Big data-driven booking consolidation and scheduling of launching service in Singapore Port

Yun Hui Lin
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
liny@ihpc.a-star.edu.sg

Ping Chong Chua
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
chua_ping_chong@ihpc.a-star.edu.sg

Xiao Feng Yin
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
yinxf@ihpc.a-star.edu.sg

Zizhe Wang
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
wang_zizhe@ihpc.a-star.edu.sg

Ning Li
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
ning_li@ihpc.a-star.edu.sg

Zhe Xiao
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
xiaoz@ihpc.a-star.edu.sg

Xiuju Fu
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
fuxj@ihpc.a-star.edu.sg

Zheng Qin
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
qinz@ihpc.a-star.edu.sg

***Abstract*—Launching services provided by launch boat (LB) operators are indispensable for vessels in port areas. In the current business practice, the operator follows a "one-trip-per-booking" method, where each booking corresponds to a LB transporting passengers to their destination. Undoubtedly, this method does not efficiently utilize the LB's capacity. A more appealing approach is to consolidate or batch multiple service bookings into a single LB, enabling it to travel to multiple destinations within one trip. Using large-scale GPS data of LBs, we conduct data-driven analysis to gain insights into LB trajectory and traveling pattern. Based on them, we propose a real-time batching algorithm to consolidate a maximum of two bookings into a task with marginal service delay. We then address the scheduling of LBs to fulfill the consolidated tasks using rule-based real-time approaches. To validate our proposed framework, we conduct a case study in Singapore Port. The results show that after implementing the data-driven batching and scheduling algorithms, we achieve a reduction of more than 25% in the traveling distances of LBs, while maintaining a high level of service quality for passengers.**

***Index Terms*—launch boat operations, big-data analytic, trajectory prediction, demand batching, task scheduling**

This paper is supported by the "Maritime AI research programme" funded by the Singapore Maritime Institute (SMI), SMI-2022-MTP-06. Any opinions, findings, conclusions and/or recommendations expressed in this report are those of the researchers and do not reflect the views of SMI.

I. Introduction

The maritime industry plays a crucial role in the foundations of major economies and transportation networks. As of 2019, approximately 80% of total trade is conducted through maritime transport [1], and there is a continuous increase in demand for these services [2]. Intercontinental trade, bulk transportation of raw materials, and the import/export of affordable food and manufactured goods heavily depend on the shipping sector. However, ensuring the reliability and efficiency of services for vessels and cargo remains an ongoing challenge for stakeholders across relevant sectors. This challenge is particularly prominent in busy ports like Singapore Port, a pivotal maritime gateway to key Asian markets situated at the crossroads of East-West trade. Singapore Port ranks

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00060

among the world's busiest transshipment hubs, with 10,364 vessel arrivals reported by the Maritime and Port Authority of Singapore in October 2023 [3]. These vessels, upon arrival, require an extensive array of services.

A vital service for vessels is the launching service. Given the considerable size and draft of vessels, they typically anchor in designated anchorage areas before proceeding to berths at specified times for operations such as transshipment. Since these anchorages are often located miles away from the shore, crew members require transportation to and from terminals or piers for both personal and business activities. This necessity has led to the establishment of launching services, utilizing small-sized boats commonly known as launch boats (LBs), to facilitate the transfer of passengers between terminals/piers and the vessels at sea. With the substantial number of vessels arriving and staying in Singapore, there is a significant demand for these launching services.

In the current launching service model, vessels initiate the process by submitting service requests and creating advanced bookings. These bookings include detailed information such as the preferred service request time, the number of passengers, and specific pick-up and drop-off locations, typically corresponding to terminals or anchorages. Subsequently, the LB operator schedules LBs and executes the operations based on the "one-trip-per-booking" practice, in which each booking is matched with a single LB tasked with transporting the passenger(s) to the intended destination. Undoubtedly, this method does not make efficient use of the capacity of the LB. A more appealing method is to consolidate or batch multiple service bookings into a single LB, which travels to multiple destinations within one trip. Implementing this consolidation strategy holds the potential to significantly enhance operational efficiency and yield positive financial results for both the service provider and their customers. This transformation of an established industry or economy, facilitated by intelligent computing platforms, aligns with the concept of "Uberisation".

The "Uberisation" of launching service underscores the importance of real-time and demand-responsive operations, presenting considerable challenges. From a practical standpoint, there are various sources of uncertainty that complicate the scheduling of LBs. The uncertainty can arise from factors such as fluctuating demand, resource availability, the constant movement of vessel and LB locations, advanced bookings, and customer modifications. For instance, service demand and bookings exhibit spatial-temporal variability and uncertainty. Customers may not be present on time for the service and may even change their booking time and location at the very last minute since the current launching service does not impose substantial penalty fees for booking changes or delayed arrivals of customers. Furthermore, some bookings are made in advance before the vessels arrive at Singapore. As a result, the exact arrival times of vessels and the pick-up location of launching service are not known to the operators. Consequently, there is a continuous influx of updates to booking information and requests from customers.

From a research standpoint, there is a noticeable gap in the availability of effective algorithms for "uberising" the launching service within the practical constraints outlined above. Notably, there is an absence of research dedicated to LB operations. Given that LBs can be considered special harbor crafts operating in the sea areas of ports, this paper aligns with the scheduling and operational planning of other harbor crafts, such as tugboats and pilot boats. In tugboat operations, tugboats play a crucial role in assisting vessels by pushing or towing at specific tug points. The successful mooring or unmooring of vessels depends on each tug point operating with sufficient horsepower. The related scheduling problem is usually formulated as a mixed-integer programming (MIP) model and then solved by off-the-shelf solvers such as CPLEX [4], tailored branch-and-cut algorithms [5], or Lagrangian relaxation heuristics [6]. Vessel pilotage is mandatory in most seaports globally. When vessels enter or exit a seaport terminal, sea pilots navigate them through the correct and safe waterways. Pilots assume control of the ship in certain waterways and assist captains in maneuvering through various challenges such as currents, sandbanks, and tidal currents. This scenario gives rise to the pilotage planning problem, involving decisions on scheduling vessel traffic in a seaport, assigning work shifts to pilots, and scheduling pilots within each work shift for vessel navigation. Similar to the tugboat problem, pilotage planning problems are often studied as MIP models and addressed using algorithms such as branch-and-price [7], [8]. Note that the tugboat and pilotage problems are NP-hard and computationally challenging even for moderate-sized problems. While existing algorithms can solve realistic-sized instances within a reasonable computation time, they still fall short of the speed required for real-time implementation. Resolution of these instances may take up to hours, which is not compatible with the "Uberisation" concept. In this context, the consolidation of bookings and the scheduling of tasks need to be executed in real-time, especially considering the frequent updates of booking requests by customers. As a result, MIP-based approaches cannot satisfy such a need. Additionally, LB operations consider the consolidation of bookings, a facet not studied in the tugboat and pilotage problems.

Motivated by the above constraints and research limitations, our goal is to develop practically implementable booking consolidation and scheduling algorithms through combined predictive and prescriptive analytics. The contributions of this paper are as follows: (i) *Predictive Analytic for LB Trajectory:* Using large-scale GPS and AIS data of LB history trajectories, we conduct analysis to gain insights into the movement and operational patterns of LBs. By examining service times, origin-destination pairs, and uncertainties, we extract valuable information for accurately predicting the traveling trajectory and distance of LBs between any pair of locations. (ii) *Two-Stage Framework for Booking Consolidation and LB Scheduling:* In addressing real-time demands, we introduce a two-stage framework. In the first stage, we propose a batching algorithm capable of consolidating a maximum of two bookings into a task. This algorithm exploits the similarity of bookings, measured in terms of the additional delay caused by batching two bookings into a single task. It operates extremely fast to meet real-time deployment needs. In the second stage, we tackle the scheduling of LBs to fulfill these consolidated tasks, presenting various rule-based real-time scheduling approaches. (iii) *Case Study Using Singapore Port Data:* To validate our proposed framework, we conduct a case study using data from a launching service provider in Singapore Port. The results demonstrate the efficiency of the batching algorithm, processing daily bookings in less than 0.4 seconds on a typical laptop. For 3,694 bookings in November 2021, the algorithm successfully consolidated 2,192 bookings (approximately 60% of total bookings) into 1,096 tasks, with an acceptable delay observed after each batching operation. Furthermore, after implementing the scheduling optimization algorithms, we achieve a reduction of more than 25% in the travel distances of LBs compared to current operational practices.

The remainder of this paper is organized as follows. Section II provides the problem description. Section III presents a summary of launch boat trajectory estimation. Subsequently, Section IV details the booking batching algorithm and the scheduling algorithm, followed by a case study in Section V. Finally, we conclude this paper in Section VI.

II. Problem description

This paper considers a LB operator in Singapore owning a fleet of LBs. At a given planing timing, there are a set of launch service bookings in the system, denoted by set $\mathcal{I} := \{1, 2, ..., I\}$. The details of each booking consists of the name of the mother vessel, the number of passengers, the pick-up and drop-off locations, the service request time (srt) that indicates the expected boarding of passengers on the LB, and the required standby time that the passengers want the LB to stand by the vessel and wait for their returning trips.

The LB operator faces the task of scheduling LBs to fulfill bookings, with the current operation relying on the "one-trip-per-booking" practice. In this approach, each booking is individually matched with a single LB, responsible for transporting passengers to their designated destination. However, this method does not utilize the LB's capacity efficiently. The operator is keen on exploring an alternative strategy: consolidating or batching multiple service bookings into a single launch boat that can travel to multiple destinations within a single trip. While this consolidation has the potential to significantly enhance operational efficiency, it is imperative to ensure that the service quality for passengers remains close to that of the "one-trip-per-booking" practice. To this end, this paper conducts both predictive and prescriptive analytics in the subsequent sections to derive an effective booking consolidation and scheduling for the launching service.

III. Prediction: Launch boat trajectory estimation based on Big GPS data

In the predictive analytics, we focus on predicting the trajectories and traveling distances of LBs during the booking fulfillment process, which are crucial inputs to subsequent tasks such as booking consolidation and LB scheduling. Essentially, given any two positions in the port and the nearby sea area, we predict the traveling distance (and time) of LBs. Indeed, there are historical trajectories stored in the dataset that can possibly be used as references; however, they are not directly applicable to the "unseen" positions.

To this end, we utilize the Waterway Pattern-Mining Framework developed in [9]. The framework takes large-scale historical GPS data as inputs and generates the waterway pattern of LBs, reflecting the LBs' navigating routines shaped by multiple aspects, such as maritime traffic operations, planning, LB maneuverability, and water's hydrographical features representing practical sea routes through voluminous real-world navigational instances. Fig 1(a) shows the generated waterway pattern of LBs in the Singapore port using massive GPS data of LBs from the years 2021 and 2022. The area covered in blue represents the waterway of LBs. Then, given two positions in the covered area, the framework returns a predicted path and an estimated distance following the "most likely" pattern. It is deployed as a callable API to support real-time usage.

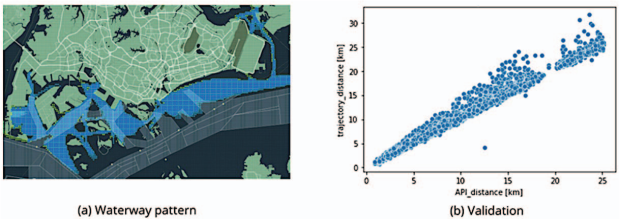

Fig. 1. Launch boat trajectory and travelling distance analysis with GPS data.

It is important to validate the results of the Waterway Pattern-Mining Framework. In our GPS dataset, we have records of the actual trajectory distances for some position pairs. We conduct validation by comparing the actual "trajectory distance" and the distances generated by the API of the Waterway Pattern-Mining Framework ("API distance"). Fig 1(b) shows the comparison. After excluding the 1.2% of outlier data, the API distance is, on average, 5.56% shorter than the trajectory distance. This implies that the accuracy of the predicted distance is, on average, 94.44%, which is high enough for practical usage. We will thus use this approach in the subsequent prescriptive analytics (interested readers are referred to a similar approach [10]).

IV. Prescription: Booking Consolidation and Launch Boat Scheduling

A. Batching algorithm

In the existing LB operation, each LB is allocated to a single booking to transport passengers to a specific destination. However, this conventional practice underutilizes the LB capacity and results in inefficient resource utilization. In this paper, we study the approach for booking consolidation and create a fast and efficient algorithm. Based on the input and domain knowledge from the industry, we place our emphasis on consolidating a maximum of **two** bookings into a task. In this context, it involves considering two key factors: i) the **capacity** of the LB should not be violated. For a typical LB, the capacity is 12 passengers; and ii) the **similarity** of the two bookings should be high.

Here, the similarity S is an index that measures how "similar" two bookings are in terms of the additional service delay caused by batching them into a single task under feasible LB capacity. Intuitively, when two bookings are batched together, it may result in additional delays for passengers since the LB must accommodate extra routes for pick-ups or drop-offs at different locations, in contrast to the traditional "one-trip-per-booking" approach. If the additional delay introduced by batching remains relatively small, then the similarity index is considered to be high. However, if the delay exceeds a predefined threshold τ (e.g., 30 or 45 minutes), the similarity index is set at zero, indicating that batching is not a viable option due to the excessive delay it would cause.

Algorithm 1 presents how we compute the similarity matrix $\mathbf{S}$ given a list of bookings in set $\mathcal{I}$. For an illustration, suppose

Algorithm 1 Computing similarity under delay threshold τ.

1: **Input:** A list of bookings $\mathcal{I}$, delay threshold τ.
2: Initialize $\mathbf{S} = 0$
3: **for** $i \in \mathcal{I}$ and $k \in \mathcal{I}$ **do**
4: Compute the number of passengers on board OB_{ik}
5: **if** $\text{OB}_{ik} \leq 12$ **then**
6: Estimate T_i^o, T_k^o, T_{ik}^b, and T_{ki}^b
7: Compute $\Delta_{ik} = T_{ik}^b - T_i^o$ and $\Delta_{ki} = T_{ki}^b - T_k^o$
8: **if** $\Delta_{ik} < \tau$ and $\Delta_{ki} < \tau$ **then**
9: $S_{ik} = S_{ki} = 2 - \frac{\Delta_{ik}}{\tau} - \frac{\Delta_{ki}}{\tau}$
10: **Return:** $\mathbf{S}$.

that we want to compute the similarity of Booking i and Booking k, denoted as S_{ik}. Let the estimated (ideal) service end times of Booking i and Booking k without batching be T_i^o and T_k^o, computed by the associated service request times plus the operation times in the fulfillment process. Let the service end times of Booking i and Booking k when they are batched together be T_{ik}^b and T_{ki}^b. Then, the additional delay of Booking i caused by batching it with Booking k is $\Delta_{ik} = |T_{ik}^b - T_i^o|$. Similarly, the additional delay of Booking k caused by batching it with Booking i is $\Delta_{ki} = |T_{ki}^b - T_k^o|$.

If one or more of the following conditions appear: (i) $\Delta_{ik} \geq \tau$; (ii) $|\Delta_{ki}| \geq \tau$; or (iii) LB capacity is not met, then the batching does not satisfy the delay threshold or the capacity of LB. In this case, S_{ik} is set to 0. Otherwise, $S_{ik} = 2 - \Delta_{ik}/\tau - \Delta_{ki}/\tau$, which is a positive number between 0 and 2. Note that, as indicated in Line 8, $\mathbf{S}$ is symmetric.

Given $\mathbf{S}$, we then proceed to determine the batching decision. Note that only those booking pairs with positive similarity indices are considered feasible batches. Algorithm 2 presents a greedy batching algorithm where we iteratively put two bookings (m, n) with the maximum similarity index into a batch (Lines 4-5) and then update the similarity matrix by setting the similarity indices related to bookings (m, n) to 0 so that bookings (m, n) would not be considered in the subsequent iteration. This process is repeated until all elements in the remaining similarity matrix are 0, i.e., there does not exist any feasible batch.

Algorithm 2 Greedy heuristic for similarity-based batching.

1: **Input:** List of bookings $\mathcal{I}$ and Similarity matrix $\mathbf{S}$.
2: Initialize the set of batches $\mathbf{B} = \emptyset$.
3: **while** $\max_{(i,k)} S_{ik} > 0$: **do**
4: $(m, n) = \arg\max_{(i,k)} S_{ik}$
5: $\mathbf{B} = \mathbf{B} \cup \{(m, n)\}$
6: $B_{mi} = B_{im} = B_{in} = B_{ni} = 0, \forall i \in \mathcal{I}$
7: **end while**
8: **Return:** $\mathbf{B}$.

Algorithms 1 and 2 run fast enough to meet real-time deployment needs. In our testing, it takes less than 0.4 seconds to process 100 bookings on a typical laptop, which is approximately the average number of daily bookings for a "bigger-than-average" launch boat service provider in Singapore Port. Note that when estimating the delays, we compare the service fulfillment time of a booking without batching and the service fulfillment time of a booking when batches with other bookings. Clearly, the traveling distance and time in each leg of the fulfillment process need to be estimated, and this can be done using the API distances that have been developed in the predictive analytic of launch boat trajectory prediction in Section III.

B. Scheduling algorithm

The batching algorithm transforms the original bookings into a set of service tasks, involving one or two bookings and an estimated service fulfillment time. Figure 2 summarizes the process of how we schedule the LB for task fulfillment.

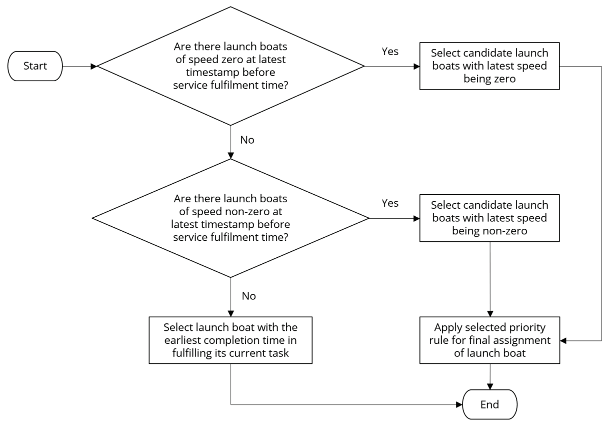

Fig. 2. Overview of selection of launch boat.

Since the exact location of LB is uncertain prior to initiating each task, it is advisable to ensure LBs are ready for service before the designated service fulfillment time. This is represented by the status of an LB exhibiting a speed value of zero at the most recent timestamp preceding the service fulfillment time for a chosen task. In cases where there is no LB with the latest speed recorded as zero, LBs with the latest non-zero speed are considered. Subsequently, a set of priority rules is employed to make the final selection of an LB from the candidate list.

Priority rules, serving as criteria for LB selection, are used. Their major advantage is the ease of implementation, coupled with low time complexities, rendering them suitable for real-time ad-hoc event scheduling [11]. The proposed priority rules leverage predicted travel time and distance along the travel path, outlined in Fig 3. These rules are articulated as follows:

- Smallest total travel time (**STT**): Select the LB with the smallest total predicted travel time from its current location to the final destination of the last booking;
- Earliest arrival time (**EAT**): Select the LB that arrives at the first pickup location of the first booking the earliest;
- Earliest arrival time with smallest travel time from current location to the first pickup location (**EATSTF**): Select the LB that arrives at the first pickup location of the first booking the earliest with the smallest total predicted travel time from its current location to the first pickup location;
- Earliest arrival time with smallest total travel time (**EATSTT**): Select the LB that arrives at the first pickup location of the first booking the earliest with the smallest total predicted travel time from its current location to the final destination of the last booking; and
- Earliest arrival time with the shortest distance from the current location to first pickup location (**EATSDF**): Select the LB that arrives at the first pickup location of the first booking the earliest with the shortest calculated distance from its current location to the first pickup location.

Fig. 3. Travel path of launch boat to fulfil task.

It is important to note that when evaluating LB options, the distance traveled from the initial pickup location to the final destination remains consistent regardless of the chosen LB. Therefore, a sufficient criterion is to compare the distance covered from the current location of each LB to the initial pickup location. In instances where no LB is available at the service fulfillment time (indicating that all LBs are currently engaged in tasks), the selection will prioritize the LB with the earliest completion time in fulfilling its ongoing task for the subsequent task assignment.

V. Case Study

To test the effectiveness of the algorithms, we conduct a case study based on a LB booking company in Singapore Port. We use its booking data from November 1st to November 30th, 2021, comprising a total of 3,694 initial bookings. There are 30 LBs ready to be utilized. In the current practice , the LBs travel a total distance of 69,318 km to fulfill these bookings.

A. Batching result

We first employ the batching algorithm on 3,694 initial bookings. Table I presents the summary of results under different delay thresholds τ. Here, $\#N^2$ is the number of batch tasks, i.e., each is formed by consolidating two bookings together. In contrast, $\#N^1$ is the number of non-batch tasks, i.e., each is a single booking that cannot be batched with any other bookings and has to be fulfilled separately.

We observe that applying a 30-minute delay threshold, there are 1,096 tasks with 2 bookings and 1,502 tasks with 1 booking. This outcome indicates that 59.33% of the bookings can be successfully batched. Intuitively, if we increase the delay threshold τ, then more bookings can be batched together. In particular, when $\tau = 90$, the batching rate is near 75%.

TABLE I
Summary of batching results of 3694 bookings in November 2021 under different delay threshold τ.

τ	$\#N^2$	$\#N^1$	Batching Rate [%]
30	1096	1502	59.33
45	1227	1240	66.43
60	1301	1092	70.44
90	1380	934	74.72

The obtained results highlight that the prevailing "one-trip-per-booking" approach significantly underutilizes the potential for enhancing operational efficiency through booking consolidation. Even by introducing a 30-minute threshold, a notable batching rate can be achieved. It is noteworthy that, according to feedback and domain expertise from our industrial partner in this research, the customers of the launching service exhibit a relatively low sensitivity to delays compared to users of land taxi and ride-sharing services. A 30-minute delay appears to be both reasonable and acceptable from the customers' perspective. This insight further underscores the practicality and viability of our proposed approach, which not only improves efficiency but also aligns with the acceptable service standards from the customer's point of view.

B. Scheduling result

Utilizing the batching results under the 30-minute threshold, we simulate the fulfillment of associated tasks employing different priority rules. In Table II, a summary of the total distance traveled is presented. The total distance covered by LBs is reduced by approximately 25% under all rules except EAT. EAT, not factoring in the shortest distance traveled but prioritizing the earliest arrival at the first pickup point, results in a 7.27% higher total distance traveled compared to the current practice. These findings highlight that the combination of a well-selected batching algorithm with an effective scheduling algorithm can significantly decrease the overall distance traveled by LBs.

TABLE II
SUMMARY OF TOTAL DISTANCE TRAVELLED.

Priority Rules	Current Practice [km]	With Consolidation [km]	Difference [%]
STT	69,318.09	51,977.43	-25.02
EAT		74,358.28	+7.27
EATSTF		51,824.76	-25.24
EATSTT		51,648.08	-25.09
EATSDF		51,865.17	-25.18

Note that the decrease in distance traveled primarily benefits the operator. In practice, it is also important to avoid a substantial reduction in service levels for passengers. In this context, we evaluate the service level by assessing the tardiness of each booking. Tardiness is computed as the difference between the service request time for each booking and the actual service fulfillment time. As such, the start time of each booking needs to be taken note of during the simulation run.

As indicated in Table III, priority rules focusing on the earliest arrival time (specifically prioritizing reaching the first pickup point promptly) exhibit superior performance in terms of tardiness measures. More than 3,440 launch boat bookings boast tardiness of less than five minutes, constituting over 93% of the total bookings. In adherence to the 30-minute delay threshold for booking consolidation, priority rules linked to earliest arrival time—namely, EAT, EATSTF, EATSTT, and EATSDF—manage to keep maximum tardiness just slightly above 30 minutes.

TABLE III
NUMBER OF BOOKINGS WITH TARDINESS BELOW AND ABOVE FIVE MINUTES.

Priority Rules	Number of bookings with tardiness $\in$ [in minutes]		Maximum value of tardiness [in minutes]
	$[0, 5]$	$(5, \infty)$	
STT	3408	286	40.9
EAT	3447	247	30.3
EATSTF	3446	248	30.3
EATSTT	3447	247	30.3
EATSDF	3564	130	30.3

In summary, the above findings demonstrate that our approach to booking consolidation and scheduling significantly reduces the travel distance of LBs during task fulfillment while maintaining a high level of service quality for passengers.

VI. CONCLUSION

Recognizing the limitations of the current launching services operation in Singapore Port, we explored an alternative approach to booking consolidation and scheduling. Our goal was to enable the fulfillment of a maximum of two bookings in a single trip. To achieve this, we introduced a real-time batching algorithm designed to consolidate bookings without introducing excessive additional delays. Subsequently, we addressed LB scheduling for these consolidated tasks using various rule-based real-time approaches. Through a case study conducted in Singapore Port, the results illustrated that the proposed booking consolidation and scheduling significantly reduces the travel distance of LBs during task fulfillment while upholding a high level of service quality for passengers.

This paper has a few acknowledged limitations. Firstly, we focused exclusively on the consolidation of two bookings into a task. Investigating the consolidation of more bookings could provide valuable insights, but service levels should not be significantly compromised. Secondly, the batching and scheduling processes are carried out sequentially. An integrated approach could further enhance the performance and effectiveness of the "Uberisation" concept.

ACKNOWLEDGMENT

The Institute of High Performance Computing research team would like to thank Mr Chua and James of ShipsFocus for their invaluable advice and extraordinary support. Their expertise is essential in conducting the project effectively. We would also like to thank Justin from ShipsFocus in accompanying us for the LB ride to the anchorage, and his insights into the LB operations and the digital platform.

REFERENCES

[1] S. N. Sirimanne, J. Hoffman, W. Juan, R. Asariotis, M. Assaf, G. Ayala, H. Benamara, D. Chantrel, J. Hoffmann, A. Premti, *et al.*, "Review of maritime transport 2019," in *United Nations conference on trade and development, Geneva, Switzerland*, 2019.

[2] H. N. Psaraftis, "The future of maritime transport," *International Encyclopedia of Transportation; Elsevier: Amsterdam, The Netherlands*, pp. 535–539, 2021.

[3] Maritime and P. A. of Singapore, "Maritime performance," 2023.

[4] L. Kang, Q. Meng, and K. C. Tan, "Tugboat scheduling under ship arrival and tugging process time uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 144, p. 102125, 2020.

[5] X. Wei, S. Jia, Q. Meng, and K. C. Tan, "Tugboat scheduling for container ports," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102071, 2020.

[6] S. Jia, S. Li, X. Lin, and X. Chen, "Scheduling tugboats in a seaport," *Transportation Science*, vol. 55, no. 6, pp. 1370–1391, 2021.

[7] L. Wu, S. Jia, and S. Wang, "Pilotage planning in seaports," *European Journal of Operational Research*, vol. 287, no. 1, pp. 90–105, 2020.

[8] L. Xiao, P. Sun, L. Wang, and M. He, "Multi-class pilot dispatching problem with stochastic pilotage duration," *International Journal of Production Research*, pp. 1–21, 2023.

[9] Z. Xiao, X. Fu, L. Zhao, L. Zhang, T. K. Teo, N. Li, W. Zhang, and Z. Qin, "Next-generation vessel traffic services systems—from "passive" to "proactive"," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 1, pp. 363–377, 2022.

[10] A. Zamuda and J. D. H. Sosa, "Success history applied to expert system for underwater glider path planning using differential evolution," *Expert Systems with Applications*, vol. 119, pp. 155–170, 2019.

[11] J. C. Tay and N. B. Ho, "Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems," *Computers & Industrial Engineering*, vol. 54, no. 3, pp. 453–473, 2008.