscoring RPMDP's capability to enhance both feasibility and cost-efficiency.

(ii) **Extended Decision-Making Time:** Under the one-step decision pattern of basic MDP, when the current action is finished, the corresponding tugboat will be idle for a new decision. Thus, the decision time is restricted to reduce the tugboat idle time, inevitably leading to increased downtime and computational load, sacrificing the decision quality. In contrast, the RPMDP extends the decision-making window. This extension allows for immediate adjustment of work plan when new requests are confirmed, providing more time to optimize decisions.

(iii) **Reduced Computational Complexity:** The basic MDP employs a one-step decision pattern, triggering a new state with each tugboat task completion. This approach significantly increases the number of states and, consequently, the computational burden. In contrast, the RPMDP adopts a work-plan approach, updating decisions only when new service requests are received. Since each request can generate multiple tasks, the number of updates in the RPMDP is much lower than in the basic MDP, significantly reduces the number of states and streamlines computational process.

(iv) **Seamless Integration of Planning Outputs:** The basic MDP often exhibits a disconnect between problem formulation and planning outputs, typically using a one-step decision that is projected by the output of planning methods and not fully utilizing planning outputs. The RPMDP, however, integrates these outputs completely at each decision epoch, effectively connecting problem formulation and planning methods and thus facilitating the solution process.

(v) **Alignment with Operational Expectations:** The RPMDP supports the operational expectations of tugboat companies by providing continuous visibility into future actions, allowing each tugboat to operate under a clear and consistent plan. This approach not only meets the operational demands of dynamic and stochastic environments but also ensures that the RPMDP aligns closely with real-world management requirements.

These enhancements position the RPMDP as an advanced framework capable of addressing the complexities of the DTDSP more effectively than traditional MDP models, providing significant operational advantages in real-world applications.

4. Anticipatory ADP Method

The MDP is challenged by vast state and decision spaces, a phenomenon known as the curse of dimensionality (Powell, 2022; Jia et al., 2022). RPMDP, with its intricate work plans, exacerbates this complexity, rendering exact optimization methods impractical for real-time applications. Furthermore, the stochastic nature of RPMDP escalates computational demands, rendering real-time solutions impractical (Archetti et al., 2018; 2020). In response, we employ an AADP method focused on evaluating cost-to-go at each state (as detailed in Equation (1)). Additionally, we explore three widely recognized methods for this evaluation, grounded in existing literature:

(i) Approximation of value function: Involves training a model (e.g., regression or neural network) to estimate the value of post-decision states. This method often struggles to accurately represent the complex relationships in post-decision states, leading to suboptimal performance (Bertsimas and Demir, 2002).

(ii) Approximate value iteration: Estimates the value of post-decision states via iteration using, for example, lookup tables. Due to the large volume of post-decision states, this method typically necessitates the aggregation of state spaces, posing challenges in selecting suitable attributes for effective aggregation (Ulmer et al., 2018).

(iii) Post-decision rollout algorithm: Iteratively refines the value of the current post-decision state using simulation techniques (Ulmer et al., 2019). This method uses simulated events to advance the current post-decision state to subsequent states, facilitating decision-making through basic policies. The simulation proceeds for a specified duration, ultimately determining the value of the post-decision state.

With comparative analysis, the rollout algorithm has proven effective in estimating current post-decision values. Recognizing that the rollout algorithm operates online within constrained times, we additionally leverage a novel IZS technique for improved efficiency. An overall framework of the solution approach is depicted in Figure 5.

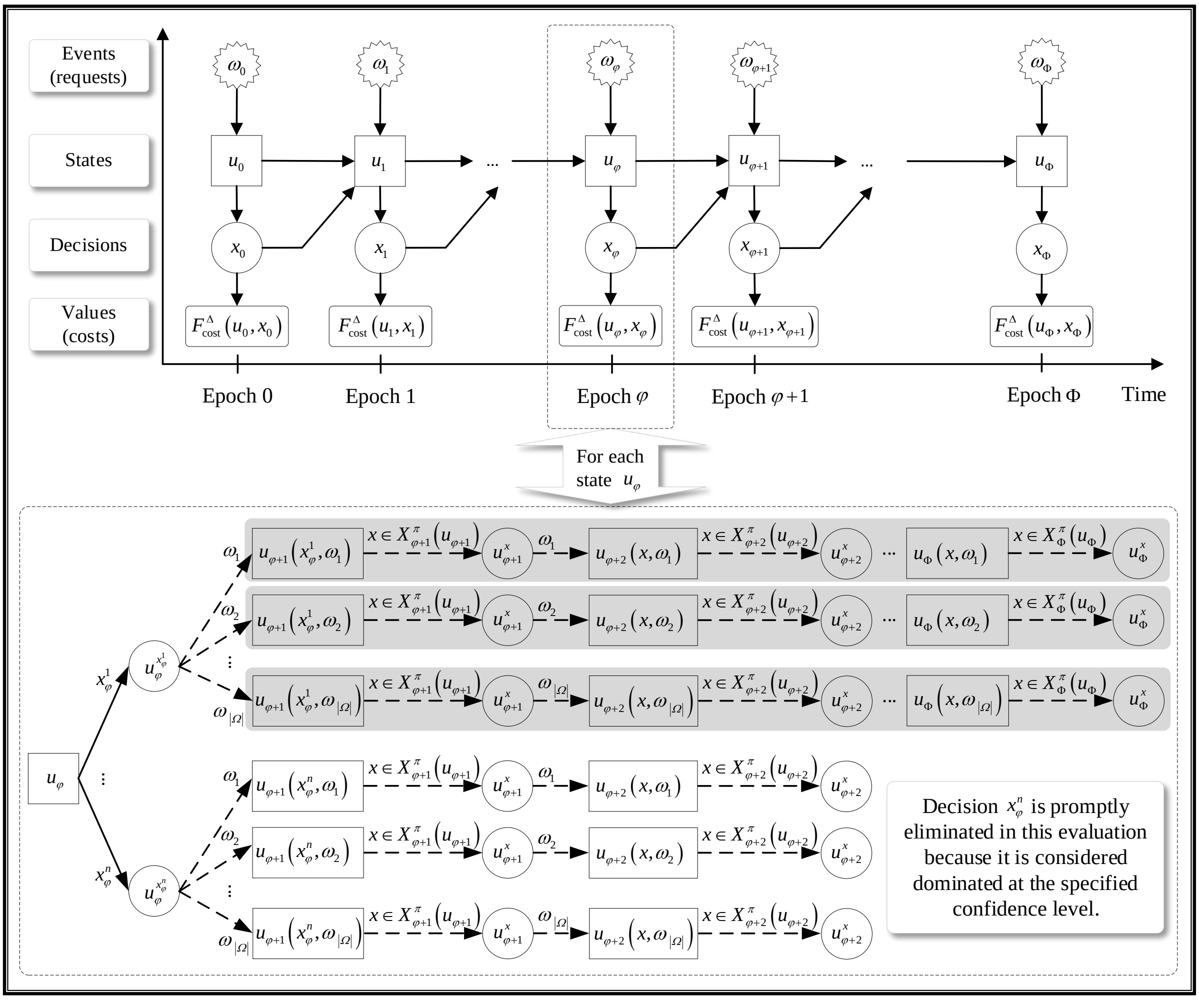


Figure 5. The dynamic decision process with AADP.

4.1. Task assignment strategy

Upon confirmation of a new service request, associated tugging tasks are assigned to eligible tugboats as outlined in Table 3. The RPMDP supports dynamic updates to the tugboat work plan by incorporating new tasks. Given the complexities of the fleet routing problem - an NP-hard challenge - along with the need for coordinating multiple tugboats to serve a single ship, heuristic algorithms become inevitable for real-time task insertion in practical-sized instances. Empirical evidence from Ghiani et al. (2012) and our own experiments confirm that simple insertion methods are as effective as more complex scenario-based approaches (Bent and Van Hentenryck, 2004) for integrating new tasks. Utilizing this approach, tasks are initially incorporated using a cost-effective insertion method to construct a preliminary work plan, which is subsequently refined using a tabu search algorithm. This two-phase process enhances dynamic task assignment by first inserting tasks efficiently to meet efficiency requirements and then optimizing the work plan through tabu search.

Phase I: Task insertion. Upon confirmation of a new service request, new tugging tasks are generated for tugboats, necessitating integration into the existing work plan. Each route is evaluated to determine the feasibility and cost-effectiveness of incorporating the new task. Some routes may be empty, representing tugboats yet to be deployed, thus expanding the insertion possibilities. The cheapest insertion method is initially utilized to determine the optimal insertion point within a route, aiming to minimize the incremental cost. This method offers two key advantages: (i) it facilitates rapid feasibility assessments and updates to the work plan, crucial for time-sensitive online decision-making; and (ii) it aligns with the tugboat company's practice of planning routes and then integrating new tasks as they arise. Costs incurred from tugboat sail and, if deploying new tugboats, fixed costs, are incorporated into the operational expenses.

Importantly, tasks from the same request must adhere to synchronization constraints. Specifically, Constraint (3) ensures that all tasks for a berthing ship are completed simultaneously, meaning the ship and all its assigned tugboats reach the destination berth together, as shown in Figure 2(a). This constraint also stipulates that the arrival at the berth must fall within a predefined time window. Similarly, Constraint (4) requires that all tasks for an unberthing ship begin concurrently, ensuring all tugboats start servicing the ship simultaneously, and that the start time aligns with the designated time window.

$$
\begin{cases}
\max\limits_{e \in \mathcal{E}_s} \left\{ t_{\text{end}}^{e} \right\} - \min\limits_{e \in \mathcal{E}_s} \left\{ t_{\text{end}}^{e} \right\} = 0 \quad \forall s \in S_{\text{ber}} \\
\underline{t}_s \le t_{\text{end}}^{e} \le \overline{t}_s \quad \forall s \in S_{\text{ber}}, e \in \mathcal{E}_s
\end{cases}
\tag{3}
$$

$$
\begin{cases}
\max\limits_{e \in \mathcal{E}_s} \left\{ t_{\text{start}}^{e} \right\} - \min\limits_{e \in \mathcal{E}_s} \left\{ t_{\text{start}}^{e} \right\} = 0 \quad \forall s \in S_{\text{unb}} \\
\underline{t}_s \le t_{\text{start}}^{e} \le \overline{t}_s \quad \forall s \in S_{\text{unb}}, e \in \mathcal{E}_s
\end{cases}
\tag{4}
$$

Phase II: Work Plan Optimization. This phase utilizes a tabu search method to refine the existing work plan, systematically examining neighboring solutions. Algorithm 1 conducts an iterative search that adheres to synchronization constraints for tugboats serving the same ship. It optimizes the current work plan for all tugboats through two key procedures $SortTasks(\cdot)$ and $Neighbour(\cdot)$, emphasizing avoiding poor local optimums.

Algorithm 1. $TabuSearch(x)$

Input: Work plan x

Output: The best plan so far x^*

1: $x^* \leftarrow x$ and $counter \leftarrow 0$

2: procedure $SortTasks(x, \mathcal{E}_{\text{unf}})$, where $\mathcal{E}_{\text{unf}}$ is the set of unfulfilled tasks in work plan x

3: **for** each task $e \in \mathcal{E}_{\text{unf}}$ according to the order in $L_{\mathcal{E}}$ **do**

4:     **while** $counter < n_{\max}$

5:         **if** $Neighbour(x, e)$ is true, **then**

6:             $x^\dagger \leftarrow x_e$, and the updated routes r and r_e in x_e are tabu for task e during m iterations

7:             **if** $F_{\text{cost}}^{\Delta}\left(u, x^\dagger\right) < F_{\text{cost}}^{\Delta}\left(u, x^*\right)$, where u denotes the current state, **then**

8:                 $x^* \leftarrow x^\dagger$, and $counter \leftarrow 0$

9:             **else**

10:                 $counter \leftarrow counter + 1$

11:             **end if**

12:         **end if**

13:     **end while**

14: **end for**

15: return x^*

Given a set of unfulfilled tasks $\mathcal{E}_{\text{unf}}$ in the current work plan x, procedure $SortTasks(\cdot)$ constructs a task list $L_{\mathcal{E}}$ by sorting tasks $\mathcal{E}_{\text{unf}}$, so that the first task in $L_{\mathcal{E}}$ is of the largest cost-saving rate when removing task e from work plan x (see Algorithm 2).

Algorithm 2. $SortTasks\left(x, \mathcal{E}_{\text{unf}}\right)$

Input: Work plan x and task set $\mathcal{E}_{\text{unf}}$

Output: Task list $L_{\mathcal{E}}$

for each task $e \in \mathcal{E}_{\text{unf}}$ **do**

calculate $c_x^e = c_{ve} + c_{ew} - c_{vw}$, where $c_{\bullet *}$ is the cost to perform tasks $\bullet$ and $*$, while v and w are the predecessor and successor of task e in the work plan, respectively; noting v may be the tugboat base if e is the first task of the tugboat, similarly, w may be the tugboat base if e is the last task of the tugboat

calculate $\rho_e = c_x^e / c_e$, where c_e is the tugboat sail cost to perform task e from the starting location o_e to destination d_e

end for

sort tasks $\mathcal{E}_{\text{unf}}$ in nonincreasing order of the ρ_e values, and return the sorted task list $L_{\mathcal{E}}$

For each task $e \in \mathcal{E}_{\text{unf}}$, the tabu search algorithm terminates when $n_{\max}$ iterations are performed without improving the current best plan x^*. Based on numerical experiments, $n_{\max}$ is set to be $\left|\mathcal{E}_{\text{unf}}\right|$. Procedure $Neighbour\left(x, e\right)$ incurs a move from the current plan x to a neighbor plan x' by removing task e from route r_e (i.e., a route including task e) and feasibly inserting task e into another route r (see Algorithm 3). When task e is removed from route r_e, it is tabu for m iterations to prevent reinsertion into route r_e. Similarly, when task e is inserted into route r, it is tabu for m iterations to prevent removing task e from route r. To generate a better plan, it is observed that the value of m is dependent on the value of $\left|\mathcal{E}_{\text{unf}}\right|$ and the number of tugboat routes $\left(n_r\right)$ in the current plan. According to numerical tests, we assign m an integer number randomly within the range $\left[3, \sqrt{10+n}\right]$, where $n = \left|\mathcal{E}_{\text{unf}}\right| + n_r$ if $\left|\mathcal{E}_{\text{unf}}\right| + n_r < 65$ and $n = \left(3/2\right)\left(\left|\mathcal{E}_{\text{unf}}\right| + n_r\right)$ if $\left|\mathcal{E}_{\text{unf}}\right| + n_r \geq 65$. A fundamental rationale behind this m-value setting is that, when the number of unfulfilled tasks (i.e., $\left|\mathcal{E}_{\text{unf}}\right|$) and tugboat routes (i.e., n_r) is large, the likelihood of a more thorough exploration increases (i.e., the value of m is larger). The above range produces a moderate value of m, effectively preventing cycling without hindering the generation of high-quality plans. Moreover, the aforementioned tabu operations might be excessively strong and forbid a good neighbor plan. Therefore, we investigate the possibility to remove task e from the route which is tabu for task e, or insert task e into the route which is tabu for task e. However, such a neighbor plan is accepted only if it is better than the best plan achieved so far.

Algorithm 3. $Neighbour(x,e)$

Input: Work plan x and task e

Output: Neighbour plan x_e

$BestValue \leftarrow \infty$
for each tugboat route $r \in x$ **do**
    if route r is not tabu to insert task e and route r_e is not tabu to remove task e, **then**
        if task e can be feasibly inserted into route r, **then**
            let x' be the plan obtained from x by removing e from r_e and inserting e into r, $f = F_{\text{cost}}^{\Delta}(u,x')$, where u denotes the current state
            if $f < BestValue$, **then**
                $BestValue \leftarrow f$, and $x_e \leftarrow x'$
            end if
        end if
    else
        if task e can be feasibly inserted into route r, **then**
            let x' be the plan obtained from x by removing e from r_e and inserting e into r, $f = F_{\text{cost}}^{\Delta}(u,x')$, where u denotes the current state
            if $f < BestValue$ and $f < F_{\text{cost}}^{\Delta}(u,x^*)$, **then**
                $BestValue \leftarrow f$, and $x_e \leftarrow x'$
            end if
        end if
    end if
end for
return x_e

4.2. Waiting strategy

The waiting strategy developed aims to optimize system-wide tugboat operation costs by proactively positioning tugboats at locations likely to receive imminent service requests. This reduces non-towage sailing and increases the efficiency of each deployed tugboat. Specifically, if a work plan results in a tugboat arriving early for a task, a waiting action is scheduled to allow for potential inclusion of new tasks from newly confirmed requests. This strategy not only minimizes unnecessary travel but also enhances the likelihood of a tugboat covering more tasks, thereby reducing both fixed and operational costs. Note that the problem with more than one server (e.g., tugboat in our work) to find the optimal waiting plan has proven to be NP-hard (Branke et al., 2005). Therefore, there is an inevitable need for heuristic methods to solve practical-size instances for real-time waiting decisions (Thomas, 2007). Furthermore, our waiting strategy is evaluated against five conventional strategies through sensitivity analysis in numerical experiments, offering a detailed comparative effectiveness assessment.

To decide a waiting action between two consecutive tasks, an initial step is to identify a proper waiting location. The waiting location could be the end location of the previous task, or a different waiting point. To determine candidate waiting points, we sample service requests and keep feasible tasks that can be inserted between the two consecutive tasks. These tasks are then divided into k clusters using the k-means clustering algorithm (see Algorithm 4), and a waiting point is defined as the centroid of task starting locations in each cluster. Particularly, in the k-means clustering algorithm, we classify a feasible task into a cluster if its starting location is closer to the cluster's centroid than other centroids. Thus, a set of candidate waiting points is calculated by alternating between: (i) clustering feasible tasks based on the current centroids, and (ii) calculating centroids based on the current clusters of feasible tasks. Based on the service area layout of the Singapore container port, parameter k is set to six in numerical experiments. Moreover, waiting time needs to be determined for the waiting location. Waiting time is critical for decision performance: (i) if the waiting time is too short, there is no sufficient time to realize potential service requests; (ii) if the waiting time is too long, one tugboat may wait a long time without new requests. To strike a balance, a tugboat is planned to wait at a location for time $t_{\text{wait}} = \overline{t} - t$, where $\overline{t}$ represents the earliest time when the tugboat can leave the waiting location without early arrival at the next task, and t denotes the arrival time at the waiting location.

Algorithm 4. Task clustering and waiting point calculation

Input: Simulated tasks $\tilde{\mathcal{E}}$ and parameter k

Output: A set of waiting points $\mathcal{F} = \{ f_1, f_2, ..., f_k \}$

1: keep simulated tasks $\bar{\mathcal{E}}$ $\left(\bar{\mathcal{E}} \in \tilde{\mathcal{E}}\right)$, that are individually feasible to be inserted between the previous task and the next task
2: initialize waiting points $\mathcal{F} = \{f_1, f_2, ..., f_k\}$ randomly
3: **for** each task $e \in \bar{\mathcal{E}}$ **do**
4: find the waiting point f_i $(i = 1, ..., k)$ closest to the starting location of task e, and assign task e to cluster i
5: **end for**
6: **for** each cluster i **do**
7: update waiting point f_i, which is the centroid of starting locations of tasks in cluster i
8: **end for**
9: repeat Lines 3-8 until convergence

Given a set of candidate waiting points and the associated waiting times, the following aspects are considered to determine a waiting action. (i) *Probability of task realization in the neighborhood of each waiting point.* Due to the stochasticity of future service demands, one tugboat may wait for nothing at one waiting point. Thus, the probability of task realization in the neighborhood of waiting point should be sufficiently high. The neighborhood of waiting point is a sub-region divided by drawing a line between task clusters (achieved by Algorithm 4) so that the distance from the line to the nearest task starting location is maximized. The probability of realizing at least n tasks whose starting location in the neighborhood of waiting point f during time t_{wait} is represented by $P_f\left(n, t_{\text{wait}}\right)$. (ii) *Distribution of idle tugboats.* The location of idle tugboats needs to be considered, so that the number of idle tugboats in the neighborhood of waiting point f matches the expected number of new tasks during time t_{wait}. To avoid ineffective relocation of tugboats, the value of n in $P_f\left(n, t_{\text{wait}}\right)$ should not be less than n_f, which is the number of idle tugboats in the neighborhood of waiting point f during time t_{wait}. (iii) *Cost-effectiveness of tugboat sailing to the waiting point.* It is a bet to guide one tugboat to a far waiting point due to the sail cost. Thus, the cost-effectiveness of tugboat sailing to each waiting point should be considered. Consequently, we further refine the value of n in $P_f\left(n, t_{\text{wait}}\right)$ to be ςn_f, where $\varsigma = \max\{\ln \lambda_{\text{ratio}}, \lambda\}$, λ_{ratio} is the ratio of distance between the previous task and the waiting point to distance between the two consecutive tasks, and λ is a user-defined parameter and set to be one by default to avoid ς is too small. The rationale is that, if the sail cost to a waiting point is higher, there should be more tasks realized around the waiting point during the waiting time. This would reduce the risk of insufficient tasks when the tugboat waits at the waiting point.

Finally, to derive the benefit of repositioning an idle tugboat to waiting point f, the realization probability $P_f\left(\varsigma n_f, t_{\text{wait}}\right)$ of ςn_f tasks in the neighborhood during the associated waiting time t_{wait} should be sufficiently high. To this end, we utilize a threshold value τ_{thres}: if condition $P_f\left(\varsigma n_f, t_{\text{wait}}\right) \geq \tau_{\text{thres}}\left(f \in \mathcal{F}\right)$ is satisfied, the tugboat waits at the waiting point $f \in \mathcal{F}$ that has the highest probability $P_f\left(\varsigma n_f, t_{\text{wait}}\right)$; otherwise, the tugboat waits at the end location of the previous task. After parameter tuning, the value of τ_{thres} is set to 0.85 in our experiments. As concluded in the sensitivity analysis, the proposed waiting strategy has a significant improvement in cost-saving compared with conventional waiting strategies.

4.3. IZS-based two-phase rollout algorithm

Utilizing the developed task assignment and waiting strategies, we can calculate a set of feasible decisions $X_{\varphi}^{\pi}\left(u_{\varphi}\right)$ for a given state u_{φ}. In this subsection, we devise an efficient algorithm to select the decision x from decision space $X_{\varphi}^{\pi}\left(u_{\varphi}\right)$ to minimize the expected system-wide tugboat operation cost. The algorithm proceeds by first calculating the marginal cost $F_{\text{cost}}^{\Delta}\left(u_{\varphi}, x_{\varphi}\right)$ of each decision $x_{\varphi} \in X_{\varphi}^{\pi}\left(u_{\varphi}\right)$, representing the first term of Equation (1). Next, it estimates the cost-to-go from the post-decision state $u_{\varphi}^{x_{\varphi}}$, forming the second term of Equation (1). Then, we select the decision x_{φ} with the smallest sum of these two terms, as illustrated in Equation (1). Because $F_{\text{cost}}^{\Delta}\left(u_{\varphi}, x_{\varphi}\right)$ is deterministically computed based on decision x_{φ} and state u_{φ}, the primary challenge resides in evaluating the cost-to-go from post-decision state $u_{\varphi}^{x_{\varphi}}$. Due to extensive decision and transition spaces, the use of heuristic methods to estimate cost-to-go is inevitable. Thus, we employ the rollout algorithm for forward-looking scenario analysis and integrate an IZS-based strategy, inspired by Ulmer et al. (2019) but applied differently, to streamline the evaluation process. A basic rollout algorithm to evaluate the cost-to-go from a post-decision state is as follows:

(i) Given a decision $x_\varphi \in X_\varphi^\pi\left(u_\varphi\right)$, and a problem realization (also known as *sample path*) $\omega \in \Omega$, the cost-to-go from post-decision state $u_\varphi^{x_\varphi}$ is defined as $F^\pi\left(u_\varphi^{x_\varphi}, \omega\right) = \sum_{i=\varphi+1}^{\Phi} F_{\text{cost}}^{\Delta}\left(u_i, X_i^\pi\left(u_i\right), \omega\right)$, where $F_{\text{cost}}^{\Delta}\left(u_i, X_i^\pi\left(u_i\right), \omega\right)$ is the marginal cost incurred by decision $X_i^\pi\left(u_i\right)$ at state u_i on problem realization ω. For example, the first grey row in Figure 5 is the procedure to calculate $F^\pi\left(u_\varphi^{x_\varphi^1}, \omega_1\right)$, in other words, $F^\pi\left(u_\varphi^{x_\varphi^1}, \omega_1\right)$ is the sum of marginal cost incurred by each decision in the first grey row. To fulfill the requisites of efficiency in online decision-making, a singular decision is derived at each state during the evaluation of cost-to-go (i.e., $\left|X_i^\pi\left(u_i\right)\right|=1$, $\varphi+1 \le i \le \Phi$), by cheapest insertion for task assignment and always waiting at the end location of the previous task until no early arrival at the next task.

(ii) For the decision $x_\varphi^1 \in X_\varphi^\pi\left(u_\varphi\right)$, the cost-to-go from post-decision state $u_\varphi^{x_\varphi^1}$ is estimated as $F^{\pi_r}\left(u_\varphi^{x_\varphi^1}\right) = \left|\Omega\right|^{-1} \sum_{\omega\in\Omega} F^\pi\left(u_\varphi^{x_\varphi^1}, \omega\right)$, namely, the average cost-to-go over problem realizations Ω, where notation π_r represents the rollout policy which is generated by the rollout algorithm. For example, cost-to-go from post-decision $u_\varphi^{x_\varphi^1}$ (i.e., $F^{\pi_r}\left(u_\varphi^{x_\varphi^1}\right)$) is the average of values, each of which is a cost-to-go calculated by a grey row in Figure 5.

(iii) The decision at state u_φ is determined as $X_\varphi^{\pi_r}\left(u_\varphi\right) = \underset{x_\varphi \in X_\varphi^\pi\left(u_\varphi\right)}{\operatorname{argmin}} \left\{F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right) + F^{\pi_r}\left(u_\varphi^{x_\varphi}\right)\right\}$. Particularly, when the tugging service system occupies state u_φ, a set of feasible decisions $X_\varphi^\pi\left(u_\varphi\right)$ are generated using the task assignment and waiting strategies. For each decision $x_\varphi \in X_\varphi^\pi\left(u_\varphi\right)$, marginal cost $F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right)$ is calculated and the state is transitioned to post-decision state $u_\varphi^{x_\varphi}$, then cost-to-go $F^{\pi_r}\left(u_\varphi^{x_\varphi}\right)$ is evaluated using the rollout algorithm. A decision is chosen that minimizes the sum of marginal cost $F_{\text{cost}}^{\Delta}\left(u_\varphi, x_\varphi\right)$ and expected cost-to-go $F^{\pi_r}\left(u_\varphi^{x_\varphi}\right)$.

Specifically, rollout policy π_r is a chain of rollout decision rules $\left(X_0^{\pi_r},...,X_\varphi^{\pi_r},...,X_\Phi^{\pi_r}\right)$, where $X_\varphi^{\pi_r}$ is induced by rollout policy π_r at state u_φ. Note that anticipation is used in the rollout algorithm to evaluate cost-to-go by building a scenario-and-decision tree, accordingly, the accuracy of cost-to-go evaluation is significantly improved.

We enhanced the basic rollout algorithm by incorporating the IZS to develop the IZS-2-R algorithm (see Algorithm 5), significantly speeding up the evaluation process. The IZS-2-R algorithm promptly eliminates dominated decisions, where a decision is considered dominated if its expected cost-to-go substantially exceeds that of others. The IZS-2-R algorithm operates based on three parameters: α, δ, and θ_0. Specifically, $1-\alpha$ signifies the confidence level ensuring the selected decision likely generates the lowest system-wide cost, when the system-wide cost of the best decision is at least δ lower than that of the second-best decision. Parameter δ defines the indifference zone, namely, the negligible difference in cost-to-go between two decisions. Particularly, if the difference between the cost-to-go of decision x and that of the best decision currently known is within δ, we believe that this difference is not significant, and decision x will be reserved for subsequent evaluation. Parameter θ_0 is the number of evaluations conducted on each problem realization in the initial stage (i.e., Phase I in Algorithm 5). Decisions $X_\varphi^\pi\left(u_\varphi\right)=\{x_1,...,x_n\}$ in the input of Algorithm 5 are the n most cost-efficient plans generated by the task assignment and waiting strategies. Moreover, $W_{x_i x_{i'}}(\theta)$ determines how far the cost of decision x_i can be higher than the cost of any other decision $x_{i'} \in X_\varphi^\pi\left(u_\varphi\right)$ without being removed from $X_\varphi^\pi\left(u_\varphi\right)$. The value of $W_{x_i x_{i'}}(\theta)$ decreases monotonically as the number of evaluations θ increases. Based on numerical experiments and motivated by Kim and Nelson (2001), set $W_{x_i x_{i'}}(\theta)=$ $\max\left\{0,\delta\left(2\beta\eta(\theta_0-1)\sigma_{x_i x_{i'}}^2/\delta^2-\theta\right)/(2\beta\theta)\right\}$ for the DTDSP, where $\sigma_{x_i x_{i'}}^2$ is the variance of cost differences between decisions x_i and $x_{i'}$. According to the problem feature, define $\sigma_{x_i x_{i'}}^2=$ $\sum_{j=1,...,|\Omega|;\theta=1,...,\theta_0}\left(F_{\text{cost}}^{\Delta}\left(x_i,x_{i'},\omega_j,\theta\right)-\bar{F}_{\text{cost}}^{\Delta}\left(x_i,x_{i'}\right)\right)^2/(\theta_0-1)$, where $F_{\text{cost}}^{\Delta}\left(x_i,x_{i'},\omega_j,\theta\right)$ denotes the difference between marginal costs, which are incurred by decisions x_i and $x_{i'}$ on problem realization ω_j and epoch indicated by θ, $\bar{F}^{\Delta}_{\text{cost}}\left(x_i, x_{i'}\right)$ is the difference between mean marginal costs incurred by decisions x_i and $x_{i'}$, over problem realizations Ω and all epochs indicated by the number from 1 to θ. Moreover, we evaluate different choices of parameters β and η, and find that $\beta = 1$ and $\eta = \left[\left(2\alpha / (n-1)\right)^{-2/(\theta_0 - 1)} - 1\right] / 2$ are typically good choices. After parameter tuning, we set parameter n (i.e., $\left|X^{\pi}_{\varphi}\left(u_{\varphi}\right)\right|$) to be $\left|\mathcal{H}\right|$; the cardinality of set Ω (i.e., $|\Omega|$) to be 100; and parameter θ_0 to be $\lfloor 0.4(\Phi - \varphi) \rfloor$. Numerical experiments demonstrate that decisions in space $X^{\pi}_{\varphi}\left(u_{\varphi}\right)$ can be efficiently estimated by the IZS-2-R algorithm, and then decision $x^* \in X^{\pi}_{\varphi}\left(u_{\varphi}\right)$ is eventually identified according to Equation (1) satisfying optimality in expectation.

Algorithm 5. IZS-based two-phase rollout algorithm

Input: Current state u_{φ}, set of feasible decisions $X^{\pi}_{\varphi}\left(u_{\varphi}\right) = \{x_1, ..., x_n\}$, set of problem realizations $\Omega = \{\omega_1, ..., \omega_q\}$, basic policy π, confidence level $1-\alpha$, indifference zone δ, and the number of evaluations θ_0 in Phase I for each problem realization

Output: The best decision $x^* \in X^{\pi}_{\varphi}\left(u_{\varphi}\right)$ in expectation for the current state u_{φ}

1: $i \leftarrow 1$, $j \leftarrow 1$, and $\theta \leftarrow 1$
2: # Phase I: Initial rollout (Lines 3-12)
3: **for** $i \le n$ **do**
4: $\quad$ $\sigma_i \leftarrow u_{\varphi}$ and $F_i \leftarrow F^{\Delta}_{\text{cost}}\left(u_{\varphi}, x_i\right)$
5: $\quad$ **for** $j \le |\Omega|$ **do**
6: $\quad\quad$ **for** $\theta \le \theta_0$ **do**
7: $\quad\quad\quad$ $u_{x_i} \leftarrow \left(\sigma_i, x_i\right)$, $\sigma_i \leftarrow \left(u_{x_i}, \omega_j\right)$, $x_i \leftarrow X^{\pi}\left(\sigma_i\right)$, $F_i \leftarrow F_i + F^{\Delta}_{\text{cost}}\left(\sigma_i, x_i\right) / |\Omega|$, and $\theta \leftarrow \theta + 1$
8: $\quad\quad$ **end for**
9: $\quad\quad$ $j \leftarrow j + 1$
10: $\quad$ **end for**
11: $\quad$ $i \leftarrow i + 1$
12: **end for**
13: # Phase II: Follow-up rollout as needed (Lines 14-29)
14: **if** $\sigma_i = u_{\Phi}$, **then**

$x^* \leftarrow \arg\min_{x_i \in X_\varphi^\pi(u_\varphi)} F_i$, return x^*, and terminate this algorithm

else

$\mathcal{X}_\varphi^{\text{refine}} = \left\{ x_i \mid x_i \in X_\varphi^\pi\left(u_\varphi\right) \text{ and } F_i \le F_{i'} + W_{x_i x_{i'}}(\theta),\ \forall x_{i'} \in X_\varphi^\pi\left(u_\varphi\right) \right\}$, $X_\varphi^\pi\left(u_\varphi\right) \leftarrow \mathcal{X}_\varphi^{\text{refine}}$

if $\left| X_\varphi^\pi\left(u_\varphi\right) \right| = 1$, **then**

$x^* \leftarrow x \in X_\varphi^\pi\left(u_\varphi\right)$, return x^*, and terminate this algorithm

else

for each decision $x_i \in X_\varphi^\pi\left(u_\varphi\right)$ **do**

$j \leftarrow 1$

for $j \le |\Omega|$ **do**

$u_{x_i} \leftarrow (\sigma_i, x_i)$, $\sigma_i \leftarrow \left(u_{x_i}, \omega_j\right)$, $x_i \leftarrow X^\pi(\sigma_i)$, $F_i \leftarrow F_i + F_{\text{cost}}^\Delta(\sigma_i, x_i)/|\Omega|$, and $j \leftarrow j+1$

end for

end for

$\theta \leftarrow \theta + 1$

end if

end if

repeat Phase II until the termination of this algorithm

5. Numerical Experiments

We set two goals for the numerical experiments. First, we validate the effectiveness and efficiency of our RPMDP and AADP by comparing them against three benchmarks. The first benchmark, a MILP model solving a hindsight problem with all requests confirmed at the beginning, acts as a theoretical cost lower bound for the DTDSP. The second, addressing a dynamic myopic problem without anticipating future demands, showcases the value of anticipation. The third, the current method used by a leading tugboat company, highlights the advantages of our approach in terms of cost savings and timely service. The second objective is to derive managerial insights for tugboat operations, involving sensitivity analysis on factors like task assignment strategy, waiting strategy, degree of dynamism, tugboat base layout, and problem size. These experiments are crucial for providing practical recommendations for tugboat deployment and scheduling. Experiments are performed using real operational data from the Singapore container port, implemented in Python 3.7, with Gurobi 9.0.1 acting as a general-purpose solver, on Intel(R) Xeon(R) CPU E5-2680 at 2.40 GHz with 32 GB RAM under

Linux 4.4.0.

5.1. Problem instances

Problem instances are generated using operational data of the Singapore container port, one of the busiest container ports in the world. The instance generation is summarized in three steps as follows. This generation method allows to simulate service demands across a continuous 1,000-day period. Then, different time segments (e.g., 12 hours) can be sampled from the 1,000-day simulation to create instances and problem realizations as needed.

(i) Generating origin-destination pairs: Utilizing AIS data, which tracks ships and tugboats every 10 seconds, we identify tugboat escorts from the spatial-temporal relationships and speed coincidences. From this data, we abstract 20 origin-destination pairs for berthing movements (where origin is the initial waypoint of a berthing ship and destination is the arrived berth) and 20 origin-destination pairs for unberthing movements (where origin is the departing berth and destination is the final waypoint) using clustering techniques tailored to the port's layout.

(ii) Simulating requests and time windows: We simulate service requests for each origin-destination pair by calibrating the intervals between service starts using historical data, confirming an exponential distribution for each pair. Time windows at berths are set based on simulated arrival and departure times, typically maintaining a 30-minute window referring to the practice.

(iii) Calibrating sail and service times: Sail times between locations and service times at berths are calibrated based on historical AIS data. Unlike conventional models that assume uniform tugboat speeds, our approach accounts for variable speeds among different tugboat types. Service times at berths are calculated as the difference between arrival and settle-down times for berthing, and between start-up and departure for unberthing.

To derive managerial insights for real-world conditions in busy ports, we conduct a sensitivity analysis by adjusting key parameters in the problem settings:

(i) Dynamism degree: Defined as the ratio of dynamic requests (confirmed during the service time horizon) to total requests, varying from 0.1 to 0.9. This reflects the proportion of requests not confirmed at the beginning of service time horizon (detailed in Section 5.4.3).

(ii) Tugboat base layout: We explore the operational impact of different tugboat base layouts by adjusting the number of base clusters from one to six, analyzing the operational efficiencies of dispersed versus centralized arrangements (see Section 5.4.4).

(iii) Problem size: The shift of tugboats is 12 hours and up to 14 hours when required in the Singapore container port, thus the service time horizon is maximally set to 14 hours. The Singapore container port experienced 17,514 container ship calls in 2019, translating to 28 ship calls every 14 hours, thus our models accommodate 28 berthing and 28 unberthing requests to mirror this activity, referred to as "28-28". To examine variations in demand, problem sizes range from 10-10 to 35-35 (refer to Section 5.4.5).

5.2. Benchmark solution methods

To demonstrate the effectiveness and efficiency of the developed modeling and solution methods, three benchmark methods are used for comparison.

(i) Deterministic deployment and scheduling (DDS): This approach uses a MILP model to solve a hindsight TDSP, where all service requests are presumed confirmed at the start of the service horizon, allowing for the formulation of a complete tugboat work plan upfront. This method provides a theoretical lower bound on system-wide costs by excluding dynamic and stochastic elements of service demands (detailed in Appendix B).

(ii) Dynamic deployment and scheduling without anticipation (DDSWA): Extending DDS, this method iteratively solves the MILP model. Initially, it assigns tasks of all confirmed requests. Upon new request confirmations, it reoptimizes for unfulfilled tasks, updating the work plan accordingly. Without anticipation, this method highlights the benefits of proactive future service demand anticipation, contrasting it with our advanced methods.

(iii) Current method used by the tugboat company: tugboat work plans are manually crafted by the Tugboat Mission Command Center (TMCC). This manual process can introduce inefficiencies, such as misassigning large tugboats to small ships or assigning tugboats to distant ships, resulting in feedback and dissatisfaction from the tugboat master and a need for plan revisions. These issues are illustrated by significant instances of tugboats arriving too early or late (see dark blue bars in Figure 6), which underscore the advantages of our optimization methods in enhancing cost-efficiency and timeliness.

5.3. Solution performance

We evaluate the effectiveness of our methods by comparing them against benchmark solutions across various settings. For unbiased insights, we configure problem instances using two representative tugboat base layouts, four problem sizes, and a moderate dynamism degree of 0.5. We generate 20 random instances for each configuration and average the results. Specifically, we compute the average total fixed cost, total sail cost, total cost, and response time for each instance. Response time, defined as the interval between the service start time for a ship and the earliest possible start time, serves as a secondary metric. While not the primary optimization goal in the tugboat industry, where meeting berth time window constraints is paramount, response time provides insights into service timeliness and helps in deriving managerial implications.

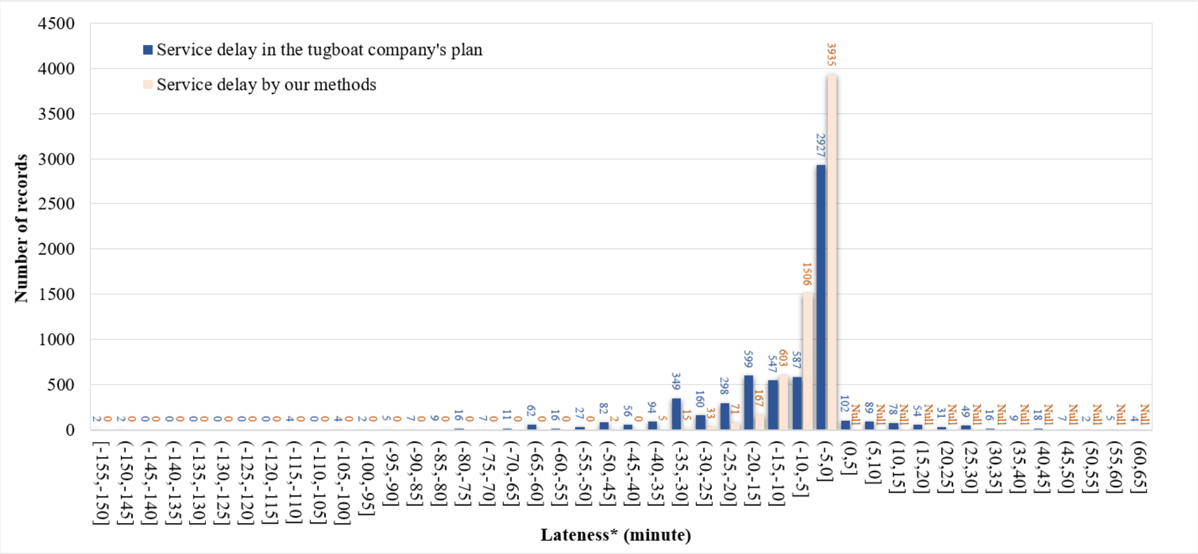
For each performance measure (see Table 5), we report the gap between the value obtained by our methods and that obtained by the benchmark methods. The results demonstrate that the developed RPMDP and AADP perform well in solving the DTDSP. In particular, the gap in total cost can be reduced to 8.4% compared with the theoretical lower bound for practical-size problem instances (e.g., 30-30) in the busy container port, and this gap can be further reduced for small-size instances. Additionally, the benefit of anticipating future service demands to reduce system-wide cost is demonstrated, via a comparative analysis with the DDSWA method. It is observed that the total cost is reduced by 14.9% with service demand anticipation. This significant cost reduction is primarily attributed to the routing and, more notably, the waiting decisions informed by the anticipation of future service demands, which effectively decrease non-towage sailing and the associated sailing costs. From the “Response time” column, it is observed that response times by the developed methods are shorter than those by the DDS method, indicating tugboats can provide service more timely by the developed methods than by the DDS method. Although response times by the developed methods are slightly larger than those by the DDSWA method, the solution by the developed methods is much more economical. An increase of 10% in response time (about 1.5 minutes) compared with the DDSWA method is deemed acceptable, particularly when it is compensated by a significant reduction in operation costs, translating into daily savings of thousands of dollars. Moreover, response times by our methods are within 15 minutes, fulfilling the Singapore container port's governmental standard for service timeliness (Maritime & Port Authority of Singapore, 2023). In summary, the developed RPMDP and AADP can effectively reduce system-wide tugboat operation cost to satisfy dynamic and stochastic requests with anticipating future service demands.

Table 5. Comparisons in tugboat operation costs and service timeliness.

No. of tugboat base clusters	Problem size*	Our method vs. DDS method				Our method vs. DDSWA method			
		Total fixed cost (%)	Total sail cost (%)	Total cost (%)	Response time (%)	Total fixed cost (%)	Total sail cost (%)	Total cost (%)	Response time (%)
1	15-15	106.4	104.3	105.3	85.1	93.4	92.6	92.9	110.5
	20-20	107.6	105.4	106.3	83.7	92.2	91.5	91.7	111.3
	25-25	109.2	106.3	107.2	81.6	90.6	89.8	90.2	111.9
	30-30	111.2	107.6	108.7	78.3	89.1	87.8	88.3	112.4
6	15-15	105.6	103.9	104.8	82.5	90.7	89.6	90.1	107.8
	20-20	107.0	105.1	105.9	80.7	89.2	88.4	88.7	108.2
	25-25	108.7	106.0	106.8	78.2	87.5	86.5	87.0	108.6
	30-30	110.7	107.4	**108.4**	74.7	85.8	84.6	**85.1**	109.1

*Notes: 15-15, 20-20, 25-25, and 30-30 represent “15 berthing requests and 15 unberthing requests”, “20 berthing requests and 20 unberthing requests”, “25 berthing requests and 25 unberthing requests”, and “30 berthing requests and 30 unberthing requests”, respectively.

We also compare our methods with the third benchmark method - the tugboat company's approach, using problem instances corresponding to June 2019. Specifically, we consider a sequence of 48 instances, each with a 15-hour service time horizon and moderate dynamism degree of 0.5. Calculating total fixed cost, total sail cost, total cost, and response time for our methods' plan and the tugboat company's plan, the gaps are 90.9%, 89.7%, 90.2%, and 98.1%, respectively. Namely, results show a 9.8% (equivalent to thousands of dollars per day) reduction in total operation cost by our method compared to the tugboat company's method. The high operation cost of the tugboat company's method is due to low tugboat utilization, indicated by many significant tugboat early arrivals at tasks (see Figure 6). Consequently, more tugboats will be deployed, increasing fixed costs. Additionally, the tugboat company's method often fails to allocate sufficient sail time for tugboats, resulting in tugboat rush to the next task at high speeds (rather than an economic speed such as 8 knots), which incurs higher sail cost (i.e., fuel consumption) for the covered distance. In contrast, our methods exhibit no service delays, unlike the notable delays by the tugboat company's method (see Figure 6). In summary, our method significantly outperforms the tugboat company's method in both cost-saving and service timeliness.



*Notes: A negative lateness value means an early arrival at one task, and a positive lateness value means a late arrival at one task.

Figure 6. Comparison in tugboat service timeliness by our methods and the tugboat company's method.

Subsequently, we demonstrate that the computing capability of our methods can meet the requirements of busy container ports. Figure 7 shows the computation time of different-size instances, in particular, the total computation time, number of states, and average computation time per decision are depicted in Figure 7 (a), (b), and (c), respectively. As explained in Section 5.1, the practical-size problem instance in the Singapore container port is 28-28 requests during a 14-hour service time horizon. Figure 7 illustrates that our methods can efficiently solve practical-size problem instances within 10 minutes per decision on average (see Figure 7 (c)). This time efficiency surpasses that of the method currently used by the tugboat company and proves to be satisfactory for tugboat deployment and scheduling in one of the busiest container ports.

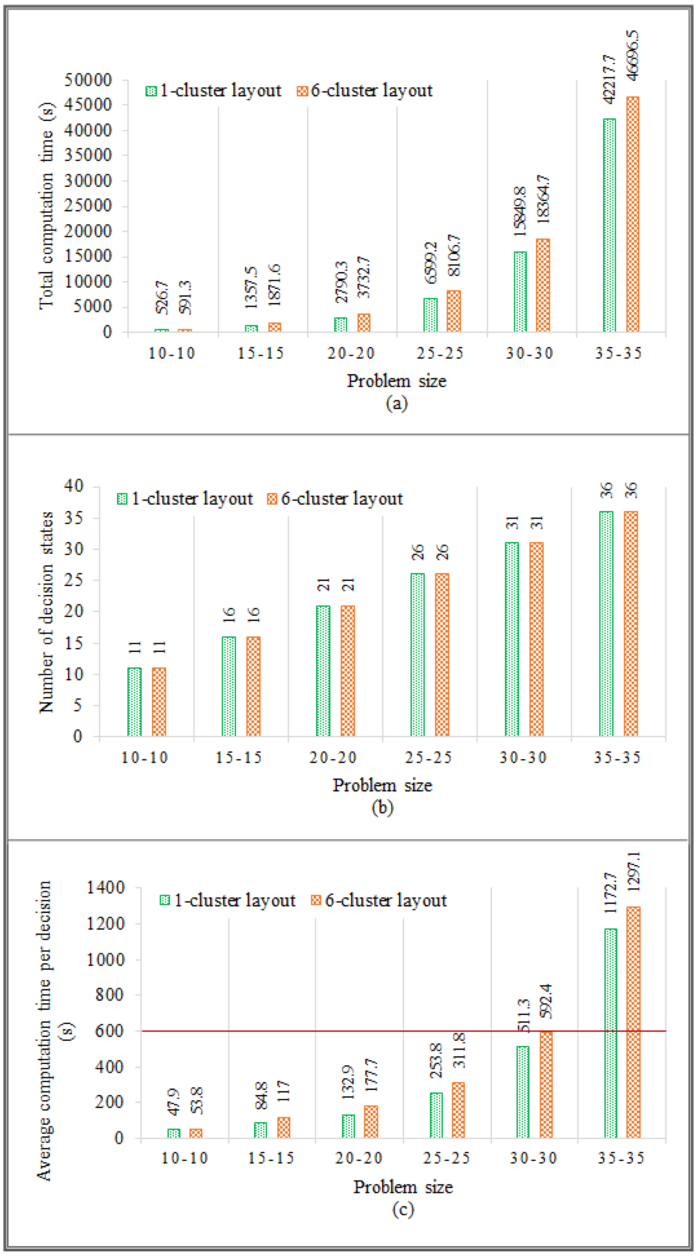


Figure 7. Computing capability of the developed RPMDP and AADP.

5.4. Sensitivity analysis

Tugboat deployment and scheduling are influenced by several factors including task assignment strategy, waiting strategy, dynamism degree, tugboat base layout, and problem size. To derive managerial insights and assess the impact of these factors on operational costs, service timeliness, and computation time, we perform sensitivity analysis. For unbiased evaluation, we generate 20 random instances for each setting and average the results.

5.4.1. Task assignment strategy

We examine the effect of four task assignment strategies. First, the sample-scenario planning (SSP) strategy, which performs well in Bent and Van Hentenryck (2004). The SSP constructs a plan to satisfy confirmed requests and simulated requests, then simulated requests are removed from the work plan, aiming to reserve vacancies in the current plan and enable incorporation of future requests. Moreover, Ghiani et al. (2012) highlight that the basic insertion methods have comparable performance to that of more computationally intensive strategies. This insight is employed to incorporate new requests into the current plan based on cheapest insertion, followed by improvement methods to enhance the plan. In summary, four task assignment strategies are compared: (i) SSP, (ii) cheapest insertion (CI), (iii) combination of cheapest insertion and tabu search (CI-TS), and (iv) combination of cheapest insertion, tabu search, and IZS-2-R (CI-TS-IR). Note that CI-TS-IR is the strategy used in the developed AADP (see Section 4), while the other three strategies (i.e., SSP, CI, and CI-TS) are used as benchmark strategies.

The results for instances with two representative tugboat base layouts are reported in Table 6. For each base layout, CI-TS-IR achieves the lowest total cost with acceptable computation time, showing it can explore solution space effectively. The fundamental reason is that CI-TS-IR gives work plans more chances to be optimized via the cheapest insertion, tabu search, and rollout algorithm; meanwhile, IZS is used to expedite the solution process. Overall, the total cost increases sequentially under strategies of CI-TS-IR, SSP, CI-TS, and CI. Though SSP achieves low costs comparable to CI-TS-IR, SSP requires much more computation time, making it unacceptable for real-time decisions. Specifically, the improvement in cost-saving by CI-TS-IR is more significant under the dispersed tugboat base layout, compared with that achieved under the centralized tugboat base layout, indicating that the advantage of CI-TS-IR can be exploited better with a dispersed tugboat base layout.

Table 6. Impact analysis of task assignment strategies (dynamism degree = 0.5, problem size = 30-30).

No. of tugboat base clusters	Task assignment strategy	Total fixed cost	Total sail cost	Total cost	No. of tugboats deployed	Response time (minute)	Computation time (second)
1	SSP	40700	80750.1	121450.1	19	10.5	30191.9
	CI	42600	127108.3	169708.3	20	8.3	661.7
	CI-TS	42500	92189.2	134689.2	20	9.1	1987.2
	CI-TS-IR	42000	74197.6	116197.6	20	10.6	15849.8
6	SSP	40500	78743.2	119243.2	19	9.7	35121.7
	CI	42400	124223.3	166623.3	20	7.8	791.6

	CI-TS	42200	88442.7	130642.7	20	8.1	2320.5
	CI-TS-IR	40200	71045.8	111245.8	19	8.9	18364.7

5.4.2. Waiting strategy

There is rich literature on waiting decisions. Based on a comprehensive review of related studies, we adopt five conventional waiting strategies from the literature (Larsen et al., 2004; Mitrović-Minić and Laporte, 2004; Thomas, 2007) as benchmarks to evaluate the proposed waiting strategy. The waiting location under each conventional strategy is determined as: (i) Current - the current location when the tugboat completes a task, (ii) Nearest - the nearest waiting point to the current location, (iii) Largest - the waiting point with the largest expected number of new tasks, (iv) Longest - the waiting point with the longest allowable waiting time, and (v) Highest - the waiting point with the highest probability of a new task. The waiting time under each conventional waiting strategy is determined as: the tugboat waits at the waiting location until no early arrival at the next task.

As reported in Table 7, the proposed waiting strategy outperforms benchmark strategies, reducing the total cost by 17.95% on average. The proposed strategy is better than conventional strategies because it comprehensively considers the probability of future requests, the distribution of idle tugboats, the sail cost to a waiting point, and the available waiting time at a waiting location. Notably, the proposed strategy is especially good in reducing the number of deployed tugboats, demonstrating the capability of reducing fixed costs to operate a tugboat fleet. The total cost increases sequentially under conventional strategies of Nearest, Current, Largest, Longest, and Highest. Compared with strategy Current, the other five waiting strategies need to calculate waiting points, and thus more computation time is needed. In addition, we observe that the proposed waiting strategy performs better under the dispersed tugboat base layout, namely, the improvement in cost-saving is more significant under the dispersed base layout. This comparative analysis covers the impact on tugboat operation costs, response time, and computation time to demonstrate the superiority of the proposed waiting strategy. In summary, our strategy enables effective waiting decisions for practical-size instances in a reasonable time, meeting the requirement for dynamic decisions in busy container ports.

Table 7. Impact analysis of waiting strategies (dynamism degree = 0.5, problem size = 30-30).

No. of tugboat base clusters	Waiting strategy	Total fixed cost	Total sail cost	Total cost	No. of tugboats deployed	Response time (minute)	Computation time (second)

1	Proposed	42000	74197.6	116197.6	20	10.6	15849.8
	Current	44600	93075.3	137675.3	21	9.1	10159.2
	Nearest	44400	83543.5	127943.5	21	10.3	15396.8
	Largest	44500	99075.9	143575.9	21	8.8	14966.5
	Longest	46000	99927.1	145927.1	22	8.4	14694.8
	Highest	46200	101446.6	147646.6	22	7.9	14569.1
6	Proposed	40200	71045.8	111245.8	19	8.9	18364.7
	Current	42200	92281.5	134481.5	20	8.1	11667.5
	Nearest	42200	82278.8	124478.8	20	8.8	18127.4
	Largest	44000	96038.6	140038.6	21	7.7	17981.7
	Longest	44300	98714.5	143014.5	21	7.6	17655.2
	Highest	44500	100540.2	145040.2	21	7.2	17568.9

5.4.3. Dynamism degree

We examine the impact of dynamism degree on problem solutions. Table 8 presents the comparison of instances with varying dynamism degrees from 0.1 to 0.9 under two representative tugboat base layouts. For each base layout, as the dynamism degree increases, the total cost increases because more tugboat fixed costs and sail costs are incurred to account for larger uncertainty. The rate of change in tugboat fixed cost is larger than that in sail cost, indicating that fixed cost is more sensitive to the level of uncertainty. As the dynamism degree increases, tugboats will respond faster to customer ships, due to fewer tasks will be assigned to each deployed tugboat and this makes it possible to start service for each ship more timely. Under a larger dynamism degree, more requests are confirmed during the service time horizon, which results in more states and thus more computation time is required. Moreover, compared with the centralized base layout, the dispersed base layout is more tolerant to uncertainty of problem realization, exhibiting smaller changes in total costs given the same variation in dynamism degree.

Table 8. Impact analysis of dynamism degrees (problem size = 30-30).

No. of tugboat base clusters	Dynamism degree	Total fixed cost	Total sail cost	Total cost	No. of tugboats deployed	Response time (minute)	Computation time (second)
1	0.1	37100	74564.2	111664.2	18	13.4	5169.9
	0.3	39300	73903.7	113203.7	19	11.8	8339.7
	0.5	42000	74197.6	116197.6	20	10.6	15849.8
	0.7	42100	79182.6	121282.6	20	9.5	32359.6
	0.9	44900	82922.1	127822.1	21	8.7	69379.4
6	0.1	36700	71070.9	107770.9	18	10.9	5990.2
	0.3	39000	69986.9	108986.9	19	9.8	9763.6
	0.5	40200	71045.8	111245.8	19	8.9	18364.7
	0.7	41900	73463.4	115363.4	20	8.1	36056.4
	0.9	44700	76350.1	121050.1	21	7.5	74230.5

5.4.4. *Tugboat base layout*

We quantitatively assess the impact of tugboat base layout on solution performance by varying the number of base clusters from one to six. This variation represents different degrees of base layout discretization. To examine the impact of changing base layout on different performance metrics (e.g., total cost), we use the centralized one-cluster base layout as a baseline, and present relative values for other base layouts in a heatmap. Thus, the trend and degree of change in each performance metric (i.e., each column) can be easily found (see Figure 8). Our findings reveal that the total cost will be reduced if a more dispersed base layout is provided, especially when the number of base clusters increases from one to two. Meanwhile, the response time decreases by 16.04% (from 10.6 minutes to 8.9 minutes) when enlarging the number of base clusters from one to six, that is, a more dispersed base layout can speed up the tugboat response to customer ships. These results quantitatively show that more dispersed base layouts can benefit tugboat operations, in terms of reducing total costs and response times. Notably, the impact of varying the number of base clusters is less significant, when the number of base clusters exceeds four. Moreover, with enlarging the discretization degree of tugboat base layout, more tentative matches between tugboats and tasks will be investigated for decision-making, correspondingly, more computation time will be required to calculate solutions.

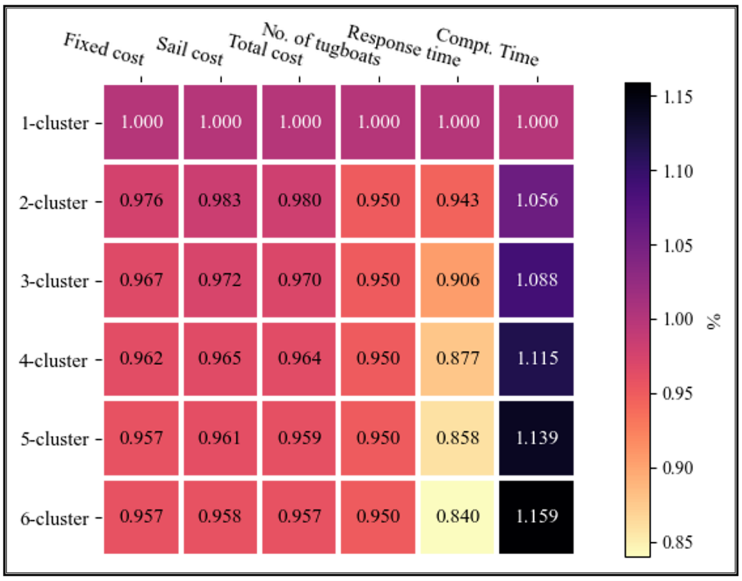


Figure 8. Impact analysis of tugboat base layouts (dynamism degree = 0.5, problem size = 30-30).

5.4.5. *Problem size*

We investigate the impacts of problem size on solution performance. The results for two representative tugboat base layouts are reported in Table 9. For each base layout, it is quantificationally observed that more operation costs are incurred with larger problem sizes. Specifically, both fixed cost and sail cost conform to a nondecreasing trend with the increase in problem size, while the increment in sail cost is more significant than that in fixed cost on average. Notably, there are "economies of scale" in tugboat operations, that is, the average cost to fulfill one task decreases when there are more tasks to be fulfilled during the service time horizon. The change in response time is related to the problem size and the number of tugboats deployed. Notably, the number of states increases when the problem instance enlarges, which requires more computation time to calculate solutions. Moreover, the dispersed base layout exhibits a bit less sensitivity to problem size expansion, exhibiting smaller growth in operation costs given the same increment in problem size.

Table 9. Impact analysis of problem sizes (dynamism degree = 0.5).

No. of tugboat base clusters	Problem size	Total fixed cost	Total sail cost	Total cost	No. of tugboats deployed	Response time (minute)	Computation time (second)
1	10-10	23800	35365.6	59165.6	11	6.9	526.7
	15-15	30100	45210.1	75310.1	14	7.5	1357.5
	20-20	34400	55301.2	89701.2	16	8.2	2790.3
	25-25	38500	64915.8	103415.8	18	9.1	6599.2
	30-30	42000	74197.6	116197.6	20	10.6	15849.8
	35-35	44200	83555.8	127755.8	21	12.2	42217.7
6	10-10	23700	33900.4	57600.4	11	6.1	591.3
	15-15	30100	43922.1	74022.1	14	6.5	1871.6
	20-20	34200	53046.3	87246.3	16	7.1	3732.7
	25-25	36400	60293.4	96693.4	17	7.8	8106.7
	30-30	40200	71045.8	111245.8	19	8.9	18364.7
	35-35	42100	80889.5	122989.5	20	10.3	46696.5

6. Conclusions

In this paper, we propose the new problem DTDSP, which arises from daily port transportation and distinguishes from existing deployment and scheduling problems due to unique challenges (e.g., time-varying service demand). In the DTDSP, service requests are not fully known initially but dynamically confirmed when the current deployment and scheduling plan is being executed. This requires the anticipation of future service demands when updating the current plan in order to reduce the system-wide tugboat operation costs. To formulate the DTDSP, we develop the reactive and proactive RPMDP model, which incorporates plans on both task assignment and waiting decisions. To calculate solutions, we develop the AADP method to dynamically update deployment and scheduling plans for real-time decisions, resorting to an IZS-2-R algorithm to expedite the solution process. The developed modeling and solution methods are compared against two well-established benchmark methods and the method currently used by the leading tugboat company, showing superior effectiveness and efficiency in generating high-quality solutions within reasonable computation times for major hubs like the Singapore container port. In addition, managerial insights are investigated via sensitivity analysis of various problem settings to provide helpful references for tugboat operations in practice. Notice that the DTDSP generalizes deployment and scheduling problems via incorporating a range of features (e.g., stochastic service demands, dynamic routing and waiting, time-varying service demands, multi-route coordination to serve a single customer), thus the developed modeling and solution methods can be referred to solve other dynamic and stochastic problems on vehicle deployment via adjusting or relaxing specific constraints by analogy.

Future research could explore two promising extensions of this work. Firstly, the investigation of a robust TDSP merits attention. This would account for variability in tugboat sailing times and focus on ensuring service reliability within a probabilistic framework. Utilizing distributionally robust optimization could prevent overfitting to historical data, thereby enhancing the method's out-of-sample performance. Secondly, strategic fleet planning for tugboats offers significant potential. Optimizing the fleet size and composition - through acquisitions, sales, and charters - poses a substantial challenge due to the long-term nature of these decisions and inherent uncertainties. An advanced model that integrates strategic, tactical, and operational considerations could provide valuable insights for fleet composition in anticipation of future service needs.

Acknowledgments

This study is supported by the Singapore Maritime Institute via the project SMI-2022-MTP-09. Any opinions, findings and conclusions or recommendations expressed in this study are those of the author(s) and do not reflect the views of the Singapore Maritime Institute. We acknowledge the leading tugboat company, PSA Marine, for providing data on daily tugboat operations and their current deployment and scheduling approach. The knowledge shared by PSA Marine is essential for defining the research problem. Their method serves as one benchmark, showcasing the advantages of our developed methods.

Appendices

Appendix A. Notations for the DTDSP

(i) Sets

$\mathcal{S}$	Set of ships, specifically, $\mathcal{S}=\mathcal{S}_{\text{ber}}\cup\mathcal{S}_{\text{unb}}$, where $\mathcal{S}_{\text{ber}}$ is the set of berthing ships, $\mathcal{S}_{\text{unb}}$ is the set of unberthing ships; for presentation simplicity, denote s as the tugging service request made by ship $s\in\mathcal{S}$
$\mathcal{P}$	Set of waypoints of berthing ships, specifically, $\mathcal{P}_s=\left(p_{s1},p_{s2},...,p_{s\Gamma_s}\right)$ are waypoints of berthing ship s, where $p_{s\gamma}$ is the γth waypoint to be visited by ship s, and $\mathcal{P}=\bigcup_{s\in\mathcal{S}_{\text{ber}}}\mathcal{P}_s$
$\mathcal{D}$	Set of waypoints of unberthing ships, specifically, $\mathcal{D}_s=\left(d_{s1},d_{s2},...,d_{s\Gamma_s}\right)$ are waypoints of unberthing ship s, where $d_{s\gamma}$ is the γth waypoint to be visited by ship s, and $\mathcal{D}=\bigcup_{s\in\mathcal{S}_{\text{unb}}}\mathcal{D}_s$
$\mathcal{B}$	Set of berths, specifically, $\mathcal{B}_{\text{ber}}=\left\{b_s \mid s\in\mathcal{S}_{\text{ber}}\right\}$ are berths associated with berthing ships, $\mathcal{B}_{\text{unb}}=\left\{b_s \mid s\in\mathcal{S}_{\text{unb}}\right\}$ are berths associated with unberthing ships, and $\mathcal{B}=\mathcal{B}_{\text{ber}}\cup\mathcal{B}_{\text{unb}}$
$\mathcal{H}$	Set of tugboats, specifically, $\mathcal{H}_s$ are tugboats that have adequate tugging power to serve ship s, $\mathcal{H}_\varphi$ are tugboats available at epoch φ
$\mathcal{O}$	Set of tugboat bases, $\mathcal{O}=\left\{o_h \mid h\in\mathcal{H}\right\}$
$\mathcal{F}$	Set of waiting points
$\mathcal{N}$	Set of locations in port waters, $\mathcal{N}=\mathcal{O}\cup\mathcal{P}\cup\mathcal{D}\cup\mathcal{B}\cup\mathcal{F}$
$\mathcal{E}$	Set of tugging tasks generated by service requests, specifically, $\mathcal{E}_s$ are tasks generated by request s, $\mathcal{E}_\varphi$ are confirmed but unfulfilled tasks at state u_φ, and $\mathcal{E}_{\text{unf}}$ are confirmed but unfulfilled tasks at the current state
$\mathcal{U}$	Set of states during decision process, specifically, u_0 is the initial state and u_Φ is the final state
Ω	Set of problem realizations, $\Omega=\bigcup_{\varphi=0,1,...,\Phi}\Omega_\varphi$

(ii) Parameters

$[0,T]$	Service time horizon
λ_h	Fixed cost of tugboat h
c_h	Unit-time sail cost of tugboat h
t_s	Confirmation time of service request s
Γ_s	The number of tugboats required by ship s
e	Task, $e \in \mathcal{E}$, specifically, k_e is the tugboat type required by task e, o_e is the starting location of task e, and d_e is the destination location of task e
$t_{i,j}^s$	Sail time of ship s between locations i and j
$t_{i,j}^h$	Sail time of tugboat h between locations i and j
φ	Decision epoch during the decision process, specifically, φ_0 is the initial epoch and φ_Φ is the final epoch
u_φ	State at decision epoch φ
$[\underline{t}_s, \overline{t}_s]$	Service time window at berth b

(iii) Decision variables

x	Decision, specifically, x_φ is the decision at epoch φ, i.e., a plan on tugboat deployment and scheduling; moreover, $x = (x_h)_{h \in \mathcal{H}}$ where x_h is the individual plan for tugboat h
u_φ^x	Post-decision state when decision x is taken at state u_φ
$t_{\text{start}}^e, t_{\text{end}}^e$	Start and end times of task e
r	Tugboat route, specifically, r_e is the tugboat route including task e
π	Policy, essentially, π is a sequence of decision rules $\left(X_0^\pi, ..., X_\varphi^\pi, ..., X_\Phi^\pi\right)$, each of which is a function that maps a state u_φ to a decision $x_\varphi \in X_\varphi^\pi\left(u_\varphi\right)$, moreover, π^* is an optimal policy

Appendix B. MILP for the hindsight TDSP

Additional notations used for the hindsight TDSP:

(i) Sets

$\mathcal{N}'$ Set of locations in port waters, $\mathcal{N}' = \mathcal{O} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{B}$

$\mathcal{A}$ Set of arcs among locations $\mathcal{N}'$

$\delta^-(i)$ Set of predecessor locations of location i

$\delta^+(i)$ Set of successor locations of location i

(ii) Parameters

$t_{\text{arrive}}^{s,p_{s1}}$ The arrival time of berthing ship $s \in S_{\text{ber}}$ at the first waypoint p_{s1}

$I(i)$ Index of waypoint i

$\bar{M}$ A large number

(iii) Decision variables

y_h Equals 1 if tugboat h is deployed during the service time horizon, and 0 otherwise

$x_{i,j}^h$ Equals 1 if tugboat h sails on arc (i, j), and 0 otherwise

$t_{\text{arrive}}^{s,i}$ Arrival time of ship s at location i ($s \in S_{\text{ber}}$, $i \neq p_{s1}$)

$t_{\text{arrive}}^{h,i}$ Arrival time of tugboat h at location i

t_{depart}^{i} Departure time from location i

The hindsight TDSP is formulated as a MILP model on directed graph $\mathcal{G} = (\mathcal{N}', \mathcal{A})$ (see Figure 4), in which $\mathcal{N}' = \mathcal{O} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{B}$ denotes the set of locations and $\mathcal{A} = \{(i, j) \mid i, j \in \mathcal{N}', i \neq j\}$ denotes the set of arcs. As depicted in Figure 2 (a) for example, berthing ship $s \in S_{\text{ber}}$ sails through a sequence of waypoints $\mathcal{P}_s$ when approaching a destination berth. For each location $i \in \mathcal{P}_s$, $I(i)$ is the index of the waypoint denoted by location i, i.e., $I(i) = \gamma$ if location i is the γth waypoint of berthing ship $s \in S_{\text{ber}}$. Similarly, unberthing ship $s \in S_{\text{unb}}$ requires tugging service when departs from berth b_s and sails through a sequence of waypoints $\mathcal{D}_s$ as depicted in Figure 2 (b) for example. For each location $i \in \mathcal{D}_s$, $I(i)$ is the index of the waypoint denoted by location i, i.e., $I(i) = \gamma$ if location i is the γth waypoint of unberthing ship $s \in S_{\text{unb}}$. The MILP model for the hindsight TDSP is formulated as follows.

Model [DDS]: min $\sum_{h\in\mathcal{H}} \lambda_h y_h + \sum_{h\in\mathcal{H}} \sum_{i,j\in\mathcal{N}',i\neq j} c_h t^h_{i,j} x^h_{i,j}$ (A1)

s.t. $\begin{cases} \sum_{h\in\mathcal{H}} \sum_{j\in\delta^+(i)} x^h_{i,j} = I(i) \\ \sum_{h\in\mathcal{H}} \sum_{j\in\delta^+(b_s)} x^h_{b_s,j} = \Gamma_s \end{cases} \quad \forall s\in S_{\text{ber}}, i\in\mathcal{P}_s$ (A2)

$$\begin{cases} \sum_{h\in\mathcal{H}} \sum_{j\in\delta^+(b_s)} x^h_{b_s,j} = \Gamma_s \\ \sum_{h\in\mathcal{H}} \sum_{j\in\delta^+(i)} x^h_{i,j} = \Gamma_s - I(i) + 1 \end{cases} \quad \forall s\in S_{\text{unb}}, i\in\mathcal{D}_s \tag{A3}$$

$$\sum_{j\in\mathcal{N}'} x^h_{i,j} = 0 \quad \forall h\in\mathcal{H}, i\in O\setminus\{o_h\} \tag{A4}$$

$$\sum_{j\in\delta^+(o_h)} x^h_{o_h,j} = y_h \quad \forall h\in\mathcal{H} \tag{A5}$$

$$\sum_{j\in\delta^+(i)} x^h_{i,j} = \sum_{j'\in\delta^-(i)} x^h_{j',i} \le 1 \quad \forall h\in\mathcal{H}, i\in\mathcal{N}' \tag{A6}$$

$$\begin{cases} x^h_{p_{s\gamma},p_{s\gamma+1}} - \sum_{j\in\mathcal{N}'} x^h_{p_{s\gamma},j} \ge 0 \\ x^h_{p_{s\Gamma_s},b_s} - \sum_{j\in\mathcal{N}'} x^h_{p_{s\Gamma_s},j} \ge 0 \end{cases} \quad \forall s\in S_{\text{ber}}, h\in\mathcal{H}, \gamma = 1,\ldots,\Gamma_s - 1 \tag{A7}$$

$$\begin{cases} x^h_{b_s,d_{s1}} - \sum_{i\in\mathcal{N}'} x^h_{i,d_{s1}} \ge 0 \\ x^h_{d_{s\gamma},d_{s\gamma+1}} - \sum_{i\in\mathcal{N}'} x^h_{i,d_{s\gamma+1}} \ge 0 \end{cases} \quad \forall s\in S_{\text{unb}}, h\in\mathcal{H}, \gamma = 1,\ldots,\Gamma_s - 1 \tag{A8}$$

$$t^{s,j}_{\text{arrive}} = t^i_{\text{depart}} + t^s_{i,j} \quad \forall s\in S_{\text{ber}}, i\in\mathcal{P}_s, j\in\delta^+(i) \tag{A9}$$

$$t^{s,j}_{\text{arrive}} = t^i_{\text{depart}} + t^s_{i,j} \quad \forall s\in S_{\text{unb}}, j\in\mathcal{D}_s, i\in\delta^-(j) \tag{A10}$$

$$\underline{t}_s \le t^{s,b_s}_{\text{arrive}} \le \bar{t}_s \quad \forall s\in S_{\text{ber}} \tag{A11}$$

$$\begin{cases} t^{h,p_{s\gamma+1}}_{\text{arrive}} \ge t^{s,p_{s\gamma+1}}_{\text{arrive}} - \bar{M}\left(1 - x^h_{p_{s\gamma},p_{s\gamma+1}}\right) \\ t^{h,b_s}_{\text{arrive}} \ge t^{s,b_s}_{\text{arrive}} - \bar{M}\left(1 - x^h_{p_{s\Gamma_s},b_s}\right) \end{cases} \quad \forall s\in S_{\text{ber}}, h\in\mathcal{H}, \gamma = 1,\ldots,\Gamma_s - 1 \tag{A12}$$

$$\begin{cases} t^{h,d_{s1}}_{\text{arrive}} \ge t^{s,d_{s1}}_{\text{arrive}} - \bar{M}\left(1 - x^h_{b_s,d_{s1}}\right) \\ t^{h,d_{s\gamma+1}}_{\text{arrive}} \ge t^{s,d_{s\gamma+1}}_{\text{arrive}} - \bar{M}\left(1 - x^h_{d_{s\gamma},d_{s\gamma+1}}\right) \end{cases} \quad \forall s\in S_{\text{unb}}, h\in\mathcal{H}, \gamma = 1,\ldots,\Gamma_s - 1 \tag{A13}$$

$$t^{h,j}_{\text{arrive}} \ge t^i_{\text{depart}} + t^h_{i,j} - \bar{M}\left(1 - x^h_{i,j}\right) \quad \forall h\in\mathcal{H}, i\in O\cup\mathcal{D}\cup\mathcal{B}_{\text{ber}}, j\in\delta^+(i) \tag{A14}$$

$$0 \le t^{h,i}_{\text{arrive}} \le T \quad \forall h\in\mathcal{H}, i\in\mathcal{N}' \tag{A15}$$

$$t^i_{\text{depart}} \ge t^{s,i}_{\text{arrive}} \quad \forall s\in S_{\text{ber}}, i\in\mathcal{P}_s\cup\{b_s\} \tag{A16}$$

$$t_{\text{depart}}^{i} \geq t_{\text{arrive}}^{h,i} - \bar{M}\left(1-\sum_{j\in\delta^{-}(i)} x_{j,i}^{h}\right) \quad \forall s \in \mathcal{S}_{\text{ber}}, h \in \mathcal{H}, i \in \mathcal{P}_{s} \cup \{b_{s}\} \tag{A17}$$

$$\underline{t}_{s} \leq t_{\text{depart}}^{b_s} \leq \overline{t}_{s} \quad \forall s \in \mathcal{S}_{\text{unb}} \tag{A18}$$

$$t_{\text{depart}}^{i} \geq t_{\text{arrive}}^{h,i} - \bar{M}\left(1-\sum_{j\in\delta^{-}(i)} x_{j,i}^{h}\right) \quad \forall s \in \mathcal{S}_{\text{unb}}, h \in \mathcal{H}, i \in \{b_{s}\} \cup \mathcal{D}_{s} \tag{A19}$$

$$t_{\text{depart}}^{i} \geq t_{\text{arrive}}^{s,i} \quad \forall s \in \mathcal{S}_{\text{unb}}, i \in \mathcal{D}_{s} \tag{A20}$$

$$0 \leq t_{\text{depart}}^{o_h} \leq T \quad \forall h \in \mathcal{H} \tag{A21}$$

$$x_{i,j}^{h} = 0 \quad \forall s \in \mathcal{S}_{\text{ber}}, h \in \mathcal{H} \setminus \mathcal{H}_{s}, i \in \delta^{-}(j), j \in \mathcal{P}_{s} \tag{A22}$$

$$x_{i,b_s}^{h} = 0 \quad \forall s \in \mathcal{S}_{\text{unb}}, h \in \mathcal{H} \setminus \mathcal{H}_{s}, i \in \delta^{-}(b_{s}) \tag{A23}$$

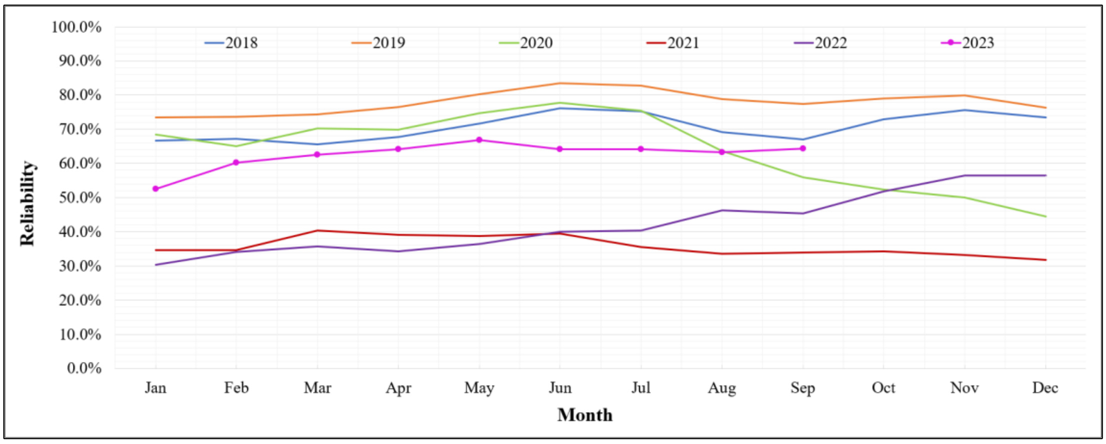
$$t_{\text{arrive}}^{s,i}, t_{\text{arrive}}^{h,i}, t_{\text{depart}}^{i} \geq 0 \quad \forall s \in \mathcal{S}, h \in \mathcal{H}, i \in \mathcal{N}' \tag{A24}$$

$$y_{h}, x_{i,j}^{h} \in \{0,1\} \quad \forall h \in \mathcal{H}, i, j \in \mathcal{N}' \tag{A25}$$

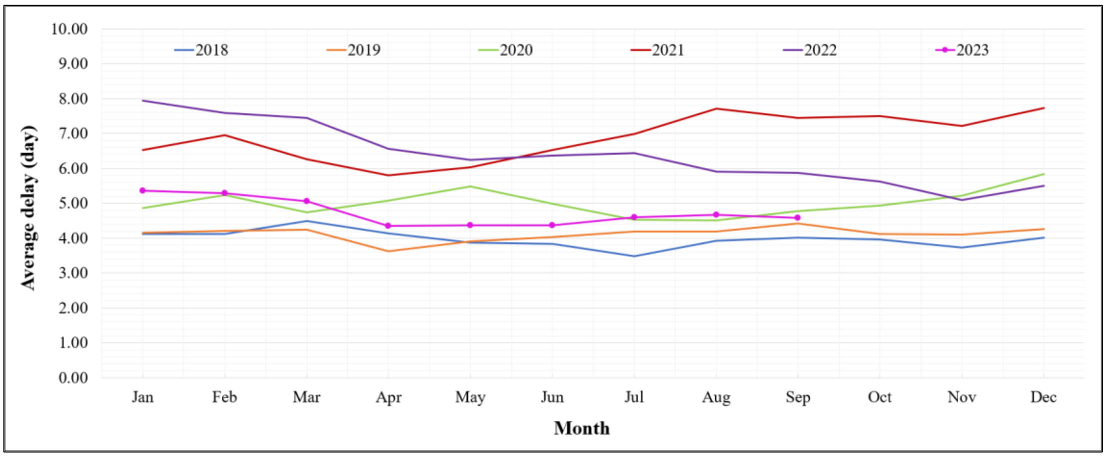
The objective function (A1) is to minimize the total tugboat operation cost, including fixed cost and sail cost. All the constraints can be classified into six categories. (i) *Constraints (A2-A8) are about tugboat flow.* The number of tugboats required to serve each berthing ship is ensured by Equation (A2). The number of tugboats required to serve each unberthing ship is ensured by Equation (A3). Equation (A4) shows that one tugboat can only depart from and return to its base. Equation (A5) marks the tugboat that is used during the service time horizon for calculating the fixed cost. Equation (A6) is the tugboat flow balance constraint. Inequality (A7) ensures that tugboat h is used by berthing ship $s \in \mathcal{S}_{\text{ber}}$ until arrives at destination b_s, if tugboat h has started service for berthing ship s. Similarly, Inequality (A8) ensures that tugboat h is used by unberthing ship $s \in \mathcal{S}_{\text{unb}}$ until arrives at the designated location, if tugboat h has started service for unberthing ship s. (ii) *Constraints (A9-A11) are about ship arrival times.* Equations (A9-A10) ensure that the ship arrival time at location j equals the departure time at the last location i plus ship sail time $t_{i,j}^{s}$. Inequality (A11) shows that the arrival time of ship $s \in \mathcal{S}_{\text{ber}}$ at berth b_s should be limited by time window $[\underline{t}_s, \overline{t}_s]$. (iii) *Constraints (A12-A15) are about tugboat arrival times.* Inequalities (A12-A13) guarantee that the arrival of tugboat h at one location cannot be earlier than that of the ship being served by tugboat h. Inequality (A14) guarantees that the tugboat arrival time at location j equals the departure time at the last location i plus tugboat sail time $t_{i,j}^{h}$. Inequality (A15) limits the tugboat arrival time at one location by the service time horizon, specifically, tugboat h should return to base o_h no later than time T. (iv) *Constraints (A16-A21) are about departure times.* Inequalities (A16-A17) show that the departure time from locations $\mathcal{P} \cup \mathcal{B}_{\mathrm{ber}}$ cannot be earlier than the arrival time at the location. Inequality (A18) ensures that the departure time from berths $\mathcal{B}_{\mathrm{unb}}$ is limited by the time window. Inequality (A19) shows that the departure time from locations $\mathcal{B}_{\mathrm{unb}} \cup \mathcal{D}$ cannot be earlier than the tugboat arrival time at the locations. Inequality (A20) ensures that the departure time at locations $\mathcal{D}$ cannot be earlier than the ship arrival time at the locations. Inequality (A21) shows that tugboat departure time from bases O should be limited by the service time horizon, specifically, tugboat h cannot depart from base o_h before time 0. (v) *Constraints (A22-A23) describe ship-tugboat compatibility.* Particularly, only tugboats that are capable of serving ship $s \in S$ can be assigned to ship s. (vi) *Constraints (A24-A25) specify the domains of decision variables.*

Appendix C. Stochasticity in container ship arrival and departure times

The stochasticity of container ship arrival times at ports is evidenced across tens of trade lanes and shipping companies (Madsen, 2023). Specifically, the reliability of port-calling schedule is shown in Figure A1(a), and the average delay of late container ships is reported in Figure A1(b). For another example, a considerable deviation in ship arrivals was observed at the Ningbo-Zhoushan Port (a world-class port), where less than 20% of container ships arrived at the latest ETA, while the remaining 80% experienced early or late arrivals spanning several hours (see Table A1), as reported by Yu et al. (2018). Moreover, the stochasticity in ship arrival and departure times has also been corroborated by studies including Umang et al. (2017), Yu et al. (2018), Guo et al. (2021), Xiang et al. (2021), and Rodrigues et al. (2022). Obviously, the stochasticity of ship arrival times at ports and operation times at terminals influences the stochasticity of ship departure times (Umang et al., 2017; Guo et al., 2021; Xiang et al., 2021; Rodrigues et al., 2022). Consequently, the stochasticity in both ship arrival and departure times incurs the stochasticity of tugging service start time. Additionally, the stochastic nature such as service start times, has also been recognized by our industry collaborator, PSA Marine, a leading tugboat company operating in the Singapore container port. In summary, the stochasticity in ship arrival and departure times and the associated service start times is prevalent, and the anticipation can yield practical benefits to enhance port operations (Umang et al., 2017; Yu et al., 2018; Guo et al., 2021; Xiang et al., 2021; Rodrigues et al., 2022; Zhang et al., 2022).



(a) Reliability of port-calling schedule for container ships.



(b) The average delay of late container ships.

Figure A1. The stochasticity of container ship arrival times (Madsen, 2023).

Table A1. Deviations in container ship arrival times (adapted from Yu et al. (2018)).

Total arrivals	Minimum deviation (hour)	Maximum deviation (hour)	Mean deviation (hour)	Standard deviation (hour)
2066	-42.7*	37.5*	0.5	4

*Notes: A negative value indicates an early arrival, and a positive value indicates a late arrival.

Appendix D. An example of RPMDP solution

An example of RPMDP solution for the DTDSP details the management of service requests during a service time horizon, as outlined in Table A2. Initially, request s and request s''' have been confirmed at the beginning of service time horizon, request s'' is confirmed at time 01:00, subsequently, request s' is confirmed at 02:30. There are four tugboats available to serve ships, including two type-B tugboats (tugboats h and h') and two type-M tugboats (tugboats h'' and h'''). The unit-time sail cost of type-M tugboats and type-B tugboats are 1.5 and 2, respectively. The deployment and scheduling of these tugboats in response to the requests are illustrated in Figure A2.

(i) *Initial decision state* u_0. In the top-left panel, the tugging service system occupies the initial decision state u_0, where an initial decision is made for the currently confirmed requests. Specifically, tugboats h and h' are scheduled to serve unberthing ship s''', while tugboats h'' and h''' are scheduled to serve berthing ship s. The dashed line indicates subsequent actions to perform assigned tasks. In the RPMDP, if the current work plan leads to an early arrival at one task (or base), a waiting action is decided for the tugboat before such arrival. We use a line starting and ending at the same location to represent a waiting action. It should be pointed out that, tugboat h' is decided to wait at an idle point f based on anticipation of future service demands, and tugboat h'' waits at base $o_{h''}$ for a certain time to avoid early arrival at the first task (see the top-left panel in Figure A2). The subsequent post-decision state $u_0^{x_0}$ is depicted in the top-right panel in Figure A2.

(ii) *Decision state* u_1. The confirmation of request s'' leads to a new decision state (i.e., u_1), accordingly, a decision is made to update the current tugboat work plan to incorporate request s'' (see the mid-left panel in Figure A2). This decision assigns the new task of request s'' to tugboat h'', with anticipation of future service demands. The subsequent post-decision state $u_1^{x_1}$ is illustrated in the mid-right panel in Figure A2.

(iii) *Decision state* u_2. The confirmation of request s' triggers a new decision state. The new decision assigns the first task of ship s' to tugboat h', and assigns the second task of ship s' to tugboat h.

Finally, all the tugboats return to the base at the end of service time horizon. Notice that all the four ships (i.e., s, s', s'', and s''') are served with satisfying the time windows at berths. The sail times of tugboats h, h', h'', and h''' are shown in Table A3, and the sail cost is 2×(215+325) + 1.5×(220+200) = 1710.

Table A2. An example of service-request realization.

Request	Berthing: Request confirmation time	Berthing: Time window of ship arrival at berth	Berthing: Time duration of berthing operation	Berthing: Required tugboats	Request	Unberthing: Request confirmation time	Unberthing: Time window of ship departure from berth	Unberthing: Time duration of unberthing operation	Unberthing: Required tugboats
s	00:00	[02:20, 02:50]	Task 1: 100 minutes Task 2: 60 minutes	2 type-M	s'''	00:00	[00:30, 01:00]	Task 1: 60 minutes Task 2: 100 minutes	2 type-B
s'	02:30	[04:30, 05:00]	Task 1: 100 minutes Task 2: 60 minutes	2 type-B	s''	01:00	[01:40, 02:10]	Task 1: 60 minutes	1 type-M

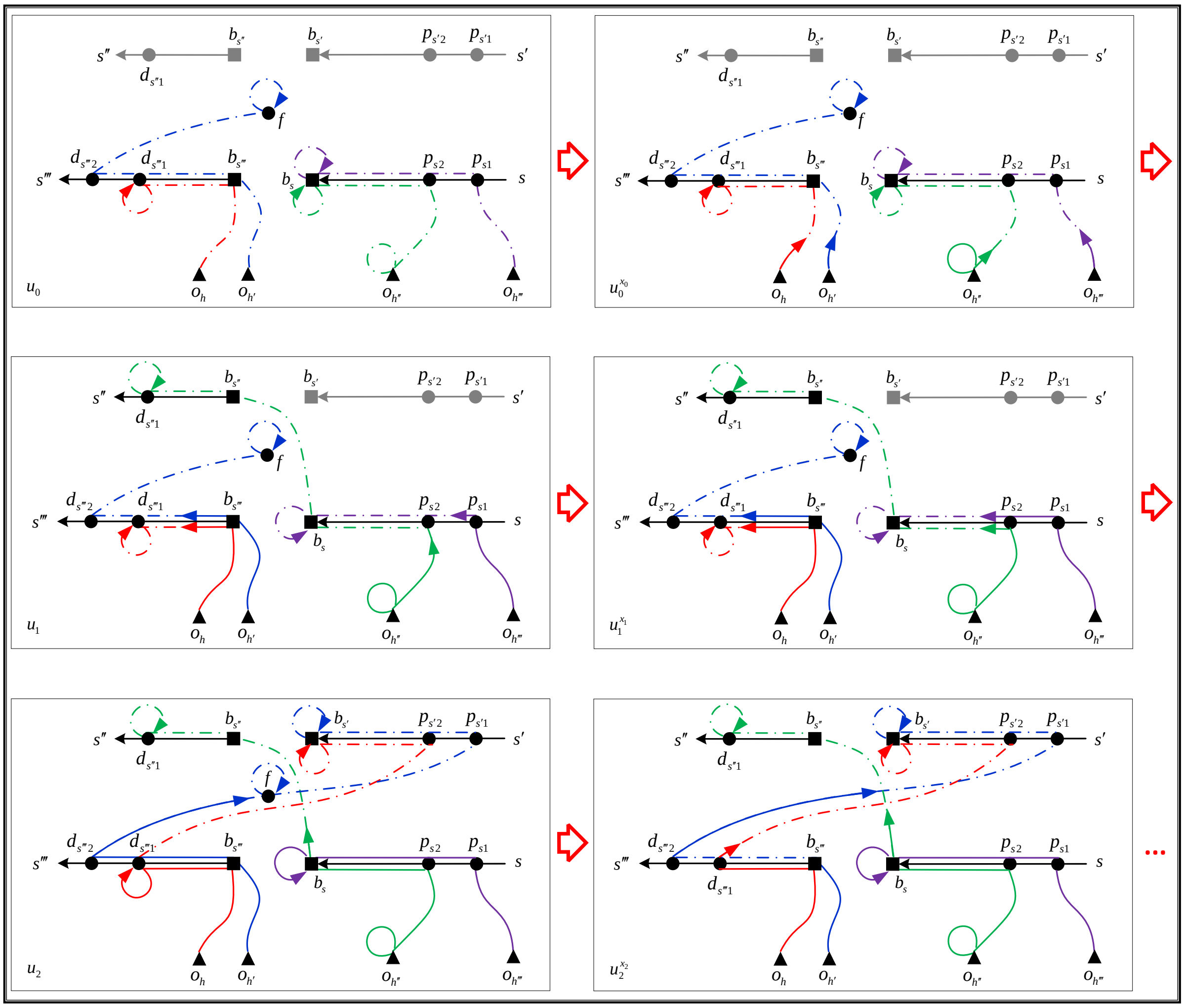


Figure A2. A RPMDP solution for the DTDSP.

Table A3. Sail times of tugboats in the RPMDP solution.

Tugboat	Link 1	Link 2	Link 3	Link 4	Link 5	Link 6	Total
h	o_h - $b_{s'''}$: 35 m	$b_{s'''}$ - $d_{s'''1}$: 60 m	$d_{s'''1}$ - $p_{s'2}$: 30 m	$p_{s'2}$ - $b_{s'}$: 60 m	$b_{s'}$ - o_h : 30 m	-	215 m
h'	$o_{h'}$ - $b_{s'''}$: 35 m	$b_{s'''}$ - $d_{s'''2}$: 100 m	$d_{s'''2}$ - f : 20 m	f - $p_{s'1}$: 40 m	$p_{s'1}$ - $b_{s'}$: 100 m	$b_{s'}$ - $o_{h'}$: 30 m	325 m
h''	$o_{h''}$ - p_{s2} : 40 m	p_{s2} - b_s : 60 m	b_s - $b_{s''}$: 30 m	$b_{s''}$ - $d_{s''1}$: 60 m	$d_{s''1}$ - $o_{h''}$: 30 m	-	220 m

h'''	$o_{h'''}$ - p_{s1}: 45 m	p_{s1} - b_s: 100 m	b_s - $o_{h'''}$: 55 m	-	-	-	200 m

Appendix E. An example of basic MDP solution

For a comparative analysis with the RPMDP, Figure A3 illustrates an example of a basic MDP solution using the same service request scenarios detailed in Table A2. The basic MDP operates on a one-step action basis at each decision state, which means it does not plan for a sequence of subsequent actions; typically, only one tugboat is assigned to a single task at each decision state. Task assignments are determined at the completion of the current task, so that more information (e.g., new requests, tugboat availability) is revealed for the task assignment decision, and the decision will be likely more cost-efficient. The detailed steps of the basic MDP are presented as follows.

(i) *Initial decision state* u_0. The initial state incurs at the beginning of service time horizon (see the top-left panel in Figure A3). The initial decision directs tugboats h and h' to serve unberthing ship s''', while tugboats h'' and h''' to serve berthing ship s. The post-decision state $u_0^{x_0}$ is shown in the top-right panel in Figure A3.

(ii) *Decision state* u_1. At time 01:35, tugboat h arrives at the designated waypoint and finishes the current task (see the mid-left panel in Figure A3). Thus, a new decision state (i.e., state u_1) arises. The new decision directs tugboat h to serve unberthing ship s'' according to the myopic optimization. The post-decision state $u_1^{x_1}$ is shown in the mid-right panel in Figure A3.

(iii) *State* u_2. At time 02:15, tugboat h' arrives at the designated waypoint and finishes the current task (see the bottom-left panel in Figure A3), which leads to a new decision state u_2. Correspondingly, the new decision guides tugboat h' to wait at the current location. The post-decision state $u_2^{x_2}$ is depicted in the bottom-right panel in Figure A3.

Subsequently, when one task is finished by one tugboat, a decision state emerges, and a decision is made to deploy the tugboat. Finally, all the tugboats return to the base at the end of service time horizon. It is observed that three ships (i.e., s, s'', and s''') are served with respecting the time windows at berths; however, berthing ship s' cannot be served by these four tugboats with satisfying the time window at the berth, because there are not enough type-B tugboats available to serve berthing ship s' due to time limits. To measure tugboat operation cost by the basic MDP, we relax the constraint on the end of time window for berthing ship s', and thus we assign tugboats h and h' to serve ship s' (tugboat h performs the task starting from $p_{s'1}$ and tugboat h' performs the task starting from $p_{s'2}$). The sail times of tugboats h, h', h'', and h''' are shown in Table A4, and the sail cost incurred by the solution is 2×(352+280) + 1.5×(130+200) = 1759.

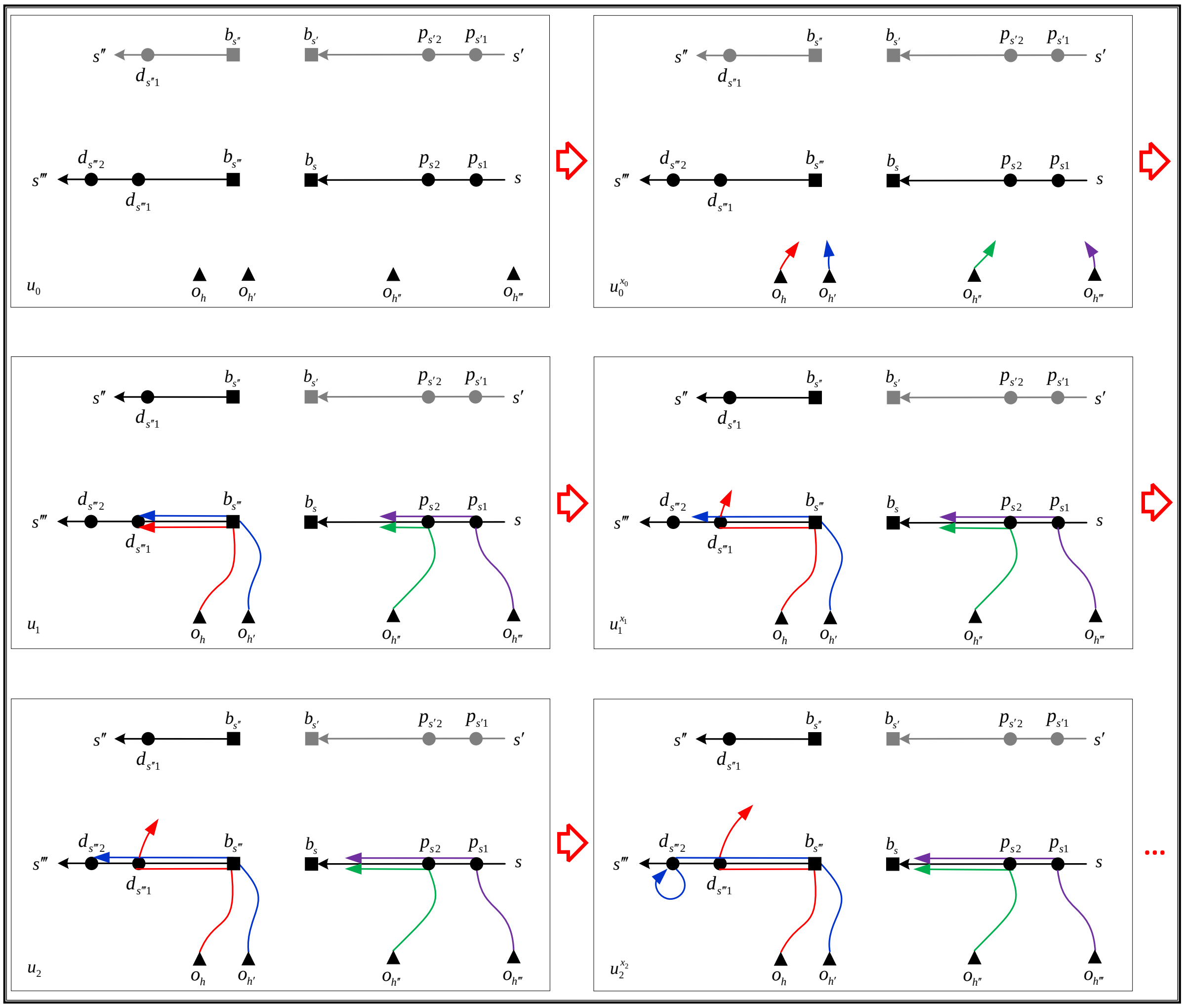


Figure A3. A solution by the basic MDP for the DTDSP.

Table A4. Sail times of tugboats in the solution by the basic MDP.

Tugboat	Link 1	Link 2	Link 3	Link 4	Link 5	Link 6	Link 7	Total
h	o_h - $b_{s'''}$: 35 m	$b_{s'''}$ - $d_{s'''1}$: 60 m	$d_{s'''1}$ - $b_{s''}$: 7 m	$b_{s''}$ - $d_{s''1}$: 60 m	$d_{s''1}$ - $p_{s'1}$: 60 m	$p_{s'1}$ - $b_{s'}$: 100 m	$b_{s'}$ - o_h : 30 m	352 m
h'	$o_{h'}$ - $b_{s'''}$: 35 m	$b_{s'''}$ - $d_{s'''2}$: 100 m	$d_{s'''2}$ - $p_{s'2}$: 55 m	$p_{s'2}$ - $b_{s'}$: 60 m	$b_{s'}$ - $o_{h'}$: 30 m	-	-	280 m

h''	$o_{h''}$ - p_{s2}: 40 m	p_{s2} - b_s: 60 m	b_s - $o_{h''}$: 30 m	-	-	-	-	130 m
h'''	$o_{h'''}$ - p_{s1}: 45 m	p_{s1} - b_s: 100 m	b_s - $o_{h'''}$: 55 m	-	-	-	-	200 m

References

Abou Kasm, O., Diabat, A., Bierlaire, M., 2021. Vessel scheduling with pilotage and tugging considerations. Transp. Res. Part E: Logist. Trans. Rev. 148, 102231.

Archetti, C., Feillet, D., Mor, A., Speranza, M.G., 2018. An iterated local search for the Traveling Salesman Problem with release dates and completion time minimization. Comput. Oper. Res. 98, 24–37.

Archetti, C., Feillet, D., Mor, A., Speranza, M.G., 2020. Dynamic traveling salesman problem with stochastic release dates. Eur. J. Oper. Res. 280, 832–844.

Bent, R.W., Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Oper. Res. 52, 977–987.

Bertsimas, D., Demir, R., 2002. An Approximate Dynamic Programming approach to multidimensional knapsack problems. Manage. Sci. 48, 550–565.

Branke, J., Middendorf, M., Noeth, G., Dessouky, M., 2005. Waiting strategies for dynamic vehicle routing. Transp. Sci. 39, 298–312.

Chen, S., Wang, F., Wei, X., Tan, Z., Wang, H., 2020. Analysis of tugboat activities using AIS data for the Tianjin Port. Transp. Res. Rec. 2674, 498–509.

Ghiani, G., Manni, E., Thomas, B.W., 2012. A comparison of anticipatory algorithms for the dynamic and stochastic traveling salesman problem. Transp. Sci. 46, 374–387.

Guo, L., Wang, J., Zheng, J., 2021. Berth allocation problem with uncertain vessel handling times considering weather conditions. Comput. Ind. Eng. 158, 107417.

Hao, L., Jin, J. G., Zhao, K., 2023. Joint scheduling of barges and tugboats for river–sea intermodal transport. Transp. Res. Part E: Logist. Trans. Rev. 173, 103097.

Ilati, G., Sheikholeslami, A., Hassannayebi, E., 2014. A simulation-based optimization approach for integrated Port resource allocation problem. PROMET - Traffic&Transp. 26, 243–255.

Jia, S., Cui, H., Chen, R., Meng, Q., 2022. Dynamic container drayage with uncertain request arrival times and service time windows. Trans. Res. Part B: Methodol. 166, 237-258.

Jia, S., Li, S., Lin, X., Chen, X., 2021. Scheduling tugboats in a seaport. Transp. Sci. 55, 1370–1391.

Kang, L., Meng, Q., Tan, K.C., 2020. Tugboat scheduling under ship arrival and tugging process time uncertainty. Transp. Res. Part E: Logist. Trans. Rev. 144, 102125.

Kim, S.-H., Nelson, B.L., 2001. A fully sequential procedure for indifference-zone selection in simulation. ACM Trans. Model. Comput. Simul. 11, 251–273.

Larsen, A., Madsen, O.B.G., Solomon, M.M., 2004. The A Priori dynamic traveling salesman problem with time windows. Transp. Sci. 38, 459–472.

Lee, C.-G., Epelman, M.A., White, C.C., III, Bozer, Y.A., 2006. A shortest path approach to the multiple-vehicle routing problem with split pick-ups. Trans. Res. Part B: Methodol. 40, 265–284.

Lee, C.-Y., Song, D.-P., 2017. Ocean container transport in global supply chains: Overview and research opportunities. Trans. Res. Part B: Methodol. 95, 442–474.

Liu, M., Luo, Z., Lim, A., 2015. A branch-and-cut algorithm for a realistic dial-a-ride problem. Trans. Res. Part B: Methodol. 81, 267–288.

Liu, Z., 2009. Hybrid evolutionary strategy optimization for port tugboat operation scheduling. Proc. IEEE Intell. Inf. Technol. Appl. 511–515.

Madsen, N.H., 2023. URL https://www.sea-intelligence.com/press-room/235-schedule-reliability-improves-to-64-4 (accessed 5.16.24).

Maritime & Port Authority of Singapore, 2023. URL https://www.mpa.gov.sg/port-marine-ops/marine-services/pilotage-and-towage/towage (accessed 5.16.24).

Mitrović-Minić, S., Laporte, G., 2004. Waiting strategies for the dynamic pickup and delivery problem with time windows. Trans. Res. Part B: Methodol. 38, 635–655.

Petris, M., Pellegrini, P., Pesenti, R., 2022. Models and algorithms for an integrated vessel scheduling and tug assignment problem within a canal harbor. Eur. J. Oper. Res. 300(3), 1120-1135.

Powell, W.B., 2022. Reinforcement learning and stochastic optimization: A unified framework for sequential decisions. John Wiley & Sons, Nashville, TN.

PSA Marine, 2023. URL https://www.psamarine.com/wp-content/uploads/2023/09/PSAM-General-

Operating-Conditions-Updated-13-Sep-2023.pdf (accessed 5.16.24).

Rodrigues, F., Agra, A., 2022. Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey. Eur. J. Oper. Res. 303, 501–524.

Soeffker, N., Ulmer, M.W., Mattfeld, D.C., 2022. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. Eur. J. Oper. Res. 298, 801–820.

Thomas, B.W., 2007. Waiting strategies for anticipating service requests from known customer locations. Transp. Sci. 41, 319–331.

Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Hennig, M., 2019. Offline-online approximate dynamic programming for dynamic vehicle routing with stochastic requests. Transp. Sci. 53, 185–202.

Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Thomas, B.W., 2020. On modeling stochastic dynamic vehicle routing problems. EURO J. Transp. Logist. 9, 100008.

Ulmer, M.W., Mattfeld, D.C., Köster, F., 2018. Budgeting time for dynamic vehicle routing with stochastic customer requests. Transp. Sci. 52, 20–37.

Umang, N., Bierlaire, M., Erera, A.L., 2017. Real-time management of berth allocation with stochastic arrival and handling times. J. Sched. 20, 67–83.

Wang, S., Kaku, I., Chen, G.Y., Zhu, M., 2012. Research on the modeling of Tugboat Assignment Problem in container terminal. Adv. Mat. Res. 433–440, 1957–1961.

Wang, S., Zheng, K., Zheng, J., Hu, W., 2010. An improved discrete particle swarm optimization for tugboat scheduling problem in container terminal. Proc. IEEE 3rd Int. Conf. Inf. Manage., Innov. Manage. and Ind. Eng. 593–596.

Wang, S., Zhu, M., Kaku, I., Chen, G., Wang, M., 2014. An improved discrete PSO for tugboat assignment problem under a hybrid scheduling rule in container terminal. Math. Probl. Eng. 2014, 1–10.

Wang, X., Liang, Y., Wei, X., Chew, E. P., 2023. An adaptive large neighborhood search algorithm for the tugboat scheduling problem. Comput. Ind. Eng. 177, 109039.

Wei, X., Jia, S., Meng, Q., Tan, K.C., 2020. Tugboat scheduling for container ports. Transp. Res. Part E: Logist. Trans. Rev. 142, 102071.

Wei, X., Meng, Q., Lim, A., Jia, S., 2021. Dynamic tugboat scheduling for container ports. Marit. Policy

Manage. 1–23.
Xiang, X., Liu, C., 2021. An expanded robust optimisation approach for the berth allocation problem considering uncertain operation time. Omega 103, 102444.
Xu, Q., Mao, J., Jin, Z., 2012. Simulated annealing-based ant colony algorithm for tugboat scheduling optimization. Math. Probl. Eng. 2012, 1–22.
Yang, Z. Y., Cao, X., Xu, R. Z., Hong, W. C., Sun, S. L., 2024. Applications of chaotic quantum adaptive satin bower bird optimizer algorithm in berth-tugboat-quay crane allocation optimization. Expert Syst. Appl. 237, 121471.
Yu, J., Tang, G., Song, X., Yu, X., Qi, Y., Li, D., Zhang, Y., 2018. Ship arrival prediction and its value on daily container terminal operation. Ocean Eng. 157, 73–86.
Zhang, A., Zheng, Z., Teo, C.-P., 2022. Schedule reliability in liner shipping timetable design: A convex programming approach. Trans. Res. Part B: Methodol. 155, 499–525.
Zhen, L., Wang, K., Wang, S., Qu, X., 2018. Tug scheduling for hinterland barge transport: A branch-and-price approach. Eur. J. Oper. Res. 265, 119–132.
Zhong, H., Zhang, Y., Gu, Y., 2022. A Bi-objective green tugboat scheduling problem with the tidal port time windows. Trans. Res. Part D: Trans. Environ. 110, 103409.