



HAL
open science

Stochastic quay partitioning problem

Jakub Wawrzyniak, Maciej Drozdowski, Éric Sanlaville

► **To cite this version:**

Jakub Wawrzyniak, Maciej Drozdowski, Éric Sanlaville. Stochastic quay partitioning problem. European Journal of Operational Research, 2026, 328 (2), pp.415-429. <10.1016/j.ejor.2025.07.043>. <hal-05292753>

HAL Id: hal-05292753

<https://normandie-univ.hal.science/hal-05292753v1>

Submitted on 30 Mar 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License



Discrete Optimization

Stochastic quay partitioning problem

 Jakub Wawrzyniak^a, Maciej Drozdowski^a,* , Éric Sanlaville^b
^a Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland

^b Université Le Havre Normandie, Univ Rouen Normandie, INSA Rouen Normandie, Normandie Univ, LITIS, UR 4108, F-76600 Le Havre, France


ARTICLE INFO

Keywords:

Scheduling
 Maritime container terminal design
 Ship traffic modeling
 Berth allocation problem
 Stochastic optimization

ABSTRACT

In this paper we consider the problem of dividing a quay of a container terminal into berth segments so that the quality of service for future ship arrivals is as good as possible. Since future arrivals are unknown, the alternative solutions are evaluated on various arrival scenarios generated for certain arrival intensity from a stochastic model referred to as a ship traffic model (STM). This problem will be referred to as a stochastic quay partitioning problem (SQPP). SQPP is defined by an STM, arrival intensity, quay length and a set of admissible berth lengths. Evaluation of an SQPP solution on one scenario is a problem of scheduling the arriving vessels on the berths, which is a classic berth allocation problem (BAP). In SQPP the sizes of BAP instances that must be solved by far exceed capabilities of the methods presented in the existing literature. Therefore, a novel approach to solving BAP is applied. Tailored portfolios of algorithms capable of solving very large BAP instances under limited runtime are used. Features of SQPP solutions are studied experimentally: patterns in selected berth lengths, dispersion of solutions quality and solutions similarity. We demonstrate, that partitioning a quay into equal-length berths is not the best approach. The largest vessel traffic is dominating in defining best quay partitions, but dedicating quays for shorter vessels give lower dispersion of solution quality. A set of algorithms to partition a quay is proposed and evaluated: methods based on integer linear programming (ILP) to match vessel classes arrival intensities with berth availability, hill climber, tabu search and a greedy approach. Only under high arrival intensity can these methods show their prowess. ILP methods have an advantage of low solution evaluation cost. Tabu is most flexible, but at high evaluation costs. To the best of our knowledge, SQPP is posed and solved for the first time in the operations research.

1. Introduction

Maritime container logistics is an essential element of the global economy. A crucial place where transport changes mode are quays and berths of maritime container terminals. Although the stretch of a quay is a continuous edge between land and water and ships can be positioned anywhere along this stretch, a concept of discrete berth is widely used. The existence of berths simplifies ship and crane positioning, designing and controlling stowage in the storage yard, planing and running the terminal internal transportation network. Thus, berths are practical for managerial reasons. However, dividing a quay into berths has been studied relatively little. In this paper we consider partitioning a quay into berths so that the overall vessel waiting time is minimized while the arriving traffic is defined stochastically. This problem will be referred to as the stochastic quay partitioning problem (SQPP). SQPP can be considered in long-term strategic port planning, when a terminal is designed or an existing one is expanded. Since ports and container terminals are designed for many years to come, the consequences of

bad quay partitioning can be costly. SQPP is practically important in ports with long quays. For example, Yangshan deep water port of Shanghai has a 5.6 km-long quay, after the recent extension (Shanghai International Port Group, 2025; Tang, 2024). Pasir Panjang terminals in Singapore have over 14 km quays, while sole PPT 2 quay is 3 km-long and has 9 berths. The Tuas Mega Port is expected to have 26 km quay (Maritime and Port Authority of Singapore, 2024). Hence, quay partitioning is a relevant problem. The goal of this paper is to evaluate the potential for improving port performance by quay partitioning and to propose quay partitioning algorithms.

Let us outline the SQPP setting (cf. Fig. 1) and the reasons behind it. A solution of an SQPP defines a *quay partition* which is a vector of berth lengths. The partition should be evaluated on the future ship traffic. Since it is unknown, SQPP solutions are evaluated on test scenarios simulating future ship stream. The scenarios are generated according to a ship traffic model (STM) determining, e.g., ship length distributions, service time distributions, periodicity of ship arrivals,

* Corresponding author.

E-mail addresses: Jakub.Wawrzyniak@cs.put.poznan.pl (J. Wawrzyniak), Maciej.Drozdowski@cs.put.poznan.pl (M. Drozdowski), Eric.Sanlaville@univ-lehavre.fr (É. Sanlaville).

<https://doi.org/10.1016/j.ejor.2025.07.043>

Received 9 September 2024; Accepted 21 July 2025

Available online 5 August 2025

0377-2217/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

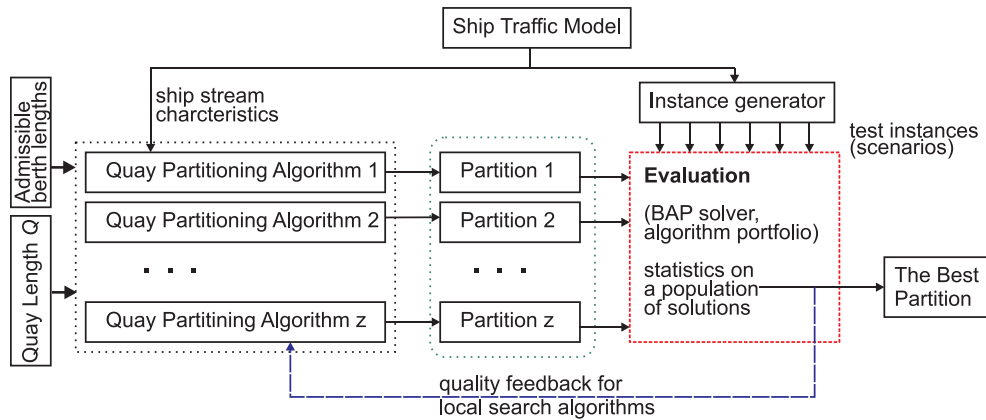


Fig. 1. SQPP solution workflow.

and their correlations. The STM used in this paper was built on the basis of real data (Wawrzyniak et al., 2022) and will be described in more detail in Section 3.2. Evaluating a partition on some test scenario consists in scheduling ships on berths and assessing quality of service, e.g., ship waiting time. This step is equivalent to solving a berth allocation problem (BAP) instance. For each partition a population of scenarios and their evaluation scores are collected. The partition with the best central tendency score (e.g. mean) is considered the best SQPP solution. A measure of the score dispersion (e.g. standard deviation of the waiting time) may be an auxiliary criterion.

An SQPP is essentially different than the logistic and operational research problems studied in the past. We deem that to evaluate an SQPP solution it is inevitable to: (1) refer to solving BAP, (2) consider very large instances, (3) use specialized methods. Firstly, solving BAP is necessary because ships are discrete objects that have to be scheduled according to the existing arrival patterns. Terminal capacity studies often refer to TEU volumes treated as continuous medium. If we treat work at quay as divisible volume of TEUs, then this is similar to preemptive scheduling with berths as processors and vessels as tasks. However, container mother ships are not moved between berths as tasks between processors in preemptive schedules. Hence, nonpreemptive schedules are required. It is known that nonpreemptive schedules can be by $\approx 33\%$ longer than preemptive ones (Coffman & Garey, 1993). When vessels are approaching and departing from the quay, the quay length may become *fragmented*, i.e. divided into many segments too short to accommodate an arriving ship. A simplified perception of a continuously divisible TEU volume medium, allows for exploiting such fragmented quay space, which is obviously infeasible. The distinction between vessels as discrete, or continuously divisible objects spanning along a quay is equivalent to the distinction between contiguous and non-contiguous parallel task scheduling (Bładek et al., 2015). It is known that the contiguous schedules can be by 25%–100% longer than the non-contiguous ones (Bładek et al., 2015). Consequently, simplified estimates ignoring the discrete spatial and temporal nature of the vessels can be very inaccurate in assessing a quay partition. Secondly, very large BAP instances need to be solved due to very long planning horizons. For example, in 2016 container terminals in Le Havre, Shanghai, Singapore, served 2271, 11606, 18494 vessel arrivals, respectively (Wawrzyniak et al., 2021, 2022). Thus, the considered BAP instances are unavoidably large while the instances routinely solved in the existing literature are notably smaller, see Table 1. This may result in unacceptable computational complexity. A further difficulty is that applying just one BAP algorithm exposes the optimization process to the deficiencies (such as worst-case behaviors) of that single algorithm. Thus, there is a need for specialized methods to solve large BAP instances reliably while obeying time limits dictated by admissible computational costs. These difficulties can be mitigated by using algorithm portfolios. We will use the results of Wawrzyniak

et al. (2020) where selecting algorithm portfolios solving very large BAPs under runtime limit was considered.

We build on the earlier research in solving very large BAPs (Wawrzyniak et al., 2020), modeling vessel traffic (Wawrzyniak et al., 2022) and exactly solving the deterministic version of QPP (Wawrzyniak et al., 2024). However, this paper not only gathers previous works to present a complete solving methodology to a new and very complex problem, but also sets new goals and provides new insights into the terminal design in the light of realistic ship traffic models. The main contributions of this paper can be summarized as follows:

- Quay partitioning problem is formulated in the stochastic way for the first time.
- A methodology of solving SQPP is proposed.
- SQPP algorithms of various kinds: greedy, MIPs matching traffic feature histograms with quay space availability, two metaheuristics, are given and evaluated.
- The impact of traffic intensity on the features of SQPP solutions is studied on the examples of Le Havre and Shanghai STMs.
- Limitations of partitioning the quay into equal length berths is analyzed. Further organization of this paper is as follows. The related literature is presented in the next section. SQPP problem is defined formally in Section 3. Algorithms solving SQPP are introduced in Section 4. Features of SQPP solutions are studied in Section 5. Performance of SQPP algorithms is evaluated in Section 6. The last section concludes the paper. The notations are summarized Table 2.

2. Related work

Quay partitioning is a strategic seaside design decision which to the best of our knowledge was hardly studied before. This may be attributed to (1) defining berths on a quay does not involve any immediate cost, (2) there is a self-imposing solution of equal length berths (equipartitioning). Consequently, the existing literature is related to quay partitioning only in a very general sense of port design. Container terminal design has links in many specific engineering and operational research problems, see e.g., surveys (Gharehgozli et al., 2020; Kastner et al., 2020; Thoresen, 2003) and a short summary in Wawrzyniak et al. (2024). This paper builds on the research in port simulation, traffic modeling, berth allocation problem, complexity of deterministic quay partitioning problem, and algorithm selection problem. It therefore merges three lines of research in these fields (Wawrzyniak et al., 2024, 2020, 2022).

Port simulation for assessing throughput and risk was surveyed in Olba et al. (2018). Most of the referred studies focus on nautical aspects, whereas terminal operations and ship arrival processes are represented only with some limited resolution. The terminal operations are represented as processing times derived from assumed distributions, e.g., Erlang (Gokkus & Yildirim, 2015; Groenvelde, 2006),

Table 1
Selected BAP literature.

Problem type	No. of vessels	No. of berths	Year	Ref.
BAP	50	10	2003	Hansen and Oğuz (2003)
BAP	35	10	2005	Cordeau et al. (2005)
BAP	200	20	2008	Hansen et al. (2008)
BAP	200	10	2010	Cheong et al. (2010)
BAP	60	13	2011	Buhrkal et al. (2011)
BAP+QCSP	200	4	2017	Li et al. (2017)
terminal	110	16	2017	Schepler et al. (2017)
BAP	100	8	2018	Dulebenets et al. (2018)
BAP	40	2	2020	El Hammouti et al. (2020)
BAP	60	16	2021	Xiang and Liu (2021)
BAP+CS	60	43	2021	Liu et al. (2021)
BAP	60	13	2024	Korekane et al. (2024)
BAP+QCSP	40	24	2025	Wang et al. (2025)
HCBAP	45	56	2025	Lyu et al. (2025)
BAP	60	4	2025	Ma and Pu (2025)
BAP+SSE	50	10	2025	Yue et al. (2025)

QCSP — quay crane scheduling problem, terminal — whole terminal with BAP as a subproblem, CS — vessel sequencing in a navigation channel, HCBAP — Horizontal Collaborative Berth Allocation Problem, SSE — Shore Side Electricity.

normal (Olba et al., 2017), or service time model based on cargo transfer speed (Ö.Üğurlu et al., 2014; Scott et al., 2016). Ship arrival times are drawn from negative exponential (Groenvelde, 2006), uniform (Ö.Üğurlu et al., 2014) distributions or using historical data (Scott et al., 2016). In this paper we represent ship arrival and service times by using distributions derived from historical data (Wawrzyniak et al., 2022). A more diverse set, both with respect to distribution types and their parameters, is used (see Section 3.2). In particular, the majority of ships arrive periodically with some dispersion of the return time, which to our best knowledge was not attempted before in the simulation studies. We take into account the dependence of the return and service times on the vessel size class which was neglected in the earlier studies. Furthermore, distinctive features of representative ports were analyzed in Wawrzyniak et al. (2022). This allows to choose an STM of the most similar port in the initial planning period when the port is not yet operational.

In order to determine the number of berths for minimum ship waiting and idle berth costs, queuing theory was used. In Gokkus and Yildirim (2015) ship inter-arrival time was modeled by fitting Poisson distribution to historical data. Similarly, the service time was fit by Erlang distribution. In Huang and Wu (2005) a survey of queuing theory models for berth number calculation is given. An M|M|m model is used to derive a relationship between the number of berths m and traffic intensity. Such a queuing theory approach is a simple model of a far more complex process because, e.g., specific ship size class arrival and service times are neglected, scheduling disciplines are very basic, berths are identical, one ship occupies a whole berth.

Though the berth allocation problem (BAP) is not the subject of this paper, SQPP is related to BAP because an SQPP solution, i.e. a quay partition, is a BAP input. Moreover, the evaluation of a partition on one scenario equivalently solves a BAP instance. Studies on BAP are reviewed in Bierwirth and Meisel (2010, 2015), Kovač (2017), Rodrigues and Agra (2022), Stahlbock and Voß (2008). BAP classes were distinguished on the basis of (1) quay layout, (2) vessel arrival type, (3) service time attributes and (4) performance measure (Bierwirth & Meisel, 2010). The quay layout types are: continuous, discrete, hybrid. In the continuous layout vessels can be positioned anywhere along the quay. In the discrete case the quay is partitioned into berths such that at most one ship occupies a berth. Hybrid quay organization allows to assign more than one vessel in a berth. It was discovered that discrete berth layout may be less robust than the hybrid layout (Wawrzyniak et al., 2024). In this paper we assume hybrid quay layout and berths can be occupied by at most two ships, which prevents berth length fragmentation (Błądek et al., 2015; Wawrzyniak et al., 2020). In the BAP formulations vessels arrive over time or all of them are ready at the same time. We assume that vessels arrive over time. Service times, as the BAP attribute, can be fixed and known in advance, or may

depend on ship berthing position, crane assignment and scheduling. In this paper it is assumed that for a given scenario ship service times are known in advance and fixed, while the set of scenarios has statistical features of real service times. A number of optimality criteria were considered in the past. We assume weighted flow time as the quality measure. According to the notation introduced in Bierwirth and Meisel (2010) the BAP version used in this paper to evaluate quay partitions is denoted $hybr|dyn|fix|\sum weight$. In Rodrigues and Agra (2022) authors review and propose taxonomy for BAP with uncertainty. Sources of uncertainty and methods of dealing with it can be the same both in BAP and in SQPP. According to the taxonomy of Rodrigues and Agra (2022) we assume a proactive stochastic programming approach, with uncertain arrival and handling times represented with probability distributions which can be rendered as scenarios, with weighted waiting time as objective, hybrid quay allocation and we use heuristic methods. Let us note that beyond similarities with stochastic BAP there are also differences stemming from the change in time horizon. Stochastic BAP solution fixes berth allocation for vessels in the proactive phase, expecting variations in arrival and processing times. SQPP solution fixes the partition during the proactive phase, but not the berth allocation that can be recomputed later during the reactive or online phase.

A large number of works for solving the BAP were proposed in the literature. A set of examples is presented in Table 1, without claims to be exhaustive. In Table 1 a short name of the considered problem, numbers of ships and berths, year of publication, are collected. The reader is kindly referred to surveys (Bierwirth & Meisel, 2010, 2015; Kovač, 2017; Rodrigues & Agra, 2022) for further BAP results and algorithms. Table 1 asserts a gap between BAP sizes routinely solved in the existing literature, and the ones that have to be solved in our case. For example, the number of ship arrivals in 2016 ranged from 471 (in Gdańsk) to 18494 (Singapore) (Wawrzyniak et al., 2021, 2022). It can be argued that only short weekly patterns of arrivals suffice because ship arrivals are periodic. Indeed, 2-, 3-week return intervals seem natural, but 5-, 7-, 11-week return times are also quite common, especially on Asia-Europe, Europe-America lines (Wawrzyniak et al., 2021, 2022). The least common multiple of these periods, necessary to grasp interactions between returns on different lines, easily spans a year. Furthermore, there might be large differences between expected and effective vessel arrival times, thus making periodic models less attractive (Vernimmen et al., 2007; Wawrzyniak et al., 2022).

Choosing algorithms to evaluate a quay partition on a traffic scenario is subject of algorithm selection problem (ASP). Initially ASP was formalized by Rice (1975) as a problem of determining the best algorithm for a given instance of some problem given some performance measure. Later an idea of solving a problem not by one algorithm but by an algorithm portfolio emerged (Kadioglu et al., 2011; O'Mahony et al., 2008). In Piechowiak et al. (2022), Wawrzyniak et al. (2020) algorithm

Table 2
Summary of notations.

a_j	fraction of ship size class j in the whole number of arrivals
B	set of admissible berth lengths
c_h^{sv}	ship h service completion time for scenario s solved by algorithm v
d	number of scenarios
e	number of ship size classes in the ship traffic model
f	number of admissible berth lengths, $f = B $
$F(K, s, v)$	mean flow time for partition K , scenario s , scheduling algorithm v
$F(K)$	mean flow time for partition K averaged over all evaluations
g	number of algorithms in a portfolio
i	weekly arrival intensity [a/w]
K	partition, vector of berth lengths frequencies
k_i	number of berth length λ_i occurrences in a partition
λ_i	admissible berth length i
L_j	upper end of ship length in size class j
L_h^s	length of the ship in the h th arrival in scenario s
L_{max}	maximum vessel length
m	number of berths in K , $m = \sum_{i=1}^f k_i$
n_s	number of vessel arrivals in scenario s
p_h^s	service time of arrival h in scenario s
Q	quay length
r_h^s	arrival (ready) time of ship h in scenario s
S	ship traffic model (STM)
$\sigma(K)$	standard deviation in partition K evaluations
w_h^s	weight (value) of servicing ship h in scenario s

portfolios covering a set of training instances with the best solutions which could be found in a given runtime limit, subject to the minimum computational cost of the chosen algorithms, were proposed. In this paper we use results of Wawrzyniak et al. (2020) where algorithm portfolios for large BAP instances were considered (see Section 3.3). To evaluate a partition, a customized portfolio of algorithms effective both in solution quality and in time performance will be used. Further results on ASP and algorithm portfolios can be found in Ewald (2010), Kotthoff (2014), Piechowiak et al. (2022), Smith-Miles (2009), Souravlias et al. (2021).

A deterministic version of SQPP, i.e. assuming one arrival scenario, can be solved using mixed integer linear program formulations (Wawrzyniak et al., 2024). On the current computing platforms in a few hours instances with a dozen of vessels and a handful of berths can be solved to optimality. 100-vessel instances can be solved without optimality guarantee, which corresponds with fortnight traffic.

3. Stochastic quay partitioning

3.1. Problem formulation

An instance of SQPP is defined by: quay length Q , set $B = \{\lambda_1, \dots, \lambda_f\}$ of admissible berth lengths, ship traffic model (STM) S and ship arrival intensity i . Without loss of generality we assume $\lambda_1 < \dots < \lambda_f$, and for simplicity of exposition, that Q is divisible by λ_1 . S is a stochastic process generating ship traffic as scenarios. The STM models the fraction a_j of ship size class j in the total number of arrivals, ship arrival times, processing times and weights. Arrival intensity i is an average frequency of vessel arrivals taken over the considered time horizon. A scenario is a realization of an STM. For some scenario s , the number of ship arrivals n_s , ship arrival time r_h^s , service time p_h^s , length L_h^s and weight w_h^s (profit from serving the ship) are determined, for ships $h = 1, \dots, n_s$. This will be explained in more detail in Section 3.2. We assume that under any scenario there is an upper limit L_{max} on vessel lengths. Currently $L_{max} = 400$ m is the size of the largest container mother ships. Necessary safety clearance between ships should be included in their lengths.

A solution of SQPP is a partition of quay length Q into berths which can be represented as vector $K = (k_1, \dots, k_f)$ of berth lengths $\lambda_1, \dots, \lambda_f$ frequencies. By definition $Q \geq \sum_{i=1}^f \lambda_i k_i$ and $L_{max} = \max_{h=1}^{n_s} \{L_h^s\} \leq \max_{i=1}^f \{\lambda_i : k_i > 0\}$, i.e., for all ships sufficiently long berths exist. The total number of berths in some SQPP solution is $m = \sum_{i=1}^f k_i$.

Let d denote the number of STM scenarios, c_h^{sv} ship h service completion time under scenario s obtained by some scheduling algorithm v . A classic scheduling objective is mean weighted flow time (MWFT): $F(K, s, v) = \sum_{h=1}^{n_s} w_h^s (c_h^{sv} - r_h^s) / \sum_{h=1}^{n_s} w_h^s$ for scenario s and scheduling algorithm v . For d scenarios and a portfolio of g scheduling algorithms a population of $d \times g$ values of $F(K, s, v)$ is obtained. We will use the average of the MWFTs over the scenarios and algorithms:

$$F(K) = \frac{1}{dg} \sum_{s=1}^d \sum_{v=1}^g F(K, s, v) = \frac{1}{dg} \sum_{s=1}^d \sum_{v=1}^g \frac{\sum_{h=1}^{n_s} w_h^s (c_h^{sv} - r_h^s)}{\sum_{h=1}^{n_s} w_h^s} \quad (1)$$

as the key quality criterion. We use average over algorithms because it is a quality central tendency indicator, which is more reliable than, e.g., the results in one best algorithm. Standard deviation $\sigma(K)$ in the population of values $F(K, s, v)$ will be the secondary criterion, used only to assess the impact of traffic intensity in Section 5.2.

A deterministic version of quay partitioning, assuming knowledge of future ship arrivals (hence with only one scenario), is in class NP because it suffices to guess the partition and the schedule for the single scenario. Since BAP is NP-hard (Lim, 1998), also deterministic version of quay partitioning is NP-hard even for a fixed quay partition K and one scenario. When a partition can be flexibly chosen, the deterministic quay partitioning is also NP-hard, even for equal service times, all vessels simultaneously ready and a schedule two time units-long (Wawrzyniak et al., 2024). To better illustrate the decision problem at hand, consider the following example: $Q = 1200$, $B = \{200, 400, 600\}$. In some arrival scenario s vessel lengths are $L_1^s = 400$, $L_2^s = 300$, $L_3^s = 300$, $L_4^s = 300$, $L_5^s = 300$, $L_6^s = 300$, $L_7^s = 200$, $L_8^s = 200$, weights are equal to vessel lengths, all vessels are ready at time zero, service times are all equal to a time unit. In Fig. 2 optimal schedules (obtained for any exact algorithm v) for the three partitions are shown. In Fig. 2a equal partitioning $K_1 = (0, 3, 0)$ results in $F(K_1, s, v) = 4000$. Partition $K_2 = (0, 0, 2)$ allows for $F(K_2, s, v) = 3400$ (Fig. 2b). A perfectly bad partition $K_3 = (4, 1, 0)$, with only one long berth and all other berths as short as possible, has $F(K_3, s, 0) = 6800$ (Fig. 2c).

3.2. Ship traffic model

The STMs allow to generate scenarios of various arrival intensities i in a given time horizon. As said, arrival intensity i is an average frequency of vessel arrivals over the considered time horizon and will be expressed in arrivals per week (a/w for short). For the purposes of the current study, a container ship traffic model has been developed in Wawrzyniak et al. (2021, 2022). Here we present the basic elements of the STM used, to make this paper self-contained and to give the reader an idea what is an STM. The traffic model was built on the basis of historical automatic identification system (AIS) data gathered for eight ports of the world. Hence, the further study reflects practical cases. The set of ships has been clustered into $e = 7$ length classes. For each ship length class, service time and arrival time models were fit into the data. Though the STM has a fixed structure, ports differ and the classes are port-specific. Examples of class j length range upper limits L_j and fractions a_j in the total ship number are shown in Table 3. We will use the simplest ship length model assuming that all ships in the class j have length rounded up to L_j . Ship service time model is derived from probability distribution of fractions p_h^s / L_h^s , i.e., service time (hours) per ship length (meters). Lognormal, logistic, γ distributions are the most common for this ratio. The model in Wawrzyniak et al. (2022) distinguishes returning and non-returning ships. Because a vast majority of ships are returning, in this paper we will consider returning ships only. Consequently, the set of vessels as physical objects is smaller than the number of arrivals. Ship return times were modeled as Gaussian probability mixtures. In the mixture, each component is a normal distribution with some given mean (return period), standard deviation and mixing proportion. An example of STM for class 1, comprising the shortest ships in Le Havre is shown in Table 4. In our tests ship weights were generated as $w_h^s = L_h^s p_h^s (0.5 + rnd(1))$ where

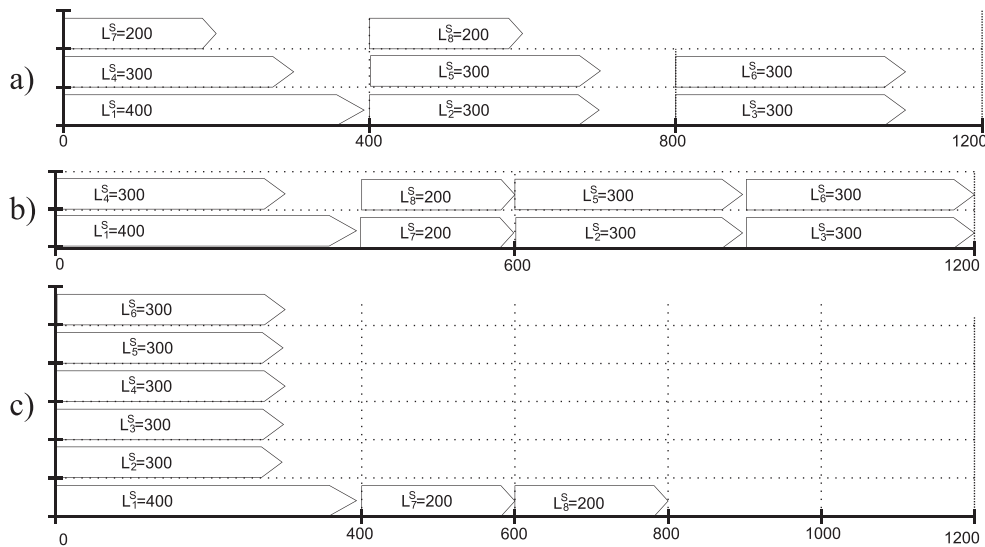


Fig. 2. Example of quay partitioning (a) equal partitioning, (b) flexible partitioning, (c) perfectly bad solution.

Table 3

STM vessel size classes examples: length range upper limits L_j (in meters) and fractions a_j in the total vessel number. On the basis of Wawrzyniak et al. (2022).

Port\class	1	2	3	4	5	6	7
Le Havre	140,0.095	210,0.148	245,0.147	278,0.176	300,0.155	368,0.232	399,0.046
Los Angeles	224,0.128	261,0.101	279,0.138	295,0.193	305,0.138	335,0.225	399,0.078
Shanghai	101,0.055	148,0.357	183,0.195	237,0.136	297,0.155	348,0.088	367,0.013
Singapore	192,0.246	198,0.173	225,0.121	262,0.118	302,0.172	345,0.088	400,0.081

Table 4

STM for Le Havre ship size class 1 (shortest ships).

0.0951	a_1 — fraction of class 1 ships in the whole set of container ships
[118,140]	range of ship length in class 1 [meters]
lognormal	type of p_h^s/L_h^s distribution [hours per meter]
-2.44661	parameter of lognormal distribution, expected value of $\ln(p_h^s/L_h^s)$
0.68494	parameter of lognormal distribution, standard deviation of $\ln(p_h^s/L_h^s)$
6	number of components in the Gaussian mix representing ship return times, the components are defined by mixing proportion, mean value [days], standard deviation [days]. The components are given below:
(0.1151, 4.6301, 0.8343) (0.1740, 15.5915, 2.0145) (0.0685, 21.0844, 1.0550) (0.2201, 36.8575, 22.4692) (0.3928, 42.0277, 0.7392) (0.0296, 201.0784, 89.3193)	

$rnd(1)$ is a pseudorandom number generator with uniform distribution in range $[0,1]$. The total number of vessels in a scenario is a product of time horizon length and arrival intensity ι . In our case one year is the time horizon. Full model parameters and example scenarios can be found at Wawrzyniak et al. (2021). The assumptions on the STM and the obtained results are further discussed in Wawrzyniak et al. (2022).

Let us observe that STM parameters may not only be used to generate test scenarios, but also to guide construction of SQPP solutions (Section 4.2).

3.3. Partition evaluation

In order to evaluate quality of partition K on an STM-generated scenario s a BAP instance must be solved. Since SQPP emerges in long strategic port planning, arrival numbers n_s are large. Moreover, BAP is NP-hard (Lim, 1998) and evaluating K may require long runtimes. Furthermore, it is necessary to avoid exposing the evaluation process to the peculiarities of just one BAP algorithm. Therefore, to resolve the above issues we will apply algorithm portfolios tailored to a limited runtime and high quality of the solutions. Selecting algorithms portfolios for very large BAPs under runtime limit was tackled in Wawrzyniak et al. (2020). According to the methodology developed in Wawrzyniak et al. (2020) portfolios covering the training instances with the best solutions which could be found by certain runtime limits, at minimum computational cost were constructed. Consequently, the portfolios comprise

only a small subset of the available algorithms which uniquely solved some test instances to the best solution found at the given time limit. 72 algorithms were evaluated on sets of training instances.

For the further study a portfolio of $g = 8$ algorithms constructed for runtime of at most 50 s at $n_s = 10000$ was selected (compare Fig. 6 in Wawrzyniak et al., 2020). The portfolio comprises heuristic algorithms: SPT-Prio, SPTGI-Prio, SPTGI-La2, GISPT-Prio, SAF-Prio, SAF-La2, RND-Prio, RND-La5. The first part of the algorithm name describes the sorting rule, i.e. the sequence in which ships are considered. For example, SPTGI orders ships according to the shortest processing times p_h^s first (SPT) and then according to the decreasing weights w_h^s (greatest importance, GI). SAF stands for the smallest area $p_h^s L_h^s$ first. RND is a random order. The second part of the name details which ships are considered for the assignment to the berths. ‘‘Prio’’ means that at ready times and ship service completion times the first ready ship is chosen according to the sorting rule. Algorithms with ‘‘Lak’’ take into account also the first k future ship arrivals and choose from them according to the sorting rule. The chosen ship is assigned to the shortest feasible berth available earliest. All these algorithms apply a hybrid berth layout with at most two vessels in a berth.

In Table 5 examples of evaluation runtimes for annual traffic with various weekly arrival intensities are given. For example, intensity 100 a/w is roughly equivalent to scheduling $n_s = 5200$ arrivals. In the tests shown in Table 5 the set of berth length was $B = \{100, 200, 300, 400, 500, 600\}$ m. There were 10 scenarios for each

Table 5
Evaluation runtimes in seconds.

intensity [a/w]	20	50	100	200	500	1000
Le Havre, $Q = 3500$ m, runtime per scenario						
median	0.48	3.52	43.69	271.58	1814.19	6011.90
SIQR	0.18	2.50	16.88	48.56	220.44	296.65
Le Havre, $Q = 3500$ m, runtime per scenario and algorithm						
median	0.06	0.44	5.46	33.95	226.77	751.49
SIQR	0.02	0.31	2.11	6.07	27.56	37.08
Shanghai, $Q = 5000$ m, runtime per scenario						
median	0.92	2.92	7.68	31.89	1132.50	3973.68
SIQR	0.07	0.10	0.38	29.99	255.66	388.10
Shanghai, $Q = 5000$ m, runtime per scenario and algorithm						
median	0.11	0.36	0.96	3.99	141.56	496.71
SIQR	0.01	0.01	0.05	3.75	31.96	48.51

SIQR–Semi-inter-quartile range.

port and intensity. All non-dominated partitions were tested (cf. Section 4.1). Computational tests presented in this paper were conducted in Poznań Supercomputing and Networking Center, on Altair cluster with Intel Xeon Platinum 8268 CPUs, each with 48 cores and 4 GB RAM per core. On the one hand, the computations were quite extensive. Over 43800 processor-hours were spent in the tests shown in Table 5. On the other hand, the runtimes per algorithm and per scenario are short, all evaluations of d scenarios by g algorithms are mutually independent and were run in parallel.

Let us observe that there may be further advantages of using algorithm portfolios to evaluate partitions. At the early strategic level of designing a container terminal, it is unknown what methods will be used by the terminal operators to assign vessels to the berths. A population of evaluations by various algorithms exposes the quay design process to a greater variety of possible scheduling methods, and thus, builds a broader view on potential dispersion of the partition scores under the future, yet currently unknown, BAP algorithms.

4. Partitioning algorithms

In this section we introduce algorithms constructing partitions K for the given quay length Q , set of admissible berth lengths B and STM S . The algorithms implement the general methodology of SQPP solution outlined in the Introduction and depicted in Fig. 1.

4.1. Brute-force enumeration

The number of possible partitions grows exponentially with the number of berth sizes and can be upper-bounded by $O(Q^f / \prod_{i=1}^f \lambda_i)$. The actual number of interesting partitions is smaller because some partitions are dominated. Let $K = (k_1, \dots, k_f)$ be a partition. If $\exists t : \lambda_t < Q - \sum_{i=1}^f k_i \lambda_i$ then K is dominated by $K' = (k_1, \dots, k_t + 1, \dots, k_f)$. Let us remind that there must be at least one berth of length greater or equal L_{max} . In Table 6 numbers of non-dominated partitions are given for some exemplary quay lengths and sets B of admissible berth lengths. The berth lengths in B were chosen from range [100,600]m with various resolutions: 200 m, 100 m, 50 m, 25 m. It can be concluded, that the number of non-dominated partitions is amenable to brute-force evaluation for contemporary quay lengths and small sets B (e.g. Q up to 5 km and berth lengths with 100 m resolution). In the following this method will be presented under short name BF.

4.2. Histogram matching and integer programming

Suppose a histogram of ship size classes importance is given as a vector (q_1, \dots, q_e) for ship lengths $\{L_1, \dots, L_e\}$, where $q_j \in [0, 1]$ (how to obtain the q_j s will be explained in the following). The quay can be partitioned so that fractions of Q dedicated to particular ship

lengths match histogram (q_1, \dots, q_e) as closely as possible. Let δ denote the biggest difference between the required value q_j and the actual fraction of the quay length where ships of length L_j can be moored. Let u_{ij} be the number of length L_j vessels which can be positioned at berths of length λ_i . Admissible values of u_{ij} are: 0 for $\lambda_i < L_j$, 1 when $L_j \leq \lambda_i < 2L_j$ and 2 for $2L_j \leq \lambda_i$. Solution vector $K = (k_1, \dots, k_f)$ approximating importance vector (q_1, \dots, q_e) can be obtained by solving the following integer linear program:

$$\text{ILP1 : min } \delta \tag{2}$$

$$\text{subject to: } \sum_{i=1}^f k_i \lambda_i = Q \tag{3}$$

$$\sum_{\{i: \lambda_i \geq L_{max}\}} k_i \geq 1 \tag{4}$$

$$\sum_{j=\ell}^e q_j - \sum_{j=\ell}^e \sum_{\{i: \lambda_i \geq L_j\}} u_{ij} k_i L_j / Q \leq \delta(e - \ell + 1) \text{ for } \ell = 1, \dots, e \tag{5}$$

$$k_i \in Z^+ \text{ for } i = 1, \dots, f \tag{6}$$

In ILP1 δ, k_i are decision variables, $\lambda_j, L_j, u_{ij}, Q$ are constants. The greatest divergence δ from the required ship length frequency is minimized. Constraint (3) guarantees that the chosen berth lengths fit on the quay and no space is left unused. Eq. (3) can be satisfied with equality because λ_1 divides Q . By constraint (4) there is at least one berth at least as long as the longest ship. Value $\sum_{\{i: \lambda_i \geq L_j\}} u_{ij} k_i L_j / Q$ in (5) is the fraction of quay length at which ships of length L_j can be positioned. On the left-hand-side of (5) the summation over $j = \ell, \dots, e$ and $\lambda_i \geq L_j$ represents the fact that vessels of length $L_\ell < \dots < L_e$ may be positioned at the same berths. For the same reason δ is multiplied by $(e - \ell + 1)$ on the right-hand-side of (5). Thus, constraints (5) set the distance δ between the required importance of vessel length ℓ, \dots, e and fraction of the space at the quay obtained by these lengths.

In an alternative integer linear program referred to as ILP2 the biggest difference between the required value q_j of availability and the actual fraction of the quay length where ships of length L_j can be moored, is bounded for each length separately. Then, the following constraints:

$$q_j - \sum_{\{i: \lambda_i \geq L_j\}} u_{ij} k_i L_j / Q \leq \delta \text{ for } j = 1, \dots, e \tag{7}$$

substitute (5) in (2)–(6).

Now let us discuss obtaining the importance histograms. In Section 3.2 ship size classes j were introduced with their relative frequencies $a_j, p_j / L_j$ distributions and upper ends of ship lengths in a class. The length range upper ends can serve as vessel lengths L_j in the above formulations. A method matching class frequency a_j histogram by use of ILP1 or ILP2 will be referred to as ILP1f and ILP2f, respectively. A histogram of area in time×space for class j occupation can be built by

Table 6
Number of non-dominated partitions for exemplary ports.
Berth lengths in [100 m, 600 m] with $L_{max} = 400$ m

<i>B</i> with resolution	200 m	100 m	50 m	25 m
Gdańsk $Q = 660$ m	2	4	11	31
Le Havre $Q = 3.5$ km	32	2052	187 444	66871400
Shanghai $Q = 5$ km	64	8958	2 470 285	4579673758
$Q = 10$ km	233	188 625	641225038	*
$Q = 26$ km	1473	16 738 836	*	*

* — enumeration abandoned after 24 h.

assuming $q_j = \overline{p_j/L_j} * L_j^2 / \sum_{l=1}^7 (\overline{p_l/L_l} * L_l^2)$, where $\overline{p_j/L_j}$ is average service time per unit of vessel length for class j . A method matching this kind of histogram will be called area matching and referred to as ILP1a and ILP2a for ILP1 and ILP2, respectively. Integer linear programs were solved by CPLEX version 12.8.0.0.

4.3. Big berths first

Big Berths First algorithm (BBF in short) builds as many as big berths as possible. The algorithm proceeds from the longest berths λ_f to the shortest λ_1 . For the considered berth length λ_i and the remaining quay size Q' , set $k_i = \lfloor Q'/\lambda_i \rfloor$ and $Q' = Q' - k_i \lambda_i$, for $i = f, \dots, 1$.

4.4. Hill climber

Let us observe that high solution evaluation cost prevents applying population-based metaheuristics. Conversely, trajectory-based metaheuristic, such as local search improving solutions by gradually modifying them, are better suited to solve SQPP. Hill climber (HC for short, see Algorithm 1) is a local search method which proceeds to the first-found better solution and stops if no improving solution exists. All local search methods need defining: explicit representation of the solution, the set of admissible solution changes called moves, a starting solution and a strategy to choose moves. The solutions are represented here as berth length frequency vectors. In Algorithm 1, K is the currently tested solution, and K^* is the best found one. The set of all solutions attainable by executing moves on the current solution are called neighborhood. There are two types of moves: split and merge. In the split move, berth of length λ_i is split into two shorter berths of length λ_x, λ_y such that $\lambda_x + \lambda_y = \lambda_i$. All such pairs x, y are tested. In the merge move, for a pair of lengths λ_x, λ_y the pair is substituted by the berth length λ_i satisfying $\lambda_x + \lambda_y = \lambda_i$. Again, all pairs x, y are tested. The set of moves is denoted Admissible_moves in the HC pseudocode. Algorithm BBF provides the starting solution. HC stops in a local optimum, that is, after enumerating all Admissible_moves without solution improvement.

4.5. Tabu search

Tabu search is also a local search method which extends a hill climber by use of memory to prevent accessing already visited solutions (cf. Algorithm 2). To prevent accessing already visited solutions, indexes of executed moves are stored in a queue called Tabu. A move stored in the Tabu queue can be undone only if it improves the best solution found. For practical usability, our tabu search algorithm is limited to 401 evaluations of the objective (1) and at most 24 h runtime, including evaluations (collectively denoted as run_limits in Algorithm 2). These limits were established as not restricting arriving in the best solutions found in the initial tests of the algorithm. If no improving solution is found, then the best non-improving solution K'' is assumed as the current one (line 20). Tabu search stops if all moves are in Tabu, and hence blocked, or on the run_limits.

Algorithm 1 Hill_Climber(Q, B, STM, i)

```

1:  $K^* \leftarrow BBF(Q, B);$  //greedy initial solution
2:  $F(K^*) \leftarrow Evaluate(K^*, STM, i);$  //evaluate the initial solution
3: while true do
4:   for  $v$  in Admissible_moves do
5:      $improved \leftarrow false;$ 
6:      $K \leftarrow Move(K^*, v);$  //build the next partition
7:      $F(K) \leftarrow Evaluate(K, STM, i);$  //evaluate partition  $K$ 
8:     if  $F(K) < F(K^*)$  then
9:        $K^* \leftarrow K; improved \leftarrow true; break;$  //better partition found, exit for loop
10:    end if
11:  end for
12:  if not improved then
13:    return  $K^*$ ; //all Admissible_moves evaluated, local optimum found
14:  end if
15: end while

```

Algorithm 2 Tabu_Search(Q, B, STM, i)

```

1:  $K^*, K' \leftarrow BBF(Q, B);$  //greedy initial solution
2:  $F(K^*), F(K') \leftarrow Evaluate(K^*, STM, i);$  //evaluate the initial solution
3: while not run_limits do
4:    $K'' \leftarrow \emptyset; F(K'') \leftarrow \infty; v_n \leftarrow \emptyset$  //  $K''$  is the best non-improving solution;
5:   for  $v$  in Admissible_moves do
6:      $improved \leftarrow false;$ 
7:      $K \leftarrow Move(K', v);$  //build the next partition
8:      $F(K) \leftarrow Evaluate(K, STM, i);$  //evaluate partition  $K$ 
9:     if  $F(K) < F(K^*)$  then
10:       $K^* \leftarrow K; improved \leftarrow true;$  //better partition found
11:       $Tabu.append(v); break;$  //remember move  $v$  as tabu, exit for loop
12:     else if  $F(K) < F(K'')$  and  $v \notin Tabu$  then
13:        $K'' \leftarrow K; v_n \leftarrow v;$  // remember the best non-improving solution
14:     end if
15:   end for
16:   if not improved then
17:     if  $K'' = \emptyset$  then
18:       break; //all Admissible_moves are tabu, exit while loop
19:     else
20:        $K' \leftarrow K''; Tabu.append(v_n);$  //use the best non-improving solution  $K''$ 
21:     end if
22:   end if
23: end while
24: return  $K^*$ ;

```

Table 7
Overview of BF results.

Le Havre STM, real traffic in 2016: 2271 arrivals					
intensity ι [a/w]	20	50	100	200	500
$\min_s \{n_s\}$	1052	2610	5214	10 409	26 016
$\max_s \{n_s\}$	1085	2657	5259	10 452	26 051
$F(K^*)$	25.24	25.89	409.21	4664.34	17 438.73
$F(K^*) + \sigma(K^*)$	25.91	26.32	607.73	5079.50	18 373.24
No. of sol. in $[F(K^*), F(K^*) + \sigma(K^*)]$	1180	79	27	32	23
No. of BF solutions	2052	2052	2052	2052	2052
Shanghai STM, real traffic in 2016: 11606 arrivals					
intensity ι [a/w]	20	50	100	200	500
$\min_s \{n_s\}$	1093	2639	5229	10 430	26 034
$\max_s \{n_s\}$	1148	2709	5286	10 518	26 127
$F(K^*)$	25.26	24.17	25.32	27.70	4197.84
$F(K^*) + \sigma(K^*)$	29.70	27.86	27.90	31.37	4579.60
No. of sol. in $[F(K^*), F(K^*) + \sigma(K^*)]$	8880	8200	6455	749	109
No. of BF solutions	8958	8958	8958	8958	8958

a/w — arrivals per week

K^* — partition with the minimum score (1), i.e., the best one.

4.6. Random

This algorithm adds berths of random length to solution K until exhausting the remaining free quay length. This procedure is repeated 10 times and will be referred to as *Rnd10*. An average $F(K)$ from all generated layouts is returned. No particular solution of RND10 will be used. Rnd10 is a reference algorithm for other algorithms performance comparisons.

5. Features of SQPP solutions

In this section we analyze features of the SQPP solutions obtained by brute force partition enumeration. Thus, optimum partitions are known. The test setting follows the description in Sections 3.2 and 3.3. In more detail, two STMs representing Le Havre and Shanghai were used. For Le Havre quay length was $Q = 3500$ m and for Shanghai it was set to $Q = 5000$ m (the quay length operating at the time the traffic data were collected). In both cases the set of admissible berth lengths was $B = \{100, 200, 300, 400, 500, 600\}$ meters. The numbers of non-dominated partitions were 2052 for Le Havre and 8958 for Shanghai, respectively, as shown in Table 6. One year traffic was generated with intensity levels 20, 50, 100, 200, 500 a/w. Let us observe that intensity $\iota = 500$ a/w exceeds the current traffic of Le Havre and Shanghai 11 and 2.2 times, respectively, and hence goes into the realm of potential future demands. Hence, this study may shed some light on the readiness of these ports to serve future progresses of the local economy. For each intensity level, 10 test scenarios were generated. Thus, each partition was evaluated on 5 arrival intensity levels, each with 10 test scenarios s , solved by 8 BAP scheduling algorithms v . This created, for each partition and intensity level, a population of 80 evaluations. $F(K)$ as defined in (1) and standard deviation of $F(K, s, v)$ were calculated for each such population. The results are summarized in Table 7. In this table the range of vessel numbers in the generated test scenarios ($\min_s \{n_s\}, \max_s \{n_s\}$), range of *MWFT*s from the best partition score $F(K^*)$ to the best partition plus one standard deviation $F(K^*) + \sigma(K^*)$, number of partitions in range $[F(K^*), F(K^*) + \sigma(K^*)]$ and the total number of verified partitions are given.

It can be concluded from the results in Table 7 that traffic intensities of 20 arrivals per week (a/w) for Le Havre, and up to 100 a/w for Shanghai, are easy to handle. This is indicated by low and similar values of the best $F(K^*)$ in these arrival intensity ranges. Furthermore, the number of partitions in range $[F(K^*), F(K^*) + \sigma(K^*)]$ near the smallest $F(K^*)$, is large (especially in Shanghai). It means that at this low traffic it is easy to schedule the arriving vessels on nearly any quay partition (except for the degenerate solutions) because there is an excess of available berthing space, quay partitioning is actually immaterial. Hence, there are many partitions which are nearly as

good as the best one. We will discuss structure of the good and the particularly bad partitions in Sections 5.1 and 5.2. Intensities 50 a/w in Le Havre and 200 a/w in Shanghai are border cases because low values of $F(K^*)$ are achievable, but only by a small set of partitions. Let us note that the 2271 arrivals that actually happened in Le Havre in 2016 is roughly intensity 42 a/w, and the 11606 arrivals in Shanghai is roughly 223 a/w. Thus, both ports seem to be close to the border traffic intensities, and schedules for the arriving vessels are not hard to construct on reasonably chosen quay partitions. With even greater arrival intensity, the number of partitions with their $F(K)$ score close to the best one tends to decrease. Thus, quay partitioning matters in vessel quality of service.

5.1. Patterns in berth length selection

In Fig. 3 general tendencies in the selection of berth lengths are shown. In the pictures berth lengths from set $B = \{100, 200, 300, 400, 500, 600\}$ are shown along the horizontal axis. The average number of berths of a given length in some subset of solutions is shown along the vertical axis. The standard deviation of the chosen number of berths of a given length is also shown. Three types of partition subsets are shown in Fig. 3. The leftmost column displays the 20 best partitions, the central column shows the partitions in the range $[F(K^*), F(K^*) + \sigma(K^*)]$, where $F(K^*)$ is the score of the best partition and $\sigma(K^*)$ a standard deviation of its evaluations. The rightmost column displays berth length frequencies in the 50 worst solutions. STMs and vessel traffic intensities are shown in rows. The first two rows show results for the Le Havre STM, and the last two rows comprise results for the Shanghai STM. Light traffic (20 a/w) is depicted in the first and the third row. Intensive traffic (500 a/w) is presented in the second and the fourth row. It can be observed that in general the best solutions (left column) tend to have many large berths (i.e. capable of hosting the longest vessels with $L_{max} = 400$ m). Note, however, that in these solutions (left column), not one but many berth lengths are chosen yet in various frequencies. Conversely, the worst partitions have many short (100 m) and very few large berths, which seems the perfect worst solution. These general observations are moderated by traffic intensities and STMs. In very low traffic (Fig. 3 first and third row) there are many partitions in the range $[F(K^*), F(K^*) + \sigma(K^*)]$ (cf. Table 7), performing relatively well and the strength of the distinction between good and bad solutions in this situation is small. Consequently the numbers of berths chosen in the population defined by range $[F(K^*), F(K^*) + \sigma(K^*)]$ for low traffic, represent frequencies of berth lengths in the whole set of partitions rather than the frequencies of berth lengths in good solutions. If we select even more restrictively the 20 best solutions at light traffic (left column, first and third row) then tendencies in good solutions become more visible: In Le Havre

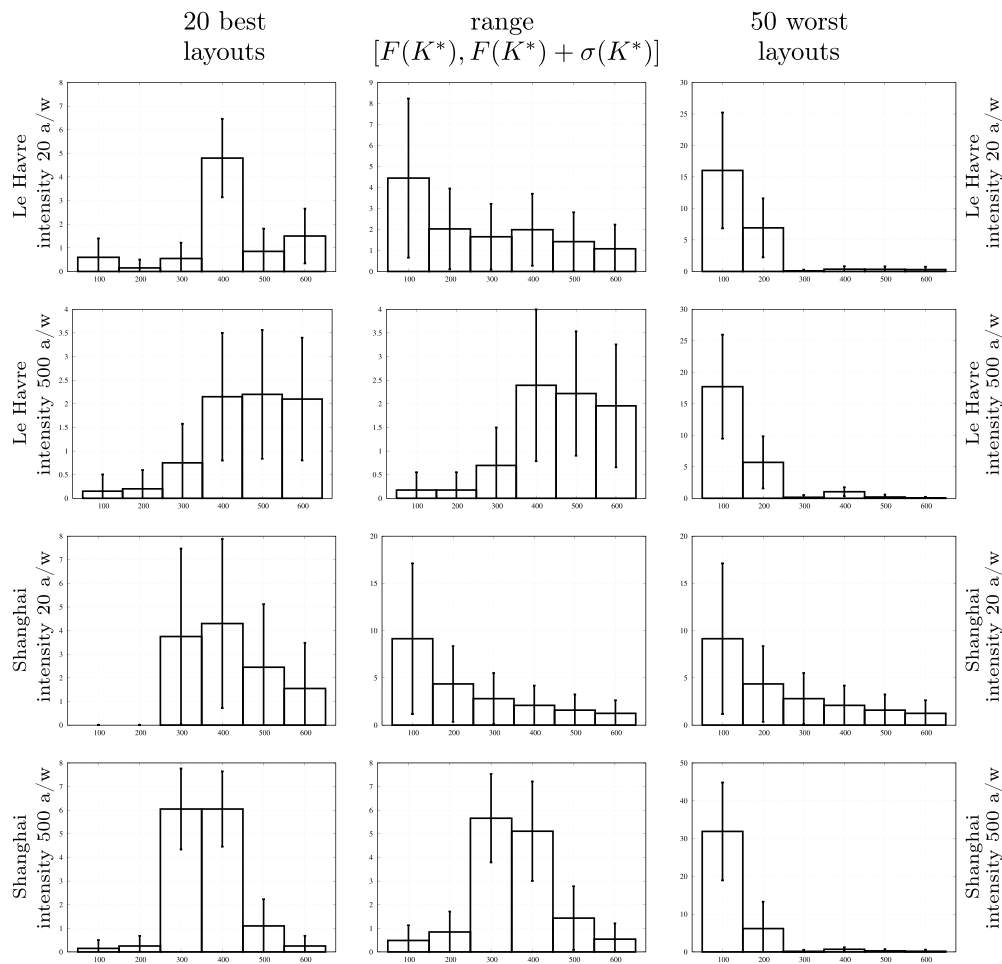


Fig. 3. Numbers of berth lengths from set $B = \{100, 200, 300, 400, 500, 600\}$ meters in subsets of partitions for Le Havre, Shanghai, intensities 20 a/w and 500 a/w.

berths with $L_{max} = 400$ m are most frequently chosen, in Shanghai berth lengths 300 m and longer are selected. The difference between Le Havre and Shanghai can be explained by different mix of vessel lengths in the STMs: Shanghai has more short vessels than Le Havre. For the intensive traffic (Fig. 3 second and fourth row) for the population of solutions in range $[F(K^*), F(K^*) + \sigma(K^*)]$ (central column) the results resemble the restricted set of the 20 best solutions (left column). It can be concluded that for intensive traffic, good partitions have to be selected carefully, their number is small (see Table 7) and hence they are more like the 20 best solutions. The chosen numbers of certain length berths differ in Le Havre and Shanghai because their STMs differ in the mix of long and short vessels. In Le Havre large berths ($\lambda_i \geq 400$ m) dominate in the best solutions because Le Havre is the first oceanic port in western Europe and large mother ships are quite frequent. In Shanghai, berth lengths 300 m and 400 m are the most frequently selected because Shanghai Yangshan port is located in the estuaries of two large rivers and short ships are more common than in Le Havre (Wawrzyniak et al., 2022). Note that in both ports not one but a few berth lengths are chosen in the 20 best solutions.

5.2. Quality dispersion

In Fig. 4 dispersion $\sigma(K)$ vs solution quality $F(K)$ for all partitions is shown. Along horizontal axes $F(K)$ is shown. Standard deviation $\sigma(K)$ of $F(K, s, v)$ in the population of 80 evaluations is shown along the vertical axes. Each point represents one partition. Colors show the number of large berths, that is $|\{\lambda_i : \lambda_i \geq 400\text{m}\}|$. The choice of $F(K)$ as a partition quality measure (horizontal axis) is natural because $F(K)$

is a quality of vessel service. Let us note that partitions incurring large dispersion of service quality are hard to manage: there are scenarios where such partitions perform well but also scenarios where quality of vessel service is low, there are scheduling algorithms which manage the scenarios on such partitions well, but also algorithms that are not equally effective. Thus, standard deviation of $F(K, s, v)$ is also a useful indicator of the partition goodness, and partitions with low dispersion should be preferred. In Fig. 4a,b results for low intensity traffic, and in Fig. 4c,d for high intensity traffic, are shown, respectively. It can be observed that differences between good and bad partitions can be large, even two orders of magnitude differences both in $F(K)$ and in $\sigma(K)$ are possible. Thus, optimization of quay partition is relevant. Generally, partitions with high quality of service (low $F(K)$) are also the partitions with low $\sigma(K)$. This is particularly true for low intensity traffic (Fig. 4a,b) — by improving $F(K)$ also $\sigma(K)$ is improved. In the case of low intensity traffic many points overlap in Fig. 4a,b, which indicates that there are many solutions similar in the criteria of $F(K)$ and $\sigma(K)$. This observation is in line with the earlier find (cf. Table 7) that at low traffic there are many similar solutions, in particular similar to the one with best $F(K)$. Colors in Fig. 4a,b show that partitions with small number of large berths are particularly bad because they do not differ much from the worst-case solution comprising one berth of length $L_{max} = 400$ m and all remaining berths as short as possible (100 m).

For high intensity traffic (Fig. 4c,d) similar observations can be made: while improving $F(K)$ also dispersion of $F(K, s, v)$ is decreasing, solutions with inadequate number of large berths are the worst. However, in high intensity traffic the partitions are less similar to each other, in the performance sense, which is marked by more dispersed

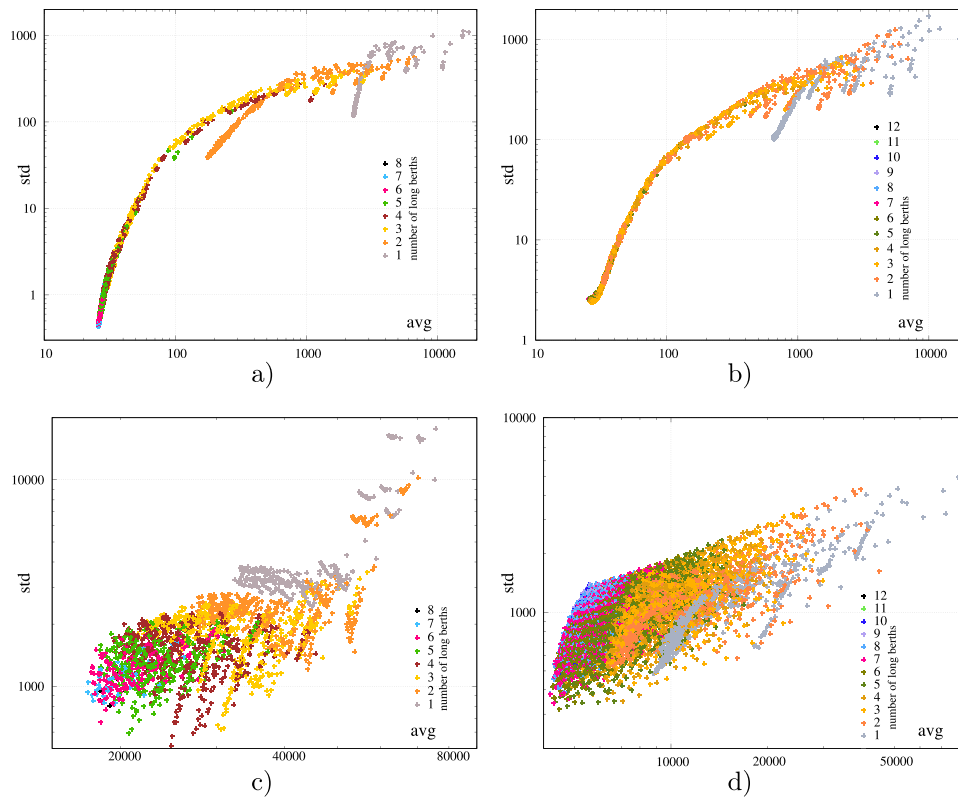


Fig. 4. MWFT average–standard deviation diagram including all brute-force solutions. (a) Le Havre intensity 50 a/w, (b) Shanghai intensity 100 a/w, (c) Le Havre intensity 500 a/w, (d) Shanghai intensity 500 a/w.

cloud of points showing the solutions. If $F(K)$ and $\sigma(K)$ are two criteria of partition quality, then it can be observed that a Pareto-front of non-dominated solutions exist when traffic is intensive. The non-dominated solutions in Fig. 4c,d are presented in Table 8. Beyond the values of $F(K)$, $\sigma(K)$, also the partitions and the number of large berths are shown in Table 8. For example, partition (0, 0, 0, 2, 3, 2) for Le Havre has 2 berths of length 400 m, 3 berths of length 500 m, and 2 berths 600m-long. It can be observed that in Le Havre partitions with the smallest $F(K)$, all berths can accommodate the longest vessels ($L_{max} = 400$ m). However, this impacts dispersion of $F(K, S, \nu)$ because traffic of large and short vessels interfere on these large berths. Adding a few shorter berths allows to slightly separate short vessels from the large ones. Thanks to this, standard deviation of MWFT decreases, yet $F(K)$ increases. In Shanghai similar phenomena emerge, yet the traffic of shorter vessels in Shanghai is more intensive (in relation to large vessels) than in Le Havre, and consequently the partition with the smallest $F(K)$ already has berths that are shorter than L_{max} .

5.3. Solution similarity

In this section we analyze probabilities that some solutions of SQPP turned out good by random coincidence. More formally, for pairs of partitions K_1, K_2 probabilities (p -values) in Student's t -test for paired samples are calculated with the null hypothesis that the performance of the two partitions is the same. The samples are paired in this sense that the results in 80 evaluations $F(K_1, s, \nu)$, $F(K_2, s, \nu)$ over the set of 10 scenarios s and 8 algorithms ν are matched. Low p -values mean that the two partitions performed differently and the dominance of one solution over the other is not a coincidence. Conversely, high p -values indicate that two partitions K_1, K_2 performed similarly and the order of the two solutions with respect to $F(K_1), F(K_2)$ could be different. The p -value is not only a score allowing to reject a null hypothesis, but can also be interpreted as a measure of similarity between the two solutions K_1, K_2 .

For brevity of exposition we will refer to the p -values as indicators of similarity.

In Table 9 dominance of the best partition is demonstrated using its p -values in the test with the second-best solution (denoted $p_{1 \rightarrow 2}$). It can be seen that the first and the second-best solutions are often similar in performance because the p -values happen to be greater than 0.3. In the case of Shanghai and $\iota \in \{20, 50\}$ the number of solutions close to the best one is large (cf. Table 7) so that over 800 best solutions were indistinguishable in their evaluations. Consequently, p -values could not be calculated. Despite this computational inconvenience, this situation shows that at low intensity traffic best solutions for Shanghai are also very similar in the performance sense. The number of solutions which p -values are at least 0.01 in the paired test with the best solution is also shown (denoted $p_{1 \rightarrow i} \geq 0.01$). It can be concluded that the set of solutions similar to the best one is not large and its cardinality decreases with arrival intensity.

The results of pairwise similarity tests for the 100 best solutions are shown in Fig. 5 for Le Havre. The partitions K are put along the axes in the order of increasing $F(K)$. The p -values for pairs of partitions are indicated in color: values smaller than 0.01 are shown as white points, values greater than 0.99 are marked red, values in range [0.01,0.99] are shown as shades of green. It can be seen that the tendency of forming groups of solutions performing similarity depends on the traffic intensity. The solutions differ most for arrival intensity 100 a/w (Fig. 5c). Blocks of similar solutions can be seen both for the lighter (Fig. 5a,b) and for more intensive traffic (Fig. 5d,e). For the light traffic there is abundant free space at the quay and it is possible to build good schedules on many alternative partitions. Hence, groups of similarly performing partitions almost naturally exist. This situation echoes results shown in Table 7. In heavy traffic space at the quay is over-subscribed and vessels inevitably have to wait. Some groups of partitions perform similarly because they do not offer any alternatives to the waiting. The border case of arrival intensity $\iota = 100$ a/w (Fig.

Table 8
MFFT average–standard deviation non-dominated solutions at $\iota = 500$ a/w.

Le Havre			
K	$F(K)$	$\sigma(K)$	No. large berths
(0,0,0,2,3,2)	17 438.7	934.5	7
(0,0,0,1,5,1)	17 762.7	904.2	7
(0,0,1,4,2,1)	17 847.7	852.7	7
(0,0,2,1,5,0)	18 411.3	671.2	6
(0,2,2,0,5,0)	20 713.2	590.4	5
(1,4,2,0,4,0)	24 781.3	515.8	4
Shanghai			
K	$F(K)$	$\sigma(K)$	No. large berths
(0,0,6,8,0,0)	4197.8	381.8	8
(0,0,7,6,1,0)	4202.1	380.3	7
(1,0,7,7,0,0)	4297.4	340.2	7
(2,0,8,6,0,0)	4477.1	318.8	6

$B = \{100, 200, 300, 400, 500, 600\}$ admissible berth lengths in meters.

Table 9
The best solution dominance in Student's t -test.

Le Havre					
intensity [a/w]	20	50	100	200	500
$p_{1 \rightarrow 2}$	0.037	0.343	9.31E–23	1.46E–12	0.159
No. sol. $p_{1 \rightarrow i} \geq 0.01$	6	2	0	0	1
Shanghai					
intensity [a/w]	20	50	100	200	500
$p_{1 \rightarrow 2}$	nan	nan	0.655	0.394	0.462
No. sol. $p_{1 \rightarrow i} \geq 0.01$	nan	nan	23	3	1

$p_{1 \rightarrow i}$ — p -value for the test that the 1st- and i th-best solutions perform the same.
 No. sol. $p_{1 \rightarrow i} \geq 0.01$ — number of solutions with $p_{1 \rightarrow i} \geq 0.01$.
 nan — over 800 best solutions are indistinguishable in quality, p -values could not be computed.

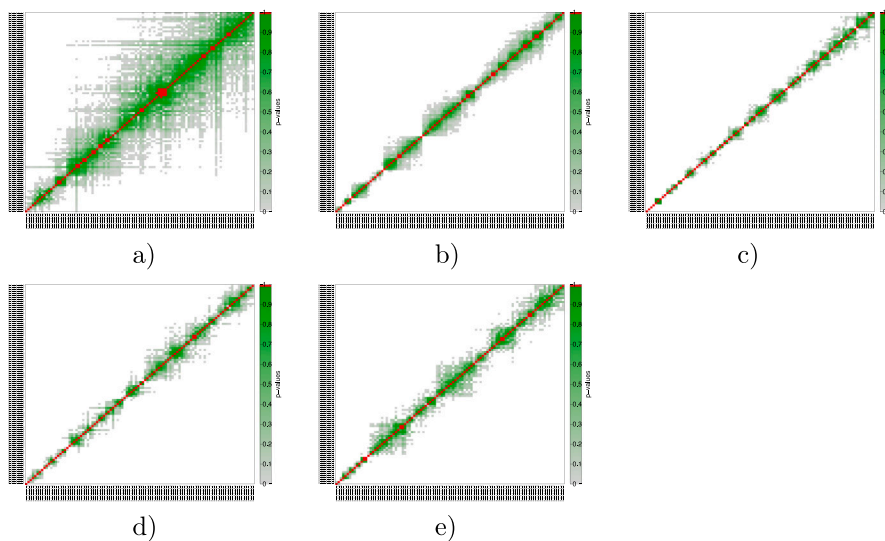


Fig. 5. p -values of a t -test that pairs of partitions perform the same. 100 best partitions, Le Havre, arrival intensity (a) 20, (b) 50, (c) 100, (d) 200, (e) 500 a/w.

5c) distinguishes solutions the most because traffic is intensive enough to fill the available quay and it is also light enough to allow avoiding vessel wait by the virtue of a good partition choice. Solution similarity phenomena for Shanghai were analogous, however, emerging at higher intensities ι .

It can be concluded that the set of solutions with similar performance pattern as the best solution is not large (cf. Table 9), and its size decreases with traffic intensity. Groups of similarly performing solution

appear mostly at light traffic (Fig. 5a) and to a lesser extent also at heavy traffic (Fig. 5e).

5.4. Equipartition

By *equipartition* we understand a type of solution in which the quay is divided into equal length berths. The chosen berth length λ_i must be at least as long as $L_{max} = 400$. Since the chosen berth lengths λ_i

Table 10
Equipartition solutions for intensity 500 a/w.

Le Havre			
K	$F(K)$	$\sigma(K)$	No. large berths
(0,0,0,0,1,5)	18 086.3	1499.4	6
(0,0,0,0,7,0)	18 792.4	882.8	7
(0,0,1,8,0,0)	19 726.5	1085.7	8
Shanghai			
K	$F(K)$	$\sigma(K)$	No. large berths
(0,1,0,0,0,8)	6494.7	1465.6	8
(0,0,0,0,10,0)	5549.0	1385.4	10
(0,1,0,12,0,0)	4695.9	671.1	12

$B = \{100, 200, 300, 400, 500, 600\}$ admissible berth lengths in meters

may not divide Q , the remaining quay length is assigned to as long berth lengths as possible. Note that the BBF algorithm proposed in Section 4.3 builds only one of possible equipartition solutions. The performance of equipartition layouts for arrival intensity 500 a/w is shown in Table 10. With respect to $F(K)$ equipartition solutions are not optimum, but they are not among the worst either (cf. Table 8 vs Table 10 and Fig. 4). However, their solution quality dispersion score $\sigma(K)$ is significantly worse than for the solutions on $F(K), \sigma(K)$ Pareto-front. In conclusion, equipartition is not recommended as better solutions are readily available.

6. Evaluation of the partitioning algorithms

In this section we report on the performance of the algorithms for SQPP. The algorithms were evaluated on the STM, Q pairs: Hamburg STM, $Q = 6000$ m, Hamburg STM, $Q = 7500$ m, Hamburg STM, $Q = 9000$ m, Le Havre STM, $Q = 3500$ m, Rotterdam STM, $Q = 4000$ m, Rotterdam STM, $Q = 5000$ m, Rotterdam STM, $Q = 6000$ m, Shanghai STM, $Q = 5000$ m, Singapore STM, $Q = 10000$ m, Singapore STM, $Q = 14000$ m, Singapore STM, $Q = 20000$ m, Singapore STM, $Q = 26000$ m, with admissible berth lengths $B = \{100, 200, 300, 400, 500, 600\}$ meters and for arrival intensities $i \in \{20, 50, 100, 200, 500, 1000\}$ a/w. The methods of scenario generation were the same as described in Section 3.2. Due to high complexity, only the instances of Le Havre $Q = 3500$ m, Shanghai $Q = 5000$ m for intensities $i \in \{20, 50, 100, 200, 500\}$ were solved by brute force approach.

Algorithms can be evaluated on the criteria of computational cost and solution quality. The computational cost has two components: algorithm own runtime and the cost of solution evaluations. Generally the former is negligible compared with the latter. For example, histogram matching methods from Section 4.2 were executed in at most 0.02s on a standard PC with i7-8550U@1.80 GHz CPU, while the partition evaluations on a high-performance server CPU can last over 751s (see Table 5). Furthermore, computational effort of the evaluations can be arbitrarily resized by changing the number of test scenarios and the size of the scheduling algorithms portfolio. Hence, time performance of the algorithms should rather be assessed by the number of calls to the solution evaluation procedure. For this complexity measure algorithms from Section 4 can be ordered as follows: histogram matching and BBF have cost of one evaluation, Rnd10 has cost of 10 evaluations. Example numbers of evaluations of hill climber are shown in Table 11 and of tabu search in Table 12. Thus, the hill climber used from 5 to 203 evaluations, and tabu search used 54 to 401 evaluations (the fixed maximum value), varying with the STM, Q , and arrival intensity. Brute force enumeration has the highest evaluations cost as shown in Table 6. This cost is growing polynomially with the Q/λ_i ratios but exponentially with the number f of admissible berth lengths. Returning to Tables 11 and 12, HC and tabu search were run 5 times to collect the data for these tables. It could be observed that for small intensities $i \leq 200$ a/w tabu stops on the iteration limit, while for larger intensities on 24 h runtime limit.

The second algorithm performance measure is quality of the solutions as defined in (1). As observed in Section 5, for low arrival

intensities quay partitioning is immaterial and any partition is equally good. Consequently, differences between solutions built by the algorithms are negligible if arrival intensity is low. For this reason we do not report on the cases where the differences in solution scores are insignificant. Precisely, the results are not reported, as not distinguishing the algorithms in a significant way, if $(\max_{v \in \mathcal{A}} \{F(K_v)\} - \min_{v \in \mathcal{A}} \{F(K_v)\}) < \min_{v \in \mathcal{A}} \{\sigma(K_v)\}$, where \mathcal{A} is the set of algorithms and K_v is a solution constructed by $v \in \mathcal{A}$. Quality of the results including brute force approach are presented in Table 13, and for larger intensities in Table 14 (excluding BF). For the brute force method the optimum value $F(K^*)$ is shown. For Rnd10, average of the 10 random solutions are shown under $F(K)$. For the hill climber and tabu search the median of $F(K)$ in the run repetitions is shown. The best non-brute force algorithm results are shown in boldface.

It can be seen in Tabs 13 and 14 that for low arrival intensities (Le Havre, $i = 50$, Singapore $Q=26000$ m and $Q=20000$ m, $i = 500$) quality of the solutions does not differ much. For larger intensities, our algorithms have solution values significantly below Rnd10. If the optimum solution K^* is known (Table 13), then there are algorithms with solutions values close to $F(K^*)$. This signifies that our methods indeed work and provide good solutions. For the cases where tabu search did not build the best solutions, ILP2 is the dominating algorithm, with ILP2f being best twice and ILP2a in 5 other cases. BBF has more complex behavior because for Le Havre it builds the best heuristic solutions at intensities $i \leq 100$ a/w, while at $i = 500$ a/w BBF is only 3.7% worse than tabu search which provided the best solution. For other STMs BBF fails and is even worse than Rnd10. Such variable BBF performance occurs because BBF builds partitions with as long berths as possible, which is effective in Le Havre with dominating long mother ships, but is counterproductive when shorter vessels are very common.

Again, it may be argued that the above results were obtained by a random coincidence. In order to verify this, p -values in the paired samples Student's t -test were calculated for each pair of the algorithms with the null hypothesis that the performance of the two algorithms is the same. The samples are paired in this sense that results $\{F(K_a), F(K_b)\}$ of a pair $\{a, b\}$ of algorithms on the same instance (STM, i, Q, B) of SQPP can be matched and compared. Conventionally, the results of the Student's t -test are used as an indicator whether a null hypothesis can, or cannot, be rejected at certain confidence level. Yet, for brevity of presentation we will present the p -values in terms of algorithm similarity measure. Low p -values indicate that probability of obtaining the given dominance between two algorithms results as a random coincidence is low. Hence, low p -values signify that the two considered algorithms perform differently. The results of the paired algorithm test for similarity are presented in Fig. 6a. For many algorithm pairs p -values are in a range of (0.1,0.2) which is larger than confidence levels of 0.05 or 0.1 conventionally assumed to reject a statistical hypothesis. Yet, it is hard to assess whether probabilities in range (0.1,0.2) are high or low. Consider for example the brute force (BF) and Rnd10 pair. These algorithms may be considered different because BF provides the optimum solution while Rnd10 solutions may be considered unbiased, i.e. tending neither toward good nor toward bad solutions. But even

Table 11
Example number of evaluations by hill climber (min,median, max).

STM, $Q \setminus t$	20	50	100	200	500	1000
Le Havre, $Q = 3.5$ km	6,6,9	6,6,6	6,6,6	6,6,6	14,14,14	8,8,8
Shanghai, $Q = 5$ km	5,6,12	5,6,10	10,10,51	69,69,69	60,60,60	60,60,60
Singapore, $Q = 14$ km	5,5,5	5,5,15	5,6,17	30,35,38	182,190,203	60,60,65
Singapore, $Q = 26$ km	5,5,5	6,6,13	5,13,16	5,5,15	89,99,131	51,51,55

Table 12
Example number of evaluations by tabu search (min,median, max).

STM, $Q \setminus t$	200	500	1000
Le Havre, $Q = 3.5$ km	401,401,401	317,392,401	66,75,129
Shanghai, $Q = 5$ km	270,299,401	146,146,315	98,101,113
Singapore, $Q = 14$ km	113,401,401	319,319,401	54,60,60
Singapore, $Q = 26$ km	401,401,401	401,401,401	137,154,154

Table 13
 $F(K)$ for algorithmically built SQPP solutions.

Le Havre, $Q = 3500$ m									
t	BF	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
50 a/w	25.945	25.945	25.959	25.959	25.959	26.082	28.283	25.945	25.945
100 a/w	409.17	409.17	613.12	548.60	612.87	675.55	1403.85	409.17	409.17
200 a/w	4664.4	4778.5	4924.2	4804.5	4925.0	5101.4	6464.9	4778.5	4664.4
500 a/w	17 437	18 085	18 353	17 851	18 351	18 343	22 619	17 770	17 437
Shanghai, $Q = 5000$ m									
t	BF	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
200 a/w	27.689	50.952	29.337	29.320	31.743	31.741	36.371	27.976	27.689
500 a/w	4199.1	6492.6	4693.5	4201.8	4879.9	4880.7	5377.8	4433.1	4199.1

bold — the best solution by a *non-brute force* algorithm.

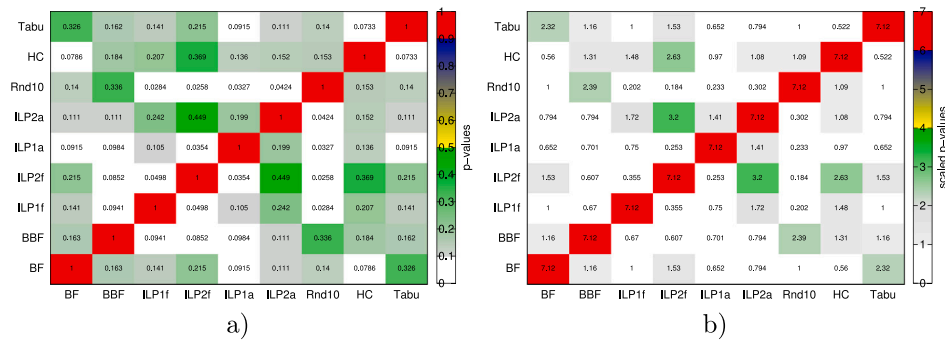


Fig. 6. Algorithm similarity in Student's t -test. (a) p -values, (b) p -values scaled to BF vs Rnd10 p -value. Higher values indicate more similar algorithms.

this algorithm pair has p -value=0.1403. Therefore, it is plausible to assume that algorithm pairs with p -values smaller than 0.1403 may be considered at least as different as BF-Rnd10 pair. The results scaled to p -value=0.1403 as a unit, are shown in Fig. 6b. It can be seen that in most of the cases the algorithms perform at least as differently as {BF,Rnd10} pair. {ILP2a,ILP2f} are the most similarly performing methods. This is intuitively expected because in 75% of tested instances the difference between solutions of ILP2a and ILP2f was less than 2.7%. The next most similarly performing pair is {ILP2f,HC}. Also in this case, in majority of instances results are quite close to each other: in 75% of instances the solution quality difference is below 4.3%. Thus, generally our algorithms are distinct methods performing differently, yet there are instances on which some pairs obtain similar results.

This section can be concluded by observing that the presented algorithms have their own domains of good performance: BBF is good for ports with many large vessels (e.g. Le Havre). ILP2f and ILP2a can

be recommended when traffic is intensive and solution evaluations are costly. Tabu search is good in all cases, but its main limitation is high solution evaluation cost.

7. Conclusions

In this paper we introduced the quay partitioning problem in a stochastic setting (SQPP). A ship traffic model is an element defining SQPP. An important component in solving the SQPP is the evaluation of quay partitions. To evaluate the partitions we used a purposefully designed portfolio of algorithms selected for short runtime and good quality solutions at the chosen time limits. The nature of SQPP solutions was extensively analyzed on the examples of Le Havre and Shanghai traffic models. It appears that at low traffic intensity most of the quay partitions are almost equally good. Only under intensive traffic, can the alternative solutions to SQPP show their superiority and the more intensive the traffic is, the more different the partitions performances are.

Table 14
F(K) for algorithmically built SQPP solutions.

Le Havre, <i>Q</i> =3500 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	38 774	38 776	37 894	38 785	38 791	44 093	38 436	37249
Shanghai, <i>Q</i> =5000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	16571	13184	12275	13576	13576	14396	12628	12255
Rotterdam, <i>Q</i> =4000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	27969	18538	16084	18538	18538	20438	15875	15875
Rotterdam, <i>Q</i> =5000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	22003	14681	12024	14681	14395	17822	11862	11862
Rotterdam, <i>Q</i> =6000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	18 347	10 988	9314.9	10 988	10 991	13 316	9282.7	9202.3
Hamburg, <i>Q</i> =6000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	31878	28009	27011	28009	28331	29393	27908	26892
Hamburg, <i>Q</i> =7500 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	24 981	21 823	21300	21 903	21 782	23 964	22 677	21 941
Hamburg, <i>Q</i> =9000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
1000 a/w	21047	17710	17150	17710	17725	19398	18590	18182
Singapore, <i>Q</i> =10000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
500 a/w	2262	1640	1549	1640	1486	1934	1481	1461
1000 a/w	8674	7343	7193	7343	7137	7855	7273	7154
Singapore, <i>Q</i> =14000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
500 a/w	418.84	116.36	69.76	116.32	54.66	224.73	49.88	46.81
1000 a/w	5082	3903.3	3884.6	3903.6	3881.5	4530.4	4174.3	4174.3
Singapore, <i>Q</i> =20000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
500 a/w	21.112	20.296	20.113	20.298	20.049	20.477	20.033	19.990
1000 a/w	2331.4	1556.5	1438.3	1556.6	1400.4	1846.1	1730.0	1730.0
Singapore, <i>Q</i> =26000 m								
<i>t</i>	BBF	ILP1f	ILP2f	ILP1a	ILP2a	Rnd10	HC	tabu
500 a/w	19.638	19.569	19.555	19.569	19.547	19.579	19.559	19.558
1000 a/w	820.91	310.07	215.06	310.25	180.22	371.05	312.10	222.29

bold — the best solution found.

Under high traffic, solutions that are good with respect to short waiting time are also good at the dispersion of waiting time. Moreover, good partitions depend on the traffic features. For example, a solution based on building berths of equal lengths or as many big berths as possible is generally not the best, yet for specific STMs can provide satisfactory solutions. Although the largest vessel traffic has dominating impact on the patterns of the best quay partitions, dedicating also some quays for shorter vessels give lower dispersion of solution quality. We proposed a set of algorithms to construct a quay partition. Brute force enumeration of all possible quay partitions and evaluating them is to a limited degree doable, e.g., for contemporary terminals like Le Havre or Shanghai. We proposed a number of methods with lower complexity with the goal of applying them in future terminal designs allowing many more berths. It turned out that these methods can be competitive in various domains of application and tabu search seems to be the most versatile.

This study was conducted under certain assumptions, and certainly, does not answer all the quay partitioning challenges. Future work on the quay partitioning problem may cover, e.g.: (i) alternative quay partitioning algorithms, (ii) terminals with many heterogeneous quays, (iii) alternative hybrid vessel berthing organizations, (iv) incorporating costs of berths and yard division reorganization, (v) including further seaside and landside details in the terminal operations model. We

believe that the results and methods presented in this paper will help both researchers and practitioners designing container terminals of the future.

CRedit authorship contribution statement

Jakub Wawrzyniak: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Maciej Drozdowski:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Éric Sanlaville:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Funding sources

Éric Sanlaville was partially supported by the Normandy region and the European ERDF through CLASSE2 project, France. Computational

experiments were conducted in Poznań Supercomputing and Networking Center. Funding provided by Poznań University of Technology, Poland and Université Le Havre Normandie, France. Funding sources had no involvement in preparing this text.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3), 615–627.
- Bierwirth, C., & Meisel, F. (2015). A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3), 675–689.
- Błądek, I., Drozdowski, M., Guinand, F., & Schepler, X. (2015). On contiguous and non-contiguous parallel task scheduling. *Journal of Scheduling*, 18(5), 487–495.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., & Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison. *Transportation Research Part E: Logistics and Transportation Review*, 47(4), 461–473.
- Cheong, C., Tan, K., Liu, D., & Lin, C. (2010). Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180(1), 63–103.
- Coffman, E. G., & Garey, M. R. (1993). Proof of the 4/3 conjecture for preemptive vs. nonpreemptive two-processor scheduling. *Journal of the ACM*, 40(5), 991–1018.
- Cordeau, J.-F., Laporte, G., Legato, P., & Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem. *Transportation Science*, 39(4), 526–538.
- Dulebenets, M., Kavoozi, M., Abioye, O., & Pasha, J. (2018). A self-adaptive evolutionary algorithm for the berth scheduling problem: Towards efficient parameter control. *Algorithms*, 11(7), 100.
- El Hammouti, I., Lajjam, A., & El Merouani, M. (2020). Comparison of planning models for dynamic berth allocation problem using a sailfish-based algorithm. *Procedia Computer Science*, 176, 3112–3120.
- Ewald, R. (2010). Automatic algorithm selection for complex simulation problems. In *Fakultät für Informatik und Elektrotechnik* (Ph.D. thesis), Universität Rostock.
- Gharehgozli, A., Zaerpour, N., & de Koster, R. (2020). Container terminal layout design: transition and future. *Maritime Economics & Logistics*, 22, 610–639.
- Gokkus, U., & Yildirim, M. S. (2015). Determination of the optimum berth number of the izmir seaport by deterministic and probabilistic methods based on artificial neural networks. *International Journal of Hybrid Information Technology*, 8(9), 11–24.
- Groenveld, R. (2006). *Ship traffic flow simulation study for port extensions, with case extension port of rotterdam*. PIANC Congress Estoril.
- Hansen, P., & Oğuz, C. (2003). A note on formulations of the static and dynamic berth allocation problems. *Technical Report G*, 200, 3–30.
- Hansen, P., Oğuz, C., & Mladenović, N. (2008). Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191(3), 636–649.
- Huang, W.-C., & Wu, S.-C. (2005). The estimation of the initial number of berths in a port system based on cost function. *Journal of Marine Science and Technology*, 13(1), 35–45.
- Kadioglu, S., Malitsky, Y., Sabharwal, A., Samulowitz, H., & Sellmann, M. (2011). Algorithm selection and scheduling. In *Proceedings of the 17th international conference on principles and practice of constraint programming*.
- Kastner, M., Kämmerling, N., Jahn, C., & Clausen, U. (2020). Equipment selection and layout planning - literature overview and research directions. In C. Jahn, W. Kersten, & C. M. Ringle (Eds.), *Proceedings of the hamburg international conference of logistics: vol. 30, Data science in maritime and city logistics: data-driven solutions for logistics and sustainability* (pp. 485–519).
- Korekane, S., Nishi, T., Tierney, K., & Liu, Z. (2024). Neural network assisted branch and bound algorithm for dynamic berth allocation problems. *European Journal of Operational Research*, 319(2), 531–542.
- Kotthoff, L. (2014). Algorithm selection for combinatorial search problems: A survey. *AI Magazine*, 35(3), 48–60.
- Kovač, N. (2017). Survey metaheuristic approaches for the berth allocation problem. *Yugoslav Journal of Operations Research*, 27(3), 265–289.
- Li, M. W., Hong, W. C., Geng, J., & Wang, J. (2017). Berth and quay crane coordinated scheduling using multi-objective chaos cloud particle swarm optimization algorithm. *Neural Computing and Applications*, 28(11), 3163–3182.
- Lim, A. (1998). The berth planning problem. *Operations Research Letters*, 22(2), 105–110.
- Liu, B., Li, Z.-C., Sheng, D., & Wang, Y. (2021). Integrated planning of berth allocation and vessel sequencing in a seaport with one-way navigation channel. *Transportation Research Part B: Methodological*, 143, 23–47.
- Lyu, X., Lalla-Ruiz, E., & Schulte, F. (2025). The collaborative berth allocation problem with row-generation algorithms for stable cost allocations. *European Journal of Operational Research*, Online . (Accessed 7 March 2025).
- Ma, X., & Pu, X. (2025). Low-carbon berth allocation: An analysis of the effectiveness of an enhanced multi-objective artificial bee colony algorithm based on a case study. *Ocean & Coastal Management*, 261, Article 107529.
- Maritime and Port Authority of Singapore (2024). Tuas port – a smarter and greener port. <https://www.mpa.gov.sg/maritime-singapore/port-of-the-future>. Online. (Accessed 7 March 2025).
- Olba, X. Bellsolá, Daamen, W., Vellinga, T., & Hoogendoorn, S. P. (2017). Network capacity estimation of vessel traffic: An approach for port planning. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 143(5), 1–9.
- Olba, X. Bellsolá, Daamen, W., Vellinga, T., & Hoogendoorn, S. P. (2018). State-of-the-art of port simulation models for risk and capacity assessment based on the vessel navigational behaviour through the nautical infrastructure. *Journal of Traffic and Transportation Engineering (English Edition)*, 5(5), 335–347.
- O'Mahony, E., Hebrard, E., Holland, A., Nugent, C., & O'Sullivan, B. (2008). Using case-based reasoning in an algorithm portfolio for constraint solving. In *Irish conference on artificial intelligence and cognitive science*.
- Ö. Uğurlu, E. Yüsekilyildiz, & E (2014). Simulation model on determining of port capacity and queue size: A case study for BOTAS ceylan marine terminal. *TransNav. The International Journal on Marine Navigation and Safety of Sea Transportation*, 8(1), 143–150.
- Piechowiak, K., Drozdowski, M., & Sanlaville, Éric. (2022). Framework of algorithm portfolios for strip packing problem. *Computers & Industrial Engineering*, 172, Article 108538.
- Rice, J. (1975). *The algorithm selection problem: Technical report 99*, Purdue University, Computer Science Technical Reports.
- Rodrigues, F., & Agra, A. (2022). Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey. *European Journal of Operational Research*, 303(2), 501–524.
- Schepler, X., Balev, S., Michel, S., & Sanlaville, É. (2017). Global planning in a multi-terminal and multi-modal maritime container port. *Transportation Research Part E: Logistics and Transportation Review*, 100, 38–62.
- Scott, D., Taylor, D., El-Solh, S., & Elliott, T. (2016). Port simulation modelling and economic assessment. *Journal of Marine Science and Engineering*, 4(1).
- Shanghai International Port Group (2025). Our services. terminal handling. <https://en.portshanghai.com.cn/TeminalHanding/index.jhtml>. Online. (Accessed 7 March 2025).
- Smith-Miles, K. A. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1).
- Souravlias, D., Parsopoulos, K. E., Kotsireas, I. S., & Pardalos, P. M. (2021). *Algorithm portfolios advances, applications, and challenges*. Cham: Springer.
- Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: a literature update. *OR Spectrum*, 30, 1–52.
- Tang, H. (2024). Research on the development and policies of the port of Singapore harbor. *Advances in Economics, Management and Political Sciences*, 80, 275–282.
- Thoresen, C. A. (2003). *Port designer's handbook*. London: Thomas Telford.
- Vernimmen, B., Dullaert, W. E. H., & Engelen, S. (2007). Schedule unreliability in liner shipping: Origins and consequences for the hinterland supply chain. *Maritime Economics & Logistics*, 9, 193–213.
- Wang, C., Wang, Q., Xiang, X., Zhang, C., & Miao, L. (2025). Optimizing integrated berth allocation and quay crane assignment: A distributionally robust approach. *European Journal of Operational Research*, 320(3), 593–615.
- Wawrzyniak, J., Drozdowski, M., É, Sanlaville, Pigné, Y., & Guinand, F. (2024). Quay partitioning problem. *International Transactions in Operational Research*, 31(3), 1554–1584.
- Wawrzyniak, J., Drozdowski, M., & Sanlaville, É. (2020). Selecting algorithms for large berth allocation problems. *European Journal of Operational Research*, 283(3), 844–862.
- Wawrzyniak, J., Drozdowski, M., & Sanlaville, É (2021). Container ship traffic model for simulation studies - additional resources. <http://www.cs.put.poznan.pl/mdrozdowski/stm/>. (Accessed 20 January 2022).
- Wawrzyniak, J., Drozdowski, M., & Sanlaville, É. (2022). A container ship traffic model for simulation studies. *International Journal of Applied Mathematics and Computer Science*, 32(4), 537–552.
- Xiang, X., & Liu, C. (2021). An almost robust optimization model for integrated berth allocation and quay crane assignment problem. *Omega*, 104, Article 102455.
- Yue, M., Wang, Y., Guo, S., Dai, L., & Hu, H. (2025). A multi-objective optimization study of berth scheduling considering shore side electricity supply. *Ocean & Coastal Management*, 261, Article 107500.