TESIS DE LA UNIVERSIDAD
DE ZARAGOZA

2023 148

Lygia Bronneberg Vélez

Port Terminal Appointment Scheduling Problem

Director/es
Çagri Gürbuz, Mustafa

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Servicio de Publicaciones

ISSN 2254-7606

Tesis Doctoral

PORT TERMINAL APPOINTMENT SCHEDULING PROBLEM

Autor

Lygia Bronneberg Vélez

Director/es

Çagri Gürbuz, Mustafa

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Logística y Gestión de la Cadena de Suministro

2023

Port Terminal Appointment Scheduling Problem

by

Lygia Bronneberg Vélez

Bachelor of Naval Architecture and Marine Engineering, Universidade de São Paulo (2012)
Master of Transport Engineering, Universidade Federal do Rio de Janeiro (2016)

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the

MIT-ZARAGOZA INTERNATIONAL LOGISTICS PROGRAM

at the

ZARAGOZA LOGISTICS CENTER,

a research institute affiliated with the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
and the
UNIVERSITY OF ZARAGOZA

December, 2022

Author ....
Lygia Bronneberg Vélez
Ph.D. Candidate, MIT-Zaragoza International Logistics Program

Certified by ....
Mustafa Çagri Gürbuz, Ph.D.
Professor, MIT-Zaragoza International Logistics Program
Research Affiliate, MIT Center for Transportation and Logistics
Thesis Supervisor

Certified by ....
Yasel Costa Salas, Ph.D.
Professor and Ph.D. Program Director, MIT-Zaragoza International Logistics Program
Research Affiliate, MIT Center for Transportation and Logistics

Port Terminal Appointment Scheduling Problem

by

Lygia Bronneberg Vélez

Submitted to the MIT-Zaragoza International Logistics Program in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Zaragoza Logistics Center, a research institute affiliated with the Massachusetts Institute of Technology and the University of Zaragoza.

ABSTRACT

The constant increase in overseas transportation in the past few years have pushed terminal operators to search for solutions that can increase their efficiency. One current example is the resolution published by the Brazilian National Agency of Oil in July of 2022, which reinforced the importance of having a structured appointment scheduling methodology to organize ship/shore operations.

This research has three main goals: the first one is framing the appointment scheduling problem under different perspectives to help the designing process of such resolutions. The second one is providing appointment scheduling models that can support terminals in their optimization process. The third one is exploring insights about the value of information, different scheduling time frames, coordination between scheduling and operational teams, different agenda congestion profiles, uncertainty levels, berthing scheduling rules, among others.

The challenge is finding the optimized appointment plan that can maximize terminals profits, considering particularities of each operation request, uncertainties in arrival and processing times of ships, costs/earnings associated with contractual agreements and the berthing sequencing rule used by the operational teams.

In one hand there will be contributions on the managerial side through ideas that can enhance terminals performance. In the other hand there will be academic contributions through proposing appointment scheduling models that incorporate stochasticity in parameters and considers arrivals as being endogenous variables, under different agenda overlap profiles. Therefore multiple heuristics will be proposed, dealing with stochastic, integer and non-linear appointment scheduling problems (SINP). They consider clients requests, contractual agreements, delays/ processing times distributions, and a pre-defined berthing sequence rule as inputs. Based on how profitable are the operations, it defines which ships are accepted/rejected to operate, as well as the appointment date that they are expected to happen.

Due to dimensionality issues, a partition methodology is proposed called "Cluster First, Schedule Second" to reduce resolution time. The main problem is decomposed into smaller ones which are then sequentially solved via Sample Average Approximation, having the scheduling of each cluster impacting further ones. The results from the optimization models are also tested in a discrete event simulation environment that reproduces multiple restrictions encountered in congested terminals.

Finally a set of ten research questions are proposed which will guide all the experimentation process used to test multiple topics about the appointment scheduling problem of port terminals. Highlighting some of the conclusions, results show that specially overlapped terminals can have significant improvements in profit by using solvers such as the ones that will be presented. Also, giving incentives to customers could be studied to get more up front information about the operation as well as to increase flexibility in the available days. Answering clients statically showed better results, as the terminal is able to take the decision with full information. In case clients value dynamic answers, a suggestion would be offering it as a premium service to reduce the overall impact. In terms of berthing rules, FIFO presented good results for terminals with congested agendas, while by schedule rule was better in low overlap situations. In case of simultaneous arrivals, prioritizing by smallest deviations is recommended. Additionally, one surprising result is that uncertainties in arrivals can, in some cases be beneficial, but accepting time windows instead of a scheduled date is not.

RESUMEN

El constante aumento del transporte marítimo de los últimos años ha llevado a los operadores de terminales marítimos a investigar nuevas soluciones que aumenten su rendimiento. Un ejemplo actual es la resolución publicada por la Agencia del Petróleo de Brasil en julio de 2022, en la que destaca la importancia de contar con una metodología de programación de citas estructurada para organizar las operaciones buque-tierra.

Esta investigación tiene tres objetivos principales: el primero de ellos es abordar el problema de programar citas desde diferentes perspectivas para ayudar al proceso de diseño de tales soluciones. El segundo es facilitar modelos de programación de citas que puedan ayudar a los terminales en sus procesos de optimización. El tercero es estudiar diferentes planteamientos sobre el valor de la información, diferentes plazos de programación, coordinación entre los equipos operativos y de programación, diferentes perfiles de congestión de la agenda, niveles de incertidumbre y normas de programación de atraque, entre otros.

El reto consiste en encontrar un plan de cita optimizado que permita maximizar las ganancias de los terminales, considerando particularidades de cada solicitud de operación, incertidumbres en plazos de llegada y en procesamiento de los buques, los costes y ganancias vinculados a los contratos y la norma de secuenciación de atraques que utilizan los equipos operativos.

Por un lado, existirán contribuciones en la parte de gestión a través de ideas que pueden impulsar el rendimiento de los terminales. Por otro lado, existirán aportaciones académicas a través de propuestas de modelos de programación de citas que incorporan la aleatoriedad en parámetros y consideran las llegadas como variables endógenas, conforme a diferentes perfiles de solapamiento de la agenda. Por tanto, se propondrán varias heurísticas, que abordarán los problemas de programación de citas aleatorios, enteros y no lineales (SINP, por sus siglas en inglés). Tienen en cuenta las solicitudes de los clientes, los acuerdos contractuales, las distribuciones de plazos de retraso / procesamiento y la norma predefinida de secuencia de atraques como insumos. En función de lo rentable que sean las operaciones, se define qué buques se aceptan o rechazan para operar, así como la fecha de la cita que se espera que se produzca.

Debido a cuestiones de dimensionalidad, se propone una metodología de descomposición llamada "Cluster First, Schedule Second" (Primero agrupar, luego programar) con el fin de reducir el plazo de resolución. El problema principal se descompone en otros más pequeños que se resuelven de manera secuencial mediante la aproximación de la media muestral, de manera que la programación de cada grupo afecta a los siguientes. Los resultados de los modelos de optimización también se evalúan en un entorno de simulación de acontecimientos discreto que reproduce varias restricciones presentes en terminales saturados.

Por último, se propondrá un conjunto de diez preguntas de investigación que guiarán todo el proceso de experimentación utilizado para probar diferentes temas sobre el problema de programación de citas de terminales portuarios. Entre las conclusiones, cabe destacar que los resultados muestran que los terminales, especialmente saturados, pueden lograr mejoras significativas en beneficios con medidas como las que se presentarán. También, se puede estudiar dar incentivos a los clientes para obtener más información por adelantado sobre la operación, así como aumentar la flexibilidad en la disponibilidad de días. Responder a los clientes de forma estática dio mejores resultados, puesto que el terminal puede tomar la decisión con toda la información. En caso de que los clientes valoren respuestas dinámicas, una sugerencia podría ser ofrecerles un servicio superior para reducir el impacto general. En términos de normas de atraque, el método FIFO presentó buenos resultados en el caso de terminales con agendas congestionadas, mientras que la norma por programación fue mejor en situaciones con poco solapamiento. En el caso de llegadas al mismo tiempo, se recomienda priorizar en función de las desviaciones más pequeñas. Además, un resultado sorprendente es que las incertidumbres en las llegadas pueden, en algunos casos, ser beneficiosas, pero aceptar ventanas de tiempo en lugar de una fecha programada no lo es.

Thesis Supervisor:
Mustafa Çagri Gürbuz, Ph.D.
Professor, MIT-Zaragoza International Logistics Program
Research Affiliate, MIT Center for Transportation and Logistics

Acknowledgements

I especially thank my advisor Professor Mustafa Çagri Gürbuz for his dedication and guidance in all the stages of my thesis. I gratefully acknowledge the support from Massachusetts Institute of Technology (MIT) and Zaragoza Logistics Center (ZLC), including all faculty and staff, the professors from my defense committee, my colleagues and our PhD Director Professor Yasel Costa Salas. I also appreciate my managers and team (former and actual ones) and all the other departments that facilitated this process. To the partner terminal operator, my thanks for supporting this research and fully opening doors, providing not only the data but specialized knowledge and staff support.

I am also grateful to all my friends that cheered for me the whole process, with a special thanks to my dearest Nines and Pablo, which since the beginning welcomed me in their homes and hearts, making this experience unforgettable. To my amazing family, team Bronneberg and team Simon, for believing and supporting me all way long. To Glori, my adorable Spanish guide and travel partner, Spain was way more beautiful knowing it through your eyes. To Juan and Marnia, for all the support and care that was offered with opened arms. To my amazing Opa and Oma, for all their long working hours under the sun so their future generations could have access to education, I will always admire and be grateful for your faith and bravery. To Myrian, my sweet middle sister that is always by my side, no matter the distance and how hard things get. To Paula, my lovely little sister for being my rock, smiling and cheering me every time I needed encouragement. To my mother Agnes, for making all of this possible with all her hard work in raising this family with unlimited love and effort, I love you three unconditionally. And to one of the most generous people that I have met, my husband Rafael, thanks for all the nights awake dedicated to studying with me even being a thousand miles away in another time zone, I can't wait to see all the beautiful achievements that we will conquer together.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Context and Motivation

The constant increase in overseas transportation in the past few years have pushed terminal operators to search for solutions that can increase their efficiency. According to UNCTAD (2021), over 80% of the international trade volume is carried by sea, which shows the importance of ships and terminals in the global economy. Due to this high level of product flow, many terminals face congestion of ships in their facilities. In fact, the same study shows that a tanker waits in average 64 hours to start performing a loading/unloading operation in a Brazilian terminal. This number stands out when compared to tankers commercial expectation of performing a full load and unload operation in 72 hours.

There are three main figures involved in a ship/shore operation: cargo owner, shipowner and terminal operator. Those operations are ruled by contracts that define the financial and performance responsibilities of each party. One important clause sets how penalties will be applied in case of delays. Terminals are specially susceptible to this issue as it is the party that absorbs uncertainties from all the rest of the supply chain. According to Notteboom (2006), more than 90% of the delays are attributable to port access and terminal operations. Furthermore, delays in the arrival of ships, bureaucracy and adverse environmental conditions also contribute to lateness in the process, which may have a "domino effect" in future operations.

In the case a ship overstays in the port due to someone else's responsibility, an official claim for demurrage payment can be posed which is proportional to the number of days beyond the scheduled due date (Sun et al., 2021) agreed between all parties. Overstaying is a consequence of delayed starts and/or extended processing times.

In July of 2022 a new resolution from the Brazilian National Agency of Oil, ANP (2022), was published which regulates how oil terminals should offer their services reinforcing the need of a structured appointment scheduling methodology. This is one of the many initiatives to maximize the utilization of oil terminals and pipelines, by improving the access for any client interested in supplying the market. The fact that this is an ongoing problem of high complexity was a great motivator that triggered the current research.

1.2 Objective and Contributions

There are three main goals: the first one is framing the appointment scheduling problem under different perspectives to help the designing process of resolutions such as ANP (2022). The second one is providing appointment scheduling models that can support terminals in the optimization process imposed by such resolutions. The third one is exploring insights about the value of information, different scheduling time frames, coordination between scheduling and operational teams, different agenda congestion profiles, uncertainty levels, berthing scheduling rules, among others.

The literature encountered approaching the ship/shore operation planning is mostly focused on allocation problems which defines what is the best strategy for the assignment of berths (Golias, 2011), cranes (Rodrigues and Agra, 2022), tanks, trucks (Phan and Kim, 2016), yard space which should be used in the operations. Some literature also incorporates the scheduling facet, defining the berthing sequence and the dates that should be considered to optimize performance. Those type of problems normally consider the arrival pattern of ships as a given, and that terminals have basically to mold their decisions in the best way possible to accommodate the operations. This research focus on the previous planning step, on which appointments are being designed. The terminal is an active player on the decision of which operations should be accepted and when they should be scheduled, which directly impacts how those arrivals will happen, configuring an endogenous profile.

The challenge is finding the optimized appointment plan that can maximize terminals profits, considering particularities of each operation request, uncertainties in arrival and processing times of ships, costs/earnings associated with contractual agreements and the berthing sequencing rule used by the operational teams.

In one hand there will be contributions on the managerial side through insights that can enhance terminals performance and quality of ship/shore services. In the other hand there will be academic contributions through proposing appointment scheduling models to the terminal problem that incorporates both stochasticity, as well as considers arrivals as endogenous variables under different agenda overlap profiles.

1.3 Methodology

As one of the goals is to study the appointment scheduling problem under different perspectives, mathematical models will be proposed, being the main (and the first one presented) based on the frame of the resolution ANP (2022). The others will be variations of the main one exploring particularities of distinct settings.

The appointment scheduling problem approached in this thesis is stochastic, integer and non-linear (SINP). It consider clients requests, contractual agreements, delays/ processing times distributions, and a pre-defined berthing sequence rule as inputs. Based on how profitable are the operations, it defines which ships are accepted/rejected to operate, as well as the appointment date that they are expected to happen. Heuristics based on Sample Average Approximation (SAA) are proposed, which is a technique used to deal with the stochastic variables. For theoretical purpose, a partially linearized version of the model is also presented.

Due to dimensionality issues faced by this combinatorial problem a partitioning methodology is proposed called “Cluster First, Schedule Second”, inspired by the “Cluster First, Route Second” applied to vehicle routing problems (Street et al., 2015). The idea is breaking the master problem into smaller clusters allowing the solution for large scale instances, which are then sequentially solved via SAA, having the scheduling of each cluster impacting further ones.

Considering that the model was proposed under simpler assumptions about port access and operational rules, a discrete event simulation reproducing multiple restrictions encountered in congested terminals is proposed (resource and berthing/unberthing limitations, operational setups, sequencing rules, among others). This model, developed in ARENA environment, tests the appointment solutions from the optimization model in order to identify possible differences in the estimation of operations starting time and expected profit from a simpler setting to a more complex one.

1.4 Research Questions

The experimentation phase is then proposed guided by a set of research questions defined as follows:

1. *What is the best clustering technique that can be used in the partitioning methodology? And what is the impact of using those approximation methods in the total profit?*

2. *What is the value generated by the optimization model when compared to manually performed appointments?*
3. *Which differences can be acknowledged in the operational start term when comparing a simple terminal to a complex one?*
4. *What is the value of having the volume information previous to the appointment scheduling process?*
5. *What is the value of having more flexibility in the availability dates?*
6. *What is the impact of making decisions without having full information about future requests (dynamic versus static scheduling)? What is the impact of allowing past corrections in the dynamic scheduling procedure?*
7. *What is the impact of not having a scheduled date defined?*
8. *Which berthing rule provides higher expected profits?*
9. *What is the impact of having uncertainties, in arrival and processing times?*
10. *What is the impact of not having coordination between schedulers and operational teams around the berthing rule?*

1.5 Thesis Structure

The following chapters will be divided as follows: **Chapter 2** presents the literature review about ship/shore operations and appointment scheduling systems used in the development of this research. **Chapter 3** analyzes practical issues about ship/shore operations and their contract settings. **Chapter 4** presents different versions of the terminal appointment scheduling problem as well as their models, the methodology used to solve large instances and the simulation model used to verify the appointment solutions under complex settings. **Chapter 5** presents the design of the experiments as well as the results of each research question. **Chapter 6** summarizes the conclusions highlighting some possibilities of future research. **Chapter 7** has all the annexes used as complementary reading.

2 Literature Review

Scheduling applied to transportation is an old and heavily studied field and is recently gaining new attention due to crossovers with currently debated issues, such as contractual design, collaboration and coordination between stakeholders, and the application of new methodologies, including machine learning and data-driven techniques. This chapter reviews the bibliography covering those problems through different perspectives: transportation mode; level of decision making; uncertainties in parameters; type of scheduling problem; optimization variables; objective function modelling; type of restrictions and solution methodologies.

Even having seaborne shipping as a major component of international trade, historically there has been less academic interest in that mode when compared, for example, to road transportation. This low attention can be explained as a compounded result of: larger variability of problem structures (making solutions specific instead of standardized to the industry), highly uncertain environment and a conservative industry resistant to incorporating new solutions.

Those explanations also elucidate the reason why container problems are more explored in the literature than liquid bulk ones (Umang et al., 2011). The cargo itself is aggregated in standard units while in bulk is a fluid, the operation time depends basically on the number of containers and crane capacity, while in bulk it also depends on the size of the ships, their pumps, the types of product and berth infrastructure. In the bulk setting there is also the possibility of interface between products, depending on the types of products carried by the ships and the sequence they are operated (Magatão et al., 2012; Cafaro and Cerdá, 2008). Even though bulk problems have those inherent complexities, the fact that regulatory agencies are demanding new strategies to enhance efficiency (ANP, 2022) and that small improvements can guarantee expressive savings, shows that there is opportunity and industry interest in the development of solutions.

There are three levels of decision making on which scheduling can be seen in the literature applied to the maritime area: strategic level (e.g. cargo owners defining which terminals to operate with, terminals selecting which operations to accept/reject and stakeholders designing contract conditions), tactical level (e.g. ships/cargo owners defining routes and terminals agreeing on the best scheduling plan) and operational level (e.g. ships defining speed adjustments policies and terminals sequencing the use of berth/cranes and yard). At the strategic level, for example, Sun et al. (2021) defines contractual laytime conditions between cargo owners and shipowners while Maisiuk and Gribkovskaia (2014) works on fleet size definition using simulation methodology. At the tactical level, Maneengam and Udomsakdigool (2021) approaches the ship routing and scheduling problem from a sustainable perspective, and Jiang et al. (2012) explores strategies for yard management in container ports. At the operational level, Correcher et al. (2019), for example, provides a survey about berth allocation in terminals with irregular layouts and Sun et al. (2021) explores the speed adjustment decisions made by shipowners.

Most of the literature can be located in the charterer/cargo owner strategic and tactical level and also in the shipowner/terminal operational level (Cankaya et al., 2019). There is limited amount of papers approaching problems from the terminal perspective at the strategic/tactical level, which is exactly how this paper stands its contribution. Solutions in that context can be found in papers such as Fernández and Munoz-Marquez (2022); Imai et al. (2014) which explore the deterministic version of the berth template problem, defining which clients should be accepted, as well as when, how and where ships with cyclic arrivals should operate.

The ship/shore operation is the result of an agreement between three stakeholders: shipowners, cargo owners and terminal operators. There are only a few papers that focus on the contractual negotiation and the implications of its execution, such as Sun et al. (2021). Normally in those agreements there are clauses defining what is expected in terms of operational time and service level, as well as penalization fees in case those expectations are not met (one important penalty is called demurrage fee, applied to ship overstaying cases). Schofield (2015) is referred for a comprehensive review on the legal terms of those contracts and a detailed explanation of how demurrage fee is applied.

Those fees are significant when compared to transportation earnings and ships overstaying in terminals are a recurrent problem, specially when considering congested systems. Ribeiro et al. (2016) and Barros et al. (2011), for example, explore the berth allocation model for bulk terminals accounting for demurrage and despatch fees (which is the opposite of demurrage, being a bonus fee when ships operate in less time then expected). Parra (1995) and Huang and Karimi (2006) proposes a scheduling system for ship/shore operations focusing on minimizing those costs and Liang et al. (2011) analyzes the influence of demurrage on yard utilization. Ships are expected to operate according to an appointment agreed between all stakeholders.

The appointment scheduling problem (ASP) is defined by customers requesting services that should be arranged to future slots of time given a booking horizon. Those clients can be patients requiring medical care, chemotherapy sessions, hotel and car rental arrangements or even loading/unloading procedures in terminals. According to Gocgun and Puterman (2014), this type of challenge is conceptually different from allocation problems (AP), which requests are either served or rejected immediately, meaning that they are not appointed to operate in future periods of time. Allocation problems normally follow pre-defined arrival patterns (exogenous arrivals), while appointment scheduling problems normally are the ones that organize those patterns (called endogenous arrivals). Having endogenous arrivals means that any change in the appointment decision automatically impacts in changes in the arrivals and start of operations.

Those problems might focus on optimizing local variables, such as queuing time, resource utilization and service level, or global variables such as profits and costs. There might be constraints on the capacity/inventory level (Wang, 1993; Barros et al., 2011), time-windows and due-dates (Gocgun and Puterman, 2014) and environmental restrictions (Le Carrer et al., 2020; Mauri et al., 2016). Some special conditions might also appear such as the possibility of no-shows (Zacharias and Pinedo, 2014), cancellation, overbooking (Cao, 2009; Wasesa et al., 2021; Ala and Chen, 2022) and different types of customers/priorities (Golias et al., 2009; Alstrup et al., 1986).

Allocation problems are vastly explored in the maritime literature, specially under the topics of berth allocation problems (BAP) and berth template problems (BTP). The main difference between those is the fact that berth template solves a BAP for cyclical arrivals, which are very common in terminals that operate container liners. For a comprehensive review on BAP the reader is referred to Rodrigues and Agra (2022), Bierwirth and Meisel (2015) and the specific paper about bulk ports from Umang et al. (2011). For BTP it is suggested the papers from Imai et al. (2008a), Moorthy and Teo (2006) and Fernández and Munoz-Marquez (2022).

Terminals can operate a single berth or multiple ones which can have discrete, continuous or hybrid compositions (Umang et al., 2013; Mauri et al., 2016). There are some interesting literature exploring the BAP integration with yard and/or crane scheduling (Wang et al., 2018; Lee et al., 2007; Imai et al., 2008b) and pipeline sequencing (Cafaro and Cerdá, 2008; Magatão et al., 2012). There are also papers that add the scheduling factor (Zhen and Chang, 2012; Xiang et al., 2017), called berth scheduling problems, defining which is the optimal berthing sequence and when operations should happen, normally given a pre-defined arrival pattern.

The appointment scheduling problem was also explored in the maritime context, such as in Sabria and Daganzo (1989), which used queuing theory to propose predictions about expected delays on systems with ship arrival deviations, or in papers integrating port performance with truck appointment systems reviewed by Abdelmagid et al. (2022). Following that topic, Guan and Liu (2009) applied a multi-server queuing model to analyze gate congestion and truck waiting costs. Also, Chen et al. (2013) proposed a genetic algorithm that optimises the hourly quota of entry appointments with a non-stationary M(t)/Ek/c(t) queuing model. Recently the environmental perspective has also gained attention considering carbon emissions in port area (Fan et al., 2019; Schulte et al., 2017).

The most attention to appointment scheduling systems has been given in the medical setting and the first papers date back to the 60's when those were being introduced in hospitals and general practices (John, 1964; Jackson, 1964). Gocgun and Puterman (2014), for example, proposed a model to solve the appointment scheduling of patients in need of chemotherapy sessions, focusing on minimizing the deviation between the scheduled dates and the target dates provided by protocoled timetable that can guarantee maximum efficacy of the treatment. Shehadeh (2019) considered an outpatient colonoscopy scheduling problem, checking the impact of pre-procedure bowel preparation quality on the variability of the colonoscopy duration. The paper written by Mandelbaum et al. (2020) also approached an appointment problem for patient surgeries, with multiple servers and uncertainties in parameters using a data-driven robust optimization approach. Chen and Robinson (2014) combined routine patients with last-minute call patients incorporating random and heterogeneous service times and no-show rates and ancillary physician tasks. For a comprehensive understanding of challenges and opportunities on the appointment scheduling in health care the reader is referred to Gupta and Denton (2008).

The appointment scheduling problem is also found under the name of booking problem (You, 2008; Xing et al., 2019; Alstrup et al., 1986), reservation systems (Liu et al., 2015; Wang and Wang, 2019; Gerchak et al., 1996; Du and Larsen, 2017; Lindberg, 2007) and advance scheduling (Gocgun and Puterman, 2014).

Those problems can have parameters that are either deterministic or stochastic, with uncertainties normally considered in the arrivals and processing times. In the deterministic setting Phan and Kim (2016), for example, brings an interesting collaboration scheme between trucking companies and terminal operators, in an iterative methodology that defines preferred appointment windows giving information about expected queues, which showed to be robust to real cases unexpected events. Lalla-Ruiz et al. (2012) approaches the sequencing of traffic in a waterway considering the Yangtze Estuary in China as a case study. They propose a hybrid metaheuristic that incorporates Tabu Search with Path Relinking which computational experience outperformed other techniques commonly used in the literature. The paper from El-Kholany et al. (2022) proposes a constrained clustering algorithm to assign operations into time-windows using Answer Set Programming, which is a form of declarative programming oriented primarily towards NP-hard searching problems. A broad review of past literature in deterministic scheduling problems can be found in papers such as Graham et al. (1979) and Dempster et al. (1981).

Going towards stochastic papers, some of them consider uncertainties only in the arrival times, such as the paper from Umang et al. (2013) that studies a dynamic hybrid BAP in bulk ports minimizing the total service times of vessels and Moorthy and Teo (2006) that models the BTP as a rectangle packing problem on a cylinder. There are also papers that consider only uncertainties in the processing times, with deterministic arrival patterns (or that consider the arrivals exactly when operations are scheduled). The paper from Sadghiani and Motiian (2021) is an example, that explores those uncertainties by training a Deep Neural Network in an appointment scheduling problem context. Zhen (2015) also proposes a robust approach to deal with those uncertainties when limited information about probability distributions are available. Karafa et al. (2013) and Golias (2011) approach with stochastic handling times a bi-objective BAP model.

There are also papers that consider both stochasticity in arrivals and processing times such as the one written by Zhen et al. (2011) which models a berth allocation model that minimizes penalty costs of deviations from the initial schedule. Golias et al. (2014) also deals with the BAP with uncertainties in arrival and processing times although by minimizing the average and the range of the total service times. Chen and Robinson (2014) studies appointment scheduling in a medical center for a combination of routine patients (who book well in advance) and last-minute patients, considering no-show possibilities. All the appointment scheduling problems highlighted above consider arrivals as an exogenous variable.

Due to the combinatorial nature of scheduling problems, specially in cases of stochastic settings, approximations and/or heuristics/metaheuristics are used to speed up the solution process (Cankaya et al., 2019). As an example, Cunha et al. (2020) defines a mathematical model and a heuristic based on the iterated local search metaheuristic for rescheduling pipe-laying support vessels with wells activities constraints, which resulted in faster and close to optimal solutions. Umang et al. (2013) propose a heuristic method based on squeaky wheel optimization to solve a dynamic berth allocation problem in bulk ports. Mansourifard et al. (2018) proposes a heuristic policy for the sequencing of surgeries based on the newsvendor cost using real hospital data for the analysis, which showed to outperform common heuristics in practice (such as ordering the surgeries based on the variance of surgery duration and requesting all patients to be present in the beginning of the day). Mak et al. (2014) focus in the sequencing of appointments using inventory approximations through a two stages solution on which determining time allowances as buffers against random job duration is considered similar to selecting inventory levels as buffers to accommodate random demand in a supply chain. They find out that the heuristic proposed is close to optimal and that computational time is reduced substantially when compared to other approaches.

Another commonly used methodology to approximate stochastic scenarios is the sample average approximation (SAA) method studied in Verweij et al. (2003) and Mancilla and Storer (2012). In this technique the expected objective function of the stochastic problem is approximated by a sample average estimation derived from a random sample, also called Monte Carlo Simulation or Numerical Simulation. This technique was also applied to the appointment scheduling context by Sadghiani and Motiian (2021), Mak et al. (2014) and in Ho and Lau (1992). In the last reference, the goal is to minimize the expected cost of idle doctors and patients waiting, by using various scheduling rules. They consider deterministic endogenous arrivals.

Decomposition/partitioning methods are also very common, with the objective of breaking the master problem into smaller ones that can be solved faster with small deviations from the optimal results. As an example, El-Kholany et al. (2022) uses decomposition with data mining methodologies to solve a job-shop scheduling problem. Sun and Batta (1996) also approach the same topic through another decomposition method called divide and conquer on which the job shop is first decomposed into cells then a shop scheduling problem is solved through iteratively defining, solving and coordinating cell scheduling problems. They also compare scheduling with different dispatching rules, with and without decomposition, concluding that breaking the problem into smaller ones is computationally effective and brings great results. Street et al. (2015) revisit another decomposition method used on vehicle routing problems called “Cluster first, Route second” on which groups using k-medians like clustering structure aggregate customers that form certain shapes with respect to the depot. Those are then solved as traveling salesman problems with the use of CPLEX.

Figure 1 visually summarizes a network map with the keywords that most appeared in the papers collected through this research path, showing that “Berth Allocation” and “Scheduling” were the ones that stood out.

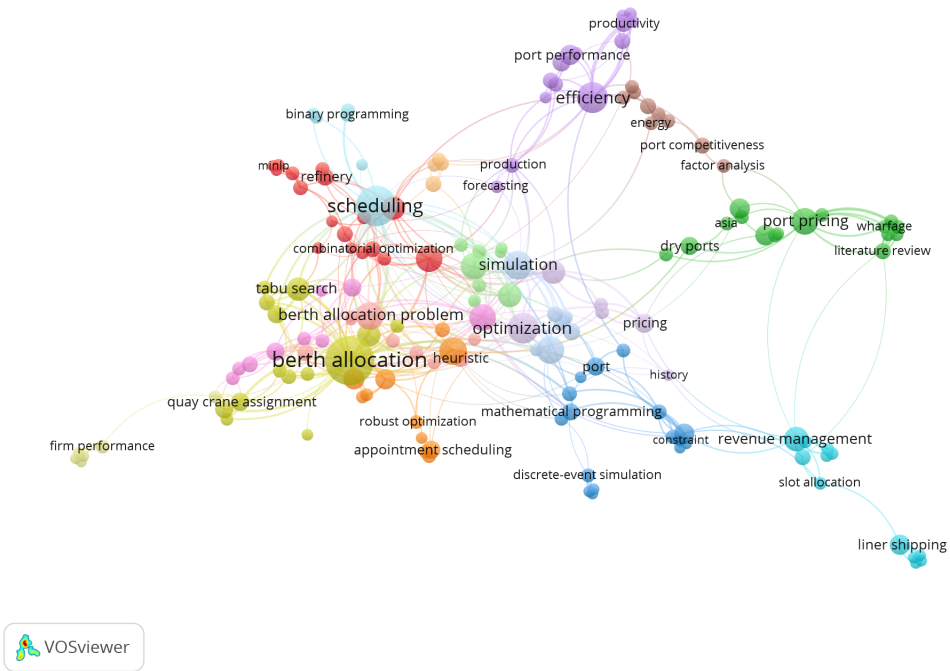

Figure 1: VOSViewer

Those topics are highly correlated and, in some cases, even complementary to the terminal appointment scheduling problem (TASP) explored in this thesis as showed by Figure 2. Normally BAP/BSP depend on given arrival patterns as an input, returning a berthing sequence as an output, while the terminal appointment scheduling under review considers the berthing rule as an input and outputs the appointments which will directly affect the arrival pattern.

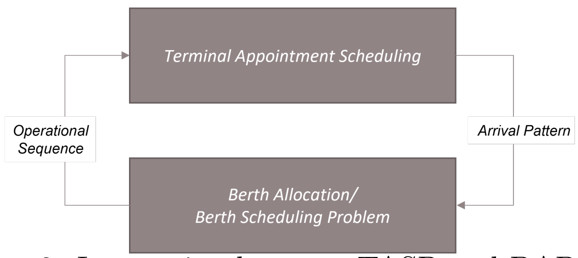

Figure 2: Interaction between TASP and BAP/BSP

In some terminals those berthing sequencing rules are defined by operational teams and not by scheduling teams, and if the schedulers perform the appointment scheduling procedure under different rules from the ones actually used, there is a greater chance that results deviate from the optimal. The alignment problem between codependent decision makers is explored in the literature under the name of coordination theory (Malone, 1988).

Table 1 summarises some of the important papers reviewed and the main characteristics collected from close related papers, reinforcing that there is no knowledge up to now of other papers that consider the terminal appointment scheduling problem under stochastic arrivals and processing times, with arrivals also being an endogenous variable, which is one of the contributions of this research.

For the experimentation phase an overlap index is proposed which allows to separate the analysis of high, medium and low overlapped requests. Additionally, to overcome the computational issue a partitioning method called "Cluster first, Schedule second" is proposed. Both the index definition and the "Cluster first, Schedule second" methodologies were not encountered in the literature reviewed.

From the managerial perspective it is also expected insights about value of information, different answering time frames, impact of uncertainties in arrivals and processing times, impact of different berthing sequencing rules, coordination between schedulers and operational teams, among others.

Table 1: Contribution PhD

Name	Authors	Journal Published	Year	Stochastic Arrivals?	Stochastic Processing Times?	Earliest?	Latest?	Allows Rejection of clients?	Defines the scheduling date?	Arrivals are endogenous?
Thesis	L. Bronneberg	-	2023	√	√	√	√	√	√	√
New formulations and solutions for the strategic berth template problem	E. Fernandez and M. Munoz-Marquez	European Journal of Operational Research	2022	X	X	X	X	√	√	X
Optimizing voyage charterparty (VCP) arrangement: Laytime negotiation and operations coordination	Q. Sun, Q. Meng, and M. C. Chou	European Journal of Operational Research	2021	n.a	√	X	X	X	X	n.a
An ILS heuristic for the ship scheduling problem: an application in the oil industry	V. Cunha, I. Santos, L. Pessoa, and S. Hamacher	International Transactions in Operational Research	2020	X	X	√	√	X	√	√
Practical approaches to chemical tanker scheduling in ports: a case-study on the Port of Houston	B. Cankaya, E. Wari, and B. Eren Tokgoz	Maritime Economics & Logistics	2019	X	X	X	X	X	√	√
The Contextual Appointment Scheduling Problem	N. S. Sadghiani and S. Motiian	Association for the Advancement of Artificial Intelligence	2019	X	√	X	X	X	X	√
The waterway ship scheduling problem	E. Lalla-Ruiz, X. Shi, and S. Voß	Transportation Research Part D	2018	X	X	√	√	X	√	X
Collaborative truck scheduling and appointments for trucking companies and container terminals	M. H. Phan and K. H. Kim	Transportation Research Part B	2016	X	X	√	√	X	√	X
Tactical berth allocation under uncertainty	L. Zhen	European Journal of Operational Research	2015	X	√	X	X	X	√	√
Dynamic scheduling with due dates and time windows: an application to chemotherapy patient appointment booking	Y. Gocgun and M. L. Puterman	Health Care Management Science	2014	X	X	√	√	√	√	√
The strategic berth template problem	A. Imai, Y. Yamakawa, and K. Huang	Transportation Research Part E: Logistics and Transportation Review	2014	X	X	X	X	√	√	X
Robust berth scheduling at marine container terminals via hierarchical optimization	M. Golias, I. Portal, D. Konur, E. Kaisar, and G. Kolomvos	Computers & Operations Research	2014	√	√	X	X	X	X	X
Sequencing and scheduling appointments with potential call-in patients	R. R. Chen and L. W. Robinson	Production and Operations Management	2014	√	√	X	X	X	√	X
A decision model for berth allocation under uncertainty	L. Zhen, L. H. Lee, and E. P. Chew	European Journal of Operational Research	2011	√	√	X	X	X	√	X
A bi-objective berth allocation formulation to account for vessel handling time uncertainty	M. M. Golias	Maritime Economics & Logistics	2011	X	√	X	X	X	√	√
A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time	X. L. Han, Z. Q. Lu, and L. F. Xi	European Journal of Operational Research	2010	√	√	X	√	X	X	X
Demurrage Costs Optimization in Ports Using Binary Mathematical Programming Models	E. Parra	TOP	1995	X	√	X	X	X	√	√

3 Oil and Derivatives Terminal Services

3.1 Terminal services and operations

The first step to understand how to optimize terminal services is by understanding how they work and what are the commitments agreed in the contracts. A port terminal consists of a dedicated space for the interconnection of land and sea, on which people and goods are in constant transit. Although infrastructure is completely different depending on the type of cargo it carries, Henesey (2006) proposes a very general four subsystem way of dividing terminal processes which is represented in Figure 3, and explained in sequence.

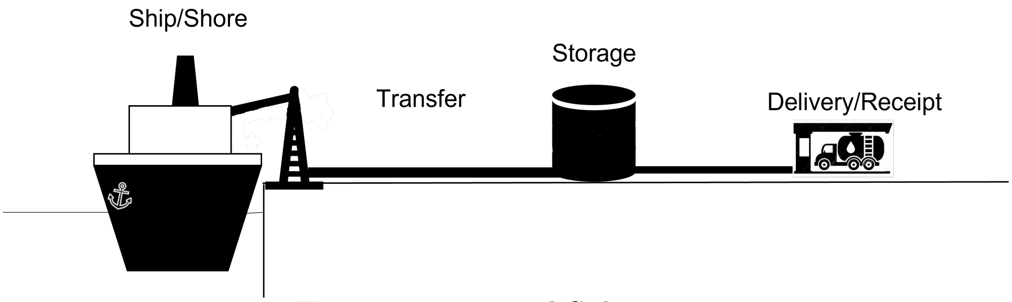

Figure 3: Terminal Subsystems

- Ship-to-Shore: the load/unload subsystem that allow movement of products between those two environments.
- Transfer: the interconnection subsystem, that allows transportation between each of the above subsystems (such as cranes, specialized vehicles, pipelines or conveyors).
- Storage: inventory subsystem.
- Delivery/Receipt: the onshore subsystem, which provides the connection with other modes of transportation such as trucks, trains and external pipelines.

The main service offered by a terminal is the product transportation (also known as cargo handling) and storage. For oil terminals, there are some secondary operations that might be offered, such as water and bunker supply, waste collection/ treatment, and oil sample analysis (Figure 4).

As seen in Figure 3, an oil terminal has its mooring infrastructure (berths or buoys), tanks (eventually) and pipelines. Depending on the terminal, there is an unique pipeline that is used for all products or there might be segregated ones for each product. Shared infrastructure (transporting more than one product through the same pipeline) may recall for product degradation (mixed interface between two products), which introduces more frequent cleaning procedures and quality tests.

The terminal can be onshore, with berths completely attached to land; or offshore, meaning that ships connect to a monobuoy or a set of buoys offshore, and product is transferred through a sub-sea pipeline between ship and onshore infrastructure. Product can be imported, exported or transferred between regions of the same country (cabotage), which might recall different quality ensurances.

There are basically two types of operations that can happen in a terminal: operations between tanks and ships (loading or unloading), and ships directly with other ships (ship-to-ship or transshipment). Loading procedure means that a ship receives cargo from other

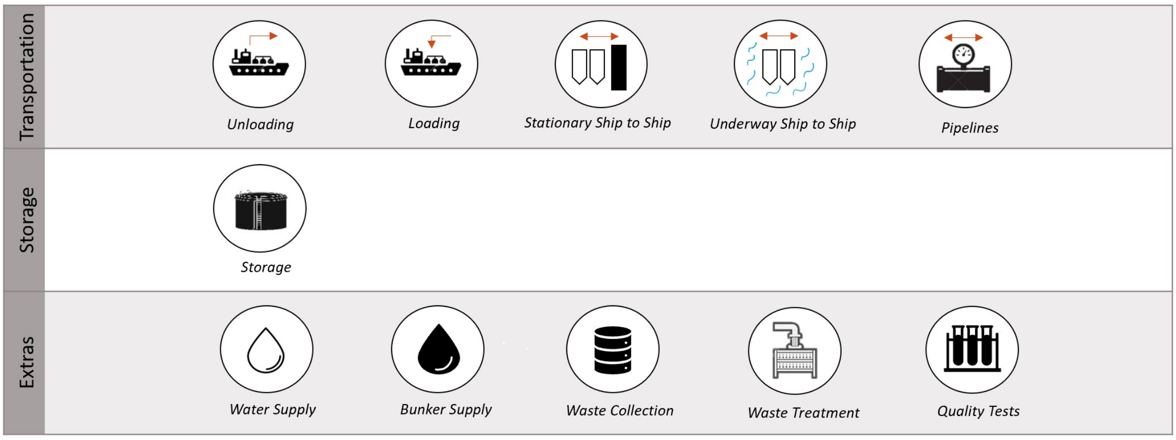

Figure 4: Types of Terminal Services

terminals/refineries via pipelines, which could be stored in tanks previously. Unloading procedure is basically the other way around, where a ship sends cargo to the terminal tanks or directly to other terminals/refineries. The reference of loading/unloading is always the ship, meaning that if a loading procedure is happening the ship is being loaded, and if unloading is happening it is the ship that is being unloaded.

A terminal can also offer its infrastructure and resources to transfer product from one ship directly to another. The first way is called stationary ship-to-ship operation (or stationary STS), on which a ship basically uses the terminal berth as a mooring point (called “principal” ship) so a second ship can moor side by side (called “secondary ship” or “pair”), and product can be transferred through hoses connecting both ships. The second way called transshipment uses two berths from the same terminal that have interconnected pipelines and product flows from one ship to the other using this interconnected infrastructure. Both ship to ship and transshipment are procedures that do not internalize product to the terminal.

Figure 5 details the main steps of a ship operation (for both loading and unloading procedures), referencing which teams are responsible for which box-process: vessel team, terminal team and external entities (boxes were drawn under the respective responsible columns). For STS and transshipment operations, all steps are done with both ships.

Some terminals operate under a first in first out approach (FIFO), and others with appointments. According to Wang (1993) the appointment is called static when all clients requests are received up to a limit date and processed at once, and dynamic when processed once per time as they come (this definition varies across authors). When scheduling a ship operation, a terminal has to define the date the ship should start the operation as well as the time-window it should take to finish considering clients restrictions.

As soon as the ship arrives to the port a “Notice of Readiness” (NOR) is sent to the terminal. If the terminal is also ready, the ship proceeds to berth directly, otherwise it needs to anchor and wait permission to enter.

In order to authorize mooring, some port authorities require a harbor pilot on board, who will guide ship maneuvering inside port region. As the ship is berthed, procedures such as: safety checks, documentation/information exchange, cargo measuring, cargo sampling and internal alignments (connections between ship/pipeline and pipeline alignments) are prepared, in order to get the system ready to pump product. If the cargo flows from the terminal to the ships, pumps from the terminal are used. If the cargo flows from the ships to the terminal, pumps from the ships are used.

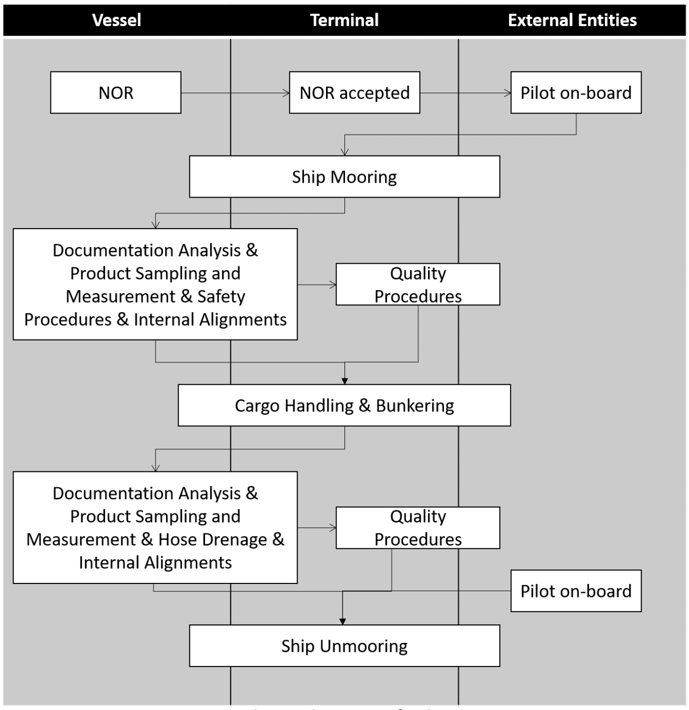

Figure 5: Flowchart of ship operation

As it was said before, it is possible that a ship requests extra services such as: bunker supply (ship fuel), water, and slop collection. In some cases those services are provided in parallel to the cargo handling, and in others they are provided in sequence. When all the product is handled, ship and terminal finalize documentation signatures, sampling and measurement procedures, as well as internal alignments, valve closures, disconnections and unmooring. Harbor pilot might be again requested to release the ship, ending operation.

There are quite many particularities impacting the flow of operations, meaning that each time is almost an unique experience, making this kind of service very complex to plan and address.

As any other service, what a terminal offers has four different characteristics (Kotler and Armstrong, 2017): intangibility (terminal/ship operation is not a physical material), inseparability (preemption is not allowed, unless in very specific cases), variability (the quality depends on stochastic variables) and perishability (time not used is never recovered).

3.2 Terminal service contracts

This section explores the players involved in a ship/shore operation, contracts and financial agreements.

A cargo owner is the party that owns the cargo and wants to transport it from point A to point B. If this transport is through sea, a ship must be hired (if the cargo owner does not have its own fleet to make transportation). The ship belongs to a ship owner, and the charterer is the figure that represents the interests of the cargo owner in the signing and planning of a Charter Party (agreement of ship service hiring).

There is another figure called ship brokers that might appear in some situations. Those are hired to intermediate negotiations, specially when cargo owners have limited contacts in a specific country and do not know the quality of the ship owners service.

The ones that effectively participate in the operation are the ones from Figure 6: ship owner, terminal operator and cargo owner (which coordinates the product logistics and the interaction between ship and terminal). Besides those, some other entities may also participate directly or indirectly in the operation such as harbor pilots (requested in some ports to maneuver ships inside port region), cargo surveyors (that provide measurement and quality tests) and agencies (which are hired by the ships to address any bureaucracy needed in shore).

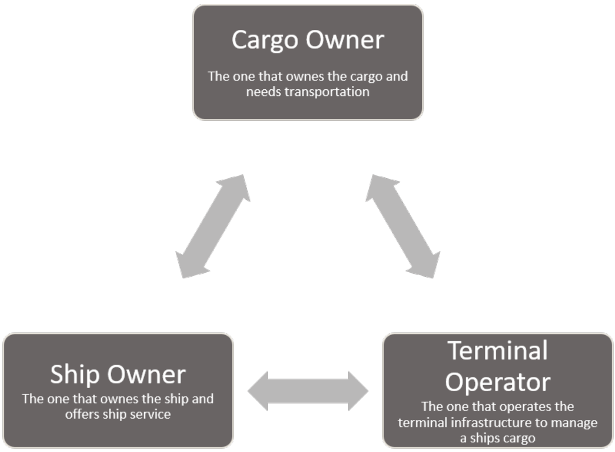

Figure 6: Parties involved in a ship/terminal operation

There are several types of shipping contracts, which are agreements between the charterer and ship owners (Sun et al., 2021), being “voyage charter party” (VCP) and “time charter party” (TCP) the most common ones, and those are celebrated by well-established formats such as ASBATANKVOY (Association of Ship Brokers and Agents (USA), 1977), SHELVOY and others. Those contracts normally have all the basic financial and efficiency related information, that should guide further operations and disputes.

One important concept that appears in all those contracts is called laytime, which is the interval that a ship will be available to handle cargo for a specific freight (Mauri et al., 2016). The goal of fixing a laytime period and providing demurrage and despatch fees is to penalize dilatoriness and reward promptitude, which was already approached in the literature by other researches such as Schofield (2015) and Ribeiro et al. (2016).

The standard laytime defined by the international guide Wordscale (2015) for full tankers operation is 72 hours for loading in the port of origin and unloading in the port of delivery (disregarding the shifting between ports), although those terms and conditions can vary from contract to contract. This is also understood by some as a benchmark that loading/unloading a full ship should take 36 hours each (or the proportional time depending on the amount carried). It is common to see this benchmark being used, specially for unloading contracts (PDVSA, 2008; REPSOL). A similar agreement has to be made with the terminal operator, on which financial benefits and operational obligations are set accordingly to the terminal limitations.

UNCTAD (2020) provides an estimation for ship freights based on both size and area they navigate. A Suezmax average rate can go from \$30,000 per day to \$150,000 depending on oil prices fluctuations, which according to REPSOL, is a good approximation of the demurrage fee.

TMRC (2021) shows a terminal agreement that sets the throughput, tankage and usage fees, with prices varying according to quantity operated (\$0.55 for each barrel, for the first 2,129,267 barrels, and \$0.10 for more volume). Another example would be from Valero Logistics Operations (2004) which charges \$0.30 per barrel, with a minimum commitment of 1,600,000 per month. Contracts with terminals can be occasional (one specific event) or long-term. It is common for long term contracts to include a “minimum throughoutput” clause which guarantees a minimum reward for the terminal independent of the facilities usage.

If more time is needed for completing the operation, someone should be accounted responsible, as the cargo owner might have a clause on the agreements to pay a demurrage fee to the ship or terminal for the extra time they were available. In order to control for that, both ship and terminal have to keep track of what was happening with the operation at all times, and file a formal complaint when a delay is being perceived due to the other party responsibility. There is also the case when the contract reward the parties involved in the operation for an operation performed in less time than expected (despatch rate).

The vast majority of contracts consider a valid Notice of Readiness (NOR) as the start of laytime period (Sun et al., 2021). For a NOR to be valid, a ship must have reached the correct proximity to port area and be ready to operate, by the time agreed in the planning period. The concept of demurrage applies only for VCP contracts, as the ship is hired for a specific voyage. In case of a TCP contract, there is not a direct fee associated to overtime for each voyage, but there is a reduction in the overall productivity as less operations are accomplished in the end.

There are a few sorts of disturbances that might cause delays in the system such as: environmental conditions (e.g. tide, waves, winds which are out of the safety range for operation), resource restrictions (e.g. lack of operational personnel, damaged pumps), delays in the arrival of ships, issues with the scheduling and coordination procedure (e.g. how ships are sequenced, time-window given), as detailed in Rahman et al. (2019). Those disturbances compromise the flow of operational activities which might generate congestion for a period of time. Any delay coming from other parties has an impact on terminals operation as they have to accommodate those uncertainties and provide service according to contractual expectations.

3.3 Chapter Summary

This chapter shows a detailed summary of how ship/shore operations are performed as well as contractually established. An operation depends on three main figures: cargo owners, ship owners and terminal operators. The need of cargo transportation of product from point A to point B triggers a set of contracts that establish safety and efficiency standards. In those agreements, both earnings and penalties are defined.

The operations considered in this research deals with fixed earning and demurrage fees, no despatch fees nor demurrage payments from other figures besides terminals, and no “minimum throughoutput”. Additionally, the scope is focused on the ship/berth operation considering that there are no restrictions on tanks and pipelines. Many of those definitions will be discussed further on and used in the conceptual design of the appointment scheduling problem. The richness of details shows how complex those operations might be, allowing multiple opportunities for future research.

4 Terminal Appointment Scheduling

4.1 Problem Setting

Whenever a client wants to operate ships in a certain terminal, formal requests have to be sent in the previous month containing information about the desired operation, such as volume, size of the ship, product, type of operation, as well as the dates that should be considered as possible dates for scheduling the start of operation (translated by an earliest and latest day). The terminal has to accept/reject the requests, as well as define an appointment date and time-window on which they are expected to happen (ANP, 2022; Valero Logistics Operations, 2004). Those decisions can be made in two different time frames: one by one as requests arrive (dynamic scheduling); or after a certain date (static scheduling) when all clients are supposed to have shared their demands. Depending on how profitable the operations are and how overlapped are the requests, more or less ships are accepted to operate.

As the oil industry is highly volatile, is common that clients do not have all those information so early on (normally volume, earliest and latest are the usually known ones), and that decisions are made by the terminal under a lot of uncertainties. Whenever a terminal sets a date and a time-window, there is an underlying agreement that if the operation does not happen according to that plan, the terminal might be requested to pay a demurrage fee for each extra day that the ship overstays in the port.

It is important to distinguish the appointment phase, which is the focus of this research, from the operational phase, which happens one month later according to a berthing rule defined by the operational team. This berthing rule can be as complex as the terminal understands is best to organize the berthing sequence. Some terminals use practical rules, such as "first in, first out" (Cankaya et al., 2019), some follow strictly the sequence given by the scheduling team (Sabria and Daganzo, 1989), others add a tolerance for delayed arrivals, others use prioritization schemes according to the importance of the client, size of the ship, limitations in the berthing/unberthing procedure, etc.

There is a lot of literature, normally under the name of berth allocation or berth scheduling problem, which focus exactly on defining those berthing sequences considering as input a pre-defined arrival pattern. This research, on the other hand, requires the definition of this berthing rule as an input and the output is the appointment plan (which directly affects the arrival pattern).

The fact that ships are scheduled in certain days do not imply that they are operated in that exact time. It all depends on the arrival dates (which are uncertain), processing times (which are also uncertain) and the berthing rule. Therefore, it is important that the scheduler is aligned with the rules currently being used by the operational team, so the appointment process is adherent to reality. A decision considering a different operational rule might lead to less profitable solutions.

The demurrage cost can be divided in two parts: start mismatch (difference between the real start and the scheduled start, counted only for ships that arrive early/on time) and operational mismatch (difference between the processing time and the time-window given). The fact that a ship waits for a berth to get free beyond the scheduled date, can be compensated by an overestimated time-window, as well as advancing the entrance of a ship can compensate an underestimated time-window.

The complexity of the problem consists in deciding who to accept/reject and when to schedule those operations in a stochastic environment which decisions not only impacts the profit but also the arrivals itself.

Considering that regulatory bodies, such as the National Petroleum Agency from Brazil, are currently framing this problem, some combinations of the exposed settings will be modelled and explored to support this process. Problem 1 presents the "main problem" with most of the definitions that will be explored over the following ones, as detailed:

- **Problem 1**: will tackle a static terminal appointment scheduling problem, on which decisions are to accept/reject requests with an appointment date definition, considering a FIFO berthing rule and available information about volume, earliest and latest.
- **Problem 2**: similar to Problem 1 except for the fact that volume information is not known when scheduling is performed.
- **Problem 3**: similar to Problem 1 except for the answering time frame which is dynamic instead of static.
- **Problem 4**: similar to Problem 1 except for the berthing rule which is "by schedule" instead of FIFO.
- **Problem 5**: will tackle a static terminal appointment scheduling problem, giving a FIFO berthing rule and information about volume, earliest and latest. The decision is reduced to accepting/ rejecting requests based on the earliest/latest informed by the clients. In this case a new demurrage and profit calculation will be discussed.

4.2 Mathematical Models and Algorithms

4.2.1 Problem 1

Consider a set of principal ships $S = (1, ..., N_{ships})$, a unique berth and a set of days on which ships can be scheduled to operate $D = (1, ..., N_{days})$. Clients have to send their requests until a specific limit date from the previous month, informing for each desired ship operation s their volume (vol_s) as well as the period they are available to be scheduled (between the earliest (e_s) and latest (l_s) days). The difference $a_s = l_s - e_s + 1$ is called availability size and impacts terminals flexibility when preparing the scheduling plan. As an example, a client wants to operate Ship s in a specific terminal next month (November). The client should send the operation request up to the 20th of October, which is the limit date established by the terminal to receive all requests of Novembers operations. The volume of Ship s should be informed, e.g $300,000m^3$, as well as the earliest and latest days on which the ship could be scheduled, e.g between day 15 and day 18 (meaning that it can be scheduled to start on day 15, day 16, day 17 or day 18).

The terminal has to evaluate all requests focusing on maximizing terminal's profit and decide which ships should be accepted/rejected to operate, the date they should be scheduled, called dx_s as well as its expected time-window for completion, called θ_s. Some terminals use a fixed and previously determined θ_s to simplify the appointment scheduling process, which will also be used in this research. The date when the ship really arrives is uncertain and is called dy_s, and the date that it really starts operation is called dz_s. There is no restriction on scheduling more than one ship in the same date even when there is only one berth in the terminal (this can be the most profitable option when compared, for example, to rejecting a ship).

It is possible that a ship arrives early, late or exactly on time to operate. Additionally, the operation can start earlier, later or exactly when it was scheduled, as summarized in Figure 7. The operation can only start when the ship arrives to the terminal ($dz_s \geq dy_s$) and takes an uncertain amount of time to finish, called processing time (or p_s).

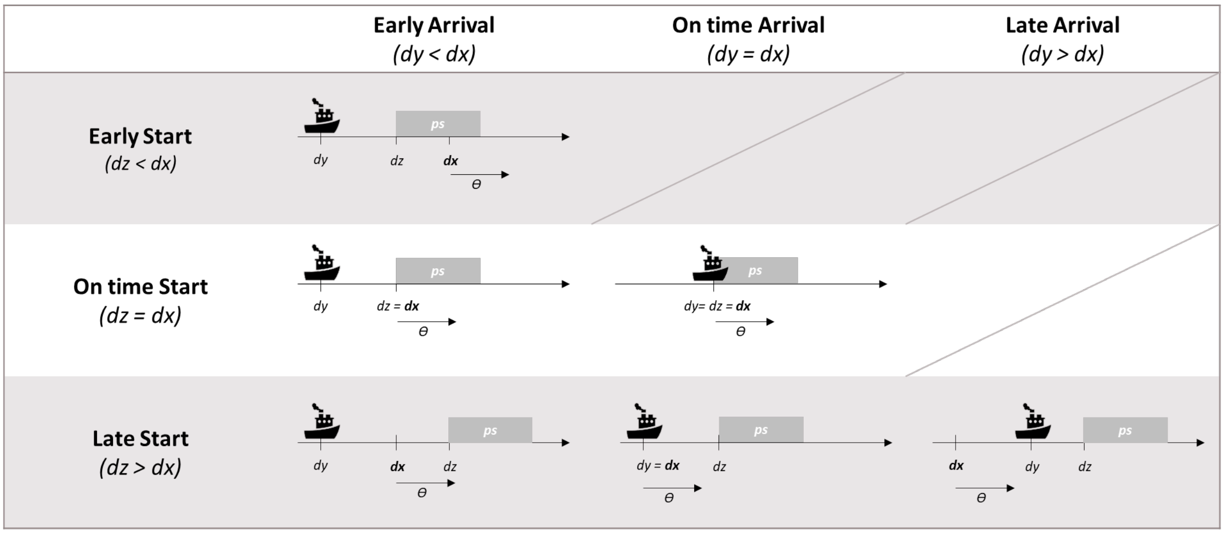

Figure 7: Arrival and Start Time Cases

All variables are integers and discretized in days (Lalla-Ruiz et al., 2012), and actions happen at the very first instant of each day. As an example, if the informed dx_s is day 6 and p_s is 3 days, means that the ship is scheduled to start its operation in the beginning of day 6 and will be processed in 3 days, meaning that in the beginning of day 9 the terminal will be available to receive another one. It is also assumed that ships can only start operating after the first day of the scheduled month (as an example, if a ship is scheduled to operate in November, but arrives in the end of October, the operation can only start after the first day of November).

The terminal charges Φ_s dollars per cubic meter transported and incurs in a demurrage penalty of Ω_s dollars per exceeded day, discounting any delay in the arrival of ships. This cost can be divided in: start mismatch ($sm_s = (dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)}$, which is the $dz_s - dx_s$ considered only when ships are not late) and operational mismatch ($p_s - \theta_s$). As the demurrage fee is always a cost and never a bonus (in this research), τ_s^+ is defined as the maximum between zero and the sum of the start mismatch and operational mismatch, which can be calculated as $\tau_s^+ = max[(dz_s - dx_s)\mathbb{1}^{(dy_s < dx_s)} + (p_s - \theta_s), 0]$. Table 2 shows some examples of how demurrage days are calculated, which are explained in detail as following:

- **Scenario 1**: Consider that a ship is scheduled to operate on day $dx_s = 4$ and the time-window offered to that operation is $\theta_s = 2$, meaning that the expected end of that operation is day $dx_s + \theta_s = 6$. The ship arrives early on day $dx_s = 2$ and the berth gets empty on day $dz_s = 3$, calling that ship to operate earlier than scheduled ($dz_s - dx_s \leq 0$). The real processing time was $p_s = 2$ and the ship left by day $dz_s + p_s = 5$, one day earlier than expected. In this first case there is no demurrage to be paid.
- **Scenario 2**: The only difference from the previous scenario is that the processing time was $p_s = 3$, one day more than the given time-window for that operation. Due to the fact that the ship berthed one day earlier, the extra day used to perform the operation is compensated, and the ship still leaves in the expected day ($dz_s + p_s = 6$). No demurrage is paid.
- **Scenario 3**: In this scenario, the operation started later than expected ($dz_s = 5 \geq dx_s = 4$), which was compensated by a faster processing time ($p_s - \theta_s = -1$). The ship also left by day $dz_s + p_s = 6$, with no need of demurrage payment.
- **Scenario 4**: This is similar to Scenario 3 but the processing time could not compensate the late start ($p_s - \theta_s = 0$), meaning that the ship left on day $dz_s + p_s = 7$ and the terminal had to pay one day of demurrage fee.

- **Scenario 5**: The ship is berthed exactly when it arrived ($dy_s = dz_s$), 2 days earlier than the scheduled date ($dz_s - dx_s = 2$), but the processing time was 3 days more than the given time window ($p_s - \theta_s = 3$). In this case the ship left on day $dz_s + p_s = 7$, and the terminal also incurred in one day of demurrage fee.
- **Scenario 6**: Finally the last example shows a different situation on which the ship arrives late ($dy_s = 5 > dx_s = 4$). In this case, even if the ship is berthed later on, it does not have the right to charge the terminal for any $dz_s - dx_s$ deviation, and the start mismatch is zero. The only term that is still considered is the operational mismatch ($p_s - \theta_s = 1$), as the terminal offered an operational time-window of $\theta_s = 2$ days and the operation lasted $p_s = 3$ days. In this case, the terminal also incurred in one day of demurrage fee.

Table 2: Days of demurrage

Situation	Scheduled Time (dx_s)	Time Window (Θ_s)	Expected End ($dx_s + \Theta_s$)	Real Arrival (dy_s)	Real Start (dz_s)	Real Processing (p_s)	Real End ($dz_s + p_s$)	Start Mismatch (dz_s - dx_s). $1^{dy_s \leq dx_s}$	Operational Mismatch (p_s- Θ_s)	Total Demurrage ($\tau^+ = max(SM + OM, 0)$)
1)	4	2	6	2	3	2	5	3-4 = -1	2-2 = 0	Max(-1,0) = 0
2)	4	2	6	2	3	3	6	3-4 = -1	3-2 = 1	Max(0,0) = 0
3)	4	2	6	2	5	1	6	5-4 = 1	1-2 = -1	Max(0,0) = 0
4)	4	2	6	2	5	2	7	5-4 = 1	2-2 = 0	Max(1,0) = 1
5)	4	2	6	2	2	5	7	2-4 = -2	5-2 = 3	Max(1,0) = 1
6)	4	2	6	5	6	3	9	LATE!	3-2 = 1	Max(1,0) = 1

The arrival of ship s is assumed to be dependent on the scheduled date and a random delay distributed around it (called $delay_s$) and is defined as $dy_s = dx_s + delay_s$. This dependence on the scheduled date is what characterizes the endogenous profile. The definition of when the ship really starts the operation (dz_s) is a result of its own arrival, previous operations and the berthing sequencing policy. This means that dz_s is also dependent on the scheduled date.

Assuming that the terminal revenue is only due to berth activities and that the only variable cost is the demurrage fee, the realised profit (P) can be calculated as the following:

$$P = \sum_{s \in S} \left[\Phi_s vol_s - \Omega_s(\tau_s^+) \right] \mathbb{1}^{(dx_s > 0)} \tag{1}$$

$$\tau_s = \underbrace{(dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)}}_{start\ mismatch} + \underbrace{(p_s - \theta_s)}_{operational\ mismatch} \tag{2}$$

If ship s is scheduled, dx_s must be between the earliest scheduled date e_s (Constraint 3) and the latest scheduled date l_s (Constraint 4), otherwise is zero. If dx_s is zero, means that the request of ship s is not accepted, then the ship never arrives and dz_s should also be zero (Constraint 5). Those constraints can be defined for any ship $s \in S$ as:

$$dx_s \geq e_s \mathbb{1}^{(dx_s > 0)} \tag{3}$$

$$dx_s \leq l_s \mathbb{1}^{(dx_s > 0)} \tag{4}$$

$$dz_s \leq Mdx_s \tag{5}$$

As previously explained, dz_s depends on the berthing sequencing rule defined externally by operational teams, which in this problem is set to be FIFO. Considering m_s as the sequence index corresponding to the operation of ship s, a general format for dz_s definition when $m_s > 1$ is given by the following recursive expression (maximum between the arrival of that ship and the end of the previous ship operation):

$$dz_s = max[dy_s, (dz_{s'} + p_{s'})|(m_{s'} = m_s - 1)] \tag{6}$$

For $m_s = 1$, the calculation of dz_s considers that operations can only start after day 1 of the operational month, and is given by:

$$dz_s = max[dy_s|(m_s = 1), 1] \tag{7}$$

This model considers thirteen input parameters and three variables, which are summarized as following:

Problem 1/ Model 1

Parameters:

- S= set of ships to be scheduled
- N_{ships} = number of ships to be scheduled
- D = set of days available for scheduling
- N_{days} = number of days from the scheduling period
- Φ_{days} = revenue fee in dollars per cubic meter transported
- Ω_{days} = demurrage fee in dollars per day
- e_s= expected earliest day that the ship s could arrive to the terminal
- l_s= expected latest day that a ship s should departure
- vol_s= expected latest day that a ship s should departure
- p_s= processing time of ship s, which follows a given statistical distribution
- $delay_s$= delay of ship s, which follows a given statistical distribution
- θ_s= time-window scheduled for ship s
- m_s= sequence index corresponding to the operation of ship s, which in this case is defined by the arrival order (FIFO).

Variables:

- dx_s is the date ship s is scheduled to arrive and start operation ($dx_s = 0$ means ship s was not scheduled/accepted to operate)

- dy_s is the date ship s arrives at the terminal, defined by $dy_s = dx_s + delay_s$
- dz_s is the date ship s starts operation

$$\underset{dx_s}{\text{maximize}} \quad E\left[\sum_{s\in S}\left[\Phi_s vol_s - \Omega_s\left((dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)\right)^+\right]\mathbb{1}^{(dx_s>0)}\right]$$

$$\begin{aligned}
\text{subject to} \quad & dx_s \geq e_s\mathbb{1}^{(dx_s>0)} && \forall s \in S \\
& dx_s \leq l_s\mathbb{1}^{(dx_s>0)} && \forall s \in S \\
& dy_s = dx_s + delay_s && \forall s \in S \\
& dz_s \leq M dx_s && \forall s \in S \\
& dz_s = max\left[dy_s, 1\right] && s|m_s = 1 \\
& dz_s = max[dy_s, (dz_{s'} + p_{s'})| \\
& \quad m_{s'} = m_s - 1] && \forall s \in S|m_s > 1 \\
& dx_s, dy_s, dz_s \geq 0 \; and \; integers
\end{aligned}$$

The problem presented is non-linear in the objective function as well as in some constraints. The reader is invited to follow the linearization process of most of the constraints detailed in "Annex 1: Linearization Process of Problem 1/ Model 1" which resulted in the following parameters and variables:

Problem 1/ Model 1 (Partially Linearized)

Parameters:

- S= set of ships to be scheduled
- N_{ships} = number of ships to be scheduled
- D = set of days available for scheduling
- N_{days} = number of days from the scheduling period
- Φ_{days} = revenue fee in dollars per cubic meter transported
- Ω_{days} = demurrage fee in dollars per day
- e_s= expected earliest day that the ship s could arrive to the terminal
- l_s= expected latest day that a ship s should departure
- vol_s= expected latest day that a ship s should departure
- p_s= processing time of ship s, which follows a given statistical distribution

- $delay_s$= delay of ship s, which follows a given statistical distribution
- θ_s= time-window scheduled for ship s
- m_s= sequence index corresponding to the operation of ship s, which in this case is defined by the arrival order (FIFO).
- M= very large number
- E= very small number

Variables:

- dx_s is the date ship s is scheduled to arrive and start operation ($dx_s = 0$ means ship s was not scheduled/accepted to operate).
- dy_s is the date ship s arrives at the terminal, defined by $dy_s = dx_s + delay_s$.
- dz_s is the date ship s starts operation.
- u_s is a binary variable which is 1 when ship s is accepted to operate, and 0 otherwise.
- w_s is a binary variable which is 1 if $dy_s \leq dx_s$ and 0 otherwise.
- z_s is a binary variable, defined by $z_s = w_s u_s$.
- h_s is an integer variable, defined by $h_s = (dz_s - dx_s)z_s$.
- v_s is an integer and greater or equal to zero variable, defined by $v_s = max(h_s + (p_s - \theta_s)u_s, 0)$.
- t_s is a binary variable which is 1 when $h_s + (p_s - \theta_s)u_s > 0$ and 0 otherwise.

$$\underset{dx_s}{\text{maximize}} \quad E\left[\sum_{s\in S}[\Phi_s vol_s - \Omega_s v_s]\right]$$

$$\begin{aligned}
\text{subject to} \quad & dx_s \geq e_s u_s && \forall s \in S \\
& dx_s \leq l_s u_s && \forall s \in S \\
& dz_s \leq M dx_s && \forall s \in S \\
& dy_s = dx_s + delay_s && \forall s \in S \\
& dx_s - dy_s \leq M w_s - E && \forall s \in S \\
& dy_s - dx_s \leq M(1 - w_s) && \forall s \in S \\
& z_s \leq w_s && \forall s \in S \\
& z_s \leq u_s && \forall s \in S \\
& z_s \geq w_s + u_s - 1 && \forall s \in S
\end{aligned}$$

$$h_s \geq -Mz_s \qquad \forall s \in S$$

$$h_s \leq Mz_s \qquad \forall s \in S$$

$$h_s \geq (dz_s - dx_s) - M(1 - z_s) \qquad \forall s \in S$$

$$h_s \leq (dz_s - dx_s) + M(1 - z_s) \qquad \forall s \in S$$

$$h_s + (p_s - \theta_s)u_s \leq Mt_s \qquad \forall s \in S$$

$$-(h_s + (p_s - \theta_s)u_s) \leq M(1 - t_s) \qquad \forall s \in S$$

$$v_s \geq h_s + (p_s - \theta_s)u_s \qquad \forall s \in S$$

$$v_s \geq 0 \qquad \forall s \in S$$

$$v_s \leq h_s + (p_s - \theta_s)u_s + M(1 - t_s) \forall s \in S$$

$$v_s \leq 0 + Mt_s \qquad \forall s \in S$$

$$dz_s = max\,[dy_s, 1] \qquad s|m_s = 1$$

$$dz_s = max[dy_s, (dz_{s'} + p_{s'})|$$

$$m_{s'} = m_s - 1] \qquad \forall s \in S|m_s > 1$$

$$dx_s, dy_s, dz_s, v_s \geq 0\; and\; integers$$
$$u_s, w_s, zs, t_s\; are\; binary$$
$$h_s\; is\; integer$$

The only constraints that were not linearized were the ones that define the dz_s calculation. In the case of this research the berthing sequencing rule is defined externally by the operational team, which adds difficulty on finding a linearized general form for dz_s. Additionally, the literature points out that appointment scheduling problems are computationally intensive (Zacharias and Pinedo, 2014) which can be explained by its combinatorial nature. Therefore, heuristics that can deal with those issues will be proposed and implemented in a free-access Touring-complete environment called "R". Initially, a methodology to find what will be called "simulated optimal solution" is presented followed by the definition of a partitioning method used to reduce resolution time (the terminology will be clear after the following examples).

In that way, the heuristic starts by enumerating all possible combinations of dx_s for all ships considering the e_s and l_s informed by clients (Table 3), including the rejection option (defined as $dx_s = NA$). As an example, the first ship presented days 4 (earliest), 5 or 6 (latest) as possibilities for scheduling the operation. The terminal has to consider those options of dates with the option of not scheduling that ship at all.

For each combination of dx_s, it is created a set of N_{scen} scenarios (a thousand, for example) of $delay_s$ and p_s randomly selected given their distributions, enabling the calculation of $dy_s = dx_s + delay_s$ (Table 4). Those scenarios will numerically simulate a sample of events.

Table 3: Numerical example of clients requests

Ships	*Availability*			
1	4	5	6	
2	2	3	4	5
3	1	2	3	

Table 4: Algorithm logic

Combinations of dx_s	dx_1	dx_2	dx_3	$E[P]$
1	4	2	1	$E[P]_1$
2	4	2	2	$E[P]_2$
3	4	2	3	$E[P]_3$
4	4	2	NA	$E[P]_4$
.				.
.				.
4*5*4 = 79	NA	NA	NA	0
				Solution: max $E[P]$

Scenarios	$delay_1$	$delay_2$	$delay_3$	p_1	p_2	p_3	dy_1	dy_2	dy_3	$m=1$	$m=2$	$m=3$
1	1	-2	-4	3	2	1	5	0	-3	3	2	1
2	-7	5	-3	4	2	1	-3	7	-1	1	3	2
3	3	1	2	1	3	5	7	3	5	2	3	1
.												
.												
1.000	.	.	.									

Ordered Scenarios	$dy_{s\|1}$	$dy_{s\|2}$	$dy_{s\|3}$	$p_{s\|1}$	$p_{s\|2}$	$p_{s\|3}$	$dz_{s\|1}$	$dz_{s\|2}$	$dz_{s\|3}$	$dx_{s\|1}$	$dx_{s\|2}$	$dx_{s\|3}$	*Profit ($P_{comb,sc}$)*
1	-3	0	5	1	2	3	1	2	5	1	2	4	$P_{1,1}$
2	-3	-1	7	4	1	2	1	5	7	4	1	2	$P_{1,2}$
3	3	5	7	3	5	1	3	6	11	2	1	4	$P_{1,3}$
.													
.													
1.000	.	.	.	.									$P_{1,1.000}$

$E[P]_1$

Having dy_s, p_s and the berthing rule (which in this case is FIFO) defined, the next step is calculating dz_s for each scenario. In this case all ships are indexed by their arrival order, from the earliest to the latest arrival. The m^{th} ship to operate is the m^{th} arrival. If multiple ships arrive at the same day, the one that requested operation first (smallest s) will be prioritized. Table 5 shows an example, in Scenario (a) the first ship to operate is Ship 3, sequenced by Ship 1 (which although arrived at the same date as Ship 2, has a smaller s) and then Ship 2. It is important to highlight that ship operations can only start after the first day of the scheduled month, even if they arrive in the end of the previous month. This can be seen in the $dz_s = 1|m_s = 1$ even knowing that $dy_s = -3|m_s = 1$. The other dz_s calculation follow the $dz_s = max[dy_s, (dz_{s'} + p_{s'})|m_{s'} = m_s - 1]$ formulation, being $dz_s = max[2, 1 + 1] = 2$ for $m = 2$ and $dz_s = max[2, 1 + 1 + 3] = 5$ for $m = 3$.

Table 5: Fifo Ordering/ Request tiebreaking rule

Scenarios	dx_1	dx_2	dx_3	$delay_1$	$delay_2$	$delay_3$	p_1	p_2	p_3	dy_1	dy_2	dy_3	*m=1*	*m=2*	*m=3*
a	4	3	3	-2	-1	-4	3	2	1	2	2	-1	3	1	2
b	4	3	3	-7	5	-3	4	2	1	-3	8	0	1	2	3
C	4	3	3	3	3	2	1	3	5	7	6	5	3	2	1

Ordered Scenarios	$dy_{s\|1}$	$dy_{s\|2}$	$dy_{s\|3}$	$p_{s\|1}$	$p_{s\|2}$	$p_{s\|3}$	$dz_{s\|1}$	$dz_{s\|2}$	$dz_{s\|3}$	$dx_{s\|1}$	$dx_{s\|2}$	$dx_{s\|3}$	*Profit ($P_{comb,scen}$)*
a	-1	2	2	1	3	2	1	2	5	3	4	3	$P_{1,1}$
b	-3	-1	7	4	2	1	1	5	7	4	3	3	$P_{1,2}$
c	3	5	7	5	3	1	3	8	11	3	3	4	$P_{1,3}$

A more refined tiebreaking rule is explored in detail in "Annex 3: Tiebreaking rule: smallest deviation first", which prioritizes requests with the smallest deviation from the scheduled date. In this case the highest priority is given to on time ships, followed by early ones (from the closer to the scheduled date to the one further), and then by the late ones (from the closer date to the schedule to the one further). Some insights about the impact of those tiebreaking rules will be explored in the experiments.

Having the dz_s calculated for all ships in each scenario, next step is defining the start mismatch and operational mismatch to calculate the total days of demurrage (τ_s^+) considering $\tau_s = (dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)$. Examples of those calculations were given previously in Table 2.

Last step is calculating the earnings by $\Phi_s vol_s$ and discounting the total cost of demurrage given by $\Omega_s(\tau_s^+)$ for all ships that were accepted to operate. The expected profit of each combination is calculated by the average of the N_{scen} scenarios. The combination with the highest $E[P]$ is called the simulated optimal solution.

This technique of creating scenarios and taking the average is also called in the literature as Sample Average Approximation (SAA).

Summarizing the algorithm presented:

Problem 1/ Algorithm 1:

- **Step 1**: For all $s \in S$, consider the e_s and l_s information to enumerate all $N_{comb} = \prod_{s \in S}(a_s + 1)$ possible combinations of dx_s including 'NA' which represent rejecting the operation of that specific ship. Set the counter $i = 1$.
- **Step 2**: Having the i^{th} combination, and given $s \in S$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 3**: For all $s \in S$, from all $j \in N_{scen}$ scenarios calculate $dy_s = dx_s + delay_s$.
- **Step 4**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of arrival of each ship s. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $mins$).
- **Step 5**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, and for all $s \in S|(m_s > 1)$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|(m_{s'} = m_s - 1)]$.
- **Step 6**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, (dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)]$
- **Step 7**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s \in S}[\Phi_s vol_s - \Omega_s(\tau_s^+)]\mathbb{1}^{(dx_s > 0)}$.
- **Step 8**: Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j \in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i + 1$. If $i \leq N_{comb}$ goes back to Step 2, otherwise goes to Step 9.
- **Step 9**: The final solution is given by the combination of $dx*_s$ that returns the maximum expected profit ($maxE[P_i]$).

It is important to observe that ordering FIFO is one of the simplest berthing rules that can be defined by operational teams. Although "R" is capable of handling complex algorithms, depending on the level of detail required, there are commercial softwares capable of creating actual digital twins, such as ARENA, ProModel, FlexSim, etc. This type of Discrete Event Simulation tools, although not used for this purpose in this research, can replace the numerical simulation if properly integrated to the optimizer. More about this topic will be explored later in this chapter.

4.2.2 Problem 2

The problem setting is very similar to what was proposed in **Problem 1**, except for the fact that volume information is not available to the terminal by the time the appointment plan is performed.

Without knowing the volume, there is no computation of real earnings which is key to profit calculation. Therefore, two different solution strategies are proposed. The first one is using Model 1 with all volumes being equal to the average encountered in the historical data. The second one is through another model with the same structure as Model 1, considering all volumes equal to 0. In this second way an additional constraint is needed to avoid the trivial solution. This constraint sets a minimum number of ships ($Min_{N_{ships}}$) that should be accepted to operate, which could also be estimated with historical data, if information is available. This model is implemented through Algorithm 2 which is defined as the following:

Problem 2/ Algorithm 2:

- **Step 1**: For all $s \in S$, consider the e_s and l_s information to enumerate all $N_{comb} = \prod_{s \in S}(a_s + 1)$ possible combinations of dx_s including 'NA' which represent rejecting the operation of that specific ship. Keep only the combinations that have more than or at least Min_N ships. Set the counter $i = 1$.
- **Step 2**: Having the i^{th} combination, and given $s \in S$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 3**: For all $s \in S$, from all $j \in N_{scen}$ scenarios calculate $dy_s = dx_s + delay_s$.
- **Step 4**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of arrival of each ship s. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $mins$).
- **Step 5**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, for all $s \in S|(m_s > 1)$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|m_{s'} = (m_s - 1)]$.
- **Step 6**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, (dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)]$
- **Step 7**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s \in S}[\Phi_s vol_s - \Omega_s(\tau_s^+)]\mathbb{1}^{(dx_s > 0)}$.
- **Step 8**: Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j \in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i + 1$. If $i \leq N_{comb}$ goes back to Step 2, otherwise goes to Step 9.
- **Step 9**: The final solution is given by the combination of $dx{*}_s$ that returns the maximum expected profit ($maxE[P_i]$).

The choice of which methodology to use depends on the historical data available, and the perception of schedulers of which one is more suitable to their context. Insights about the value of volume information and comparisons between those solutions will be explored in the experiments.

4.2.3 Problem 3

This problem tackles a different answering time frame compared to **Problem 1** and **Problem 2**. In this case, terminals have to answer clients requests immediately, without having full information about the volume and availability days from the following requests. For this problem, two solutions are given: one on which past decisions are fixed, meaning that whatever schedule was set for previous clients can not be changed; and the other one that allows adjustments in past decisions, considering the earliest/latest information given by clients and a penalization λ introduced in the objective function.

As an example of the first solution, consider that Ship 1 requests an appointment between day 2 and 5, and that the decision was to schedule it to operate on $dx_1 = 2$. Then consider that, later on, Ship 2 requests an appointment between day 1 and 3. The dx_2 has to be decided knowing that $dx_1 = 2$ is fixed.

In the other hand, if the example was from the second solution, the dx_2 would be decided considering that the appointment of Ship 1 could be adjusted for any of the other days within its earliest and latest. This means that appointments from Ship 1 could be given on $dx_1 = 2$, $dx_1 = 3$, $dx_1 = 4$ or $dx_1 = 5$, instead of only considering $dx_1 = 2$ as an option. What is never an option is rejecting a ship that was already accepted previously, or accepting a ship that was rejected.

The algorithm that does not allow past adjustments is given by Algorithm 3A and the one allowing past adjustments is given by Algorithm 3B. Both algorithms need the definition of S_{new} with all new ships to be considered at each new iteration (for this exercise S_{new} has only one ship per iteration). Those are given as following:

Problem 3/ Algorithm 3A:

- **Step 1**: Set $S_{all} = 0$.
- **Step 2**: For all $s \in S_{new}$, consider the e_s and l_s information to enumerate all options of dx_s including 'NA' which represent rejecting the operation of that specific ship. If $S_{all} = 0$, go to Step 5, otherwise go to Step 3.
- **Step 3**: For all ships $s \in S_{all}$, consider the $dx_s = dx*_s$ that was previously defined in Step 9.
- **Step 4**: Enumerate all possible combinations of dx_s considering $s \in (S_{all} \cup S_{new})$.
- **Step 5**: Set the counter $i = 1$. Define $N_{comb} = (\prod_{s \in S_{all}} 1) * (\prod_{s \in S_{new}} (a_s + 1))$ as the total number of combinations. Set $S_{all} = (S_{all} \cup S_{new})$.
- **Step 6**: Having the i^{th} combination, and given $s \in S_{all}$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 7**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios calculate $dy_s = dx_s + delay_s$.
- **Step 8**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of arrival of each ship s. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $mins$).
- **Step 9**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, for all $s \in S|(m_s > 1)$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|(m_{s'} = m_s - 1)]$.
- **Step 10**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, (dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)]$.

- **Step 11**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s\in S}[\Phi_s vol_s - \Omega_s(\tau_s^+)]\mathbb{1}^{(dx_s>0)}$.
- **Step 12**: Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j\in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i+1$. If $i \leq N_{comb}$ goes back to Step 6, otherwise goes to Step 13.
- **Step 13**: The final solution is given by the combination of $dx*_s$ that returns the maximum expected profit ($maxE[P_i]$). If there are other requests, redefine S_{new} being the group of new ships, and go back to Step 2, otherwise $dx*_s$ is the final solution.

Problem 3/ Algorithm 3B:

- **Step 1**: Set $S_{all} = 0$.
- **Step 2**: For all $s \in S_{new}$, consider the e_s and l_s information to enumerate all options of dx_s including 'NA' which represent rejecting the operation of that specific ship. If $S_{all} = 0$, go to Step 4, otherwise go to Step 3.
- **Step 3**: For all ships $s \in S_{all}$, consider the e_s and l_s information of ships that were previously accepted in Step 9 to enumerate all options of dx_s. For the ones that were already rejected in Step 9, keep the information that $dx_s = dx*_s = NA$.
- **Step 4**: Enumerate all possible combinations of dx_s considering $s \in (S_{all} \cup s_{new})$.
- **Step 5**: Set the counter $i = 1$. Define $N_{comb} = (\prod_{s\in(S_{all}|dx_s=NA)} 1)*(\prod_{s\in(S_{all}|dx_s>0)}(a_s))*(\prod_{s\in S_{new}}(a_s+1))$ as the total number of combinations. Set $S_{all} = (S_{all} \cup S_{new})$.
- **Step 6**: Having the i^{th} combination, and given $s \in S_{all}$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 7**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios calculate $dy_s = dx_s + delay_s$.
- **Step 8**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of arrival of each ship s. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $min|s|$).
- **Step 9**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, for all $s \in S|m_s > 1$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|m_{s'} = m_s - 1]$.
- **Step 10**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, (dz_s - dx_s)\mathbb{1}^{(dy_s\leq dx_s)} + (p_s - \theta_s)]$.
- **Step 11**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s\in S}[\Phi_s vol_s - \Omega_s(\tau_s^+) - \lambda_s(|dx*_s - dx_s|)]\mathbb{1}^{(dx_s>0)}$.
- **Step 12**:Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j\in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i+1$. If $i \leq N_{comb}$ goes back to Step 6, otherwise goes to Step 13.
- **Step 13**: The final solution is given by the combination of $dx*_s$ that returns the maximum expected profit ($maxE[P_i]$). If there are other requests, redefine S_{new} being the group of new ships, and go to Step 2, otherwise $dx*_s$ is the final solution.

Comparisons between those two methodologies will be also explored in the experiments.

4.2.4 Problem 4

The problem setting is very similar to what was proposed in **Problem 1**, terminal answers all clients after having all requests, the goal is to determine who is accepted/rejected as well as the scheduling date, and information about volume, earliest and latest days are available. The difference stands in the rule used by the operational team to determine the berthing sequence. In this case instead of FIFO, ships are operated exactly in the sequence informed by the schedulers (also called as "by sequence" rule), regardless if the berth is free and there are ships that could be advanced. If two ships are scheduled to operate in the same date, two tiebreaking rules are proposed: the first one prioritizing ships by their request order (the one with the smallest s first), and the second one prioritize ships by its arrival deviation from the scheduled date (which is explained in "Annex 3: Tiebreaking rule: smallest deviation first"). Table 6 shows an example of ordering with the "by schedule" rule with the request order prioritization. In Scenario (a) is possible to see that both Ship 2 and Ship 3 are scheduled to operate on day 3, and Ship 1 on day 4. In this case, according to the first tiebreaking rule defined, the sequence on which ships will be operated is Ship 2 first, Ship 3 after and then Ship 1, even knowing that Ship 3 was available before Ship 2 ($dy_3 = -1 < dy_2 = 2$).

Table 6: By Schedule Ordering/ Request tiebreaking rule

Scenarios	dx_1	dx_2	dx_3	$delay_1$	$delay_2$	$delay_3$	p_1	p_2	p_3	dy_1	dy_2	dy_3	m=1	m=2	m=3
a	4	3	3	-2	-1	-4	3	2	1	2	2	-1	2	3	1
b	4	3	3	-7	5	-3	4	2	1	-3	8	0	2	3	1
C	4	3	3	3	3	2	1	3	5	7	6	5	2	3	1

Ordered Scenarios	$dy_{s\|1}$	$dy_{s\|2}$	$dy_{s\|3}$	$p_{s\|1}$	$p_{s\|2}$	$p_{s\|3}$	$dz_{s\|1}$	$dz_{s\|2}$	$dz_{s\|3}$	$dx_{s\|1}$	$dx_{s\|2}$	$dx_{s\|3}$	Profit ($P_{comb,scen}$)
a	2	-1	2	2	1	3	2	4	5	3	3	4	$P_{1,1}$
b	8	0	-3	2	1	4	8	10	11	3	3	4	$P_{1,2}$
c	6	5	7	3	5	1	6	9	14	3	3	4	$P_{1,3}$

The algorithm considering the "by schedule" berthing rule is called Algorithm 4 and the implementation is given as the following:

Problem 4/ Algorithm 4:

- **Step 1**: For all $s \in S$, consider the e_s and l_s information to enumerate all $N_{comb} = \prod_{s \in S}(a_s + 1)$ possible combinations of dx_s including 'NA' which represent rejecting the operation of that specific ship. Set the counter $i = 1$.
- **Step 2**: Having the i^{th} combination, and given $s \in S$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 3**: For all $s \in S$, from all $j \in N_{scen}$ scenarios calculate $dy_s = dx_s + delay_s$.
- **Step 4**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of the dx_s of each ship s. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $mins$).
- **Step 5**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, for all $s \in S|(m_s > 1)$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|(m_{s'} = m_s - 1)]$.
- **Step 6**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, (dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)]$
- **Step 7**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s \in S}[\Phi_s vol_s - \Omega_s(\tau_s^+)]\mathbb{1}^{(dx_s > 0)}$.

- **Step 8**: Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j \in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i + 1$. If $i \leq N_{comb}$ goes back to Step 2, otherwise goes to Step 9.
- **Step 9**: The final solution is given by the combination of $dx*_s$ that returns the maximum expected profit ($maxE[P_i]$).

In the experiments it will be studied the comparison between the "by schedule" and the FIFO berthing rule, as well as the impact of not having coordination between decision makers towards which rule is being used.

4.2.5 Problem 5

This last problem setting is also about a static scheduling process, using a FIFO berthing rule and having information about volume, earliest and latest available. The difference stands on the fact that the decision made by the terminal is reduced to accepting/rejecting requests, no longer defining a specific scheduled date for it to happen. In this case, terminals agree that the ship will arrive uniformly distributed within the availability days (between earliest and latest), although the real arrival might have some delays. This means that dy_s is no longer endogenous and can be calculated by $dy_s = \mathcal{U}[e_s, l_s] + delay_s$.

The number of scheduling combinations is now reduced to $2^{N_{ships}}$ possibilities, as each ship can only be accepted or rejected. Table 7 shows an example of enumeration for $N_{ships} = 3$. For each of those combinations a set of N_{scen} scenarios is created (a thousand, for example) of $delay_s$ and p_s randomly selected given their distributions, enabling the calculation of $dy_s = \mathcal{U}[e_s, l_s] + delay_s$, being the uniform distribution just of integer numbers (as the hole problem is discretized in days). Those scenarios will numerically simulate a sample of events to deal with the stochasticity of the problem.

Having dy_s, p_s and the berthing rule (which in this case is FIFO) defined, the next step is calculating dz_s for each scenario. In this case all ships are indexed by their arrival order, from the earliest to the latest arrival. The m^{th} ship to operate is the m^{th} arrival. If multiple ships arrive at the same day, the one that requested operation first (smaller s) will be prioritized.

For the calculation of the demurrage cost a new setting is suggested, when ships arrive before e_s, meaning $dy_s < e_s$, the start mismatch (sm_s) should be calculated by the difference $dz_s - e_s$. If the ship arrives within e_s and l_s, meaning $e_s \leq dy_s \leq l_s$, the start mismatch should be calculated by the difference $dz_s - dy_s$. If the ship arrives after l_s, meaning $dy_s > l_s$, start mismatch is zero. Next the total days of demurrage is defined by τ_s^+ considering $\tau_s = sm_s + (p_s - \theta_s)$.

Last step is calculating the earnings by $\Phi_s vol_s$ and discounting the total cost of demurrage by $\Omega_s(\tau_s^+)$ for all ships that were accepted to operate. The expected profit of each combination is calculated by the average of the N_{scen} scenarios. The combination with the highest $E[P]$ is called the simulated optimal solution.

Table 7: Combination Acceptance/Rejection

Combinations of acceptance/rejection	dx_1	dx_2	dx_3	$E[P]$
1	1	1	1	
2	1	1	0	
3	1	0	1	
4	1	0	0	
.				.
.				.
(2*2*2) = 8	0	0	0	

The algorithm used to model this problem is given by Algorithm 5 as the following:

Problem 5/ Algorithm 5:

- **Step 1**: Considering that each $s \in S$ can be accepted ($status_s = 1$) or rejected ($status_s$ = NA), enumerate all $N_{comb} = 2^{N_{ships}}$ possible combinations of decisions available. Set the counter $i = 1$.
- **Step 2**: Having the i^{th} combination, and given $s \in S$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 3**: For all $s \in S$, from all $j \in N_{scen}$ scenarios calculate dy_s =U$[e_s, l_s] + delay_s$, being the uniform distribution just of integer numbers.
- **Step 4**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of arrival of each ship. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $mins$).
- **Step 5**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, for all $s \in S|(m_s > 1)$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|(m_{s'} = m_s - 1)]$.
- **Step 6**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the start mismatch as the following: in case when $dy_s < e_s$, $sm_s = dz_s - e_s$; or if $e_s \leq dy_s \leq l_s$, $sm_s = dz_s - dy_s$, or if $dy_s > l_s$, $sm_s = 0$.
- **Step 7**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, sm_s + (p_s - \theta_s)]$
- **Step 8**: For all $s \in S$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s \in S}[\Phi_s vol_s - \Omega_s(\tau_s^+)]\mathbb{1}^{(dx_s>0)}$.
- **Step 9**: Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j \in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i + 1$. If $i \leq N_{comb}$ goes back to Step 2, otherwise goes to Step 10.
- **Step 10**: The final solution is given by the combination of $dx*_s$ that returns the maximum expected profit ($maxE[P_i]$).

Experiments will explore the comparison between this setting, on which dy_s is not endogenous, with **Problem 1** setting on which dy_s is endogenous.

4.3 Complex dz_s calculation

Although free-coding environments such as "R" allow rich algorithm coding, there are some specialized simulation softwares (such as ARENA, ProModel, FlexSim) capable of creating digital twins with friendly graphic interfaces that can reproduce high level of details encountered in real scale problems.

This type of simulation have the same purpose as the Numerical Simulation explained in previous algorithms. It receives a set of scheduled requests with their vol_s and dx_s, and according to pre-defined operational parameters, $delay_s$ distribution and the berthing/unberthing logic it outputs the calculation of dz_s. Those results can be compared to the ones from the solver.

The simulation model is based in a case study from a Brazilian oil terminal which operates four types of product (crude oil, formation water, light derivatives and dark derivatives) and two types of operation (ship to ship and unloading). The reader is invited to "Annex 5: Detailing the Simulation Model" for an explanation on the simulation model development.

4.4 Approximate Methodology for Solving Large Scale Instances

In the previous subsections different models were presented which were implemented in "R". In those cases all combinations of scheduling agenda (considering the particularity of each problem) are enumerated and profits calculated, in order to find what is the combination that returns the higher profit (called as simulated optimal solution). The choice of enumerating and calculating all profits was taken to increase flexibility for the experimentation phase, in which not only the simulated optimal solution is needed but also other non-optimal solutions (one great example will be presented in the experiment involving ARENA). The difficulty with this kind of solution, as well as with most combinatorial problem is the "curse of dimensionality". Very small instances can easily get intractable time wise, which might derail the use of it in a real context (Cankaya et al., 2019).

The appointment scheduling, as it was designed in **Problem 1**, is of order $N_{scen} \prod_s (a_s + 1)$ which is the number of scenarios multiplied by the product of the availability sizes plus one to incorporate the case when ships are rejected ("NA"). Later on, it will be showed that finding the solution for a small instance of $N_{ships} = 8$ ships with $a_s = 5$ days can take almost 2 days, which is completely unacceptable for practical purposes.

In order to get around this timing issue, a clustering based partitioning method is proposed inspired by the "Cluster First, Route Second" methodology explored in the vehicle routing setting, adding the complexity that scheduling results of previous clusters impacts directly the scheduling decisions of the following ones.

Figure 8 illustrates the process. First step, all requests are clustered into smaller groups given a specific methodology, which will be explained in detail in one of the experiments. Then each of those clusters is scheduled taking into consideration whatever was scheduled in previous clusters (Cluster 1 is scheduled first and its result is considered as an input to the scheduling process of Cluster 2, then the schedule of ships from Cluster 1 and 2 are inputs for Cluster 3, and so on). The schedule definition set in previous clusters are never changed/adjusted.

Clustering is an important step in this partitioning method, which sets the order that requests will be considered in the scheduling process, recalling that each ship has a different reward and introduces a different cost to the system. The probability of rejecting a ship increases towards the end of the scheduling procedure, as previous ships were already scheduled.

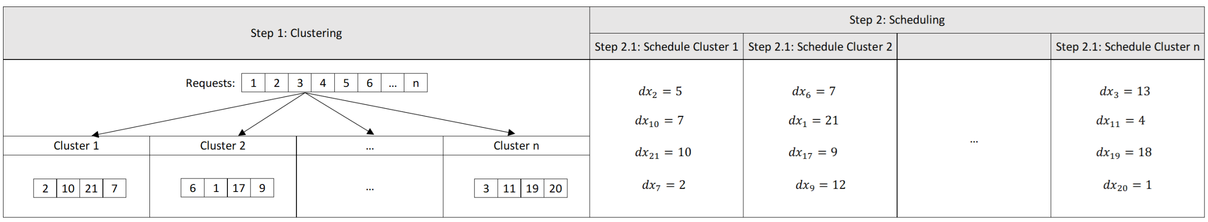

Figure 8: Partitioning Method: First Cluster, Second Schedule

This clusters can be of different sizes which will be identified as the following examples: [8] (a unique cluster of 8 ships), [4 4] (two clusters of 4 ships), [2 2 2 2] (four clusters of 2 ships), [1 1 .. 1 1] (eight clusters with 1 ship each), or any combination possible when cluster sizes are different, for example [5 3] (two clusters, one with 5 ships and the other with 3 ships).

Due to the fact that the clustering method sets the order on which ships will be scheduled, the best clustering methodology should prioritize ships with higher financial contribution. Finding this ordering is not straightforward as ships revenue is a function of volume, time spent in berth and port's congestion level (which is a consequence of the scheduling procedure itself). Therefore, the characteristics that appear to be relevant for the clustering procedure are volume, earliest/latest and availability size.

For this reason, a total of ten different clustering methodologies will be tested: two machine learning methodologies, K-means and hierarchical agglomerative clustering (HAC) which were used previously in appointment scheduling context (Yousefi et al., 2020); and other practical ordering systems, "Lowest Earliest First" (LEF)/ "Highest Earliest First" (HEF); "Lowest Volume First" (LVF)/ "Highest Volume First" (HVF); "Lowest Latest First" (LLF)/ "Highest Latest First" (HLF) and "Lowest Availability First" (LAF)/ "Highest Availability First" (HAF).

Further on, results from the "Cluster First, Schedule Second" method will show that one type of clusterization stands for its small gap with base results and easy implementation. This technique is then selected to incorporate the partitioning method.

4.5 Chapter Summary

Considering the focus recently given by regulatory agencies (ANP, 2022) to design the best frame to maximize terminals profit, five different problem settings are proposed and summarized as following:

- **Problem 1**: will tackle a static terminal appointment scheduling problem, on which decisions are to accept/reject requests with an appointment date definition, considering a FIFO berthing rule and available information about volume, earliest and latest. Two tiebreaking rules are suggested: ordering by the request number (Algorithm 1) and ordering by the smallest deviation with the scheduled date (Algorithm 1 with the adjustments explained in Annex 3).
- **Problem 2**: similar to Problem 1 except for the fact that volume information is not known when scheduling is performed. This problem will be approached by Algorithm 1 considering the mean historical volume, and Algorithm 2 which does not consider volume information at all.
- **Problem 3**: similar to Problem 1 except for the answering time frame which is dynamic instead of static. This problem will be approach by Algorithm 3A (which does not allow adjustments in previous appointments) and Algorithm 4 (which allows them).
- **Problem 4**: similar to Problem 1 except for the berthing rule which is "by schedule" instead of FIFO. Two tiebreaking rules are suggested: ordering by the request number (Algorithm 4) and ordering by the smallest deviation with the scheduled date (Algorithm 4 with the adjustments explained in Annex 3).
- **Problem 5**: will tackle a static terminal appointment scheduling problem, giving a FIFO berthing rule and information about volume, earliest and latest. The decision is reduced to accepting/ rejecting requests based on the earliest/latest informed by the clients. In this case a new demurrage and profit calculation will be discussed. This problem will be modeled by Algorithm 5.

To deal with computational issues, heuristics where proposed and implemented in a free-access Touring complete environment called "R". Initially all solutions are enumerated and the expected profit of each one is calculated by the mean of N_{scen} replications to account for the stochastic variables (delays and processing times). The solution with the highest expected profit is considered the simulated optimal solution. This technique is called by the literature as Sample Average Approximation.

This kind of methodology by itself is time-consuming which led to the development of a partitioning method called "Cluster First, Schedule Second". Ten different clustering methodologies were explored, and further on, experiments will show that one stands due to its small gap and simplicity in the implementation.

Although "R" can model complex algorithms, there are Discrete Event Simulation software created specifically to reproduce highly detailed logic, such as ARENA. A simulation model is then proposed considering restrictions encountered in real congested terminals. This model will be used to compare results of the dz_s calculation with the ones from the optimization algorithms.

5 Experiments and Results

5.1 Experiments Design

The following section presents the experiments designed to answer each of the research questions from the first chapter, exploring topics such as value of information, answering time frames, berthing rules, levels of uncertainties, coordination among decision makers, etc.

It is expected that depending on the level of congestion faced by terminals, some of the insights will have more or less impact, which will be differentiated by categorizing the appointment problems through an index (κ) that captures the degree of overlap among requests. Therefore three categories are proposed: HO (high overlap), MO (medium overlap) and LO (low overlap) based on the number of ships and earliest/latest information. The following equation defines how κ is calculated:

$$\kappa = \frac{1}{\underbrace{\frac{N_{ships}!}{2!(N_{ships}-2)!}}_{\#pairs}} \sum_{i=1}^{N_{ships}-1} \sum_{j=i+1}^{N_{ships}} \left[\frac{(l_i - e_i + 1) + (l_j - e_j + 1)}{(max(l_i, l_j) - min(e_i, e_j) + 1)}\right]$$

Considering the mean availability size of all set of requests, boundaries were empirically defined as: $\kappa \leq 0.8$ is considered LO, $\kappa \geq 1.2$ is considered HO, and anything in between is MO. Table 8 shows an example of the categorization for $N_{ships} = 3$, with requests spread over the first 11 days of the agenda. In this example, the HO case is characterized by Ship 1 with availability days from Day 1 to 4, Ship 2 with Day 3 and 4, and Ship 3 with Day 3 to 5. On the other hand, the LO case is characterized by requests more spread in the agenda on which Ship 1 requested that the operation should be scheduled within Day 1 to 4, Ship 2 on Day 5 or 6 and Ship 3 from Day 9 to 11.

Table 8: HO, MO and LO Kappa

	κ	1	2	3	4	5	6	7	8	9	10	11
HO	1.52											
MO	1.02											
LO	0.78											

In order to answer the research questions some different input exercises are defined (referred as EXA, EXB, EXC, EXD and EXE) which are explored through a HO, MO and LO perspective. For the static scheduling models, each exercise contain information about a set of requests ($N_{ships} = 8$), their respective volume vol_s and e_s and l_s information, which are described in Table 10, 11, 12, 13 and 14. The dynamic scheduling models will also consider some of those input exercises, only that requests are revealed one by one. The volumes and availability sizes used were based on a statistical analysis performed with data from a real Brazilian terminal (the ARENA experiment which will be explained later on also used the same data).

The time window should be considered fixed to all ships and equal to $\theta_s = 2$ days (as the number reported by specialists in interviews). The terminal has a revenue of \$0.50 per m^3 transported, and it has to pay a daily demurrage fee of \$25,000 for every extra day that a ship stays in the terminal. Delays are considered normally distributed with mean 0 and standard deviation of 2 days (to keep the discretization in days). This means, as shown in Figure 9, that in 19.7% of the cases ships arrive exactly on time, 17.5% arrive 1 day early and the same percentage arrive 1 day late, 12.1% arrive 2 days early and the same percentage arrive 2 days late and 0.7% arrive more than 2 days early and the same percentage arrive more than two days late.

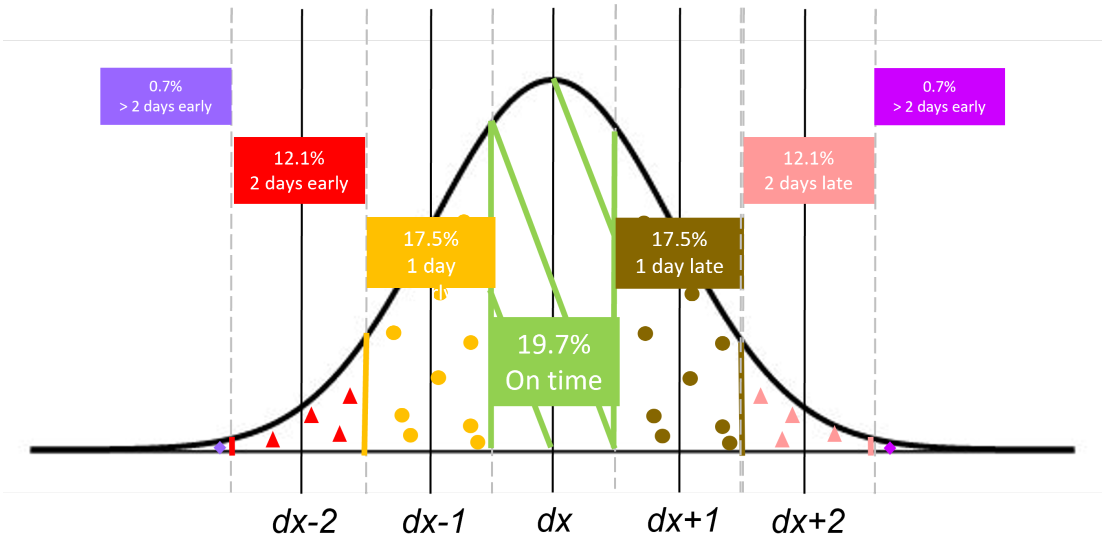

Figure 9: Normal Distribution - Delay

The processing time has also an associated normally distributed function with mean of 2 days and standard deviation of 1 day, with a difference that an operation is never smaller than one day (meaning that p_s is defined by a truncated normally distributed formulation). This means, as shown in Table 9 that approximately 17.8% of the cases ships take 1 day to operate, 45.5% of the cases ships take 2 days, 28.8% of the cases ships take 3 days and 7.9% of the cases ships take more than 3 days. Operations start at day 1 of each month, meaning that even if ships arrive earlier in the end of the previous month, they can only start to operate from the first day of the next month on. $N_{scen} = 1,000$ scenarios will be performed to account for the stochasticity of delays and processing times, which number showed converging results.

Table 9: Truncated Normal Distribution - Processing Time

Processing Times	% of Ocurrances
1 day	17.80%
2 days	45.50%
3 days	28.80%
> 3 days	7.90%

The first exercise, called EXA has all the ships with volume equal to $100,000m^3$ which correspond to an income of \$50,000 per ship, and 3 days of availability size ($a_s = l_s - e_s + 1$).

Table 10: Experiments Layout EXA

EXA - HO				$\kappa = 1.47$
	e(s)	l(s)	vol(s)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	4	6	100000	
2	2	4	100000	
3	3	5	100000	
4	1	3	100000	
5	3	5	100000	
6	2	4	100000	
7	3	5	100000	
8	1	3	100000	

EXA - MO				$\kappa = 0.89$
	e(s)	l(s)	vol(s)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	3	5	100000	
2	10	12	100000	
3	9	11	100000	
4	12	14	100000	
5	7	9	100000	
6	1	3	100000	
7	11	13	100000	
8	4	6	100000	

EXA - LO				$\kappa = 0.57$
	e(s)	l(s)	vol(s)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	2	4	100000	
2	20	22	100000	
3	5	7	100000	
4	23	25	100000	
5	10	12	100000	
6	6	8	100000	
7	15	17	100000	
8	26	28	100000	

The second exercise, called EXB has all the ships with exactly the same e_s and l_s from EXA (meaning that κ is also the same), but ships can have volumes of $100,000m^3$, $200,000m^3$ or $300,000m^3$. The average volume of all requests increased from $100,000m^3$ in EXA to $200,000m^3$. The expected revenue in those cases are $50,000m^3$, $100,000m^3$ and $150,000m^3$ respectively.

Table 11: Experiments Layout EXB

EXB - HO				$\kappa = 1.47$
	e(s)	l(s)	vol(s)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	4	6	200000	
2	2	4	200000	
3	3	5	100000	
4	1	3	300000	
5	3	5	200000	
6	2	4	300000	
7	3	5	200000	
8	1	3	100000	

EXB - MO				$\kappa = 0.89$
	e(s)	l(s)	vol(s)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	3	5	200000	
2	10	12	200000	
3	9	11	100000	
4	12	14	300000	
5	7	9	200000	
6	1	3	300000	
7	11	13	200000	
8	4	6	100000	

EXB- LO				$\kappa = 0.57$
	e(s)	l(s)	vol(s)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	2	4	200000	
2	20	22	200000	
3	5	7	100000	
4	23	25	300000	
5	10	12	200000	
6	6	8	300000	
7	15	17	200000	
8	26	28	100000	

The third exercise, called EXC has all the ships with the same volume as EXB, but the availability size of each ship can be 3, 4 or 5 days. The average availability size increased from 3 days in EXB to 4 days. κ is updated according to the new definition of e_s and l_s. This exercise is closely related to what is encountered in real settings, on which volumes and availability sizes are normally different among all requests.

It is important to highlight that is not possible to isolate completely the changes from one exercise to the other, because some parameters are correlated. For example, increasing the availability size from 3 days to 3, 4 or 5 days can be performed in multiple ways which impact not only the availability size, but directly the e_s, l_s and κ. Therefore, the index that was chosen to guide those changes in all exercises was keeping κ at the same range.

Table 12: Experiments Layout EXC

EXC - HO				κ = 1.40																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	3	7	200000																															
2	2	5	200000																															
3	3	5	100000																															
4	5	8	300000																															
5	4	6	200000																															
6	2	5	300000																															
7	5	9	200000																															
8	1	4	100000																															

EXC - MO				κ = 1.11																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	8	12	200000																															
2	10	13	200000																															
3	1	3	100000																															
4	3	6	300000																															
5	6	8	200000																															
6	10	13	300000																															
7	2	6	200000																															
8	11	14	100000																															

EXC- LO				κ = 0.64																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	20	24	200000																															
2	1	4	200000																															
3	25	27	100000																															
4	8	11	300000																															
5	28	30	200000																															
6	2	5	300000																															
7	13	17	200000																															
8	8	11	100000																															

The fourth exercise, called EXD is similar to EXC in terms of volumes but the availability sizes are all 5 days, which gives more flexibility to the terminal to choose a date in the appointment scheduling process.

Table 13: Experiments Layout EXD

EXD - HO				κ = 1.52																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	3	7	200000																															
2	1	5	200000																															
3	2	6	100000																															
4	4	8	300000																															
5	2	6	200000																															
6	1	5	300000																															
7	5	9	200000																															
8	1	5	100000																															

EXD - MO				κ = 1.13																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	8	12	200000																															
2	10	14	200000																															
3	1	5	100000																															
4	3	7	300000																															
5	6	10	200000																															
6	10	14	300000																															
7	2	6	200000																															
8	11	15	100000																															

EXD - LO				κ = 0.75																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	20	24	100000																															
2	1	5	100000																															
3	25	29	100000																															
4	8	12	100000																															
5	26	30	100000																															
6	1	5	100000																															
7	13	17	100000																															
8	7	11	100000																															

The last exercise, called EXE has the same availability sizes of 3, 4 or 5 days from EXC but the volume is the same as in EXA (equal to $100,000m^3$). The average volume is again reduced to $100,000m^3$.

Table 14: Experiments Layout EXE

EXE - HO				κ = 1.40																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	3	7	100000																															
2	2	5	100000																															
3	3	5	100000																															
4	5	8	100000																															
5	4	6	100000																															
6	2	5	100000																															
7	5	9	100000																															
8	1	4	100000																															

EXE - MO				κ = 1.03																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	8	12	100000																															
2	10	13	100000																															
3	1	3	100000																															
4	3	6	100000																															
5	6	8	100000																															
6	10	13	100000																															
7	2	6	100000																															
8	11	14	100000																															

EXE- LO				κ = 0.64																														
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	20	24	100000																															
2	1	4	100000																															
3	25	27	100000																															
4	8	11	100000																															
5	28	30	100000																															
6	2	5	100000																															
7	13	17	100000																															
8	8	11	100000																															

Table 15 summarises the main differences between all exercises considering their volumes, availability sizes and earliest/latest definitions.

Table 15: Differences Experiments

	Volume	Availabilities	Earliest/Latest
EXA	All ships with 100.000 m3	All ships with 3 days	-
EXB	2 ships with 100.000 m3; 4 ships with 200.000 m3; 2 ships with 300.000 m3;	Same as EXA	Same as EXA
EXC	Same as EXB	2 ships with 3 days; 4 ships with 4 days; 2 ships with 5 days;	-
EXD	Same as EXB	All ships with 5 days	-
EXE	Same as EXA	Same as EXC	Same as EXC

The results from applying EXA, EXB, EXC, EXD and EXE as inputs of Algorithm 1 are showed respectively on Table 16, 17, 18, 19 and 20. Those results will be called as " Base Results" as those are the ones that will be used in most experiment comparisons. The running time increases according to the number of dx_s combinations tested (meaning that the greater the availability sizes, the greater the running time). In EXA and EXB for a set of $N_{ships} = 8$ ships the running time was between one to two hours, for EXC and EXE was between eight to nine hours and for EXD between one day and a half to two days.

All experiments were performed using a Microsoft Surface with Intel(R) Core(TM) i5-1035G4 CPU @ 1.10GHz 1.50 GHz, and 2022.02.3+492 version of R software.

Those tables can be read as following: the first information is that 8 ships are operated at once (if they were operated in clusters it would be an indication with the number of ships in each cluster, such as [4 4], [2 2 2 2], etc). Next it is shown the number of scenarios tested for each combination (N_{scen}) followed by the report of the total number of dx_s combinations. The solution is the simulated optimal set of dx_s that returns the maximum profit (indicated as E[P]). Next it is showed the mean and standard deviation of $dz_s - dx_s$, first for all ships together, and then stratified by the ones that were early or on time (called E/OT) and the ones that were late (called LT).

Table 16: Result Base EXA

	EXA - HO	EXA - MO	EXA - LO
Number of Ships	**8 [8]**	**8 [8]**	**8 [8]**
# Scenarios	1,000	1,000	1,000
# Combinations	65,536,000	65,536,000	65,536,000
Solution	6 NA 5 NA 5 NA 5 1	3 12 NA 14 9 1 13 6	2 20 5 25 12 8 17 28
E[$]	$ 132,000.00	$ 214,875.00	$ 316,575.00
Dev [min; max; mean; sd]	2.0; 3.1	1.4; 2.2	0.5; 1.9
Dev E/OT [min; max; mean; sd]	0.2; 2.0	0.2; 1.6	-0.7; 1.2
Dev LT [min; max; mean; sd]	4.5; 1.9	3.2; 1.6	2.3; 1.1

Table 17: Result Base EXB

	EXB - HO	EXB - MO	EXB - LO
Number of Ships	**8 [8]**	**8 [8]**	**8 [8]**
# Scenarios	1,000	1,000	1,000
# Combinations	65,536,000	65,536,000	65,536,000
Solution	6 4 NA 2 5 4 5 NA	3 12 NA 14 9 1 13 6	2 20 5 25 12 8 17 28
E[$]	$ 501,100.00	$ 614,875.00	$ 716,575.00
Dev [min; max; mean; sd]	3.0; 3.7	1.4; 2.2	0.5; 1.9
Dev E/OT [min; max; mean; sd]	0.8; 2.6	0.2; 1.6	-0.7; 1.2
Dev LT [min; max; mean; sd]	6.3; 2.3	3.2; 1.5	2.3; 1.1

Table 18: Result Base EXC

	EXC - HO	EXC - MO	EXC - LO
Number of Ships	8 [8]	8 [8]	8 [8]
# Scenarios	1,000	1,000	1,000
# Combinations	360,000,000	360,000,000	360,000,000
Solution	7 2 NA 8 6 2 9 NA	12 13 NA 3 6 13 2 13	22 4 26 11 30 2 17 10
E[$]	$ 548,325.00	$ 617,075.00	$ 720,850.00
Dev [min; max; mean; sd]	1.9; 2.6	1.4; 2.7	0.5; 1.9
Dev E/OT [min; max; mean; sd]	0.6; 1.9	-0.1; 1.8	-0.8; 1.2
Dev LT [min; max; mean; sd]	4.0; 1.8	3.6; 1.9	2.3; 1.2

Table 19: Result Base EXD

	EXD - HO	EXD - MO	EXD - LO
Number of Ships	8 [8]	8 [8]	8 [8]
# Scenarios	1,000	1,000	1,000
# Combinations	1,679,616,000	1,679,616,000	1,679,616,000
Solution	7 2 NA 8 6 2 9 NA	9 14 NA 3 7 14 2 15	22 5 26 12 30 2 17 10
E[$]	$ 548,325.00	$ 629,225.00	$ 722,300.00
Dev [min; max; mean; sd]	1.9; 2.6	1.2; 2.2	0.4; 1.9
Dev E/OT [min; max; mean; sd]	0.6; 1.9	-0.1; 1.5	-0.8; 1.2
Dev LT [min; max; mean; sd]	4.1; 1.8	3.1; 1.4	2.3; 1.1

Table 20: Result Base EXE

	EXE - HO	EXE - MO	EXE - LO
Number of Ships	8 [8]	8 [8]	8 [8]
# Scenarios	1,000	1,000	1,000
# Combinations	360,000,000	360,000,000	360,000,000
Solution	7 NA 3 8 NA NA 9 1	12 13 1 NA 6 13 4 13	22 4 26 11 30 2 17 10
E[$]	$ 159,850.00	$ 218,875.00	$ 320,850.00
Dev [min; max; mean; sd]	1.3; 2.1	1.4; 2.6	0.4; 1.9
Dev E/OT [min; max; mean; sd]	0.0; 1.4	0.0; 1.8	-0.7; 1.2
Dev LT [min; max; mean; sd]	3.1; 1.4	3.6; 1.8	2.3; 1.1

Overall the more spread the requests are over the agenda, the smaller the rejection rates (which are indicated by "NA" reported in the "Solution" line), going from 25% in cases of HO to 0% in cases of LO. The mean deviation in operation start (given by $dz_s - dx_s$) also decreases from HO to LO cases. For ships that arrive early/on time this indicator is mostly close to 0, and for late ships they were greater then 2 days in all exercises.

Having the "Base Results" reported, the next step is going back to the research questions proposed in Chapter 1 and defining which of the input exercises should be used in each analysis. Those questions are divided in three categories: **Methodological approaches**, which are questions about large sized instances and particularities from the industry; **Dynamics between terminals and clients**, which approaches questions about different appointment scheduling structures and levels of information sharing that involves negotiation between those two parties; and finally **Terminal Operational Structure** which explores questions about the operational process. The design of each category's questions is defined as follows:

Methodological approaches

The first question is *"What is the best clustering technique that can be used in the partition methodology? And what is the impact of using those partition methods to take appointment decisions when compared to the optimal profit?"*. To answer this question all exercise settings (EXA, EXB, EXC, EXD and EXE) were considered to test ten different clustering methodologies. The results were then compared and one of them is chosen to be the cluster-based partition method proposed. For further reference this experiment will be called as ***Clustering Methodology***.

The second research question is *"What is the value generated by the optimization model when compared to manually performed appointments?"*. This question was answered through an appointment scheduling game proposed to a real team of schedulers. In this game, they should perform manually the appointment scheduling of EXA, EXB, EXC and EXE considering the setting from **Problem 1** (static scheduling), and EXC considering the setting from **Problem 3** (dynamic scheduling that allows adjustments in past appointment decisions). The results from the manual procedure are then compared to the ones given by the solvers (more specifically Algorithm 1, for the static case, and Algorithm 3A, for the dynamic case). For further reference this experiment will be called as ***Optimized versus Manual Appointment Scheduling***.

The third question is *"Which differences can be acknowledged in the dz_s calculation when comparing a simple terminal to a complex one?"*. As previously explained, there are discrete event simulation systems specialized in the recreation of complex logic decisions. In this experiment an ARENA model is developed, which reproduces the particularities of a case study terminal. The base appointment of EXC will be tested as input of the simulation model and results of expected profit and dz_s calculation will be compared. For further reference this experiment will be called as ***Complex dz_s calculation cases*** experiment.

Dynamics between terminals and clients

The fourth research question is *"What is the value of having the volume information previous to the appointment scheduling process?"*. This answer was given by comparing: 1) the results of EXB and EXC considering all vol_s being equal to the average of historical data applied as inputs of Algorithm 1; 2) the results of EXB and EXC considering all vol_s being zero applied as inputs of Algorithm 2; 3) the base results from EXB and EXC. This comparison shows how valuable is knowing the volume information before the appointment scheduling process. For further reference this experiment will be called as ***Value of Volume Information***.

The fifth question is *"What is the value of having more flexibility given by clients in the appointment scheduling dates?"*. This question is answered by comparing the base results from EXA with EXE (which have the same volume), and EXB with EXC and EXD (which also have the same volume). This comparison shows the impact of having more flexibility in the available days to decide on the appointment scheduling. For further reference this experiment will be called as ***Value of More Availability Size***.

The sixth question is *"What is the impact of answering clients as requests are placed compared to having all requests before taking those decisions (dynamic x static scheduling)? What is the impact of allowing adjustments in appointments previously set in the dynamic scheduling procedure?"*. The first part of the question is answered by a comparison between the base results having EXA, EXB, EXC and EXE as inputs (which considers static scheduling) with the results of the same exercises tested as inputs of Algorithm 3A (which considers dynamic scheduling). The difference between those results shows how valuable is having full information about all requests before taking any appointment decision, when compared to answering each client without knowing what future requests will be.

The second part of the question compares the results of exercise EXB and EXC tested as inputs of Algorithm 3B (dynamic scheduling allowing adjustments in previously set appointments) with the results of the same exercises as inputs of Algorithm 3A (already calculated for the first part of the exercise). Allowing those adjustments could be considered as a mid-term solution between answering the clients right away, but allowing the terminal to make adjustments in that decision that could guarantee better profits. Different ranges of "penalty cost" for those adjustments are also analysed. This experiment is further referenced as ***Static versus Dynamic Scheduling***.

The seventh question is *"What is the impact of not having dx_s defined?"*. In this experiment the base results of EXA, EXB, EXC, EXD, EXE are compared to the results of the same exercises applied as inputs of Algorithm 5. Both models represent different realities, in the first one dy_s is dependent on the definition of dx_s and a random variable of $delay_s$ and in the second one, there is no definition of dx_s and dy_s is calculated by an uniformly distributed function over the availability days adding a $delay_s$ (with the same statistical properties previously defined). This question tackles the endogenous/ not endogenous arrivals topic and their financial impact. For further reference this experiment will be called as dx_s ***Definition versus Acceptance Only***.

Terminal Operational Structure

The eighth question is *"Which berthing rule provides higher expected profits?"*. This test will compare the base results from exercises EXB and EXC (that consider FIFO berthing rule) with results when applying them as inputs of Algorithm 5 ("by schedule" berthing rule), showing which berthing rule performs best. For further reference this experiment will be called as ***Berthing Rules***.

The ninth question is *"What is the impact of increasing variability in processing and arrival times?"*. This test will compare results from exercises EXB and EXC tested as inputs of Algorithm 1 with standard deviation of 0, 1, 2 and 3 days in the arrival delays (which is defined by a normal distribution with mean 0) and the same standard deviations of the associated normal distribution of the processing times (which is modeled as a truncated normal distribution with mean of 2 days). The results will show the impact of increasing/decreasing the variability level of both arrival delays and processing times. For further reference this experiment will be called as ***Uncertainty Levels***.

The tenth question is *"What is the impact of not having coordination between schedulers and operational teams around the berthing rule?"*. For this test the results of exercises EXB and EXC applied as inputs of Algorithm 1 and Algorithm 4 will be used to compare optimal and non-optimal solutions. This test is important to quantify the impact of taking appointment scheduling decisions considering a different berthing rule from the one used by operational teams. For further reference this experiment will be called as ***Coordination between schedulers and operational teams***.

5.2 Research Questions

5.2.1 Methodological Approaches

A. Clustering Methodology

The results from EXA, EXB, EXC, EXD and EXE showed that running all combinations of dx_s can be time consuming even for small instances such as $N_{ships} = 8$ ships. Figure 10 shows that solution time increases with an exponential shape when increasing the number of combinations. This can be observed when comparing results from EXA/EXB (on which all ships have availability size of 3 days), with EXC (on which all ships have of 3, 4 or 5 days) and EXD (on which all ships have 5 days).

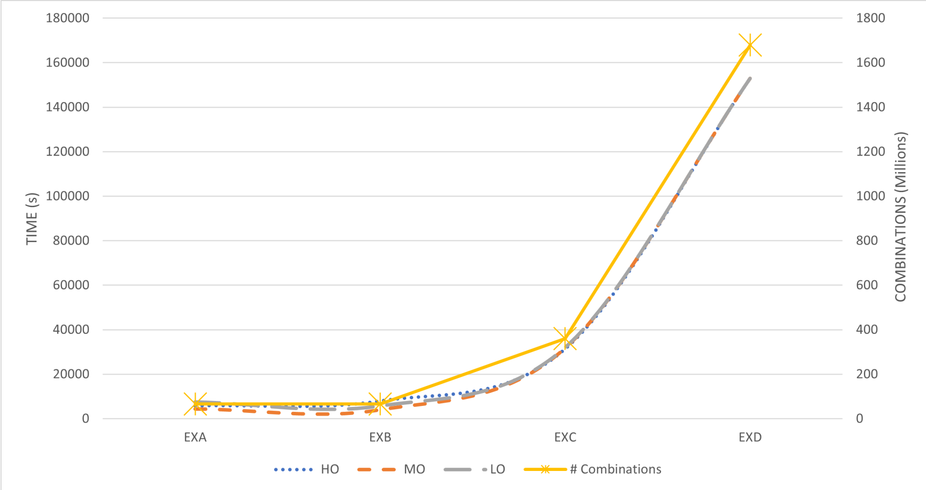

Figure 10: Time and Combinations

In order to reduce the total amount of combinations tested, a partition methodology is proposed inspired by the "Cluster First, Route Second" used in vehicle routing problems. The main idea is decomposing the main problem into smaller ones and scheduling them sequentially, which leads to an approximated result. Table 21 shows an example of a very simple clustering technique on which a set of $N_{ships} = 8$ ships are partitioned given their request order (which are showed in groups separated by a "/"), called further on as "baseline clusters".

Table 21: Clustering Order Baseline

	Clusters
[8]	1 2 3 4 5 6 7 8
[4 4]	1 2 3 4/ 5 6 7 8
[2 2 2 2]	1 2/ 3 4/ 5 6/ 7 8
[1 1 .. 1 1]	1/ 2/ 3/ 4/ 5/ 6/ 7/ 8

Following the "Cluster First, Schedule Second" methodology, each exercise (EXA, EXB, EXC, EXE) will now have an extra information defining from which cluster c_s each ship pertains. Those exercises are then used as inputs of Algorithm 1 (Partitioned Scheduling) defined in "Annex 6: Using the cluster based partition method" for the scheduling step.

The results in Figure 11 shows the difference in expected profit between the optimal solution (running all ships at once - [8]) and the baseline decomposed structures proposed ([4 4], [2 2 2 2] and [1 1 .. 1 1]).

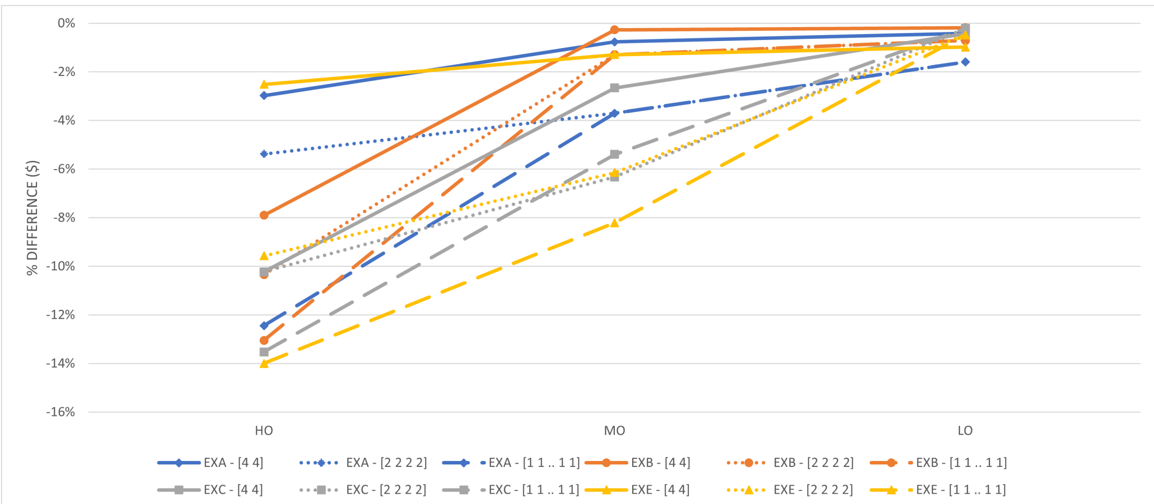

Figure 11: Comparison Partition Sizes

The experiment shows that in general decreasing cluster size has a negative impact in the expected profit. In EXA for example, for the HO case the difference between [8] and [4 4] expected profit results is -3.0%, between [8] and [2 2 2 2] is -5.4% and between [8] and [1 1 .. 1 1] is -12.4%. This difference is smaller in MO and LO cases. For EXA in the LO case, the difference between [8] and [4 4] expected profit results is -0.4%, between [8] and [2 2 2 2] is -1.6% and between [8] and [1 1 .. 1 1] is also -1.6%.

Although the solution is now approximated when compared to the optimal [8] solution, the time needed to encounter the solution decreased from more than 8 hours to minutes or even seconds in partitioned instances, as showed for EXC in Table 22.

Table 22: EXC Times and Combinations

	[8]	[4 4]	[2 2 2 2]	[1 1 .. 1 1]
EXC - HO	8.7 h	1.8 m	5.6 s	2.4 s
EXC - MO	8.9 h	1.2 m	5.6 s	2.5 s
EXC - LO	8.8 h	1.2 m	6.2 s	2.4 s
Combinations	360,000,000	1,200,000	100,000	4,000

Normally it is expected that the smaller the sizes of the clusters are the greater the gap when from the base results. There are some exceptions to that, which might happen due to the myopic nature of this kind of "divide and conquer" methodology. EXC MO is an example, on which the result of the [2 2 2 2] was \$578,045, slightly worse than [1 1 .. 1 1] which was \$583,775. Figure 12 details the reason why those kind of situations might happen.

For the first four ships the results of running Algorithm 1 (Partitioned Scheduling) considering clusters of two ships was exactly the same as considering clusters of one ship only. After that, when the schedule of Cluster 3 was being performed via [2 2 2 2], which has the requests of ship 5 and 6, the best solution encountered (considering the ones previously scheduled) was $dx_5 = 6$ and $dx_6 = 13$ (with an expected profit of \$545,700). Any other solution was worse than this one, even $dx_5 = 8$ and $dx_6 = 13$ which returns an expected profit of \$544,525.

When compared to scheduling ship 5 and ship 6 via [1 1 .. 1 1], decisions were slightly different. Because the [1 1 .. 1 1] just have information of one ship at each stage of the scheduling, the best solution for only scheduling ship 5 (considering the ones previously scheduled) was $dx_5 = 8$ with expected profit of \$427,425. This solution is better when compared to any other, even $dx_5 = 6$ which returns an expected profit of \$425,000. For ship 6 the best schedule was $dx_6 = 13$.

If there were only those 6 ships, the solution of [2 2 2 2] (which was, up to now, $dx_1 = 8$, $dx_2 = 13$, $dx_3 = 2$, $dx_4 = 5$, $dx_5 = 6$ and $dx_6 = 13$) would be better than the one given by [1 1 .. 1 1] (which was, up to now, $dx_1 = 8$, $dx_2 = 13$, $dx_3 = 2$, $dx_4 = 5$, $dx_5 = 8$ and $dx_6 = 13$).

But for both there are still 2 ships missing to be scheduled, and because in [2 2 2 2] the decision made was to schedule $dx_5 = 6$ the possible combinations of scheduling the following ships have smaller expected profit when compared to the choice made in [1 1 .. 1 1] of scheduling $dx_5 = 8$.

Summarizing, clustering into smaller problems creates myopic instances which do not consider information of ships allocated in future clusters to perform the scheduling procedure. A decision made in a previous cluster, although being the best decision up to that point, can lead to combinations with reduced expected profit in future steps.

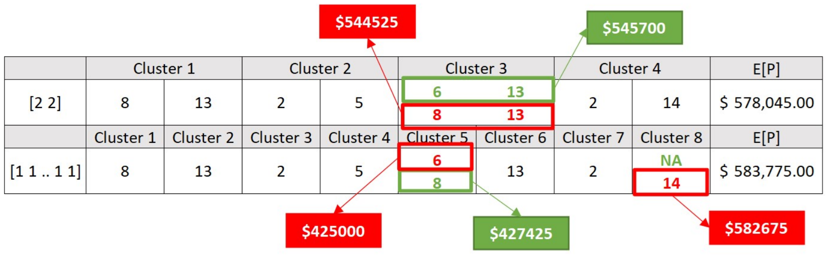

Figure 12: Myopic Results

The gap in expected profit between [8] and [4 4]/[2 2 2 2]/[1 1 .. 1 1] can be increased/ decreased depending on the clustering method used in the partition. Up to now only clustering by the request order was presented, but there are multiple ways for clustering that group of ships. The question that stands is: Which clustering methodology is more convenient for this kind of problem?

Due to the fact that the clustering method sets the order in which ships will be scheduled in the second stage, a good clustering methodology should be able to cluster ships in a decreasing rank of priority, avoiding that ships with higher financial contribution are rejected by the end of the scheduling procedure. Finding this ordering is not straightforward as ships profit is a function of volume, time spent in berth and port's congestion level (which is a consequence of the scheduling procedure itself). Therefore, ships characteristics that appear to be more relevant for the clustering procedure are volume, earliest/latest and availability size.

Ten different clustering methodologies will be evaluated, being two machine learning methodologies, called k-means and hierarchical agglomerative clustering (HAC), and the other eight are practical ordering systems, called "Lowest Earliest First" (LEF), "Highest Earliest First" (HEF), "Lowest Volume First" (LVF), "Highest Volume First" (HVF), "Lowest Latest First" (LLF), "Highest Latest First" (HLF), "Lowest Availability First" (LAF) and "Highest Availability First" (HAF).

K-means goal is to take data points and group them in a way that members from each cluster are as similar as possible between each other and different as possible from the ones in other clusters. The algorithm starts with initially selected centroids (depending on the number of clusters, k, previously defined), each one representing a different cluster. Then the distance of each data point to all centroids is calculated and data points are reassigned to the cluster with smaller distance. Then the centroids are updated as the center of that new group of data points. This is performed iteratively until no reassignment is made, which defines the final clusters.

Hierarchical agglomerative clustering is another method with a bottom-up approach. Initially each data point is considered an unique cluster and distance between all clusters is calculated (also called dissimilarities). The pair of clusters with smaller distance is then grouped, forming a unique cluster. Again distance between all clusters is calculated, and the pair of clusters with smaller distance is grouped. This is performed iteratively until the number of clusters previously defined is reached.

Both k-means and HAC algorithms are defined in "Annex 4: k-means and HAC Clustering Algorithms", which were performed considering a pre defined number of 2 clusters each. Although the number of clusters is fixed by the user, the number of ships considered in each cluster might vary (for example [5 3], one cluster with 5 ships and the other with 3 ships).

Multiple versions of those two machine learning techniques were tested given different input data, which were: 1) only earliest and latest (called E/L); 2) earliest, latest, volume and availability (called E/L/V/A); 3) for the exercises on which all requests have the same volume or availability size it was considered only the information that is different (called E/L/V - when availability is the same - or E/L/A - when volume is the same). Table 23 summarizes the results of clustering by k-means and HAC, which are the most elaborated methodologies.

Table 23: Clustering Order

	HO	MO	LO
EXA - KM2 E/L	2 4 6 8/ 1 3 5 7	1 6 8 2/ 3 4 5 7	1 3 5 6/ 2 4 7 8
EXB - KM2 E/L	2 4 6 8/ 1 3 5 7	1 6 8 2/ 3 4 5 7	1 3 5 6/ 2 4 7 8
EXC - KM2 E/L	5 6 7 8/ 1 2 3 4	3 4 5 7/ 1 2 6 8	2 4 6 7/ 8 1 3 5
EXE - KM2 E/L	2 3 6 8/ 1 4 5 7	1 2 6 8/ 3 4 5 7	2 4 6 7 8/ 1 3 5
EXB - KM2 E/L/V	2 4 6 8/ 1 3 5 7	1 6 8/ 2 3 4 5 7	1 6 8/ 2 3 4 5 7
EXC - KM2 E/L/V/A	1 4 7/ 2 3 5 6 8	3 4 5 7/ 1 2 6 8	1 2 4 6 7 8/ 3 5
EXE - KM2 E/L/A	2 3 5 6 8/ 1 4 7	1 2 6 8/ 3 4 5 7	1 2 4 6 7 8/ 3 5
EXA - HAC E/L	4 8/ 1 2 6 7 3 5	6 1 8/ 5 4 7 2 3	8 2 4/ 1 3 6 5 7
EXB - HAC E/L	2 4 6 1 3 5 7/ 8	2 3 8/ 6 1 7 4 5	8 5 6/ 1 2 4 3 7
EXC - HAC E/L	4 7/ 8 3 2 6 1 5	1 8 2 6/ 5 3 4 7	2 6 7 4 8/ 1 3 5
EXE - HAC E/L	2 3/ 8 5 4 7 1 6	1 4 2 3/ 7 5 6 8	1 7 8/ 2 4 5 3 6
EXB - HAC E/L/V	2 4 6 8/ 1 3 5 7	4 6/ 1 2 3 5 7 8	3 6 1 5/ 8 4 2 7
EXC - HAC E/L/V/A	4 5 6 7 8/ 1 2 3	1 2 3 4 5/ 5 6 7 8	1 2 3 4 5 6/ 7 8
EXE - HAC E/L/A	1 4 7/ 2 3 5 6 8	1 2 6 8/ 3 4 5 7	1 2 4 6 7 8/ 3 5

The practical ordering systems have straightforward logic (LEF/HEF, LLF/HLF, HVF/LVF, HAF/LAF). The case of high volume first (HVF), for example, requests are ordered by their volume (from the highest to the lowest) and clusterized given a specific number of groups (such as two groups of four requests [4 4]).

To perform this experiment EXA, EXB, EXC and EXE will be ordered by each of those clustering rules with the definition of a new column that defines the cluster c_s from which each ship belongs. As an example, Table 24 shows EXC ordered by HVF and clustered in two clusters of 4 requests each.

Table 24: Example of EXC clustered by HVF

EXC - HO					$\kappa = 1.40$
	e(s)	l(s)	vol(s)	cluster	
4	5	8	300000	1	
6	2	5	300000	1	
1	3	7	200000	1	
2	2	5	200000	1	
5	4	6	200000	2	
7	5	9	200000	2	
3	3	5	100000	2	
8	1	4	100000	2	

EXC - MO					$\kappa = 1.11$
	e(s)	l(s)	vol(s)	cluster	
4	3	6	300000	1	
6	10	13	300000	1	
1	8	12	200000	1	
2	10	13	200000	1	
5	6	8	200000	2	
7	2	6	200000	2	
3	1	3	100000	2	
8	11	14	100000	2	

EXC- LO					$\kappa = 0.64$
	e(s)	l(s)	vol(s)	cluster	
4	8	11	300000	1	
6	2	5	300000	1	
1	20	24	200000	1	
2	1	4	200000	1	
5	28	30	200000	2	
7	13	17	200000	2	
3	25	27	100000	2	
8	8	11	100000	2	

Having the clusters defined, next step is applying those exercises as inputs of Algorithm 1 (Partitioned Scheduling). Figure 13 shows a radar chart with the difference in expected profit from the partitioned methods tested and their optimal solutions. The closer the data is to the outer perimeter, the closer the difference is to 0%.

HO, MO and LO were analysed separately, and as it can be seen the negative impact of using different methodologies is more significant for HO cases (with differences up to 21%) and smaller for LO cases (with differences up to 8%).

In general, HAC E/L and HVF showed better results, with an negative gap of less than 4% for all experiments tested. The ones with worse results for HO were KM2 E/L, LLF, LVF and LEF and for LO were some results of LLF and HLF.

The HVF was chosen as the clustering methodology for the partition method due to the fact that it is a very simple rule, as well as intuitive to practitioners, and easy to control the sizes of clusters.

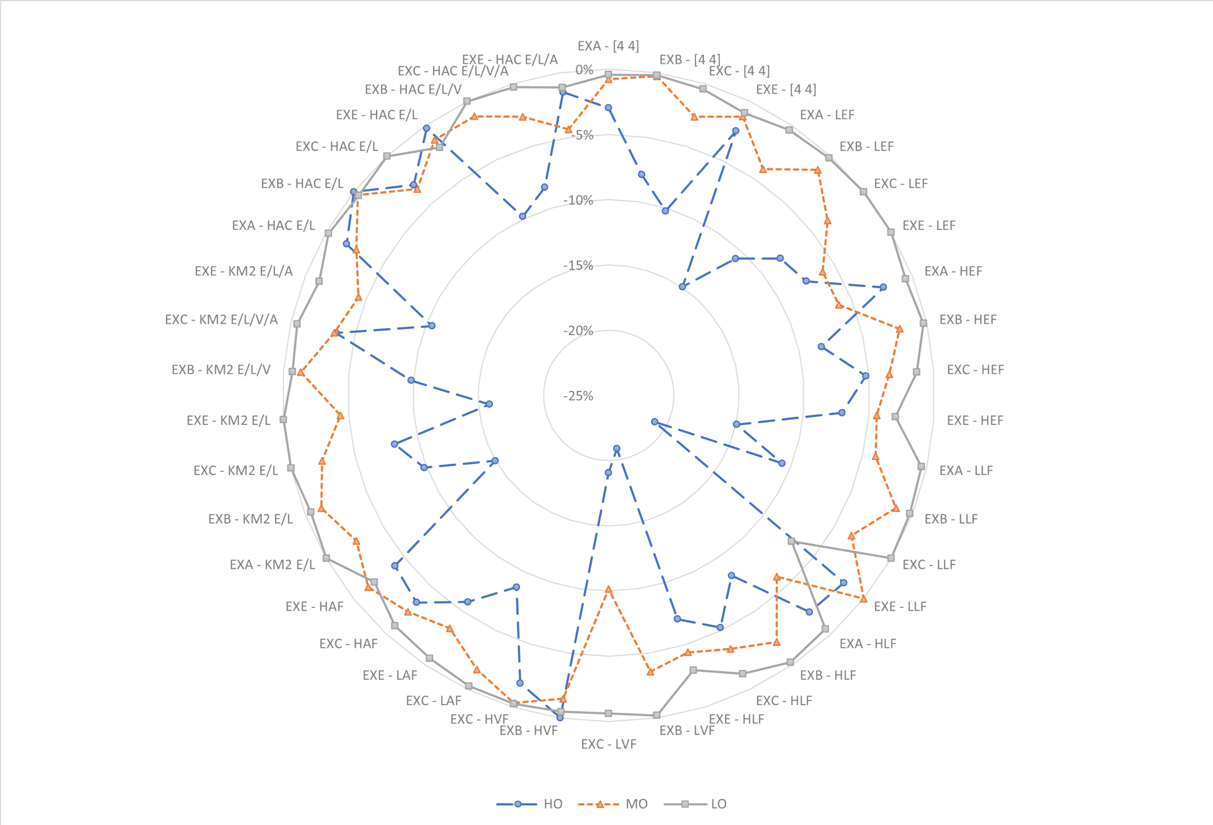

Figure 13: Comparison Clustering Methods

The answer for the research question ***What is the best clustering technique that can be used in the partition methodology? And what is the impact of using those approximation methods in the total profit?*** is that "High Volume First" (HVF) showed to be a great clustering methodology to be used in the "Cluster First, Schedule Second" partition. This technique consistently presented a small reduction in the expected profit when compared to the base solution (less than 2%).

B. Optimized versus Manual Appointment Scheduling

This experiment was proposed to compare the results given when performing the appointment scheduling manually to the optimal solution suggested by the optimization model. Therefore a group of schedulers from the case study company was asked to read the Problem 1 setting reported in "Annex 2: Scheduling Game" and follow the instructions given. In total there were 13 people playing the game separately, which lasted between 1.5 to 2.5 hours.

In the first game, requests from EXC were presented one by one and the schedulers had to make the decision of accepting/rejecting each ship and setting the dx_s, without knowing any other request from the future (simulating a dynamic scheduling process). First they performed the HO case, then they performed the MO case, and lastly the LO case.

In the second game, requests from EXA, EXB, EXC and EXE are showed (once per time, first EXA, then EXB, and so on) and schedulers had to take the decision of accepting/rejecting ships and setting the dx_s knowing all information about each set of $N_{ships} = 8$ ships (simulating a static scheduling process).

In the third game, a pre-defined solution is given and the schedulers are invited to criticize and make suggestions on top of it (this solution is actually the "Base Result" of EXC, although schedulers don't have this information). Lastly a post game questionnaire was proposed, opening the opportunity for them to share their decision logic along the process, and insights about the usability of such kind of optimization tool in their daily basis.

Results from Table 25 shows the negative impact that using manual procedure has when compared to using the solver. For the HO case the expected profit reduced up to 25% and for LO case it reduced up 4%. The impact is very similar when comparing the results of the static exercises with dynamic ones.

The post-game questionnaire showed that schedulers had a tendency of accepting most of the ships and spacing them as much as possible without any fixed rule, considering the e_s and l_s information. Additionally, there was a common (and natural) understanding that whenever ships had different volumes, the ones that where bigger should be prioritized and that smaller ones should be somehow isolated in order to avoid queues. After volume, earliest/latest and demurrage were considered the most important parameters in the appointment process.

Schedulers showed enthusiasm in working with such tool, specially as base solutions from which they could perform improvements on top of, incorporating the human knowledge about the details of operation. It was highlighted the importance of considering inventory and pipeline constraints and the interaction with refineries, which normally have priority to be served.

Table 25: Results Schedulers Game

	% Deviation
Static Deviation Mean HO	-25.46%
Static Deviation Mean MO	-16.08%
Static Deviation Mean LO	-2.43%
Dynamic Deviation Mean HO	-23.42%
Dynamic Deviation Mean MO	-14.89%
Dynamic Deviation Mean LO	-3.78%

Those numbers answer the research question ***What is the value generated by the optimization model when compared to manually performed appointments?***. It is possible to observe that specially for HO cases, using the manual procedure has a significant negative impact (up to 25%) when compared to using the solver. Results are expected to be better when considering the solver, as it can incorporate stochasticity of parameters in the calculation.

C. Complex dz_s calculation cases

As explained previously, although R and many free-coding environments allow for the development of complex algorithms, there are specialized softwares capable of creating highly detailed models such as Discrete Event Simulation tools.

The optimization models proposed from Problem 1 to Problem 5 were developed considering simpler operations with two types of berthing rules: FIFO and "by schedule". Therefore a simulation model based on a real Brazilian oil terminal was built in ARENA to compare the dz_s calculation from the solver and the simulation model, using EXC as an input. Both techniques consider similar $delay_s$ and p_s means and standard deviation, although the solver considers discretization in days, while the simulation is continuous.

The simulation model also incorporates complexities such as: the calculation of flow and time in port dependent on the type of product and type of operation; canal access (only one ship per time/per direction); minimum interval of 3 hours between harbor masters requisitions; and berthing/unberthing restrictions during the night (VLCCs/ secondary loaded ships).

Figure 14, Figure 15 and Figure 16 show the results of: the expected profit provided by the optimization tool from the optimal and sub optimal combinations with 90%, 80%, 70%, 60%, 50% and 25% from the optimal profit; and the results of the same combinations when implemented in the simulation model.

First of all, results show similar decreasing slope tendency in both simulation and solver expected profits. Additionally, it is observed, specially in HO case, that simulation results oscillate around solver ones, which indicates more randomness.

Although expected profit is an important information, it has to be evaluated together with the dz_s calculation which, in the end, is what changes directly whenever there is a different rule/logic being used. Results showed that for the HO case, some combinations returned better expected profits in the simulation, while in others they were better in the solver, with differences up to 11% in absolute numbers.

In the 90% scenario, this was observed because there were less ships being accounted for demurrage in the simulation compared to the optimization model. In the 80%, the proportion of ships that were accounted/ not accounted for demurrage was similar in both simulation and optimization, but the total start mismatch/operational mismatch perceived by the simulation was higher.

One important consideration is that there were situations encountered in the LO case, which showed almost the same expected profit in both (less then 1%), but opposite tendencies in the calculation of the start and the operational mismatch terms.

This reinforces the fact that it is important to have models able to reproduce the problem in more detail to avoid any misleading conclusions, increasing precision and reliability in the results.

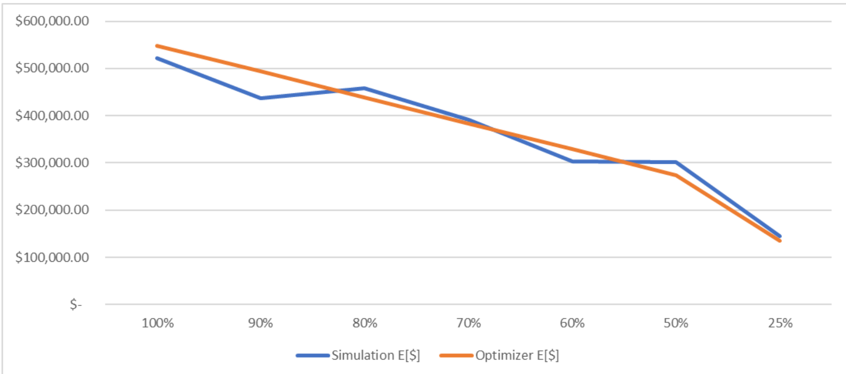

Figure 14: EXC HO Simulation versus Optimization HO

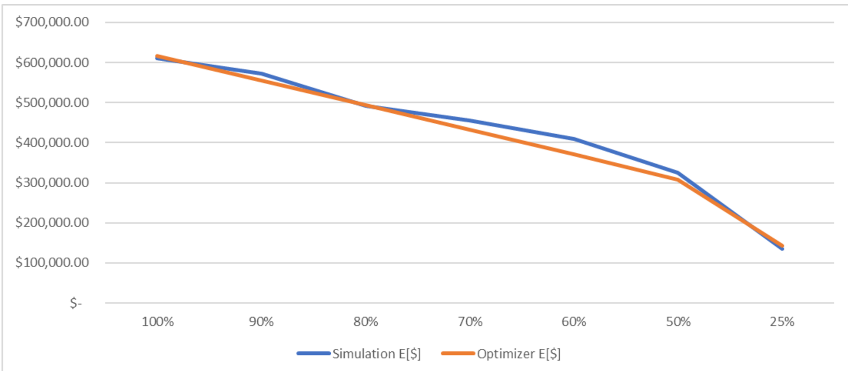

Figure 15: EXC MO Simulation versus Optimization

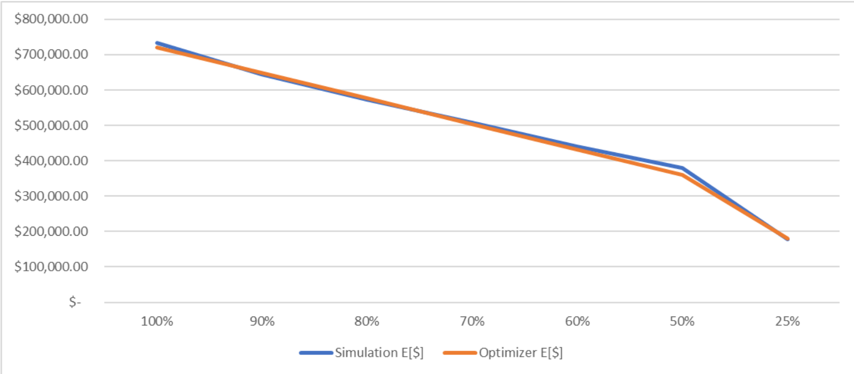

Figure 16: EXC LO Simulation versus Optimization LO

The answer for the research question ***Which differences can be acknowledged in the*** dz_s ***calculation when comparing a simple terminal to a complex one?*** was that it was observed differences in the number of ships that were accounted/not accounted for demurrage calculation and differences (and even opposite results) in the accountability of start mismatch and operational mismatch. Those results reinforce the importance of having models that can incorporate detailed settings, to increase precision and reliability in further decisions.

5.2.2 Dynamics between terminals and clients

D. Value of Volume Information

Next experiment is about the importance of terminals having the volume information when preparing the scheduling plan. In order to do that EXB and EXC are applied as inputs of 1) Algorithm 1 considering all volume data available (also called as V0); 2) Algorithm 1 with all volumes being equal to $200,000m3$, which is the mean volume encountered in the historical data (called V1); 3) Algorithm 2 (called V2) with minimum number of ships to be operated also taken from the historical data (for HO it was considered at least 6 ships, for MO 7 ships and for LO 8 ships).

Results are given in Figure 17 which shows that the impact of knowing this information previously is higher for HO cases and is dependent on the model used to define the solution (for V1 was observed a reduction up to 27% and for V2 up to 38%, when compared to V0). For the LO case not knowing the volumes didn't show any impact in the total expected profit regardless of the model used.

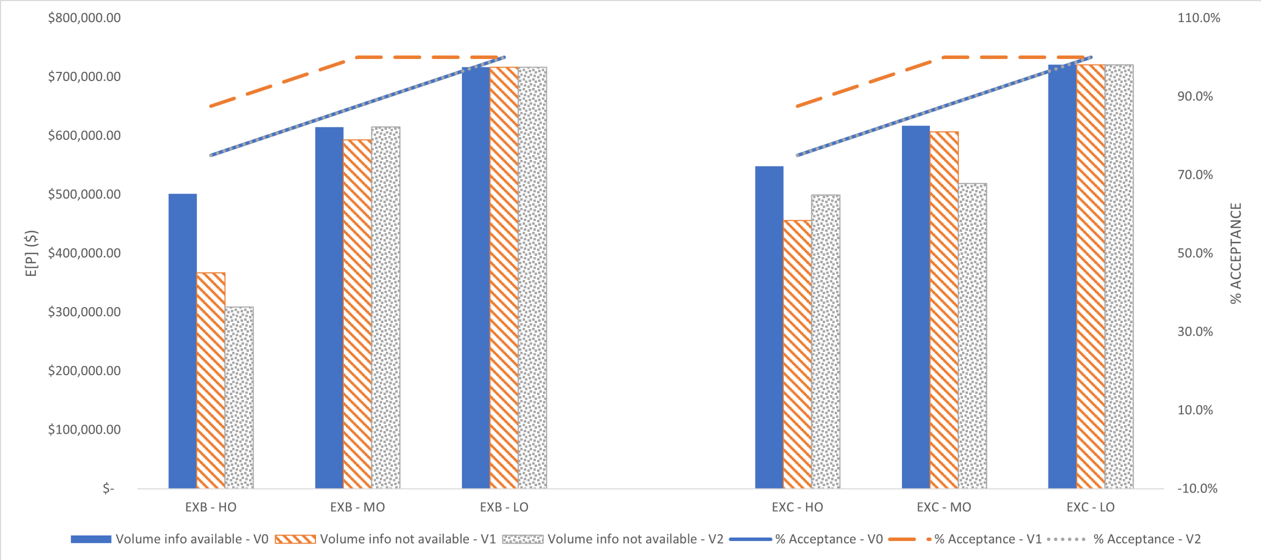

Figure 17: Comparison with and without volume information

The answer for question ***What is the value of having the volume information previous to the appointment scheduling process?*** is: in HO cases knowing volume can highly impact expected profit (up to 38%) while in LO cases no impact was observed. The choice of which model to use depends directly in which historical data is available.

E. Value of More Availability Size

This experiment aims to quantify the value perceived by the terminal whenever the requests have more available days for the scheduling process.

Therefore, EXA will be compared to EXE, and EXB with EXC and EXD. Comparisons are proposed with experiments that have the same volume, but different availability sizes. In EXA and EXB all requests have availability size of 3 days; EXC and EXE have 2 requests with availability size of 3 days, 4 requests with availability size of 4 days and 2 with availability size of 5 days; finally EXD has all requests with availability size equal to 5 days.

Figure 18 shows that increasing availability size has a positive impact on the expected profit which is more expressive in the HO cases (between 9% to 21%) than in the MO and LO ones (less than 3%). It was also observed that from EXC HO to EXD HO, the availability size was non biding.

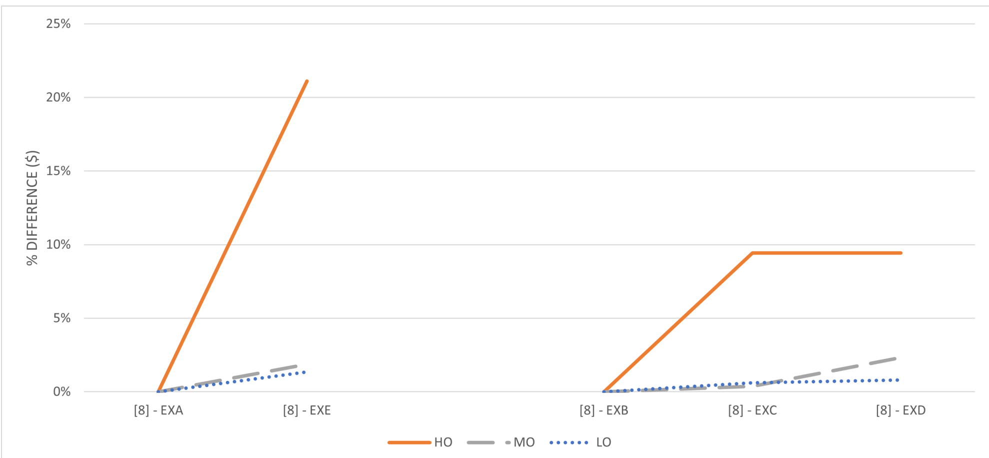

Figure 18: Comparison Different Availability Sizes

The increase in the availability size allows for solutions that are more spread over the agenda, as the ones from EXA and EXE showed in Figure 19.

EXA - HO													
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10
1	4	6	100000						X				
2	2	4	100000										
3	3	5	100000					X					
4	1	3	100000										
5	3	5	100000					X					
6	2	4	100000										
7	3	5	100000					X					
8	1	3	100000	X									

EXE - HO													
	e(s)	l(s)	vol(s)	1	2	3	4	5	6	7	8	9	10
1	3	7	100000							X			
2	2	5	100000										
3	3	5	100000			X							
4	5	8	100000								X		
5	4	6	100000										
6	2	5	100000										
7	5	9	100000									X	
8	1	4	100000	X									

Figure 19: Comparison EXA and EXE Availability Sizes

Therefore the research question ***What is the value of having more flexibility in the availability dates?*** also depends on how overlapped are the requests. Results a increase in expected profit when there is more flexibility on the available days. This increase is more pronounced in HO cases (up to 21%) than compared to MO or LO cases (less than 3%).

F. Static versus Dynamic Scheduling

Next research question evaluates how expected profit changes if the terminal provides instant responses to appointment requests, which can be perceived by some clients as a better service, as opposed to delayed responses.

One way of answering requests instantaneously is using Algorithm 1 (Partitioned Methodology), with clusters of unique requests (called previously as [1 1 .. 1 1]) in their request order, which is actually the same as using Algorithm 3A. Every time a new request arrives, the schedule already prepared (for previous clusters) is assumed as an input which can not be changed, and is analysed together with the possible combinations of schedule for the new request.

This experiment was already performed and showed in Figure 11 comparing the baseline solution for the static scheduling version [8] (in which the 8 requests are scheduled together) to the dynamic scheduling version [1 1 .. 1 1] (in which request 1 is answered, then request 2, then request 3 and so on, exactly in the order they arrive). The results showed an error up to 14% depending on the inputs used.

The difference in expected profit between answering one client per time and accumulating all to answer together is highly dependent on the order those requests arrive to the terminal. Just as an example, consider another situation on which requests are exactly the same as the ones from EXA but instead of arriving as the baseline shows, they arrive in a different order, as an example for the HO case: [request 1 = "baseline request 4", request 2 = "baseline request 8", request 3 = "baseline request 2", request 4 = "baseline request 6", request 5 = "baseline request 3", request 6 = "baseline request 5", request 7 = "baseline request 7", request 8 = "baseline request 1"]. This is the same order as if baseline requests were ordered by the lowest earliest first (LEF).

Figure 20 shows that the difference in expected profit for answering requests that arrived in LEF order was almost -25% (MO case) when compared to "baseline results", which was the higher reduction observed among all the experiments.

This reduction in expected profit is not the same for HO, MO and LO cases and the explanation is similar to the one given for myopic results (Figure 12). Decisions made in previous clusters (which here would be “for the previous requests”) have a direct impact in the total expected profit when considering future requests.

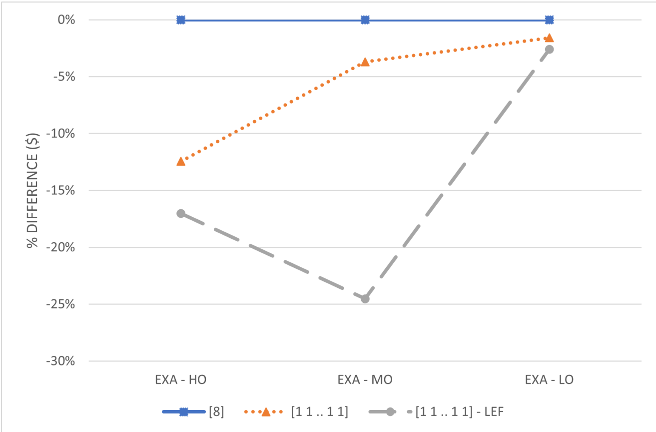

Figure 20: Comparison between requests arrival in the baseline order or LEF order

One possible solution that could be offered to clients is to answer requests as they come, allowing the terminal to perform minor adjustments later on within the earliest/latest requirements (only if necessary). In this situation requests that were already accepted can not be refused, and the ones that were refused can not be considered as possibilities. This new model (which was implemented as Algorithm 3B) takes advantage of previous findings about clustering properties and ordering to recalculate if any rescheduling is needed.

The idea is that every new request is evaluated together with all possible combinations of appointments from previously scheduled ships (considering their e_s and l_s). An additional cost is introduced in the objective function penalizing the deviation from previous appointment definitions. This is a way of discouraging the terminal from performing too many changes that could somehow impact their relationship with the clients.

To overcome the timing issue, the partition method is utilized. In every stage, the new request together with the ones already scheduled are clustered over HVF methodology in groups of 4 or less requests. As an example, consider the setting of EXB HO, on which the optimal solution proposed for $dx_1 = 4$. When the second request arrives, all combinations for scheduling request 1 (minus the “NA”, as this ship can’t be rejected anymore) are considered together with the combinations for request 2. A cost of $\$3,125$ is introduced to penalize every day that the new schedule of ship 1 deviates from the previous schedule (day 4). This cost is considered initially equivalent to 1/8 of the demurrage fee ($\$25,000/8 = 3,125$).

The optimal solution considering those combinations are $dx_1 = 4$ and $dx_2 = 2$. Then the third request arrives, again all combinations for ship 1 and 2 are considered (minus the “NA”) together with combinations of request 3, and the additional costs for rescheduling ship 1 and ship 2. The optimal solution is $dx_1 = 4$, $dx_2 = 2$ and $dx_3 = 5$. Until now rescheduling was not a profitable option.

Next, request 4 arrives and the process is repeated. The optimal solution is $dx_1 = 6$, $dx_2 = 2$, $dx_3 = 5$ and $dx_4 = 3$. This time the model chose to reschedule request 1 in order to accept request 4 and have a greater expected profit, being penalized by $(6 - 4) * 3,125 = 6,250$ dollars.

As explained before, as the algorithm enumerates all possible solutions every time a new ship is added, the resolution time grows exponentially. This is the reason why the partitioned version of the Algorithm is used, considering HVF ordering and clusters of 4 requests (or less). Until now ordering was being performed, but it didn't change the solution as all ships were being scheduled at the same time.

Continuing the example, consider that request 5 arrives, now all ships are ordered by HVF and separated in two clusters, one with the first 4 requests and the other with the remaining 1 request. All combinations are tested (minus "NA" for ships that were not rejected before) considering the rescheduling cost. The optimal solution for those 5 ships is $dx_1 = 6$, $dx_2 = 2$, $dx_3 = 5$, $dx_4 = 3$ and $dx_5 = 5$. This routine is repeated for every new ship and the problem is now solved with reduced solution time.

Figure 21 compares the difference in expected profit between the optimal solution [8], the 8 ships problem partitioned in [4 4] - HVF, the [1 1 .. 1 1] baseline and [1 1 .. 1 1] allowing past corrections with rescheduling fees of $\$6,250$, $\$3,125$ and $\$0$ (multiples of the demurrage fee) for EXB and EXC.

The results show that running [1 1 .. 1 1] with past corrections is not always better than [1 1 .. 1 1] baseline, not even if the cost for rescheduling is $0. As an example, results for EXC LO shows the solution [20 4 27 11 30 2 17 9] with expected profit of $\$71,9325$ for the [1 1 .. 1 1] baseline and [21 2 26 11 30 5 17 10] with expected profit of $\$71,8950$ for the [1 1 .. 1 1] with past corrections (and rescheduling cost of $0). This recalls the discussion about how different ordering and clustering methodologies can impact the solution. In the [1 1 .. 1 1] baseline, ordering is based on request arrivals, and in [1 1 .. 1 1] with past corrections HVF is used.

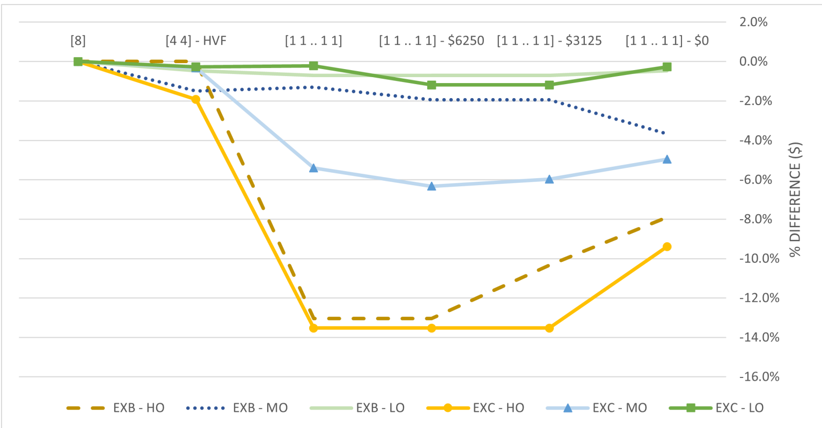

Figure 21: Comparison Dynamic Scheduling Results

Also, it is normally expected that [1 1 .. 1 1] allowing past corrections with $0 penalty has the same solution as [4 4] - HVF, as observed in EXB LO, on which both solutions are [3 20 5 25 12 8 17 28]. This is due to the fact that in the end of running [1 1 .. 1 1] with past corrections, the structure of information is equivalent to [4 4] - HVF (as both use the same ordering and clustering methodology and penalty of $0 applied to alterations). This is not the case when a ship is rejected (or even accepted when it was not supposed to) by the [1 1 .. 1 1] with past corrections, as the example EXB HO with $0. In the [4 4] - HVF case the solution is [6 4 NA 2 5 4 5 NA], Figure 22 shows that when this experiment is performed under [1 1 .. 1 1] allowing past corrections, the third ship is accepted in the third stage, and $dx_3 = 5$, which can not be rejected anymore. That is the reason why in those cases [4 4] - HVF and [1 1 .. 1 1] with past corrections do not have the same results.

EXB - HO ($ 0)	EXB - LO ($ 0)	EXC - LO ($ 0)
1) 4	1) 3	1) 20
2) **6** 2	2) 3 20	2) 20 4
3) 6 2 5	3) **2** 20 7	3) 20 4 27
4) 6 **4** 5 1	4) 2 20 7 25	4) 20 **3** 27 11
5) 6 4 5 1 5	5) **3** 20 7 25 12	5) 20 3 **25** 11 30
6) 6 4 5 **2** 5 4	6) 3 20 **5** 25 12 8	6) **21** 2 **26** 11 30 5
7) 6 4 5 2 5 4 5	7) 3 20 5 25 12 8 17	7) 21 2 26 11 30 5 16
8) 6 4 5 2 5 4 5 NA	8) 3 20 5 25 12 8 17 28	8) 21 2 26 11 30 5 **17** 10

Figure 22: Examples Rescheduling

Figure 23 shows that rescheduling starts to be a considered option if the penalty is around 1/8 of the demurrage fee. Additionally, it depends on the e_s and l_s available and in which ships were already accepted/rejected along the scheduling process, showing that the number of reschedules can vary among HO, MO and LO cases. Overall the differences comparing [1 1 .. 1 1] with [1 1 .. 1 1] allowing past corrections were lower than 6%.

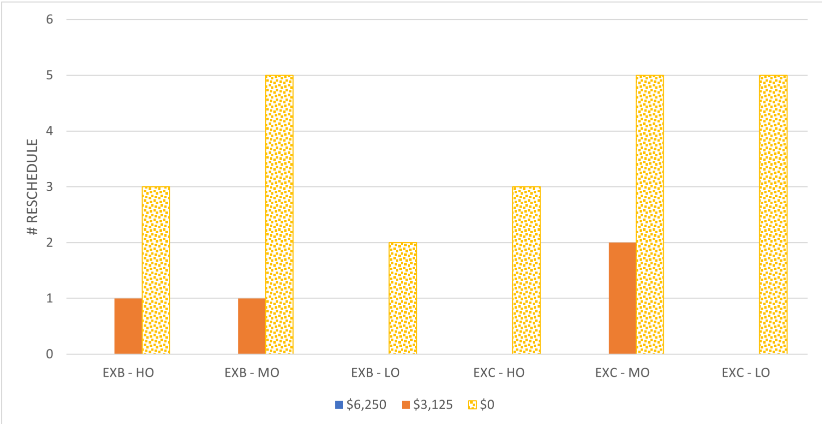

Figure 23: Number Reschedules

The research question ***What is the impact of making decisions without having full information about future requests (dynamic x static scheduling)? What is the impact of allowing past corrections in the dynamic scheduling procedure?*** is answered as following: terminals might consider answering clients right away although it might incur in a financial impact up to 25%. As a mid-term solution it was proposed a dynamic scheduling that allows past corrections, which ensures an instant response to the client, and more flexibility to the terminal searching for solutions with greater profit. The solution given by this new method showed to be slightly better (up to 6%) than [1 1 .. 1 1] depending on the penalty cost used, considered only started to be an option when the penalty was around $\$3,125$.

G. dx_s Definition versus Acceptance Only

The idea of this research question is to compare two different realities. The first one considers the main setting on which ships are accepted/rejected, dx_s is defined and dy_s is endogenous (this problem was modelled by Algorithm 1). The second one considers a setting on which the decision is reduced to only accepting/rejecting ships, with dy_s given by an uniform distribution within the availability days added to a normal distributed delay (this problem was modelled by Algorithm 5).

As in the second case there is no dx_s definition, the demurrage calculation also changes in Algorithm 5. Whenever a ship arrives before the e_s, the terminal should consider e_s as the reference for start mismatch calculation. If dy_s arrives within the availability days, the start mismatch is accounted from the moment it arrives to the terminal. If the ship arrives after l_s, there is no start mismatch accountability.

Figure 24 shows the results of EXA, EXB, EXC, EXD and EXE applied as inputs in Algorithm 1 (referred as P1 in the graph) and Algorithm 5 (referred as P5). Those results show that defining an appointment date, specially in overlapped cases is better than just accepting/rejecting that ship to operate. The first reason is because dx_s definition sets what is the best day to maximize terminals profits, and allowing arrivals to largely deviate from it, is obviously sub optimal. The second reason is that accepting a ship to arrive in any day of the available days increase the period that the ship can have start mismatch accounted.

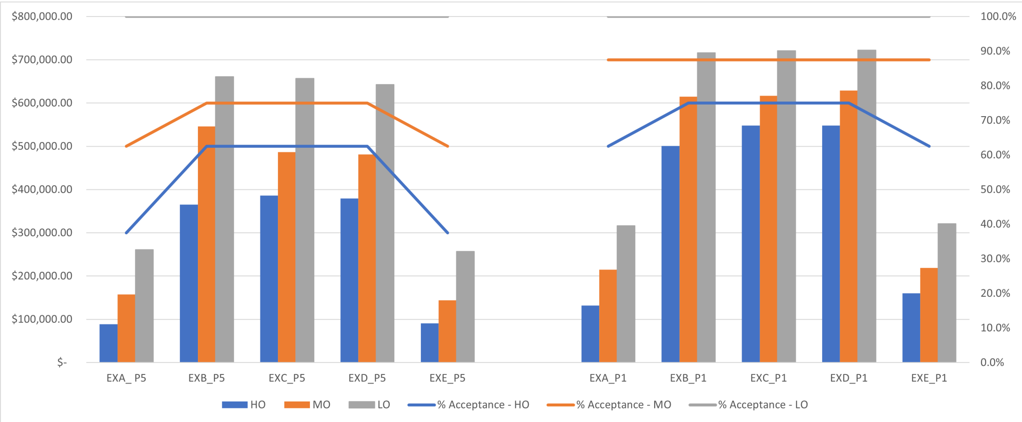

Figure 24: Impact of endogenizing dy_s

As explored in the experiment about availability sizes, the increase in the availability period shows an increase in expected profits (or at least the same results). The opposite tendency is observed when comparing EXA and EXE MO and LO cases, as well as EXB, with EXC and EXD MO and LO cases applied as inputs to Algorithm 5. In those cases, actually increasing the availability size increases the period on which the ship can be accounted for start mismatch, which reduces profit.

The answer for the research question ***What is the impact of not having*** dx_s ***defined?*** is that terminal's profit might decrease when there is no dx_s definition (depending of course in how the demurrage cost is calculated). The extreme results were perceived in EXE HO case with a reduction in expected profit of around 43% and in EXB LO 8%. All other results were in between those values.

5.2.3 Terminal Operational Structure

H. Berthing Rules

Next experiment focus on comparing two different practical berthing rules used by terminals: FIFO and "by schedule". FIFO is the typical procedure that focus on maximizing berth utilization and is encountered, for example, in many Brazilian public ports. The "by schedule" rule focus on following the sequence of operation defined by schedulers. Those rules will be analyzed considering two different tiebreaking rules, the first one prioritizes ships by their order of request (smallest s first) and the second one focus on prioritizing ships with the smallest deviation from the scheduled date (first ships that are on-time, then early ones, then late ones).

The FIFO rule is represented by Algorithm 1 and the FIFO with priorities (as it will be called by addressing the second tiebreaking rule instead of the first) is represented by Algorithm 1 with the adjustments from "Annex 3: Tiebreaking rule: smallest deviation first". The "by schedule" berthing rule is represented by Algorithm 4 and the "by schedule" with priorities is represented by Algorithm 4 with the adjustments from "Annex 3: Tiebreaking rule: smallest deviation first".

Figure 25 compares the expected profit results from EXB and EXC applied as inputs in the four algorithms (FIFO/ "by schedule" and FIFO with priorities/ "by schedule" with priorities). The first result observed is that having a "by schedule" rule in HO and MO cases can reduce the expected profit up to 16% when compared to using FIFO. In those situations on which there is high demand for berth utilization, losing time by waiting ships to arrive instead of advancing whoever is already in the terminal can propagate demurrage as a domino-effect.

The LO case has a different response, actually the "by schedule" showed slightly better results (up to 3%), when compared to FIFO. In this situation, advancing ships result in an unneeded risk for paying demurrage in a context with reduced berth utilization.

The use of the second tiebreaking rule shows to outperform (or at least give the same result) as the first one, with a greater impact in the "by schedule" cases. This rule prioritizes the operation of on-time/ early ships, which are the ones that actually incur in start mismatch.

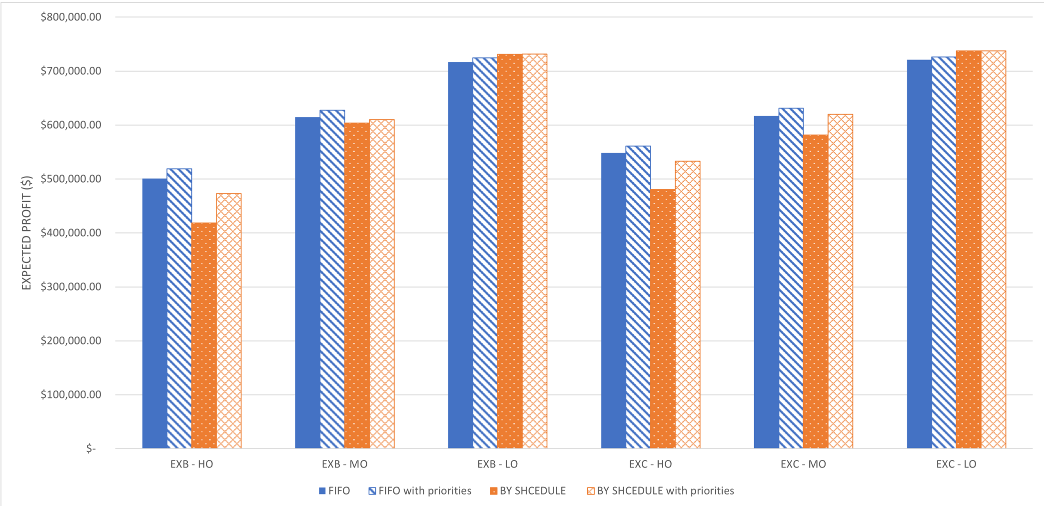

Figure 25: Comparison Different Scheduling Operational Rules

Due to the fact that the "by schedule" rule strictly follows the sequence defined by the schedulers, some terminals apply tolerances to deal with the variability in ship arrivals. This "by schedule with tolerance" rule sets that if the ship arrives until the scheduled date plus a tolerance, they are operated according to the "by schedule rule", otherwise they loose their preference and will be operated whenever the terminal understands there is no impact in other operations. As those type of rules might incur in differences on the demurrage calculation, it is suggested as future research.

The answer to the ***Which berthing rule provides higher expected profits?*** question also depends on how overlapped is the system. For HO and MO cases the FIFO rule showed better results by prioritizing berth utilization (up to 16%). In the other hand, in LO cases the "by schedule" rule showed slightly better results by not advancing operations that could wait (up to 3%). The tiebreaking rule that considers the deviation to the scheduled date showed better results (or at least the same ones) by prioritizing ships that arrive on-time and early, which are the ones that actually incur in demurrage.

I. Uncertainty Levels

In this experiment it will be explored the impact of different uncertainty levels in the processing times and in the arrival delays. Therefore, EXB and EXC will be applied as inputs of Algorithm 1 considering standard deviations of 0, 1, 2 and 3 days in the processing time (which is modelled as a truncated normal distribution with mean of 2 days) and in arrival delays (which was modelled as normal distribution with mean of 0 days).

Figure 26 shows the results for processing times. The greater the variability, the greater is the impact on the expected profit.

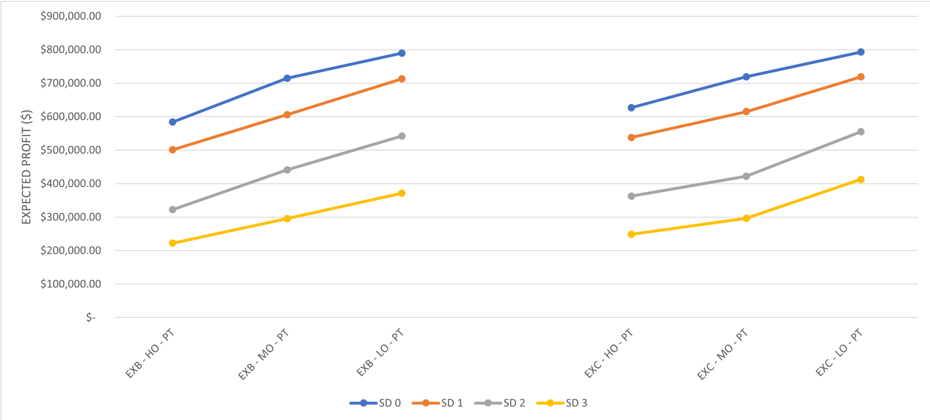

Figure 26: Comparison Different Standard Deviation in Processing Times

However, it is important to observe that those results were a consequence of not only a change in the variability but also in the means of the processing times. Table 26 shows that a change in the variability of the associated normal distribution has a direct impact in means and standard deviation of the actual truncated normal distribution, which is calculated in R via the 'rtruncnorm' formulation. This formulation requires the following parameters: the number of samples to be created (in this case $N = 1,000$), the inferior limit ($a = 1$), the superior limit ($b = Infinite$), the mean of the associated normal and its standard deviation.

RTRUNCNORM (N, a, b, Mean, SD)			
Mean	SD	Actual Mean	Actual SD
2	0	2	0
2	1	2.25	0.77
2	2	2.99	1.36
2	3	3.78	1.96

Table 26: Truncate Normal Distribution

In the case of the arrival delays, the results were surprising. Figure 27 shows that increasing the variability in the arrival delays increase the expected profit observed by terminals.

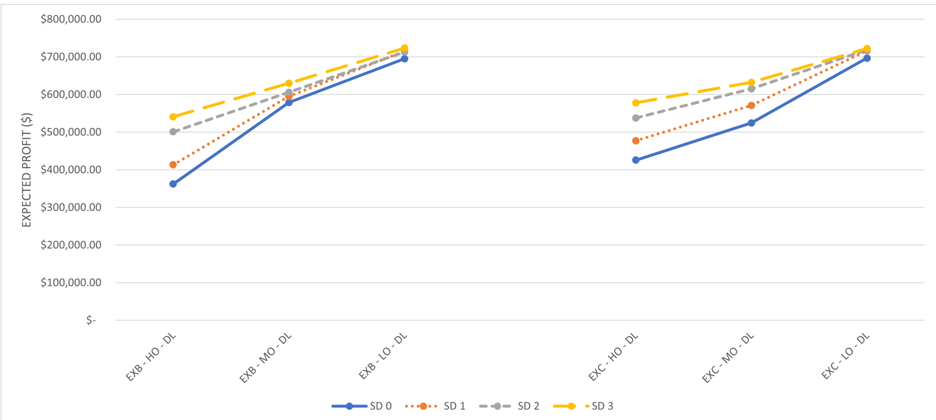

Figure 27: Comparison Different Standard Deviation in Arrival Delays

The explanation can be verified in the example of Figure 28 on which it is presented the results of $dz - dx$ and operational mismatch means, stratified by: ships that are early/on time and late; and by when the demurrage terms are accounted or not (accounted means that the total demurrage was greater than 0). Those results compared information about EXB HO with standard deviation of 2 days (SD2) and EXB HO with standard deviation of 3 days (SD3).

The reasoning behind the increase in expected profit from SD2 to SD3, is given by the fact that the mean deviation of early/on time ships that have their start mismatch accounted, reduced from 6 days to 4.37 days.

Whenever there is more uncertainty in the arrival, more ships will arrive way earlier or later than in a situation with less uncertainty. For ships that are late, the fact that they arrive even later does not have a direct impact in start mismatch, as late ships do not account for that term. On the other hand, ships that arrive earlier, give the terminal more flexibility to advance their operations, reducing possible payments of start mismatch which impacts the total profit.

EXB HO SD3	dz-dx	Operational Mismatch	Number of Ships
Early/ OnTime Accounted	4.37	0.70	1.54
Late Accounted	5.69	1.16	0.93
Early/ OnTime not Accounted	-3.35	0.15	1.85
Late not Accounted	10.59	-0.46	1.68

EXB HO SD2	dz-dx	Operational Mismatch	Number of Ships
Early/ OnTime Accounted	6.00	0.88	1.97
Late Accounted	5.23	1.08	0.86
Early/ OnTime not Accounted	-2.11	0.01	1.62
Late not Accounted	9.42	-0.42	1.54

Figure 28: Example EXB HO with different Uncertainty Levels

Next research question ***What is the impact of having uncertainties, in arrival and processing times?*** is answered as following: increasing the standard deviation of processing times reduces the expected profit. This impact was a result of the difference in means and standard deviation from the truncated normal distribution used to model the processing times. On the other hand, increasing the standard deviation of arrival delays increase the expected profit, which is a result of how the start mismatch term is calculated. On those situations more ships arrive earlier and later. Later ships do not account for start mismatch, while earlier ships give more flexibility to advance operations.

J. Coordination between schedulers and operational teams

Next experiment tests what is the impact of not having alignment between the scheduling team and the operational team about the berthing rule. In those cases the scheduling team might consider a different rule which might impact the appointment result and its expected profit. The exercise will compare the expected profits of using the appointment provided considering a FIFO berthing rule in settings that do not follow FIFO (such as FIFO with priorities, “by schedule” and “by schedule” with priorities).

In this case the same four algorithms used to answer research question about different berthing rules will be used again: Algorithm 1 (representing FIFO), Algorithm 1 with the adjustments from “Annex 3: Tiebreaking rule: smallest deviation first” (representing FIFO with priorities), Algorithm 4 (representing ”by schedule”) and Algorithm 4 with the adjustments from “Annex 3: Tiebreaking rule: smallest deviation first” (representing “by schedule” with priorities).

Figure 29 shows with filled colors the actual expected profit for the solution proposed by the FIFO model, when other operational rules are used in reality. The complement of those columns, which are indicated with a dot pattern, shows the additional expected profit in case schedulers had used the proper berthing rule.

The greater differences were observed in cases on which FIFO is used in a "by schedule"/"by schedule with priorities" setting. As an example, in EXB HO case using the base schedule in a context with a "by schedule with priorities" rule reduces the expected profit in up to 15% if compared to the case on which they use the correct rule. Overall in the LO cases the impact is smaller.

Another result is that in the FIFO with priorities case, considering the proper rule or the FIFO one, did not impact the expected profit.

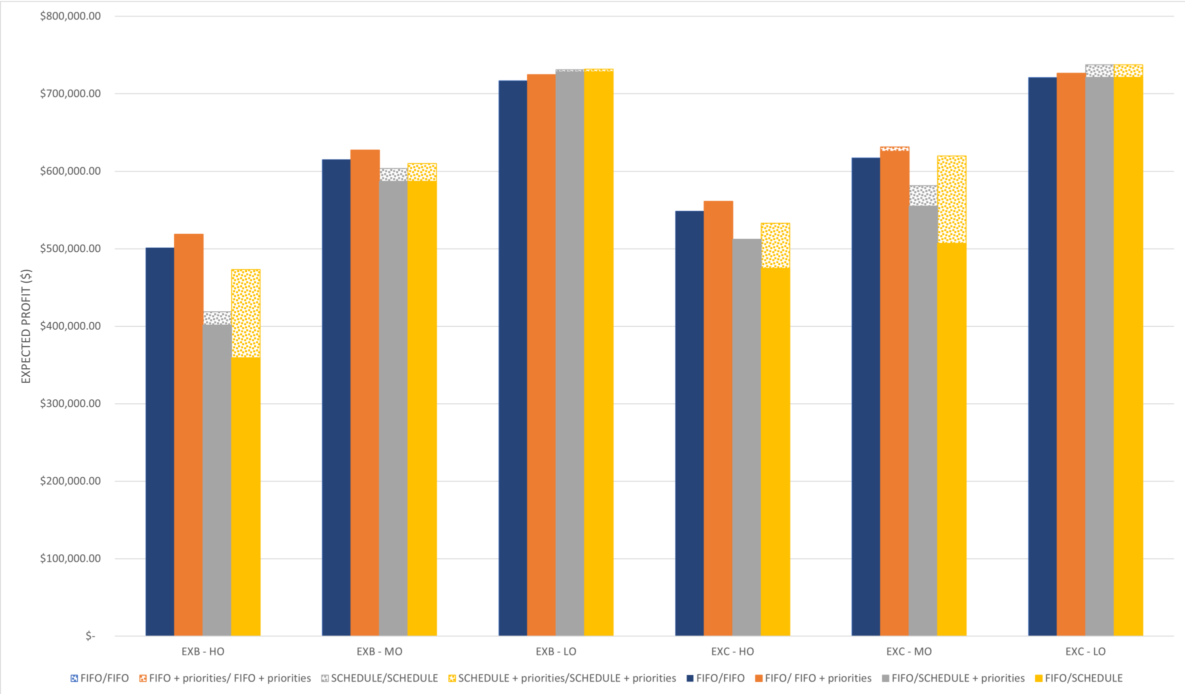

Figure 29: Comparison Coordination Problem

The answer for ***What is the impact of not having coordination between schedulers and operational teams around the berthing rule?*** depends on how overlapped is the problem, and how different the actual rule is from the one considered by the schedulers. Using FIFO in settings that follows a "by schedule" rule has greater impact when compared to using FIFO in a FIFO with priorities context. This difference is even greater when using FIFO in a "by schedule" with priorities setting. This impact is significant in HO and MO cases (up to 15%), but not very significant in LO cases.

5.3 Chapter Summary

This chapter is dedicated to answering the research questions proposed in Chapter 1. Therefore a set of experiments were initially designed as well as the exercises that should be used as inputs for the algorithms developed in Chapter 4, which were categorized in three different topics. As there are terminals that operate with different levels of congestion, an overlap index is proposed to capture any differences that those settings might have.

The result of comparing manual solutions to the simulated optimal shows the impact that such appointment scheduling tools can have, specially for HO and MO cases. Those kind of tools are able to consider multiple variables from a non linear problem and incorporate stochasticity in calculations, which is an advantage when compared to the manual procedure.

In order to solve those problems faster, a cluster based partition method was proposed in Chapter 4 and results from experiment two showed that "Highest Volume First" (HVF) had a great performance. Interestingly, ordering ships by their volumes was also observed when schedulers performed the manual appointment scheduling exercise.

Over the experiments a lot of attention was given to test the value of information in the appointment scheduling process. Results showed that volume and more flexibility in the availability days have significant impact, specially in HO cases.

It was also calculated the negative impact of answering clients right away, compared to waiting up to a certain limit date. One alternative solution explored was answering clients right away, although allowing adjustments to the scheduling previously made. Results from this alternative solution didn't show major differences in results, not even in the HO case.

Considering the operational aspect, FIFO showed better results for HO and MO cases by prioritizing berth utilization. For LO cases it was observed that the "by schedule" rule showed slightly better results by not advancing ships with early arrivals. Another experiment showed the financial impact that the lack of coordination between schedulers and operational teams can cause, which showed to be significant in HO and MO cases, but not very significant in LO cases.

One unexpected finding was that increasing the levels of uncertainty in arrivals actually increases the expected profit. This result is due to the design of the demurrage calculation which is different for early and late ships. A similar test was performed varying the standard deviation of processing times, which showed an expected tendency on which increasing variability decreases expected profit.

Another experiment compared a setting on which dx_s is defined by the scheduler and dy_s is an endogenous variable, with another setting on which dx_s is not defined and dy_s is expected to be distributed around the e_s and l_s with some delays. In this situation as the demurrage calculation is different, allowing for a more spread arrival pattern increases the start mismatch period that can be accounted, resulting in a high and negative effect in the expected profit.

The results from the solver were also compared to results from a simulation system which reproduces some of the complexities encountered in terminals. The deviation in dz_s calculation showed the importance of incorporating the most details possible so the results are adherent to the terminal's reality.

6 Conclusions and Future Research

This research was proposed with three main goals: the first one was framing the appointment scheduling problem under different perspectives to help the designing process of resolutions such as ANP (2022). The second one was providing appointment scheduling models that could support terminals in the optimization process imposed by such resolutions. The third one is exploring managerial insights about the value of information, different scheduling time frames, coordination between scheduling and operational teams, different agenda congestion profiles, uncertainty levels, berthing scheduling rules, among others.

Those objectives were accomplished through five different problem settings which were discussed in Chapter 4 and defined as the following:

- **Problem 1**: will tackle a static terminal appointment scheduling problem, on which decisions are to accept/reject requests with an appointment date definition, considering a FIFO berthing rule and available information about volume, earliest and latest. Two tiebreaking rules are suggested: ordering by the request number (Algorithm 1) and ordering by the smallest deviation with the scheduled date (Algorithm 1 with the adjustments explained in Annex 3).
- **Problem 2**: similar to Problem 1 except for the fact that volume information is not known when scheduling is performed. This problem will be approached by Algorithm 1 considering the mean historical volume, and Algorithm 2 which does not consider volume information at all.
- **Problem 3**: similar to Problem 1 except for the answering time frame which is dynamic instead of static. This problem will be approach by Algorithm 3A (which does not allow past corrections) and Algorithm 4 (which allows past corrections).
- **Problem 4**: similar to Problem 1 except for the berthing rule which is "by schedule" instead of FIFO. Two tiebreaking rules are suggested: ordering by the request number (Algorithm 4) and ordering by the smallest deviation with the scheduled date (Algorithm 4 with the adjustments explained in Annex 3).
- **Problem 5**: will tackle a static terminal appointment scheduling problem, giving a FIFO berthing rule and information about volume, earliest and latest. The decision is reduced to accepting/ rejecting requests based on the earliest/latest informed by the clients. In this case a new demurrage and profit calculation will be discussed. This problem will be modeled by Algorithm 5.

Each problem was approached via different optimization models capable of solving the appointment scheduling setting considering stochasticity in both arrivals and processing times. Additionally a clustering based partition method was provided to reduce the resolution time of large scale instances, which was detailed in Chapter 4.

Finally, in Chapter 5 experiments and analysis were based on the models proposed, allowing the answering of all research questions, and interesting managerial insights about value of information, different scheduling time frames, coordination between scheduling and operational teams, different agenda congestion profiles, uncertainty levels, berthing scheduling rules.

Highlighting some of the findings and further recommendations, results show that specially overlapped terminals can have significant improvements in profit by using solvers such as the ones presented. Also, giving incentives to customers could be studied to get more up front information about the operation as well as to increase flexibility in the available days.

Answering clients statically showed better results, as the terminal is able to take the decision with full information. In case clients value dynamic answers, a suggestion would be offering it as a premium service to reduce the overall impact.

In terms of berthing rules, FIFO presented good results for terminals with congested agendas, while by schedule rule was better in low overlap situations. In case of simultaneous arrivals, prioritizing by smallest deviations is recommended.

One surprising result is that uncertainties in arrivals can, in some cases be beneficial, but accepting time windows instead of a scheduled date is not. And lastly we showed the importance of detailed simulations whenever decisions depend on the results and not only on general tendencies.

There is no knowledge up to now about models that approached terminal appointment scheduling problems under stochastic settings which also considered endogenous arrivals. The fact that this problem is current being reviewed by the industry shows the great potential of academic evolution on the topic.

Among future researches identified, the following ones can be highlighted:

- Including iterative negotiation with clients
- Allowing for appointments outside the time windows, no-shows and cancellations (Gocgun and Puterman, 2014)
- Including despatch fees that bonify the terminal for faster operations
- Including multiple berths and integrating with berth allocation systems
- Exploring other berthing rules such as "by schedule with tolerance"
- Adding inventory and pipeline restrictions, which was highlighted in the results of the Scheduling Game
- Developing algorithms capable of solving the problems with reduced scenarios
- Allowing set up between different operations (Cunha et al., 2020) and tanks cleanup (Cankaya et al., 2019)
- Allowing spot operations that were not scheduled before, as explored by Ala and Chen (2022) in hospital settings
- Integration with other types of simulation models

7 Annex

7.1 Annex 1: Linearization Process of Model 1

Step 1: Define a new binary variable called u_s which is 1 when ship $s \in S$ is accepted to operate, and 0 otherwise. Update Equation 3, 4 and 1 by Equation 8, 9 and 10 respectively, having defined $u_s = \mathbb{1}^{(dx_s > 0)}$:

$$dx_s \geq e_s u_s \tag{8}$$

$$dx_s \leq l_s u_s \tag{9}$$

$$P = \sum_{s \in S} [\Phi_s vol_s - \Omega_s(\tau_s^+)] u_s \tag{10}$$

Step 2: Define a new binary variable called w_s which is 1 if $dy_s \leq dx_s$ and 0 otherwise. The well function of w_s is guaranteed by the addition of two constraints. If $w_s = 0$, Equation 11 shows that $dy_s \geq dx_s$. If $w_s = 1$, Equation 12 shows that $dx_s \geq dy_s$.

$$dx_s - dy_s \leq M w_s - E \tag{11}$$

M is considered a very large number

E is considered a very small number

$$dy_s - dx_s \leq M(1 - w_s) \tag{12}$$

Update Equation 2 using $w_s = \mathbb{1}^{(dy_s \leq dx_s)}$.

$$\tau_s = \underbrace{[dz_s - dx_s] w_s}_{start\ mismatch} + \underbrace{(p_s - \theta_s)}_{operational\ mismatch} \tag{13}$$

The merge of Equations 10 and 13 result in Equation 14, which non-linearity (multiplication of two binary variables $w_s u_s$) is handled in the next step.

$$P = \sum_{s \in S} [\Phi_s vol_s - \Omega_s max[(dz_s - dx_s) w_s u_s + (p_s - \theta_s) u_s, 0]] \tag{14}$$

Step 3: Define a new binary variable called $z_s = w_s u_s$. The well function of z_s is guaranteed by the addition of three constraints. If $w_s = 0$ or/and $u_s = 0$, z_s is 0 by Equation 15 or 16. If $w_s = 1$ and $u_s = 1$, z_s is 1 by Equation 17.

$$z_s \leq w_s \tag{15}$$

$$z_s \leq u_s \tag{16}$$

$$z_s \geq w_s + u_s - 1 \tag{17}$$

Update Equation 14 using $z_s = w_s u_s$.

$$P = \sum_{s \in S} [\Phi_s vol_s - \Omega_s max[(dz_s - dx_s)z_s + (p_s - \theta_s)u_s, 0]] \tag{18}$$

Now the challenge is to tackle the non-linearity caused by the multiplication of the integer variable $(dz_s - dx_s)$ and the binary variable (z_s) from Equation 18, handled in the next step.

Step 4: Define a new variable called $h_s = (dz_s - dx_s)z_s$. Four other constraints are added which guarantee the well function of h_s. In the case $z_s = 0$, $h_s = 0$ is assured by Equation 19 and 20. In the case $z_s = 1$, $h_s = (dz_s - dx_s)$ is assured by Equation 21 and 22.

$$h_s \geq -Mz_s \tag{19}$$

$$h_s \leq Mz_s \tag{20}$$

$$h_s \geq (dz_s - dx_s) - M(1 - z_s) \tag{21}$$

$$h_s \leq (dz_s - dx_s) + M(1 - z_s) \tag{22}$$

Update Equation 18 using $h_s = (dz_s - dx_s)z_s$.

$$P = \sum_{s \in S} [\Phi_s vol_s - \Omega_s max[h_s + (p_s - \theta_s)u_s, 0]] \tag{23}$$

The following step will be linearizing the $max[h_s + (p_s - \theta_s)u_s, 0]$ from the objective function (Equation 23).

Step 5: Define a new variable called $v_s = max[h_s + (p_s - \theta_s)u_s, 0]$ (which can be seen as $v_s = max[a, b]$). Also define a binary variable called t_s, which is 1 when $h_s + (p_s - \theta_s)u_s > 0$ $(a >$

b) and 0 otherwise. Six new constraints are added to ensure the well function of v_s and t_s. In the case $t_s = 1$, Equation 26 and 28, guarantee that $v_s = max[h_s + (p_s - \theta_s)u_s, 0] = h_s + (p_s - \theta_s)u_s$. In the case $t_s = 0$, Equation 27 and 29, guarantee that $v_s = max[h_s + (p_s - \theta_s)u_s, 0] = 0$.

$$h_s + (p_s - \theta_s)u_s - 0 \leq Mt_s \tag{24}$$

$$0 - (h_s + (p_s - \theta_s)u_s) \leq M(1 - t_s) \tag{25}$$

$$v_s \geq h_s + (p_s - \theta_s)u_s \tag{26}$$

$$v_s \geq 0 \tag{27}$$

$$v_s \leq h_s + (p_s - \theta_s)u_s + M(1 - t_s) \tag{28}$$

$$v_s \leq 0 + Mt_s \tag{29}$$

Update Equation 23 using $v_s = max[h_s + (p_s - \theta_s)u_s, 0]$.

$$P = \sum_{s \in S} [\Phi_s vol_s - \Omega_s v_s] \tag{30}$$

This partially linearized model has the following variables and constraints:

Problem 1/ Model 1 (Partially Linearized)

Parameters:

- S= set of ships to be scheduled
- N_{ships} = number of ships to be scheduled
- D = set of days available for scheduling
- N_{days} = number of days from the scheduling period
- Φ_{days} = revenue fee in dollars per cubic meter transported
- Ω_{days} = demurrage fee in dollars per day
- e_s= expected earliest day that the ship s could arrive to the terminal
- l_s= expected latest day that a ship s should departure
- vol_s= expected latest day that a ship s should departure

- p_s= processing time of ship s, which follows a given statistical distribution
- $delay_s$= delay of ship s, which follows a given statistical distribution
- θ_s= time-window scheduled for ship s
- m_s= sequence index corresponding to the operation of ship s, which in this case is defined by the arrival order (FIFO).
- M= very large number
- E= very small number

Variables:

- dx_s is the date ship s is scheduled to arrive and start operation ($dx_s = 0$ means ship s was not scheduled/accepted to operate).
- dy_s is the date ship s arrives at the terminal, defined by $dy_s = dx_s + delay_s$.
- dz_s is the date ship s starts operation.
- u_s is a binary variable which is 1 when ship s is accepted to operate, and 0 otherwise.
- w_s is a binary variable which is 1 if $dy_s \leq dx_s$ and 0 otherwise.
- z_s is a binary variable, defined by $z_s = w_s u_s$.
- h_s is an integer variable, defined by $h_s = (dz_s - dx_s) z_s$.
- v_s is an integer and greater or equal to zero variable, defined by $v_s = max(h_s + (p_s - \theta_s)u_s, 0)$.
- t_s is a binary variable which is 1 when $h_s + (p_s - \theta_s)u_s > 0$ and 0 otherwise.

$$
\begin{aligned}
&\underset{dx_s}{\text{maximize}} \quad E\left[\sum_{s\in S}[\Phi_s vol_s - \Omega_s v_s]\right] \\
&\text{subject to} \quad & dx_s \geq e_s u_s & \qquad \forall s \in S \\
&& dx_s \leq l_s u_s & \qquad \forall s \in S \\
&& dz_s \leq M dx_s & \qquad \forall s \in S \\
&& dy_s = dx_s + delay_s & \qquad \forall s \in S \\
&& dx_s - dy_s \leq M w_s - E & \qquad \forall s \in S \\
&& dy_s - dx_s \leq M(1 - w_s) & \qquad \forall s \in S \\
&& z_s \leq w_s & \qquad \forall s \in S \\
&& z_s \leq u_s & \qquad \forall s \in S \\
&& z_s \geq w_s + u_s - 1 & \qquad \forall s \in S \\
&& h_s \geq -M z_s & \qquad \forall s \in S \\
&& h_s \leq M z_s & \qquad \forall s \in S \\
&& h_s \geq (dz_s - dx_s) - M(1 - z_s) & \qquad \forall s \in S \\
&& h_s \leq (dz_s - dx_s) + M(1 - z_s) & \qquad \forall s \in S \\
&& h_s + (p_s - \theta_s)u_s - 0 \leq M t_s & \qquad \forall s \in S \\
&& 0 - (h_s + (p_s - \theta_s)u_s) \leq M(1 - t_s) & \qquad \forall s \in S
\end{aligned}
$$

$$v_s \geq h_s + (p_s - \theta_s)u_s \qquad \forall s \in S$$

$$v_s \geq 0 \qquad \forall s \in S$$

$$v_s \leq h_s + (p_s - \theta_s)u_s + M(1 - t_s) \forall s \in S$$

$$v_s \leq 0 + Mt_s \qquad \forall s \in S$$

$$dz_s = max\,[dy_s, 1] \qquad s|m_s = 1$$

$$dz_s = max[dy_s, (dz_{s'} + p_{s'})|$$

$$m_{s'} = m_s - 1] \qquad \forall s \in S|m_s > 1$$

$$dx_s, dy_s, dz_s, v_s \geq 0\, and\, integers$$
$$u_s, w_s, zs, t_s\, are\, binary$$
$$h_s\, is\, integer$$

7.2 Annex 2: Scheduling Game

Problem Setting

This exercise aims to reproduce terminals appointment scheduling procedure considering uncertainties in the arrival and processing times of ships. Clients have to send their requests, called s, until a specific limit date from the previous month of the ship operation, informing their volume (vol_s) as well as the period they are available to be scheduled (between the earliest (e_s) and latest (l_s) days). The difference $l_s - e_s + 1$ is called availability size and impacts terminals flexibility when preparing the scheduling plan. As an example, a client wants to operate a ship s in a terminal next month, which is November, so until the 20th of October a request should be sent, informing the volume of the operation, e.g 300,000 m^3, and the earliest and latest days on which the ship could be scheduled, e.g between day 15 and day 18 (meaning that it can be scheduled to start on day 15, day 16, day 17 or day 18).

The terminal has to evaluate all requests focusing on maximizing its profit and decide which ships should be accepted/rejected to operate, as well as the date they should be scheduled, called dx_s. The date when the ship really arrives is uncertain and is called dy_s, and the date that it really starts operation is called dz_s.

It is possible that a ship arrives early, late or exactly on time to operate. Additionally, the operation can start earlier, later or exactly when it was scheduled. The operation can only start when the ship arrives to the terminal ($dz_s \geq dy_s$) and takes an uncertain amount of time to finish, called processing time (p_s). All those information will be used to calculate the demurrage paid by the terminal for every extra day that a ship stays in the terminal, discounting any delays on ship arrivals.

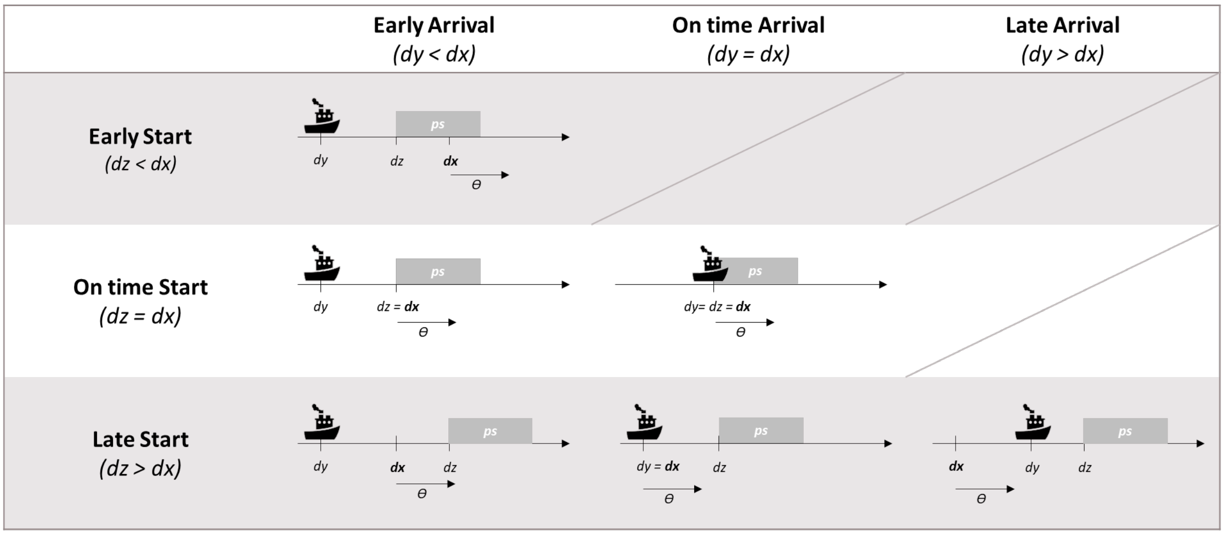

Figure 30: Arrival and Start Time Cases

Consider that the terminal has a revenue of \$0.50 per m^3 transported, and it has to pay a daily demurrage fee of \$25,000 for every extra day that a ship stays in the terminal. The extra days that a ship stays in the port is calculated by the sum of two parts: the start mismatch (dz_s - dx_s), which is the difference between the real start and the scheduled start, only counted when ships arrive early or on time (late ships are responsible for their own delays and can not charge terminals for start mismatch); and the operational mismatch ($p_s - \theta_s$), which is the difference between real processing time and the time window scheduled for that operation. The time window considered for this exercise will be fixed to all ships and equal to 2 days, meaning the terminal is scheduling 2 days of time window to finish any operation. The total profit is calculated by the sum of revenues (volume times the revenue fee per m^3) minus the costs of demurrage (total number of extra days times the demurrage fee) for all accepted ships. Table 27 shows some examples on how to calculate those extra days:

Situation	Scheduled Time (dx_s)	Time Window (θ_s)	Expected End ($dx_s + \theta_s$)	Real Arrival (dy_s)	Real Start (dz_s)	Real Processing (p_s)	Real End (dz_s + p_s)	Start Mismatch (dz_s - dx_s). $1^{dy_s \leq dx_s}$	Operational Mismatch (p_s - θ_s)	Total Demurrage ($\tau^+ = max(SM + OM, 0)$)
1)	4	2	6	2	3	2	5	3-4 = -1	2-2 = 0	Max(-1,0) = 0
2)	4	2	6	2	3	3	6	3-4 = -1	3-2 = 1	Max(0,0) = 0
3)	4	2	6	2	5	1	6	5-4 = 1	1-2 = -1	Max(0,0) = 0
4)	4	2	6	2	5	2	7	5-4 = 1	2-2 = 0	Max(1,0) = 1
5)	4	2	6	2	2	5	7	2-4 = -2	5-2 = 3	Max(1,0) = 1
6)	4	2	6	5	6	3	9	LATE!	3-2 = 1	Max(1,0) = 1

Table 27: Days of demurrage

The fact that ships are scheduled in certain days do not imply that they will be operated in that order. It is important to distinguish the scheduling phase, focus of this exercise, with the operational phase, which happens one month later, with ships arriving around the schedule dates and being operated given a sequence defined by the operational team. Consider to this exercise a "First Come, First Serve" operational policy.

The arrivals dy_s are normally distributed around dx_s with a standard deviation of 2 days. This means, as shown in Figure 31, that in 19.7% of the cases ships arrive exactly on time, 17.5% arrive 1 day early and the same percentage arrive 1 day late, 12.1% arrive 2 days early and the same percentage arrive 2 days late and 0.7% arrive more than 2 days early and the same percentage arrive more than two days late.

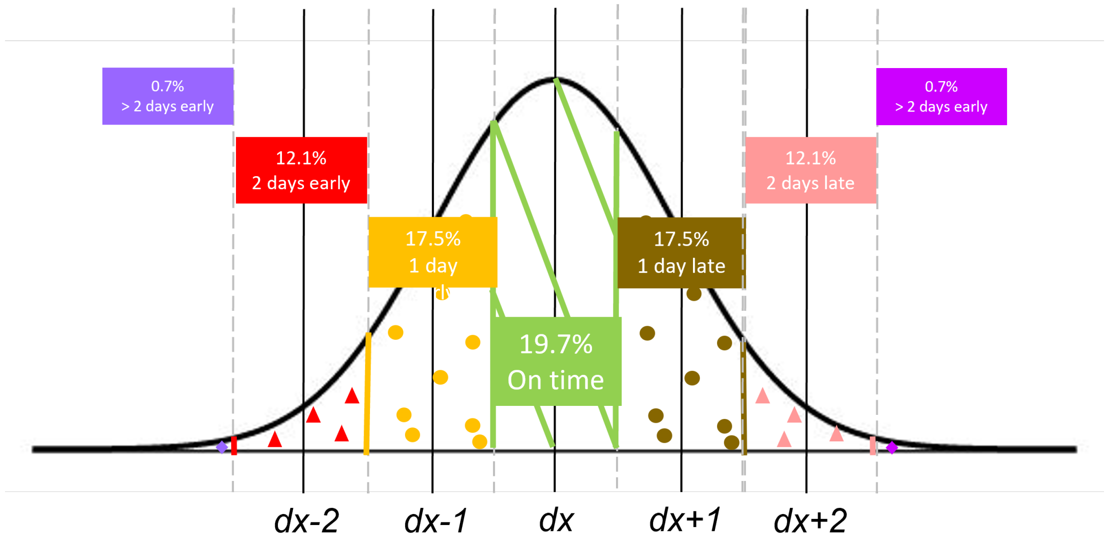

Figure 31: Normal Distribution - Delay

The processing time is also based on a normally distributed with mean of 2 days and standard deviation of 1 day, with a difference that an operation is never smaller than one day. This means, as shown in Table 28 that 17.8% of the cases ships take 1 day to operate, 45.5% of the cases ships take 2 days, 28.8% of the cases ships take 3 days and 7.9% of the cases ships take more than 3 days. Operations start at day 1 of each month, meaning that ships scheduled in November are never anticipated to operate in October.

Processing Times	% of Ocurrances
1 day	17.80%
2 days	45.50%
3 days	28.80%
> 3 days	7.90%

Table 28: Truncated Normal Distribution - Processing Time

The process of accumulating all requests up to a limit day and then defining who is accepted/rejected and when they are scheduled, is called static scheduling. There is also another version, called dynamic scheduling, on which clients receive answers immediately as they place their requests. The difference between those methodologies is that, in the first case, terminals take decisions with full information about clients requests, and in the second case, terminals make decisions once at a time, without knowing if further requests will bring more or less profit.

Exercise Instructions

Now you will play the role of the scheduler and make the scheduling decisions considering all the information previously introduced.

The first exercise will be a dynamic scheduling exercise on which requests information (volume, earliest and latest days) will be placed one by one, and you will have to accept or reject them; and if accept, define the scheduled date. In this exercise ships will have volumes of 100,000 m^3, 200,000 m^3 or 300,000 m^3 and availability sizes ($l_s - e_s$+1) of 3 days, 4 days or 5 days.

If you accept the operation, you should mark an "X" in the date that ship should be scheduled. If you don't accept that ship, please leave it blank. Remember that your goal is to maximize terminals profit. Although there is only one berth in this terminal, you are allowed to schedule ships at the same day if you believe this will return more money in the end. In some situations operations are so profitable, that you prefer to schedule ships at the same date, knowing that you will incur in demurrage and some clients will wait, than loosing the deal. We are not worried in this exercise if those decisions will have any impact in the long term relationship between clients and terminals.

Further on, there will be 5 static scheduling experiments, but this time you will know all the information at once to place your decisions.

Remember that deciding the scheduling dates will impact the arrival pattern of ships, triggering demurrage disputes in case ships overstay. Also take into account that further on the operation itself will be adjusted to a "First Come, First Serve" policy to maximize berth utilization. Feel free to order the columns of the excel sheet (to do so you can select the lines you want to order (including the header) > Go to 'Data' > Go to 'Create a filter'), if you believe ranking the ships somehow will help you answer your exercise.

As the last exercise, it will be showed the scheduling plan recommended by a system, and you are asked to propose adjustments (if you believe they are needed) that could increase the total profit. Finally you will answer a quick post game questionnaire.

Post Game Questionnaire

1. In the dynamic scheduling game, what was your strategy choosing the ships you should accept or reject? Did your strategy changed over the exercise (you started to reject more or less as the game continued)?

2. In the static scheduling game, what was your strategy choosing the ships you should accept or reject?

3. Considering now both static and dynamic exercises, did you have any fixed strategy to set the scheduled date for the accepted ships?

a) Early start schedule (schedule close to the early date)
b) Late start schedule (schedule close to the late date)
c) Medium point start schedule (schedule close to the medium point between early and late dates)
d) Consider one of the above + the time window information. **(Please inform which one of the above!)**
e) Consider one of the above + number of ships already scheduled to operate. **(Please inform which one of the above!)**
f) Another strategy. **(Please inform which one of the above!)**

4. Rank the factors that were more important to you in the overall process:

☐ Earliest/ Latest
☐ Availability size
☐ Volume
☐ Processing time
☐ Arrival time
☐ Demurrage
☐ Other **(Please inform which ones!)**

7.3 Annex 3: Tiebreaking rule: smallest deviation first

There is another possibility of tiebreaking rule used by operational teams that prioritize requests with the smaller deviation from the scheduled date. In this case the highest priority is given to on time ships, followed by early ones (from the closer to the scheduled date to the further), and lastly by the late ones (from the closer to the scheduled date to the further). In this case the step from the algorithm that defines the tiebreak rule should be substituted by: "If there is a tiebreak, consider $prior_s = dx_s - dy_s$ and order them as following: $[s|dx_s - dy_s = 0$, $s|min(prior_s^+)$, ..., $s|max(prior_s^+)$, $s|max(prior_s^-)$, ..., $s|min(prior_s^-)]$".

Figure 32 shows an example of FIFO berthing rule with the tiebreaking rule based on the smaller deviation. In this case in Scenario (a) both Ship 1 and Ship 2 arrived at the same time, but as $dx_1 - dy_1 = 2$ and $dx_2 - dy_2 = 1$, the order of the tiebreak is first Ship 2 (with the smallest deviation to the scheduled date) and then Ship 1.

Scenarios	dx_1	dx_2	dx_3	$delay_1$	$delay_2$	$delay_3$	p_1	p_2	p_3	dy_1	dy_2	dy_3	m=1	m=2	m=3
a	4	3	3	-2	-1	-4	3	2	1	2	2	-1	3	2	1
b	4	3	3	-7	5	-3	4	2	1	-3	8	0	1	2	3
C	4	3	3	3	3	2	1	3	5	7	6	5	3	2	1

Ordered Scenarios	$dy_{s\|1}$	$dy_{s\|2}$	$dy_{s\|3}$	$p_{s\|1}$	$p_{s\|2}$	$p_{s\|3}$	$dz_{s\|1}$	$dz_{s\|2}$	$dz_{s\|3}$	$dx_{s\|1}$	$dx_{s\|2}$	$dx_{s\|3}$	Profit ($P_{comb,scen}$)
a	-1	2	2	1	2	3	1	2	4	3	3	4	$P_{1,1}$
b	-3	-1	7	4	2	1	1	5	7	4	3	3	$P_{1,2}$
c	3	5	7	5	3	1	3	8	11	3	3	4	$P_{1,3}$

Figure 32: Fifo Ordering/ Smaller Deviation to Scheduled tiebreaking rule

Figure 33 shows an example of "by schedule" berthing rule with the tiebreaking rule based on the smaller deviation. In this case in Scenario (a) both Ship 2 and Ship 3 were scheduled to operate in the same date, but as $dx_2 - dy_2 = 3 - 2 = 1$ and $dx_3 - dy_3 = 3 - (-1) = 4$, the order of the tiebreak is first Ship 2 (with the smallest deviation to the scheduled date) and then Ship 3.

Scenarios	dx_1	dx_2	dx_3	$delay_1$	$delay_2$	$delay_3$	p_1	p_2	p_3	dy_1	dy_2	dy_3	m=1	m=2	m=3
a	4	3	3	-2	-1	-4	3	2	1	2	2	-1	2	3	1
b	4	3	3	-7	5	-3	4	2	1	-3	8	0	3	2	1
C	4	3	3	3	3	2	1	3	5	7	6	5	3	2	1

Ordered Scenarios	$dy_{s\|1}$	$dy_{s\|2}$	$dy_{s\|3}$	$p_{s\|1}$	$p_{s\|2}$	$p_{s\|3}$	$dz_{s\|1}$	$dz_{s\|2}$	$dz_{s\|3}$	$dx_{s\|1}$	$dx_{s\|2}$	$dx_{s\|3}$	Profit ($P_{comb,scen}$)
a	2	-1	2	2	1	3	2	4	5	3	3	4	$P_{1,1}$
b	0	8	-3	1	2	4	1	2	4	3	3	4	$P_{1,2}$
c	5	6	7	5	3	1	5	10	13	3	3	4	$P_{1,3}$

Figure 33: By Schedule Ordering/ Smaller Deviation to Scheduled tiebreaking rule

7.4 Annex 4: K-means and HAC Clustering Algorithms

Those two techniques are used as possible options of clustering methodologies, which have pre-defined libraries in R. By the end those clusters should be order by decreasing mean volume of ships from each cluster. The codes utilized in those cases were the following:

Hierarchical Agglomerative Clustering (HAC)

```
#### READING DATA ####

data_reading <- read_excel("C:/Users/Data & Analysis/Experimento_2.xlsx", sheet="EXE_LO")

options(tibble.print_max = Inf)
options(tibble.width = Inf)

data_to_use <- data_reading[,2:3]
data <- scale(data_to_use)

# Dissimilarity matrix

d <- dist(data, method = "euclidean")#

# Hierarchical clustering using Complete Linkage

hc1 <- hclust(d, method = "complete" )#

# Plot the obtained dendrogram

plot(hc1, cex = 0.6, hang = -1)
```

Figure 34: HAC clustering algorithm

K means

```
#### READING DATA ####

data_reading <- read_excel("C:/Users/Lygia/OneDrive - ZLC/Lygia and Çagri/Data & Analysis/Experimento_2.xlsx", sheet="EXE_LO")
options(tibble.print_max = Inf)
options(tibble.width = Inf)

#### CLUSTERING ####

data_to_use <- data_reading[,2:4]
total_clusters <- 2

# Z-score standardization

clustering_input <- as.data.frame(scale(data_to_use))

set.seed(777)
clusters <- stats::kmeans(clustering_input, centers=total_clusters, iter.max=20, nstart=1)
clusters$size
clusters$centers

# Visualize the clustering

fviz_cluster(clusters, clustering_input,  geom = "point", show.clust.cent = FALSE, palette = "jco", ggtheme = theme_classic())
```

Figure 35: K means clustering algorithm

7.5 Annex 5: Detailing the Simulation Model

A simulation is a simplified and controlled version of reality. The first step is designing the logic of the flow process and determining which are the entities flowing through the model, the resources used at each moment, the processes that the entities go through, the queuing positions, as well as the decisions that are made at each point.

The ships are the main entities of the system, which arrive, wait to berth, operate and leave. Those ships can be principals or secondaries (also called 'pairs') and can operate different types of product with tanks (loading and unloading), or between themselves (ship to ship or transshipment).

This model was built to allow operation with two berths, but as all the optimization models consider the appointment scheduling of a unique berth, the second berth is always considered "out of operation" for the experiment proposed. This will allow future research with multiple berths.

As soon as a principal ship enters the system, given a schedule or an arrival distribution, it goes to a common queue and wait until the berth is empty in a 'first in first out' order. There is a berthing rule that if the vessel is a VLCC full of cargo, it can only enter if there is enough time to reach the berth before the sunset, due to safety reasons. As the sunrise/ sunset vary during the year, a simplification is used fixing the entrance time window between 5 am to 15 pm for those ships to enter.

If the berth is free and there are restrictions to enter, the system calls for the next ship with no restrictions in order to maximize berth utilization. If a berth is free and the ship has no restriction, a harbor master is requested to maneuver the ship through the port access until it is completely berthed.

Figure 36 shows the first sequence of modules that represent this logic. The simulation was prepared to read the information of ships (product, type of operation, dimension, etc...) in three different ways: the first one is by reading the respective distribution and randomly selecting the characteristics of each particular ship through them; the second one is by reading the operational information of each ship from an excel file; and the third one, which will be used in the thesis, read the vol_s and the dx_s information, and select from distributions the rest of the operational characteristics and a $delay_s$ rounded from the normal distribution with mean 0 and standard deviation of 2 days also used in the optimization model.

One of the characteristics read is the number of pairs that each ship will operate with. This information is used to create the secondary ships by duplicating the principal ship. Those ships read their own information and wait a call from the principal ship whenever it is finished berthing.

Principal ships are separated into two different queues: one for ships that have restriction to berth and the others that don't.

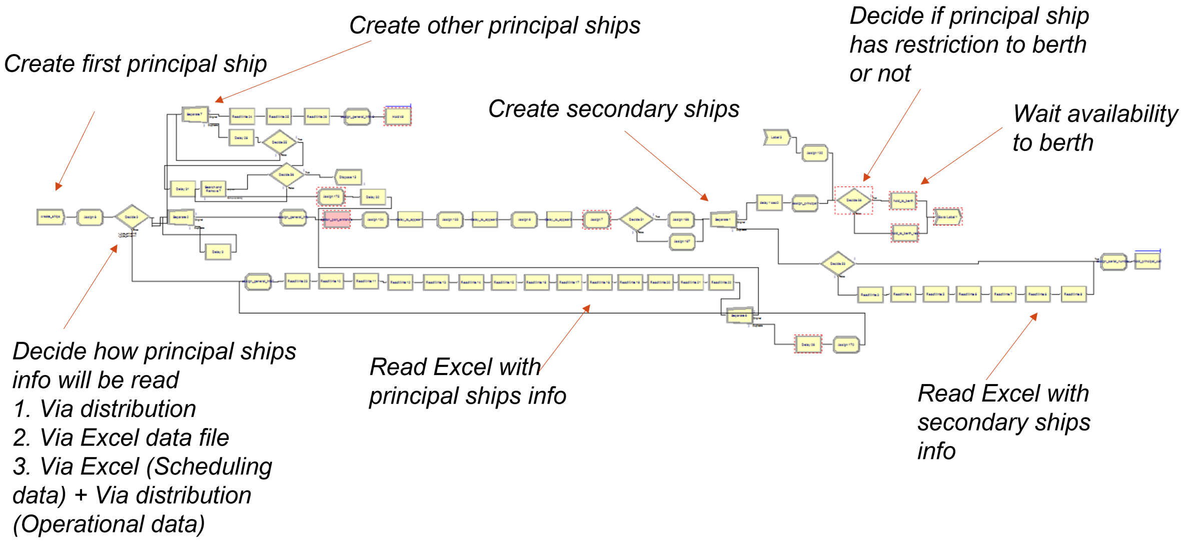

Figure 36: ARENA framework - Part 1

This simulation is agent based, meaning that there is an entity which is called *decider* who is always checking all the ships requests for berthing and unberthing and taking decisions on who should be prioritized. There is a priority sequence followed by that agent for taking actions, which is given as follows:

1. Secondary ships requesting to berth for STS operation
2. Any ship performing STS requesting to unberth that has restriction to exit
3. Any ship performing transshipment requesting to unberth that has restriction to exit (this feature was implement but not used as multiple berths are not studied in this research)
4. Any ship performing loading/unloading requesting to unberth that has restriction to exit
5. Principal ships requesting to berth with restrictions to enter
6. Principal ships requesting to berth without restrictions to enter
7. Any ship performing loading/unloading requesting to unberth that does not have restriction to exit
8. Secondary ships requesting to berth for transshipment operation (this feature was implement but not used as multiple berths are not studied in this research)
9. Any ship performing STS requesting to unberth that does not have restriction to exit
10. Any ship performing transshipment requesting to unberth that does not have restriction to exit (this feature was implement but not used as multiple berths are not studied in this research)

Figure 37 shows the logic of the agent. This entity waits until some ship needs to be berthed or unberthed to trigger the rest of the logic. Is possible that multiple ships are requesting berth/unberth at the same time, in this case the agent follows the enumerated order presented above, attending first the ships that have restriction to enter or exit the port, and then the ones that don't. Ships are signalized if they should proceed with berthing/unberthing, or if they should wait.

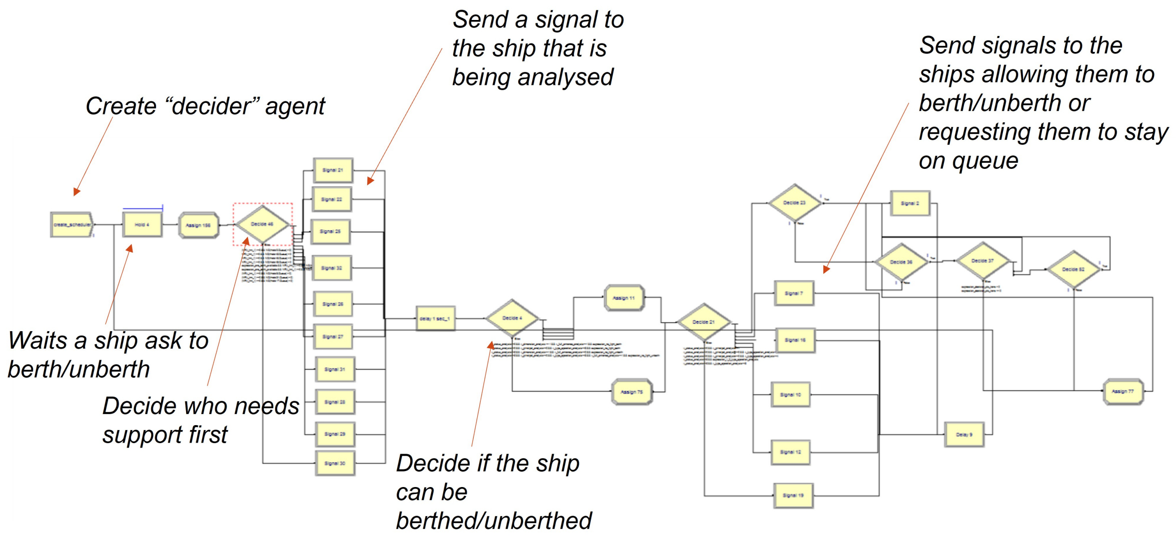

Figure 37: ARENA framework - Part 2

When an agent is processing a specific request, the ship under analysis is moved to another flowchart, on which, depending on the answer given by the agent, it either proceeds to berth, first signalizing to the secondary ships that the operation will happen (if there are any secondary ships) and then calling the harbor master; or it waits until there is no restriction to request berth again (those are shown in Figure 38). It is important to highlight that those "moves" happen in the flowchart, but physically the ship is just anchored waiting for further instruction.

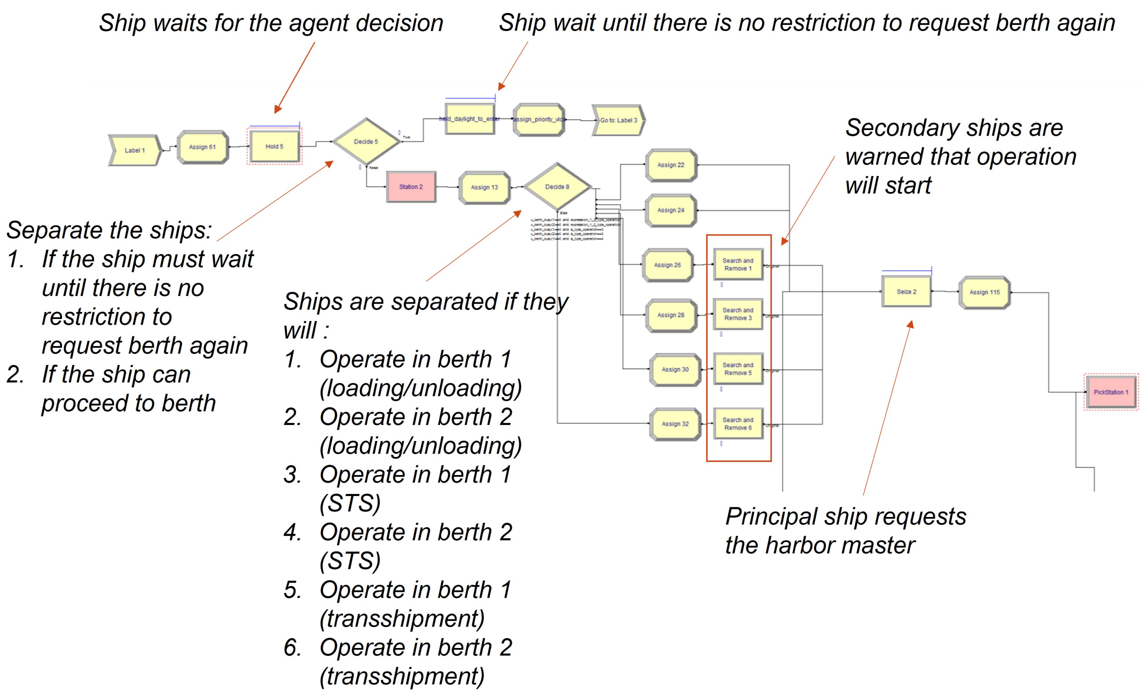

Figure 38: ARENA framework - Part 3

In this simulation the harbor master, mooring team, tanks and berths are considered resources. Having the harbor master onboard, the ship maneuvers from the queuing region to the berth, requests the mooring team and the berthing procedure begins. When the berthing is complete, harbor master and mooring team are released. There is a rule that the harbor master can only be called once every 3 hours, meaning that even when shifting and berthing takes less than 3 hours, the harbor master can not be requested again until this period is finished.

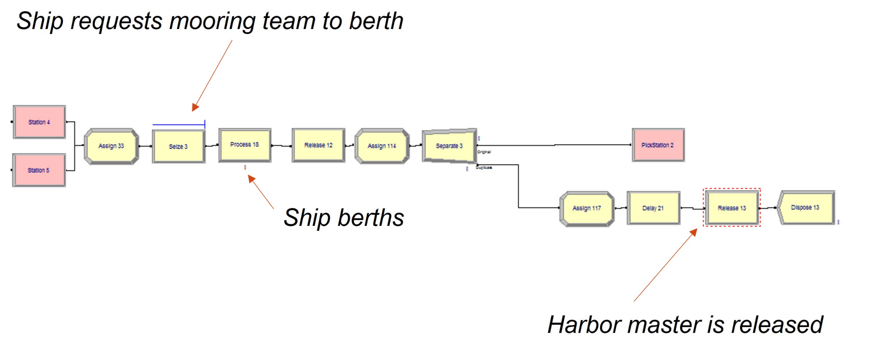

Figure 39: ARENA framework - Part 4

The secondary ships (when there are any) are sent to the flowchart of Figure 40 when warned that their operation will start, they wait until the berth is available and request their entrance to the agent.

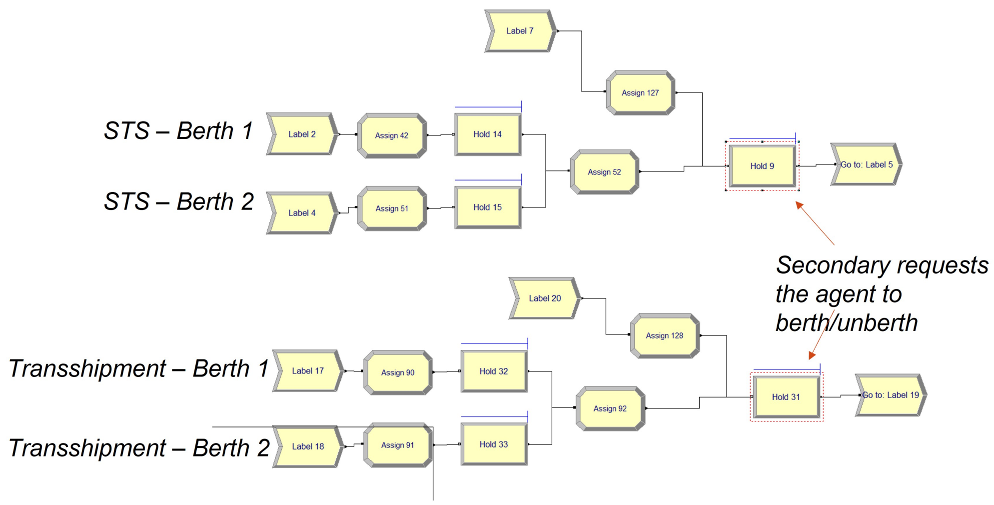

Figure 40: ARENA framework - Part 5

The agent (Figure 37) goes through the same process as explained before, and decides if the secondary ship can or can not berth. Depending on its decision the secondary ship waits until there is no restriction anymore to request again, or continues to request the harbor master as showed in Figure 41. The secondary ship enters the flowchart of Figure 39, requests assistance of the mooring team, berths and releases the harbor master and mooring team when finished.

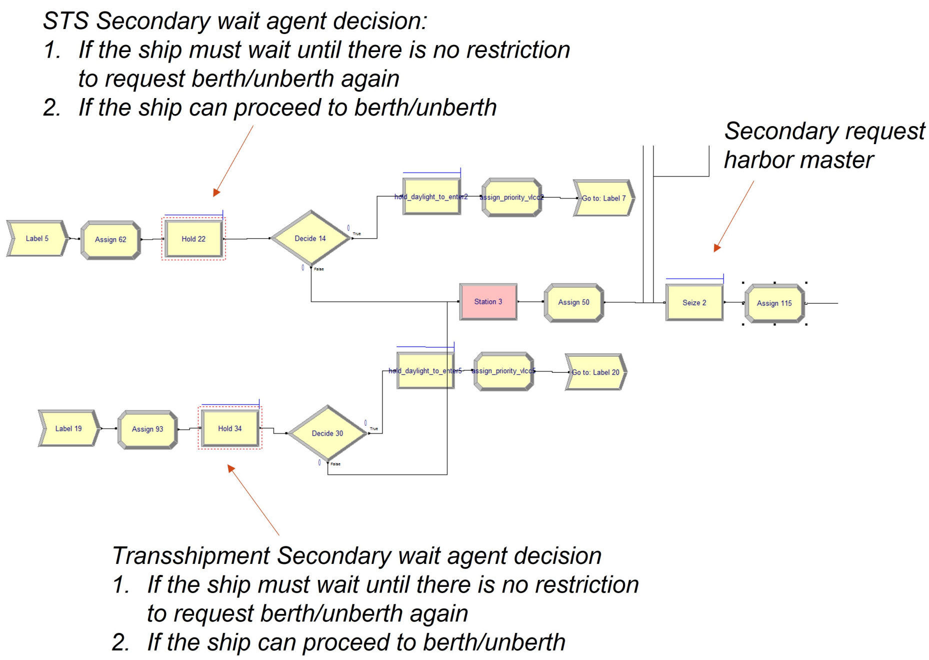

Figure 41: ARENA framework - Part 6

If the operation performed is loading/unloading, none of this last part dedicated to secondary ships is needed. In this type of operation the next step followed by the principal ship is the pre-transfer process, shown in Figure 42 where basically all the documentation, sampling and measurement of product onboard is executed. Next, the pipelines/tank of the specific product transported are requested so transfer can be performed. Next it is the post-transfer procedure, which is really similar to the pre-transfer (documentation, sampling and measurement). Finally the ship is ready to request unberth. Unberthing has also a restriction which says that full VLCC and full secondary ships operating STS, can only unberth between 5 am to 17 pm.

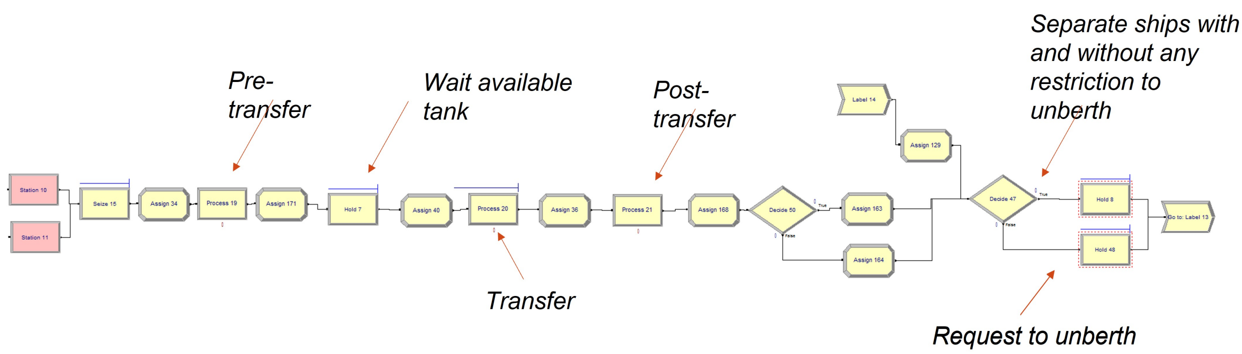

Figure 42: ARENA framework - Part 7

Whenever the agent analyses the request (and goes through the flowchart of Figure 37), the ship moves to the flowchart of Figure 43 and if the unberth is not accepted due to restrictions the ship has to wait to request again, or it continues to seize the harbor master and mooring teams to unberth.

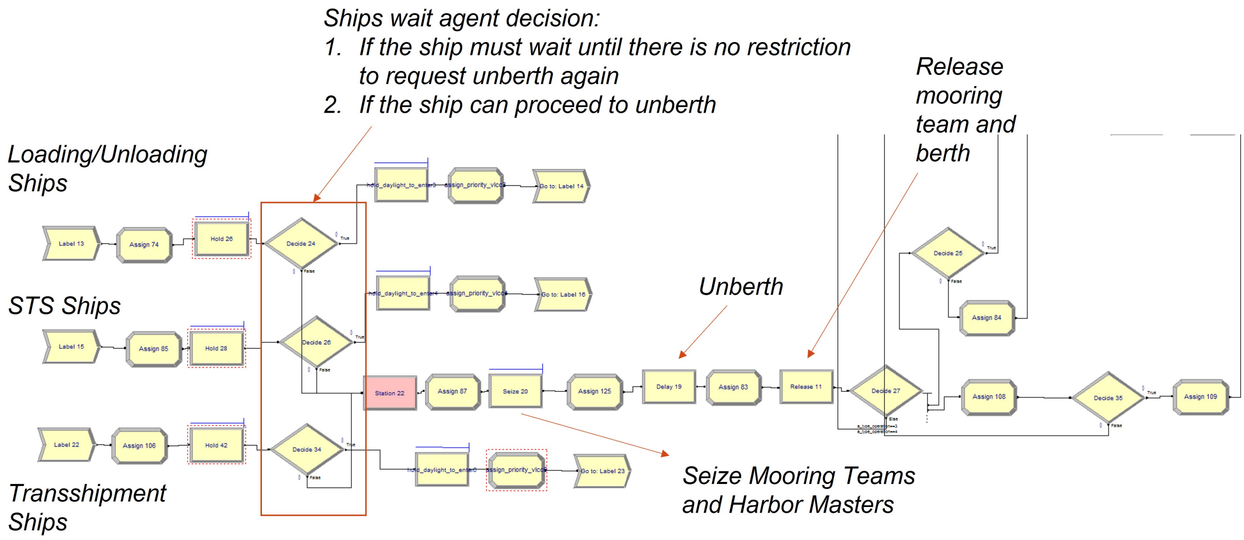

Figure 43: ARENA framework - Part 8

If the ship operating is a STS the flowchart is slightly different (Figure 44). While the principal is advancing with its pre transfer procedure, the secondary is being shifted and berthed. The transfer can only start when both ships finished their pre-transfer. After transferring, both ships have their post-transfer process. If there are still other pairs to operate, the principal ship goes back to the beginning of the process and prepares the pre-transfer for the next secondary. The secondary that already operated requests to unberth. Whenever the agent considers this request, the secondary ship goes to the flowchart of Figure 43 and if accepted to unberth it seizes the harbor master and mooring team.

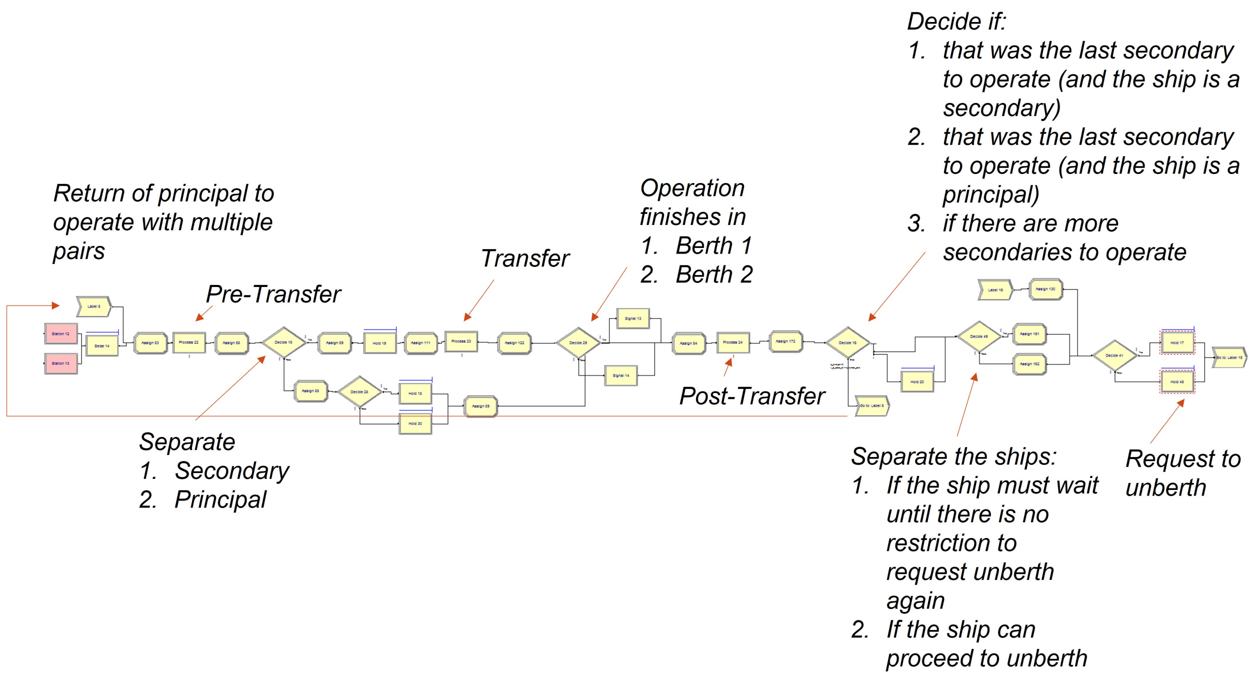

Figure 44: ARENA framework - Part 9

The last case would be if the vessel is performing a transshipment, which although modeled to allow future research, is not considered for the experiments of this research. In this case, after berthing, the principal can also start its own pre-transfer before the secondary but has to wait the secondary shift, berth and perform its own pre-transfer so they are ready to transfer cargo, as it is showed in Figure 45. After finishing the cargo transfer there is the post-transfer. Whoever finishes first can leave, unless the principal is operating with other secondaries in sequence. In that case the principal goes back in the same flowchart, starts the pre-transfer again, while waiting the first secondary to leave and the second to enter and go through the initial process. The ship that requests to unberth triggers the decision of the agent (Figure 37), which should define if it has to wait until there is no restriction to request unberth again, or if it can proceed to unberth (which is showed in Figure 43).

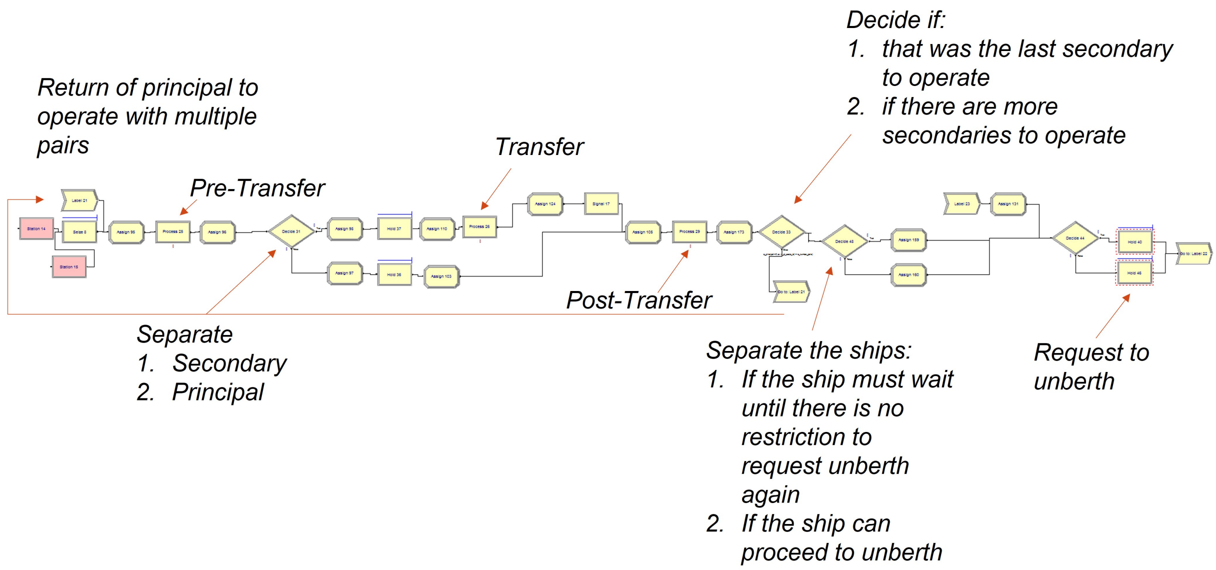

Figure 45: ARENA framework - Part 10

This simulation model was built using an academic license from software Arena, version 16.00.00003 and it has both logic and graphic interface. All major inputs are registered in an Excel file, which is read by Arena in the beginning of each replication and the outputs are written in a .csv file, allowing further data analysis.

7.6 Annex 6: Using the cluster based partition method

The following algorithm includes the extra steps needed to incorporate the cluster based partition method proposed in Chapter 4 to solve the appointment scheduling problem faster. In order to do that the vol_s, e_s, l_s and c_s should be given as input parameters. Being c_s the cluster from which each ship belongs defined by the clustering algorithm. N_c the total number of clusters, and N_k is the total number of ships in cluster k. Figure 29, shows the example of c_s definition for EXC HO, considering [4 4], [2 2 2 2] and [1 1 .. 1 1] clustered via HVF.

EXC - HO [4 4]				
	e(s)	l(s)	vol(s)	c(s)
4	5	8	300000	1
6	2	5	300000	1
1	3	7	200000	1
2	2	5	200000	1
5	4	6	200000	2
7	5	9	200000	2
3	3	5	100000	2
8	1	4	100000	2

EXC - HO [2 2 2 2]				
	e(s)	l(s)	vol(s)	c(s)
4	5	8	300000	1
6	2	5	300000	1
1	3	7	200000	2
2	2	5	200000	2
5	4	6	200000	3
7	5	9	200000	3
3	3	5	100000	4
8	1	4	100000	4

EXC - HO [1 1 .. 1 1]				
	e(s)	l(s)	vol(s)	c(s)
4	5	8	300000	1
6	2	5	300000	2
1	3	7	200000	3
2	2	5	200000	4
5	4	6	200000	5
7	5	9	200000	6
3	3	5	100000	7
8	1	4	100000	8

Table 29: Example of HVF clustering for EXC HO

Problem 1/ Algorithm 1 (Partitioned Methodology):

- **Step 1**: Set $S_{all} = 0$. Set $k = 1$. Set $N_k = \sum_{s \in S|c_s=k} 1$.
- **Step 2**: For all $s \in S|c_s = k$ consider the e_s and l_s information to enumerate all options of dx_s including 'NA' which represent rejecting the operation of that specific ship. If $S_{all} = 0$, go to Step 4, otherwise go to Step 3.
- **Step 3**: For all ships $s \in S_{all}$, consider the $dx_s = dx*_s$ that was previously defined in Step 9.
- **Step 4**: Enumerate all possible combinations of dx_s considering $S_{all} \cup (s \in S|c_s = k)$.
- **Step 5**: Set the counter $i = 1$. Define $N_{comb} = (\sum_{s \in S_{all}} 1) * (\prod_{s \in S|c_s=k}(a_s + 1))$ as the total number of combinations. Define $S_{all} = S_{all} \cup (s \in S|c_s = k)$.
- **Step 6**: Having the i^{th} combination, and given $s \in S_{all}$, create a set of N_{scen} scenarios with randomly selected $delay_s$ and p_s for each ship, considering their distributions.
- **Step 7**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios calculate $dy_s = dx_s + delay_s$.
- **Step 8**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, define an index m_s with the order of arrival of each ship. If there is a tiebreak, order first the one that requested operation first (in other words the ship with $min|s|$).
- **Step 9**: For all $j \in N_{scen}$, consider that for the $s|m_s = 1$ the calculation of dz_s is given by $dz_s = max[dy_s|(m_s = 1), 1]$, for all $s \in S|m_s > 1$, calculate $dz_s = max[dy_s, (dz_{s'} + p_{s'})|m_{s'} = m_s - 1]$.
- **Step 10**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, calculate the exceeded days of demurrage by $\tau_s^+ = max[0, dz_s - dx_s)\mathbb{1}^{(dy_s \leq dx_s)} + (p_s - \theta_s)]$
- **Step 11**: For all $s \in S_{all}$, from all $j \in N_{scen}$ scenarios, calculate the profit, called $P_{i,j}$ by the expression $\sum_{s \in S}[\Phi_s vol_s - \Omega_s(\tau_s^+)]\mathbb{1}^{(dx_s > 0)}$.

- **Step 12**: Take the expectation of the total profit from all scenarios of i^{th} combination, called $E[P_i] = \sum_{j \in N_{scen}} P_{i,j}/N_{scen}$. Set $i = i + 1$. If $i \leq N_{comb}$ goes back to Step 6, otherwise goes to Step 13.
- **Step 13**: The solution for ships $s \in S_{all}$ considering cluster k is given by the combination of $dx*_s$ that returns the maximum expected profit ($maxE[P_i]$). If $k < N_c$ go to Step 14, otherwise $dx*_s$ is the final solution.
- **Step 14**: Set $k = k + 1$. Set $N_k = \sum_{s \in S|c_s = k} 1$. Go back to Step 2.

The same adjustments performed to Algorithm 1 can be used to the other algorithms proposed.

References

A. M. Abdelmagid, M. S. Gheith, and A. B. Eltawil. A comprehensive review of the truck appointment scheduling models and directions for future research. *Transport Reviews*, 42(1): 102–126, 2022. ISSN 14645327. doi: 10.1080/01441647.2021.1955034. URL https://www.tandfonline.com/doi/abs/10.1080/01441647.2021.1955034.

A. Ala and F. Chen. Appointment Scheduling Problem in Complexity Systems of the Healthcare Services: A Comprehensive Review. *Journal of Healthcare Engineering*, 2022:1–16, 3 2022. ISSN 2040-2309. doi: 10.1155/2022/5819813. URL https://www.hindawi.com/journals/jhe/2022/5819813/.

J. Alstrup, S. Boas, O. B. G. Madsen, R. Victor, and V. Vidal. Booking policy for flights with two types of passengers. *European Journal of Operational Research*, 27:274–288, 1986. URL https://www.sciencedirect.com/science/article/abs/pii/0377221786903255.

ANP. ANP 881 Resolution. Technical report, 2022. URL https://www.in.gov.br/en/web/dou/-/resolucao-anp-n-881-de-8-de-julho-de-2022-414421806.

Association of Ship Brokers and Agents (USA). Asbatankvoy: Tanker Voyage Charter Party, 1977. URL https://shippingforum.files.wordpress.com/2012/08/asbatankvoy.pdf.

V. H. Barros, T. S. Costa, A. C. Oliveira, and L. A. Lorena. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers and Industrial Engineering*, 60(4):606–613, 2011. ISSN 03608352. doi: 10.1016/j.cie.2010.12.018. URL http://dx.doi.org/10.1016/j.cie.2010.12.018.

C. Bierwirth and F. Meisel. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3):675–689, 2015. ISSN 03772217. doi: 10.1016/j.ejor.2014.12.030. URL http://dx.doi.org/10.1016/j.ejor.2009.05.031.

D. C. Cafaro and J. Cerdá. Efficient tool for the scheduling of multiproduct pipelines and terminal operations. *Industrial and Engineering Chemistry Research*, 47(24):9941–9956, 2008. ISSN 08885885. doi: 10.1021/ie071630d.

B. Cankaya, E. Wari, and B. Eren Tokgoz. Practical approaches to chemical tanker scheduling in ports: a case study on the Port of Houston. *Maritime Economics and Logistics*, 21(4): 559–575, 12 2019. ISSN 1479294X. doi: 10.1057/s41278-019-00122-w. URL https://link.springer.com/article/10.1057/s41278-019-00122-w.

W. Cao. *An integrated overlapping and overbooking outpatient appointment scheduling system.* PhD thesis, Binghamton University, New York, 2009.

G. Chen, K. Govindan, Z. Z. Yang, T. M. Choi, and L. Jiang. Terminal appointment system design by non-stationary M(t)/E k/c(t) queueing model and genetic algorithm. *International Journal of Production Economics*, 146(2):694–703, 12 2013. ISSN 09255273. doi: 10.1016/j.ijpe.2013.09.001.

R. R. Chen and L. W. Robinson. Sequencing and scheduling appointments with potential call-in patients. *Production and Operations Management*, 23(9):1522–1538, 9 2014. ISSN 19375956. doi: 10.1111/poms.12168.

J. F. Correcher, T. Van den Bossche, R. Alvarez-Valdes, and G. Vanden Berghe. The berth allocation problem in terminals with irregular layouts. *European Journal of Operational Research*, 272(3):1096–1108, 2019. ISSN 03772217. doi: 10.1016/j.ejor.2018.07.019. URL https://doi.org/10.1016/j.ejor.2018.07.019.

V. Cunha, I. Santos, L. Pessoa, and S. Hamacher. An ILS heuristic for the ship scheduling problem: application in the oil industry. *International Transactions in Operational Research*, 27(1):197–218, 2020. ISSN 14753995. doi: 10.1111/itor.12610. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12610.

M. Dempster, J. Lenstra, and A. Rinnooy Kan. *Deterministic and Stochastic Scheduling*. D. Reidel Publishing Company, Dordrecht, 1981.

B. Du and C. Larsen. Reservation policies of advance orders in the presence of multiple demand classes. *European Journal of Operational Research*, 256(2):430–438, 1 2017. ISSN 03772217. doi: 10.1016/j.ejor.2016.06.028.

M. M. S. El-Kholany, K. Schekotihin, and M. Gebser. Decomposition-Based Job-Shop Scheduling with Constrained Clusterin. In *Practical Aspects of Declarative Languages*, pages 165–180. 2022. URL https://link.springer.com/bookseries/7408.

H. Fan, X. Ren, Z. Guo, and Y. Li. Truck scheduling problem considering carbon emissions under truck appointment system. *Sustainability (Switzerland)*, 11(22), 11 2019. ISSN 20711050. doi: 10.3390/su11226256.

E. Fernández and M. Munoz-Marquez. New formulations and solutions for the strategic berth template problem. *European Journal of Operational Research*, 298(1):99–117, 4 2022. ISSN 03772217. doi: 10.1016/j.ejor.2021.06.062. URL https://www.sciencedirect.com/science/article/pii/S0377221721005932.

Y. . Gerchak, D. Gupta, and H. Mordechai. Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science*, 42(3):321–334, 1996.

Y. Gocgun and M. L. Puterman. Dynamic scheduling with due dates and time windows: An application to chemotherapy patient appointment booking. *Health Care Management Science*, 17(1):60–76, 3 2014. ISSN 13869620. doi: 10.1007/s10729-013-9253-z.

M. Golias, I. Portal, D. Konur, E. Kaisar, and G. Kolomvos. Robust berth scheduling at marine container terminals via hierarchical optimization. *Computers and Operations Research*, 41 (1):412–422, 2014. ISSN 03050548. doi: 10.1016/j.cor.2013.07.018.

M. M. Golias. A bi-objective berth allocation formulation to account for vessel handling time uncertainty. *Maritime Economics and Logistics*, 13(4):419–441, 12 2011. ISSN 14792931. doi: 10.1057/mel.2011.21.

M. M. Golias, M. Boile, and S. Theofanis. Berth scheduling by customer service differentiation: A multi-objective approach. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):878–892, 2009. ISSN 13665545. doi: 10.1016/j.tre.2009.05.006.

R. Graham, E. Lawler, J. Lenstra, and A. Rinnooy Kan. Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, 5:287–326, 1979. URL https://www.sciencedirect.com/science/article/abs/pii/S016750600870356X.

C. Guan and R. Liu. Container terminal gate appointment system optimization. *Maritime Economics and Logistics*, 11(4):378–398, 12 2009. ISSN 14792931. doi: 10.1057/mel.2009.13.

D. Gupta and B. Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions (Institute of Industrial Engineers)*, 40(9):800–819, 9 2008. ISSN 0740817X. doi: 10.1080/07408170802165880.

L. E. Henesey. *Multi-agent systems for container terminal management.* Blekinge Institute of Technology, 2006. ISBN 9789172950993.

C.-J. Ho and H.-S. Lau. Minimizing Total Cost in Scheduling Outpatient Appointments. *Management Science*, 38(12):1750–1764, 1992. URL `https://www.jstor.org/stable/2632708`.

C. Huang and I. A. Karimi. Scheduling tanker-lightering operations in crude oil transportation. *Industrial and Engineering Chemistry Research*, 45(24):8063–8082, 2006. ISSN 08885885. doi: 10.1021/ie060379j.

A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):900–920, 2008a. ISSN 13665545. doi: 10.1016/j.tre.2007.03.003.

A. Imai, E. Nishimura, and S. Papadimitriou. Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):136–151, 2008b. ISSN 13665545. doi: 10.1016/j.tre.2006.05.002.

A. Imai, Y. Yamakawa, and K. Huang. The strategic berth template problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:77–100, 12 2014. ISSN 13665545. doi: 10.1016/j.tre.2014.09.013. URL `https://www.sciencedirect.com/science/article/abs/pii/S1366554514001677`.

R. R. P. Jackson. Design of an Appointments System. *OR*, 15(3):219–224, 1964. URL `https://www.jstor.org/stable/3007209`.

X. Jiang, L. H. Lee, E. P. Chew, Y. Han, and K. C. Tan. A container yard storage strategy for improving land utilization and operation efficiency in a transshipment hub port. *European Journal of Operational Research*, 221(1):64–73, 2012. ISSN 03772217. doi: 10.1016/j.ejor.2012.03.011. URL `http://dx.doi.org/10.1016/j.ejor.2012.03.011`.

F. John. Appointments in General Practice. *OR*, 15(3):233–237, 1964. URL `https://about.jstor.org/terms`.

J. Karafa, M. M. Golias, S. Ivey, G. K. Saharidis, and N. Leonardos. The berth allocation problem with stochastic vessel handling times. *International Journal of Advanced Manufacturing Technology*, 65(1-4):473–484, 2013. ISSN 02683768. doi: 10.1007/s00170-012-4186-0.

P. Kotler and G. Armstrong. *Principles of Marketing.* Pierson, 2017.

E. Lalla-Ruiz, B. Melián-Batista, and J. Marcos Moreno-Vega. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6):1132–1141, 2012. ISSN 09521976. doi: 10.1016/j.engappai.2012.06.001.

N. Le Carrer, S. Ferson, and P. L. Green. Optimising cargo loading and ship scheduling in tidal areas. *European Journal of Operational Research*, 280(3):1082–1094, 2 2020. ISSN 03772217. doi: 10.1016/j.ejor.2019.08.002.

L. H. Lee, E. P. Chew, K. C. Tan, and Y. Han. An optimization model for storage yard management in transshipment hubs. *Container Terminals and Cargo Systems: Design, Operations Management, and Logistics Control Issues*, 561:107–129, 2007. doi: 10.1007/978-3-540-49550-5{_}6.

C. J. Liang, J. Guo, and J. W. Zhang. Analysis of the influences of demurrage on yard utilization in container terminal. *International Conference on Management and Service Science, MASS 2011*, pages 1–5, 2011. doi: 10.1109/ICMSS.2011.5999402.

M. Lindberg. A Survey of Reservation-Based Scheduling. Technical report, Lund University, 2007. URL `https://lucris.lub.lu.se/ws/portalfiles/portal/4381632/8627270.pdf`.

W. Liu, H. Yang, and Y. Yin. Efficiency of a highway use reservation system for morning commute. *Transportation Research Part C: Emerging Technologies*, 56:293–308, 7 2015. ISSN 0968090X. doi: 10.1016/j.trc.2015.04.015.

S. N. B. Magatão, L. Magatão, H. Luis Polli, F. Neves, L. V. R. De Arruda, S. Relvas, and A. P. F. D. Barbosa-Póvoa. Planning and sequencing product distribution in a real-world pipeline network: An MILP decomposition approach. *Industrial and Engineering Chemistry Research*, 51(12):4591–4609, 2012. ISSN 08885885. doi: 10.1021/ie2010252.

Y. Maisiuk and I. Gribkovskaia. Fleet sizing for offshore supply vessels with stochastic sailing and service times. In *Procedia Computer Science*, volume 31, pages 939–948. Elsevier B.V., 2014. doi: 10.1016/j.procs.2014.05.346.

H. Y. Mak, Y. Rong, and J. Zhang. Sequencing appointments for service systems using inventory approximations. *Manufacturing and Service Operations Management*, 16(2):251–262, 2014. ISSN 15265498. doi: 10.1287/msom.2013.0470.

T. W. Malone. What is Coordination Theory? Technical report, 1988.

C. Mancilla and R. Storer. A sample average approximation approach to stochastic appointment sequencing and scheduling. *IIE Transactions (Institute of Industrial Engineers)*, 44(8):655–670, 8 2012. ISSN 0740817X. doi: 10.1080/0740817X.2011.635174.

A. Mandelbaum, P. Momčilović, N. Trichakis, S. Kadish, R. Leib, and C. A. Bunnell. Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center. *Management Science*, 66(1):243–270, 1 2020. ISSN 15265501. doi: 10.1287/mnsc.2018.3218.

A. Maneengam and A. Udomsakdigool. A set covering model for a green ship routing and scheduling problem with berth time-window constraints for use in the bulk cargo industry. *Applied Sciences (Switzerland)*, 11(11), 6 2021. ISSN 20763417. doi: 10.3390/app11114840.

F. Mansourifard, P. Mansourifard, M. Ziyadi, B. Krishnamachari, and M. Hsieh. A Heuristic Policy for Outpatient Surgery Appointment Sequencing: Newsvendor Ordering. In *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2018.

G. R. Mauri, G. M. Ribeiro, L. A. N. Lorena, and G. Laporte. An adaptive large neighborhood search for the discrete and continuous Berth allocation problem. *Computers and Operations Research*, 70:140–154, 2016. ISSN 03050548. doi: 10.1016/j.cor.2016.01.002. URL http://dx.doi.org/10.1016/j.cor.2016.01.002.

R. Moorthy and C. P. Teo. Berth management in container terminal: The template design problem. *OR Spectrum*, 28(4):495–518, 10 2006. ISSN 01716468. doi: 10.1007/s00291-006-0036-5.

T. E. Notteboom. The Time factor in liner shipping services. *Maritime Economics and Logistics*, 8(1):19–39, 2006. ISSN 1479294X. doi: 10.1057/palgrave.mel.9100148.

E. Parra. Demurrage costs optimization in ports using binary mathematical programming models. *TOP*, 3(1):145–159, 1995. ISSN 11345764. doi: 10.1007/BF02574808. URL https://link.springer.com/article/10.1007/BF02574808.

PDVSA. Crude Oil Sales Agreement. 2008.

M. H. Phan and K. H. Kim. Collaborative truck scheduling and appointments for trucking companies and container terminals. *Transportation Research Part B: Methodological*, 86: 37–50, 4 2016. ISSN 01912615. doi: 10.1016/j.trb.2016.01.006.

N. S. F. A. Rahman, M. K. Othman, I. A. Sanusi, A. M. Arof, and A. Ismail. Evaluation of Delay Factors on Dry Bulk Cargo Operation in Malaysia: A Case Study of Kemaman Port. *The Asian Journal of Shipping and Logistics*, 35(3):127–137, 2019. ISSN 20925212. doi: 10.1016/j.ajsl.2019.09.001. URL http://dx.doi.org/10.1016/j.ajsl.2019.09.001.

REPSOL. General Terms & Conditions for Sales and Purchases of Crude Oil & Petroleum Products. Technical report.

G. M. Ribeiro, G. R. Mauri, S. De Castro Beluco, L. A. N. Lorena, and G. Laporte. Berth allocation in an ore terminal with demurrage, despatch and maintenance. *Computers and Industrial Engineering*, 96:8–15, 2016. ISSN 03608352. doi: 10.1016/j.cie.2016.03.005. URL http://dx.doi.org/10.1016/j.cie.2016.03.005.

F. Rodrigues and A. Agra. Berth allocation and quay crane assignment/scheduling problem under uncertainty: a survey. *European Journal of Operational Research*, 1 2022. ISSN 03772217. doi: 10.1016/j.ejor.2021.12.040. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221721010961.

F. Sabria and C. F. Daganzo. Approximate Expressions for Queueing Systems with Scheduled Arrivals and Established Service Order. Technical report, 1989.

N. S. Sadghiani and S. Motiian. The Contextual Appointment Scheduling Problem. *Association for the Advancement of Artificial Intelligence*, 8 2021. URL http://arxiv.org/abs/2108.05531.

J. Schofield. *Laytime and Demurrage.* Lloyd's Shipping Law Library, seventh edition, 2015. URL https://www.routledge.com/Laytime-and-Demurrage/Schofield/p/book/9780367708368.

F. Schulte, E. Lalla-Ruiz, R. G. González-Ramírez, and S. Voß. Reducing port-related empty truck emissions: A mathematical approach for truck appointments with collaboration. *Transportation Research Part E: Logistics and Transportation Review*, 105:195–212, 9 2017. ISSN 13665545. doi: 10.1016/j.tre.2017.03.008.

K. S. Shehadeh. *Stochastic Optimization Approaches for Outpatient Appointment Scheduling under Uncertainty.* PhD thesis, University of Michigan, 2019.

N. Street, B. Thomas, F. Cakir, W. N. Street, and B. W. Thomas. Revisiting Cluster First, Route Second for the Vehicle Routing Problem Related papers Revisiting Cluster First, Route Second for the Vehicle Routing Problem, 2015.

D. Sun and R. Batta. Scheduling larger job shops: A decomposition approach. *International Journal of Production Research*, 34(7):2019–2033, 1996. ISSN 1366588X. doi: 10.1080/00207549608905010.

Q. Sun, Q. Meng, and M. C. Chou. Optimizing voyage charterparty (VCP) arrangement: Laytime negotiation and operations coordination. *European Journal of Operational Research*, 291(1):263–270, 5 2021. ISSN 03772217. doi: 10.1016/j.ejor.2020.09.032.

TMRC. Amorco Marine Terminal Use and Throughput Agreement. 2021.

N. Umang, M. Bierlaire, and I. Vacca. The Berth Allocation Problem in Bulk Ports. *Swiss Transport Research Conference 2011*, (April), 2011. URL `http://medcontent.metapress.com/index/A65RM03P4874243N.pdf%5Cnhttp://infoscience.epfl.ch/record/167446`.

N. Umang, M. Bierlaire, and I. Vacca. Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics and Transportation Review*, 54:14–31, 2013. ISSN 13665545. doi: 10.1016/j.tre.2013.03.003. URL `http://dx.doi.org/10.1016/j.tre.2013.03.003`.

UNCTAD. Review of Maritime Transport. Technical report, United Nations, New York, 2020.

UNCTAD. Review of Maritime Transport 2021. Technical report, United Nations, New York City, 2021.

Valero Logistics Operations. Terminal Agreement - Crude Oil Transportation and Storage, 2004.

B. Verweij, S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2 2003. ISSN 09266003. doi: 10.1023/A:1021814225969.

K. Wang, L. Zhen, S. Wang, and G. Laporte. Column generation for the integrated berth allocation, quay crane assignment, and yard assignment problem. *Transportation Science*, 52 (4):812–834, 2018. ISSN 15265447. doi: 10.1287/trsc.2018.0822.

P. P. Wang. Static and Dynamic Scheduling of Customer Arrivals to a Single-Server System. *Naval Research Logistics*, 40:345–360, 1993. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/1520-6750%28199304%2940%3A3%3C345%3A%3AAID-NAV3220400305%3E3.0.CO%3B2-N`.

X. Wang and X. Wang. Flexible parking reservation system and pricing: A continuum approximation approach. *Transportation Research Part B: Methodological*, 128:408–434, 10 2019. ISSN 01912615. doi: 10.1016/j.trb.2019.07.006.

M. Wasesa, F. I. Ramadhan, A. Nita, P. F. Belgiawan, and L. Mayangsari. Impact of overbooking reservation mechanism on container terminal's operational performance and greenhouse gas emissions. *Asian Journal of Shipping and Logistics*, 37(2):140–148, 6 2021. ISSN 20925212. doi: 10.1016/j.ajsl.2021.01.002.

Wordscale. NEW WORLDWIDE TANKER NOMINAL FREIGHT SCALE applying to the carriage of oil in bulk Code name. Technical report, 2015. URL www.worldscale.co.uk.

X. Xiang, C. Liu, and L. Miao. A bi-objective robust model for berth allocation scheduling under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 106:294–319, 2017. ISSN 13665545. doi: 10.1016/j.tre.2017.07.006. URL http://dx.doi.org/10.1016/j.tre.2017.07.006.

X. Xing, P. R. Drake, D. Song, and Y. Zhou. Tank Container Operators' profit maximization through dynamic operations planning integrated with the quotation-booking process under multiple uncertainties. *European Journal of Operational Research*, 274(3):924–946, 5 2019. ISSN 03772217. doi: 10.1016/j.ejor.2018.10.040.

P. S. You. An efficient computational approach for railway booking problems. *European Journal of Operational Research*, 185(2):811–824, 3 2008. ISSN 03772217. doi: 10.1016/j.ejor.2006.12.049.

N. Yousefi, F. Hasankhani, and M. Kiani. Appointment scheduling model in healthcare using clustering algorithms Analytic Project Management View project Organ Transplantation View project Appointment scheduling model in healthcare using clustering algorithms. Technical Report 1, 2020. URL http://www.ijocta.com.

C. Zacharias and M. Pinedo. Appointment scheduling with no-shows and overbooking. *Production and Operations Management*, 23(5):788–801, 2014. ISSN 19375956. doi: 10.1111/poms.12065.

L. Zhen. Tactical berth allocation under uncertainty. *European Journal of Operational Research*, 247(3):928–944, 2015. ISSN 03772217. doi: 10.1016/j.ejor.2015.05.079. URL http://dx.doi.org/10.1016/j.ejor.2015.05.079.

L. Zhen and D. F. Chang. A bi-objective model for robust berth allocation scheduling. *Computers and Industrial Engineering*, 63(1):262–273, 8 2012. ISSN 03608352. doi: 10.1016/j.cie.2012.03.003.

L. Zhen, L. H. Lee, and E. P. Chew. A decision model for berth allocation under uncertainty. *European Journal of Operational Research*, 212(1):54–68, 2011. ISSN 03772217. doi: 10.1016/j.ejor.2011.01.021. URL http://dx.doi.org/10.1016/j.ejor.2011.01.021.