UNIVERSIDADE FEDERAL DO MARANHÃO
Programa de Pós-Graduação em Ciência da Computação

Jorge Rodrigo Silva e Silva

Dynamic Berth Allocation Problem for Tidal Bulk Ports with Inventory Level Constraints

São Luís
2021

Jorge Rodrigo Silva e Silva

Dynamic Berth Allocation Problem for Tidal Bulk Ports with Inventory Level Constraints

Dissertation presented as a partial requirement for obtaining the title of Master in Computer Science, to the Postgraduate Program in Computer Science, at the Federal University of Maranhão.

Postgraduate Program in Computer Science

Federal University of Maranhão

Supervisor: Prof. Dr. Alexandre Cesar Muniz de Oliveira

São Luís - MA

2021

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Silva, Jorge Rodrigo Silva e.
Dynamic Berth Allocation Problem for Tidal Bulk Ports with Inventory Level Constraints / Jorge Rodrigo Silva e Silva. - 2021.
59 p.

Orientador(a): Alexandre Cesar Muniz de Oliveira.
Dissertação (Mestrado) - Programa de Pós-graduação em Ciência da Computação/ccet, Universidade Federal do Maranhão, São Luís, 2021.

1. Berth Allocation Problem. 2. Discrete Optimisation. 3. Greedy algorithms. 4. Gurobi solver. 5. Metaheuristics. I. Oliveira, Alexandre Cesar Muniz de. II. Título.

Jorge Rodrigo Silva e Silva

Dynamic Berth Allocation Problem for Tidal Bulk Ports with Inventory Level Constraints

Dissertation presented as a partial requirement for obtaining the title of Master in Computer Science, to the Postgraduate Program in Computer Science, at the Federal University of Maranhão.

Work Approved. São Luís - MA, 31 de agosto de 2021:

Prof. Dr. Alexandre Cesar Muniz de Oliveira
Supervisor
Federal University of Maranhão

Prof. Dr. Omar Andres Carmona Cortes
Internal Evaluator
Federal University of Maranhão

Prof. Dr. Ricardo de Andrade Lira Rabêlo
External Evaluator
Federal University of Piauí

São Luís - MA
2021

Acknowledgements

I would like to thank first to my family, my brothers, my mother and father that without the support and encouragement in all my path to here, the overcame of the obstacles faced would be not only more difficult, but also lonely. You are the basis and reason of my life, I am eternally grateful.
Secondly, I would like to thank the Prof.Dr. Alexandre Cesar Muniz de Oliveira. The advice, the support, the reception to the laboratory and all the moments he encouraged me to continue with his optimism and prudence. He showed not only flexibility and seriousness in dealing with the daily laboratory activities, but also wit when dealing with the problems faced during the execution of this work and the restriction imposed by the global health crisis.
Finally, I would to thank all my colleagues of laboratory and life. All the moments we share the problems created by the contingencies of life and the experiences emerged are invaluable. Thanks to all.

"Our life is what our thoughts make of it."

(Marcus Aurelius)

Abstract

Abstract

Berth Allocation Problem is a proven to be NP-Hard where a set of ships will be served by a set of berths within a given planning horizon. It's a well known optimization problem, generally related to the combinatorial optimization, having algorithms constructed specifically to deal with problems of that kind. The optimization algorithms can be classified in a general manner in exact, approximate, metaheuristic or hybridized. A common approach is to apply exact methods to solve the problem, since them guarantee the optimum solution, but some cases of a problem is very difficult to be solved by the exact path. In this case the application of approximate or metaheuristic algorithms is taken, but without the assurance of optimality. The objective of this work is to contribute to the study of the berth allocation problem in operational scenarios of bulk ports. The model employed is a discrete and dynamic version of BAP, named Berth Allocation Problem in Tidal Bulk ports with Inventory level conditions (BAPTBS). The model was executed with the Gurobi's solver, a Greedy Heuristic was proposed and also used as an initial solution constructor to the solver and to a GRASP metaheuristic and Two versions of an Evolutionary Clustering Search (ECS) was executed, on e being the standard version and the other a hybridized version using the solver as local searcher.

Keywords: Discrete Optimisation, Berth Allocation Problem, Metaheuristics, Gurobi solver, Greedy algorithms.

List of Figures

List of Tables

Acronyms

ACO Ant Colony Optimization.

AM Analyzer Module.

BAP Berth Allocation Problem.

BAPTBS Berth Allocation Problem in Tidal Bulk ports with Inventory level conditions.

DMCHBAP Dynamic Minimum Cost Hybrid Berth Allocation Problem.

EA Evolutionary Algorithm.

ECS Evolutionary Clustering Search.

GA Genetic Algorithm.

GRASP Greedy Randomized Adaptive Search Procedure.

GSPP Generalized Set Partition Problem.

IC Iterative Clustering.

IGH Iterated Greedy Heuristic.

LS Local Searcher.

MILP Mixed Integer Linear Programming.

PCA Principal Component Analysis.

PFPSP Product Flow Planning and Scheduling Problem.

PSO Particle Swarm Optimization.

QCSP Quay Crane Scheduling Problem.

SA Simulated Annealing.

TBAP Tactical Berth Allocation Problem.

VNS Variable Neighborhood Search.

Contents

1 Introduction

The Berth Allocation Problem (BAP) assumes that a set of ships are served by a set of berths within a given planning horizon, in which some variations of the problem have been proven to be NP-Hard by (LIM, 1998).

According to Bierwirth and Meisel (2010), the BAP can be categorised by its spatial attribute. It is considered discrete when the quay are partitioned and each partition (berth) can only attend one vessel at time. Otherwise, it is considered continuous, i.e., when the berth is a contiguous space, where a given vessel can be served by more than one shiploader. Others criteria can be considered and they are summarised in Table 1

Table 1 – Attributes for port classification (KOVAČ, 2017)

Spatial attribute	**Temporal attribute**	**Handling time attribute**
Discrete	Static	Fixed times
Continuous	Dynamic	Position dependent
Hybrid	Cyclic	QC assignment
Vessel Draft	Stochastic	QC scheduling

Table 1 organises the features used in classification by a common attribute. The spatial attribute differentiates by the berths' organisation in the quay space, in the discrete case the division between the berths is clear and each ship is served by only one berth. In the continuous case, there is no division between the berths and one ship can be served by more than one berth. The hybrid case merge the previous two, defining a clear separation of the berths' space and the possibility of more than one berth attend the same ship. The vessel draft case analyses the availability of mooring a ship considering the ship draft.

The temporal attribute categorises the problem by the ships' arrival. Considering the planning horizon, the static case considers that all ships are available to be served. In the dynamic case the ships will arrive within the planning horizon without a predefined interval, but the arrival is known. When the interval is defined, is the cyclic case and when it is not known is the stochastic case.

The handling time attribute deals specifically with the berth itself. Considering if has a fixed time to serve a ship or if the berth position constrains the possibility of functioning. The assignment and the scheduling are also considered. To try to solve the models in the BAP scenario the techniques used are called optimisation algorithms.

The optimisation algorithms can be classified in a general manner in exact, approximate or hybridised. The exact methods guarantee the optimal solution, if given enough time and considering the search space and the solution feasibility, but the computational cost can be prohibitive for some instances of the problem. In the case of having enough amount of information about the problem a approximate algorithm can be used to obtain a high-quality sub-optimal solutions. Hybridised optimisation algorithms try to use the best features of the two previous approaches(TALBI, 2009). The increasing processing capacity of computers and paralleling techniques have become hybridised techniques an interesting approach (STEFANELLO et al., 2011).

Many researches in BAP address specific problems such gas consumption, constraints related to the quay configuration, being adjacent or opposite, etc. The majority is dealing with discrete and dynamic version of BAP, but because the high model's heterogeneity an objective comparison between them are unavailable (KOVAČ, 2017).

A metaheuristic can be defined as an upper level general methodology used as guiding strategy in designing underlying heuristics that solve specific optimisation problems (TALBI, 2009). The attribute summary used for classification is presented in Table 2.

Table 2 – Criteria for metaheuristic classification (TALBI, 2009)

Nature inspired	Inspired by natural processes
Memory usage	Information extracted dynamically is used during the search
Stochastic	Random rules are applied during the search, different final solutions may be obtained from the same initial solution
Deterministic	Make deterministic decisions, the same initial solution will lead to the same final solution
Population-based search	A whole population of solutions is evolved
Single-solution based search	Manipulate and transform a single solution during the search

The objective of this work is to contribute to the study of the berth allocation problem in operational scenarios of bulk ports. The study is carried out from a mathematical model originally proposed by (BARROS et al., 2011) and which has been updated recently to deal with heterogeneous berths (with different throughput).

In this work, the model employed is a discrete and dynamic version of BAP, named Berth Allocation Problem in Tidal Bulk ports with Inventory level conditions (BAPTBS) (briefly named as BAPTBI or BAPTBS (BARROS et al., 2011)) that dealing with inventory constraints on different loads in bulk ports, served by heterogeneous berths in discrete tidal time windows.

The model is inspired by operation scenarios that arises in the port terminals in São Luís, as the private ones managed by Alumar and Vale. Such ports work with bulks such as coal, soybeans, bauxite, iron ore, alumina, etc (BARROS et al., 2010). Discrete time is not a limitation of the model but rather a requirement for decision-making with strict inventory control in which all levels of raw materials need to be above a contingency threshold.

In addition to the commercial solver, other algorithms are used, such as the ECS framework (Evolutionary Clustering Search (ECS)(OLIVEIRA; LORENA, 2004)), whose laboratory version is adapted for solving sequencing problems(OLIVEIRA; LORENA, 2007), and the GRASP (Greedy Randomized Adaptive Search Procedure (GRASP)(RESENDE; RIBEIRO, 2016)), proposed in this work and implemented from several greedy polynomial-time heuristics.

The main contribution of this work is the validation of the mathematical model for the heterogeneous case of Berth Allocation Problem in Tidal Bulk ports with Inventory level conditions (BAPTBS) from a comprehensive computational experiment. There have been compared computational results from the commercial solver, greedy heuristics and population metaheuristics, all applied over a set of synthetic problem instances generated from realistic situations observed in tidal bulk ports at São Luís.

The instance dataset consists of a set of small and medium-sized instances representing a scenario of up to two weeks of operations in large port terminals. The suite of heuristics and metaheuristics are capable of finding solutions compatible with the commercial solver, but in less computational time for larger instances.

The structure of this work is organised as follows. Chapter 1 introduces the general concept of the BAPTBI and a description of a port functioning. Chapter 3 describes fundamental concepts to understand the development of this work. Chapter 2 summarises related research works recently published, listing the techniques used and the different approaches constructed to solve the problem. The objective functions and the constraints of the model used in this work are explained in chapter 4. The methodology is explored in Chapter 5 and the optimisation approaches proposed to deal with the problem. Chapter 6 reports and discusses the results obtained, also specifies the performance parameters used. Finally Chapter 7 makes the conclusion and final appointments.

2 Related Works

In this chapter the mathematical models and the metaheuristic approaches to BAP are described. The models related to continuous quays and working with containers was not considered.

2.1 Mathematical Models

A model S and two extensions of it, dealing with tidal constraints, was proposed in (ERNST et al., 2017). The two extensions intended to improve performance, The S VI extension added valid inequalities to the model, leading to a tighter Mixed Integer Linear Programming (MILP) formulation, even if this means adding redundant constraints for the integer program. The TI extension added a new variable to the S model to turn the time discrete, a common assumption in the scheduling literature, this model was heavier than the previous extension and to solve this was proposed a two-phase method.

A mathematical model that deals with tidal constraints was proposed by (LIU et al., 2018), specifying aspects such allocation of sections of quays, arrival and departure of vessels. The model was executed on CPLEX and a Genetic Algorithm was proposed to solve difficult instances, with 20 or more vessels.
The model in (CORRECHER et al., 2019) deals with different berth locations, being in an opposite or adjacent side of a given berth, the quay is continuous so the model have constraints about the overlapping of berths and vessels. Some sets are calculated previously, using the available data, to reduce the total amount of variables and restrictions in the model.

A stochastic version of a previous deterministic model in (YAN et al., 2019) is based on a network berth-flow, dealing with delays on the planning horizon, the model considers different arrival times, for each vessel and for each berth of different type a network was constructed. The results shows that the stochastic model surpass the manual allocation, the deterministic model use the maximum tidal time window and, in some cases, the manual approach leaves one vessel without service within the planning horizon.

Based on the port of Jorf Lasfar (BOUZEKRI; ALPAN; GIARD, 2020), the largest in Africa, A model was built to deal with the restrictions of routes made between the storage hangars and the berths, the different water depths for allowing a reasonable draft and heterogeneous berth speeds. Even being inspired by a specific port, the model is formulated with predicates what gives great flexibility to be adapted to any kind of bulk port and improves the computational performance. The principal objective of the model is to maximise the difference between dispatch and the demurrage for all berthed vessels.

To minimise the total amount of CO_2 emission and the total travel time taken by the ships in the context of inland waterway transportation, a river system composed by a network of rivers that connect hundreds of cities and industrial areas, a bi-objective model was proposed in (MANEENGAM; UDOMSAKDIGOOL, 2021). Using branch-and-cut algorithm and a Pareto frontier generated by the ϵ-constraint method, the proposed model is solved with better results than the old method, which use heuristics that rely on the employee routing and scheduling capabilities, what cannot guarantee the quality of solutions.

Making an attempt to integrate the planning and the scheduling decisions to guarantee that products are stored and shipped within the established schedule, in (MENEZES; MATEUS; RAVETTI, 2017) a mathematical model was formulated as a Product Flow Planning and Scheduling Problem (PFPSP) solved by a column generation procedure and a branch-and-price algorithm (B&P). The results obtained show that the proposed method arrived to exact solutions in small and medium instances and so produce upper and lower bound for instances of medium and large-size in scenarios that optimisation packages are not effective.

2.2 Metaheuristic Approaches

Using decision theory and stochastic optimisation techniques, (CARRER; FERSON; GREEN, 2020) try to address tide routing problems (cargo loading and ships scheduling decisions). Considering uncertainty in the sea levels and draft of ships, the model shows robust results next to optimal than the standard approach considering fixed margins. The use of Particle Swarm Optimization (PSO) and Monte Carlos simulations was made for minimise the risk measure, defined considering the expect economic loss when compared with sea levels previously known.

An Genetic Algorithm (GA), a Simulated Annealing (SA) and an Ant Colony Optimization (ACO) were compared, by performance, through the elaboration of the best sequence of berthing aiming to minimise the penalty cost in the berth allocation in a exportation port, localised in Santa Marta (Colombia) (ATENCIO; CASSERES, 2018).

The Tactical Berth Allocation Problem (TBAP) and Quay Crane Scheduling Problem (QCSP) was addressed in (RUIZ et al., 2013) by two versions of the Variable Neighborhood Search (VNS) metaheuristic, making the comparison between VNS and Tabu Search combined with Branch and Price (TS-BP) and VNS with an exact technique called UDS, the former to solve TBAP and the latter to solve the QCSP.

An heuristic has been proposed in (BARROS et al., 2011) based on the Simulated Annealing to minimise the handling service time and to solve difficult instances with the numbers of vessels varying from 10 to 30, each one varying the number of berths.

A Continuous and Discrete BAP was formulated as Generalized Set Partition Problem (GSPP) in (LIN; TING, 2014) and a Simulated Annealing with restart-strategy heuristic (SArs) was proposed to solve the two versions and compared with state-of-art algorithms applied to them. The t-pair test was used and the SArs was equivalent in performance to the others algorithms, but was statistically superior in the continuous case.

Different versions of Variable Neighbourhood Search (VNS) applied to Dynamic Minimum Cost Hybrid Berth Allocation Problem (DMCHBAP) was studied in (KOVAČ; DAVIDOVIĆ; STANIMIROVIĆ, 2018). The Skewed version (SVNS) was superior in solution quality and computational cost when compared with the others metaheuristics and the CPLEX model implementation.

A Reduced version of VNS (RVNS) is studied in (CHEIMANOFF et al., 2020) to be applied in the continuous case of BAP. The work used three different instance datasets and the heuristics sorted the ships with distinctive criteria, the RVNS reach the optimum in almost all instances within 2 minutes against the 2 hours available to the CPLEX version used to run the model.

A Iterated Greedy Heuristic (IGH) is proposed in (LIN; YING; WAN, 2014) for minimising the handling service time, using a greedy algorithm to build the initial solution, applying construction and destruction phases for improvement. The heuristic was tested with three datasets and the results showed that the IGH is effective.

2.3 Final considerations

In Table 2.3 metaheuristics and exact methods used in the mentioned papers are summarised.

Table 3 – Metaheuristics and Exact Methods

Authors	Metaheuristics	Exact Methods
(Atencio & Casseres, 2018)	GA, SA, ACO	-
(Ruiz et al., 2013)	VNS, TS-BP	UDS
(Ruiz & Voss, 2016)	Hybrid POPMUSIC	-
(Barros et al., 2011)	Heuristic, SA	-
(Lin; Ying; Wan, 2014)	IGH	-
(Liu et al., 2018)	GA	-
(Lin & Ting, 2014)	SArs, SA, TS, CS, MA	-
(KOVAC et al., 2018)	SWO, GVNS, VND, MS-VND	-
(Cheimanoff et al., 2020)	RVNS, TS, GRASP	-
(Le Carrer et al., 2020)	-	PSO
Maneegan & Udomsakdigoo	B&C	-

There have been observed that the cases found during this literature review are, for the most part, specific to certain operational contexts around the world. There is already a certain effort to formalise methods that produce datasets that are sufficiently generic to allow a wide comparison of mathematical models and solving algorithms.

However, this work advances towards providing data for experiments and comparison of algorithms, while remaining specific characteristics of the ports in the Maranhense Gulf region that have notorious importance for the Brazilian economy.

3 Main Concepts

In this chapter, techniques related to Operations Research, useful for understanding this work, such as optimisation algorithms and metaheuristics, are given, ranging from mathematical modelling techniques and approximate algorithms, as metaheuristics. The Evolutionary Clustering Search and Greedy-Randomised Search Procedure are highlighted as interesting approaches. Principal Component Analysis is included as a statistical technique needed to a specific development to be presented further.

3.1 Mathematical programming

The area emerged in the midst of the World War II for solving problems related to inspection and repair of airplanes, the improvement of submarine destruction and stock maintenance. Some phenomenological events can be translated into mathematical formulation that allows the designer making predictions and analysis about the phenomenon (ARENALES et al., 2006).

Figure 1 – Modelling Steps (ARENALES et al., 2006)

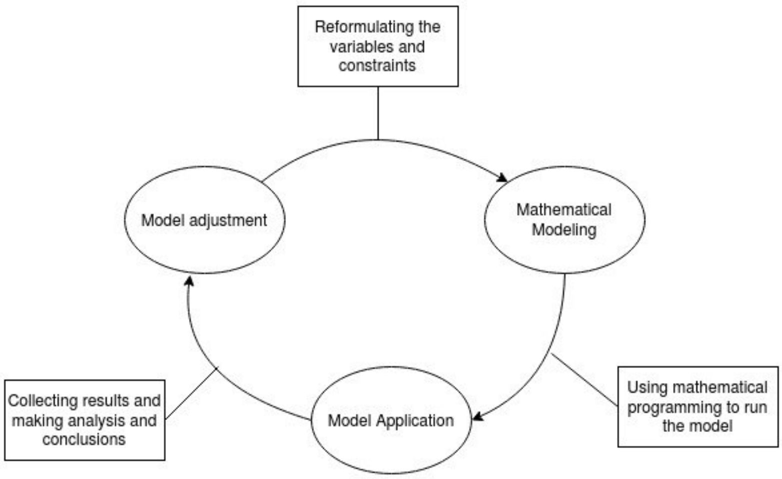


Figure 1 indicates the common steps, initiating the cycle by the mathematical modelling, to arrive in a mathematical model (ARENALES et al., 2006):

- Mathematical Modelling: accordingly to a specialist's description, the variables and constraints are identified. Definitions are made based on the performance criteria. The model is an approximated version of a real case scenario, chasing to have great fidelity.

- Model Application: the mathematical model is solved by well-known methods and algorithms. Commonly incorporated by a computing solver, a software for mathematical programming.
- Model Adjustment: the results are analysed and the model conformity is verified by a specialist. Once the results obtained are consistent with reality, the model is considered validated.

As the image suggests, the process is cyclic and can run indefinitely. The business rules change over time and the model needs to do the same to remain useful.

3.2 Linear Programming

There exist optimisations models that guide the modelling process, working as a type of paradigm, e.g., linear, non-linear, integer, constrained programming, among others (TALBI, 2009). Linear programming can be applied when the objective and constraints are linear functions.

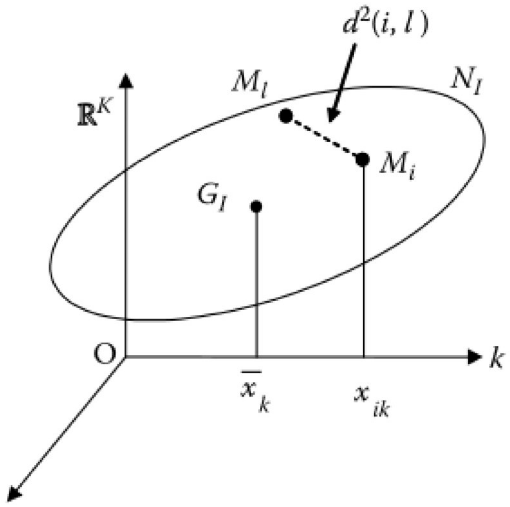
The standard linear program has the form $A\mathbf{x} = b$, subject to the non negativity assumption represented by $x \geq 0$. There are four hypothesis that need to be satisfied in a linear program(BELFIORE; FAVERO, 2013):

- Proportionality: the contribution of every decision variable related to the constraints and objective functions needs to be directly proportional to its value;
- Additivity: the sum of each decision variable individual contributions represents the totality for objective functions or every constraint function.
- Divisibility and Non-negativity: Every decision variable needs to assume values within a given positive interval, even fractional values, considering the constraints;
- Certainty: any coefficient of the objective functions or the constraints are constant and known.

3.3 Principal Component Analysis

Principal Component Analysis (PCA) is applied in a table when each row represents an individual and each column a variable, in this case specifically, the variables need to be quantitative. PCA allows studying the relationship among variables in a *K-dimensional* space, including the analysis of the intensity of the relations by a correlation factor, building synthetic variables (known as principal components) (PAGÈS, 2014).

Figure 2 – Cloud of Individuals (PAGÈS, 2014)



In Figure 2,it is demonstrated the cloud of individuals (N_i), with each individual having a profile (M_i) with coordinates x_{ik}; $k = 1, K$, being developed in the $\mathbb{R}^K$ dimensional space, having the distance between two profiles i and l is measured by the euclidean distance equation 3.1.

$$d^2(i, l) = \sum_k (x_{ik} - x_{il})^2 \tag{3.1}$$

This distance is used to measure discrepancy between profiles and the 'peculiarity' of an individual.

3.4 Metaheuristics

A meta-heuristic can be defined as high-level general guiding strategy for designing of optimization algorithms (TALBI, 2009). There are design and implementation concerns related to modeling, hardware issues, programming, and running environment. Metaheuristic algorithms can employ different search strategies, such as initial solutions, search operators, solution representations, as well ass different parameter settings, always aiming to increase algorithm robustness.

In respect to the domain analysis, there are global and partial exploration strategies. They differ in the problem subdivision. In the former case, the algorithm explores the whole search space, making a more thoroughly exploration. The latter assumes solving a given sub-problem with locally known features. Both strategies can interchange data, performing a collaborative work to construct a global solution (TALBI, 2009).

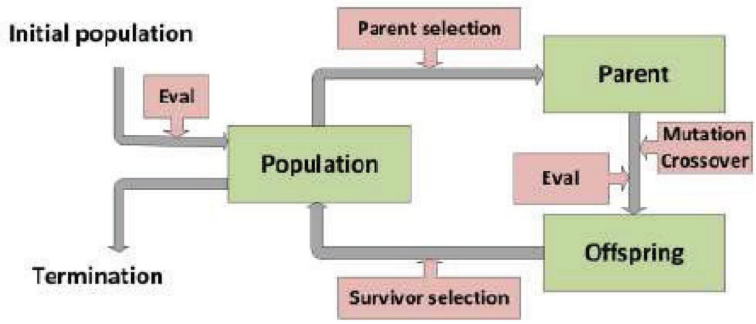
In respect to the function analysis, there are generalist and specialist, the scheme above can be expanded or re-utilized here. But previous scheme is generalist in the sense that all the algorithms involved works to solve the same optimization problem, where a specialist type is the combination of algorithms that solve different problems (TALBI, 2009).

3.4.1 Evolutionary Algorithms

An Evolutionary Algorithm (EA) can be seen as a population-based stochastic optimization algorithm. Accordingly to Kita (KITA, 2011) an EA is a general tool to solve different types of optimization problems. When the problem is poorly understood and there is no specific method to apply to it, the application of an EA is very convenient.

The EA uses the principles of evolution and natural selection, investigated by Darwin (DARWIN, 1859). When the fittest individual in the struggle to survive can pass his genome to the next generation offspring. In this perspective, the EA uses a population of individuals that evolves through generations. The population becomes more and more fitter by operations of crossover and mutation, metaphorizing nature.

Figure 3 – General schema of an EA by (KITA, 2011)



The general EA scheme, showed in Figure 3 summarises its functioning. The individual is an abstract structure that represents a candidate solution of the problem aimed to be solved. The algorithm designer needs to define the components that best deal with the fitness (evaluation) function (TALBI, 2009).

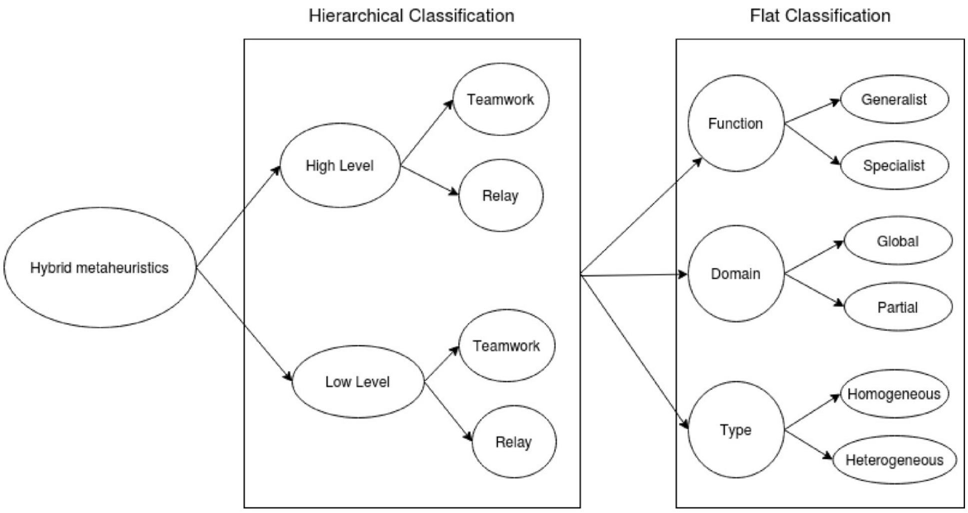
The initial population is chosen at random. Each individual must be evaluated by the fitness function. The process evolves with the selection of individuals for modifying by crossover or/and mutation. The offspring become the new population and the process repeats until a max number of generations is reached or other criteria.

3.4.2 Hybrid Metaheuristic

A main problem with metaheuristics is the *early convergence* caused by the generation of competitive solutions in the early iterations. The use of single-based metaheuristics along with population metaheuristics is a logical step seeking for algorithm improvements.

Hybrid metaheuristics are combination of metaheuristics with others metaheuristics, or mathematical programming, more used in the operations research, or constraint programming, more used in the artificial intelligence community, or machine learning/data mining techniques. Such algorithms can provide excellent search algorithms. A general taxonomy is provided by (TALBI, 2009):

Figure 4 – Classification of Hybrid Metaheuristics



The hybridisation process can be viewed in either a high-level or low-level perspective. The former consists of self-contained metaheuristics, so none direct relationship to the metaheuristics internal workings is made. The latter provide a functional composition of a single-optimisation (low level) method that replaces a given function of a metaheursitic (high level).

Besides the architectural perspective, there are more two types related to the search process itself, the so-called *relay approach*, running as a pipeline, where the executions of metaheuristics or related techniques follow a sequential order. The *teamwork approach* keeps cooperative agents evolving in parallel, each one performing a search in the solution space. The possible combinations are described as follows:

- LRH (Low-level Relay Hybrid): representing the class of built-in metaheuristics in single-based metaheuristics, having few examples.
- LTH (Low-level Teamwork Hybrid):designed to balance the exploration with exploitation,

since the population-based metaheuristics are well-suited for exploration but having weak exploitation.

- HRH (High-level Relay Hybrid): a sequence of complete metaheuristics is performed, playing different role in the search, as exploration the whole population, or sub-populations or the best solution found so far.

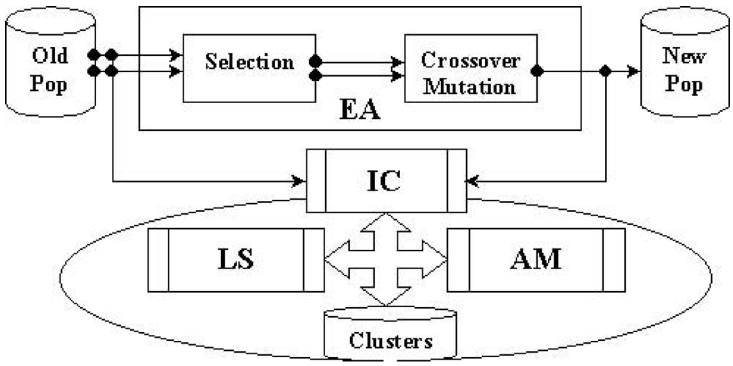
- HTH (High-level Teamwork Hybrid): a team of complete metaheuristics running in a cooperative fashion.

3.5 Evolutionary Clustering Search

The Evolutionary Clustering Search (ECS) uses clustering to find promising search areas, defining an area as an abstract search subspace delimited by its neighborhood relationship. The subspaces can be framed by clusters, defined by the tuple $\Lambda = (c, r, s)$, where r(radius) can be calculated using some distance metric.

In combinatorial optimization cases, c being the center of the cluster, initially defined randomly and further *walking* to more interesting points and s is the associated search strategy (OLIVEIRA; LORENA, 2004).

Figure 5 – ECS Standard Architecture



There are 4 ECS components : an Evolutionary Algorithm (EA), an Iterative Clustering (IC), an Analyzer Module (AM) and a Local Searcher (LS):

- EA: the evolutionary process occurs independently of the remaining parts, working through the whole process as a solution generator, a Genetic Algorithm is commonly used.

- IC: is used after the selection or updated executed by the EA, grouping individuals by the solution they represent and not in a direct process, using the center as an approximated representation of all the solutions in the cluster c

- AM: indicates a probable promising cluster, by its analysis in regular intervals of generations, using the cluster density (λ) to decide if a cluster remains or is eliminated.
- LS: is responsible for the exploitation process, being used when a new point is grouped or an promising area is detected by the AM.

The main equations used by ECS are described bellow:

$$r_t = \frac{x_{sup} - x_{inf}}{2 \cdot \sqrt[n]{|C_t|}} \tag{3.2}$$

In Eq.3.2 is calculated a common radius for all clusters, in each generation t. The $|C_t|$ indicates the current number of clusters, where initially $C_t = MC$, MC informs the pre-defined maximum number of clusters. The x_{sup} and x_{inf} are, respectively, the domain upper and lower bound of the x variable.

$$c_i^{'} = c_i + \alpha \cdot (p_k - c_i) \tag{3.3}$$

In Eq.3.3 updates the cluster center ($c_i^{'}$), by a step of the cluster(c_i) toward a selected individual p_k within it, given by $\alpha \cdot (p_k - c_i)$, being $\alpha \in [0, 1]$ is a disorder degree associated with the assimilation process.

$$\delta_i \geq PD \cdot \frac{NS}{|C_t|} \tag{3.4}$$

Equation 3.4 specifies a threshold that activates the AM to run the LS, where the δ_i is the cluster density, representing the total amount of updates and selections made by the EA, and PD indicates the pressure of density controlling the AM sensibility and NS informs the number of individuals selected for the evolution process in each generation.

3.6 GRASP

The Greedy Randomized Adaptive Search Procedure (GRASP) is a global optimisation algorithm, by repeatedly sampling stochastic greedy solutions and a refinement by local search procedure to reach a local optima. It is centred in a construction mechanism based on stochastic and greedy step-wise procedure, this approach limits the selection and order-of-inclusion of the solution elements considering the value they are expected to have.

The algorithm works as follows. It receives a value $\{\alpha \in \mathbb{Q} : 0 \leq \alpha \geq 1\}$, the closer to 1, the greedier the algorithm. A random start solution is constructed to define the first best solution, then the algorithm runs until reach the optimum. If it is known or the

Figure 6 – Pseudocode of GRASP by (BROWNLEE, 2011)

```
Input: α
Output: S_best
S_best ← ConstructRandomSolution();
while ¬ StopCondition() do
    S_candidate ← GreedyRandomizedConstruction(α);
    S_candidate ← LocalSearch(S_candidate);
    if Cost(S_candidate) < Cost(S_best) then
        S_best ← S_candidate;
    end
end
return S_best;
```

maximum number of iterations. Building a partially greedy solution and applying a local search to make a refinement, saving the best solution found in the process.

3.7 Final considerations

In this chapter, technical aspects were addressed seeking to clarify concepts inherent to mathematical and heuristic models for solving combinatorial problems. The main features of heuristic and metaheuristic algorithms were highlighted. ECS and GRASP are representative and popular metaheuristics, justifying as main options for a suite of algorithms capable of solving large instances of BAPTBI.

ECS incorporates the efficiency of population metaheuristic algorithms hybridised with powerful local search engines. However, little or no specific knowledge about the problem is incorporated. GRASP, in turn, manages to associate heuristic knowledge in the form of greedy criteria with a multi-start mechanism that makes the algorithm more robust. Both approaches ensure a good argument in terms of diversified strategies.

4 BAPTBI

In this chapter, an improvement of the model proposed by (BARROS et al., 2010) is presented. The decision variable is reformulated as well as the related constraints and objective functions.

4.1 Initial Assumptions

The model is based on the importation port scenario, having discrete tidal times ($\approx$ 12 hours) and discrete berths with different load speeds of bulks. Each ship can only moor in one berth at time, and each ship have an arrival tidal, within the maximum number of tidal time windows (TTW). The model was particularly inspired by the ALUMAR's port in São Luis, Maranhão, dealing with bulk materials like soy, iron, bauxite, coal and wheat, even so the model is more abstract than real world application; therefore, it is more academic than industrial(BARROS et al., 2011).

4.2 Input Parameters

Each instance provide a specific set of data and a pre-calculated set(h_{il}), used for calculating the objective value functions. The parameters are described bellow:

- N: set of ships;
- M: set of tidals;
- L: set of berths;
- K: set of operated goods in the yard;
- a_i: tidal arrival of ship i;
- v_l: work speed of the berth l;
- e_k: stock initially available for good k;
- c_k: production rate of good k;
- h_{il}: handling time for ship i in the berth l;
- q_{ik}: capacity of ship i of transportation of good k;

4.3 Decision Variables

The decision variable is a 3-D binary matrix defined by the elements of the sets N, M and L.

$$y_{ijl} = \begin{cases} 1 & \text{if the ship } i \text{ is allocated to TTW } j \text{ and berth } l \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

4.4 Constraints

There are four constraints related to tidal arrival of the ships, the overlap between them and the inventory restriction about the acceptable level of each good in it.

$$\sum_{j=1}^{a_i-1} \sum_{l=1}^{|L|} y_{ijl} = 0, \qquad \forall i \in N \tag{4.2}$$

In Eq.4.2 the ships cannot be moored before its arrival in the port. For this the tidal times before its effective arrival are set to 0.

$$\sum_{j=a_i}^{|M|} \sum_{l=1}^{|L|} y_{ijl} = 1, \qquad \forall i \in N \tag{4.3}$$

In Eq.4.3 the tidal times are set to 1, indicating that ships can be served in any tidal after its arrival.

$$\sum_{\substack{n=1 \\ n\neq i}}^{|N|} \sum_{\substack{m=j \\ m\leq |M|}}^{j+h_{il}-1} y_{nml} \leq (1 - y_{ijl})|N||M|, \qquad \forall i \in N, j \in M, l \in L \tag{4.4}$$

Equation 4.4 avoids the overlap of ships in the planning horizon, indicating that no ship will be moored in the berth and within the attendance of any other ship.

$$\sum_{i=1}^{|N|} \sum_{l=1}^{|L|} \sum_{z=a_i}^{j} \frac{\min(j - a_i + 1, h_{il})}{h_{il}} q_{ik} \times y_{izl} \leq j \times c_k + e_k, \qquad \forall j \in M, k \in K \tag{4.5}$$

Equation 4.5 prevents the goods rarefaction, not allowing a ship to be moored if there are not sufficient good stock to load the ship not reset the inventory levels.

4.5 Objective Functions

Two objective functions are proposed for this model, aiming to attend the minimisation of handling time and demurrage. Each function is executed apart, resulting in different solutions for the same instance.

$$\min \sum_{i=1}^{|N|} \sum_{j=1}^{|M|} \sum_{l=1}^{|L|} (j + h_{il} - a_i) \times y_{ijl} \tag{4.6}$$

Equation 4.6 represents the minimisation of the total amount of time taken by a set of ships to be moored and unberth in the port. This is calculated by the summation of all the handling time of all ships, the same ship can have different handling times by changing the berth which it is docked.

$$\min \sum_{i=1}^{|N|} \sum_{j=1}^{|M|} \sum_{l=1}^{|L|} d_i(j + h_{il} - a_i) \times y_{ijl} \tag{4.7}$$

The demurrage objective function is represented by Eq.4.7, it is a weighted version of the previous equation. Even seeming similar, the equations put different challenges to the model, not only needs to minimise the gap between the tidal time of the mooring of a ship and its arrival time, but also the weight associated with it.

4.6 Final considerations

The mathematical model under study incorporates several improvements over the previously published version, such as the use of a single decision variable, avoiding extra constraints for model linearisation. In addition, the model also provides for the possibility of berths with different throughputs. In bulk ports, berths with higher loading speeds, in general, are allocated to larger ships.

The mathematical model maintains its original characteristic, which is the discretisation of the quay and planning horizon, which allows carrying out inventory control at each time unit. The model does not lose generality when representing the time unit as TTW since this is a concept that can mean from 12 hour tidal windows or smaller time intervals, depending on the tide conditions of each port.

5 Methodology

In this chapter, aspects of the work related to exact and approximating solvers, instance generation, experimentation, and validation of results are described.

5.1 Solvers

There are some solvers available to deal with the optimisation problems that operational research deals with, such as Gurobi[1] and CPLEX [2]. Commercial solvers are generally based on exact methods coming from mathematical programming. However, there are several metaheuristic frameworks based on heuristic techniques, as ParadisEO [3], GALib [4] and BRKGA [5].

5.1.1 Commercial solver

Exact methods, usually commercial solvers, play an important role in validating heuristic algorithms, as they serve as a baseline for comparing results.
In this work we use the Gurobi's solver, version 8.1 with academic license to run the model as a Mixed Integer Linear Programming (MILP) with default configurations to build the baseline used to compare the results of different algorithms applied to the problem.

This particular solver was chosen by the resources available, the detailed documentation and the library provided in many programming languages. Also the academic license is available, allowing to use the solver in a single computer or a university's local-area network [6].

The solver uses simplex or barrier for continuous models and branch-and-cut for MILP [7]. The user can modify a great number of parameters allowing different configurations, still making possible a performance analysis about these different configurations in solving the set of instances.

1 https://www.gurobi.com/resource/starting-with-gurobi/
2 https://www.ibm.com/support/pages/downloading-ibm-ilog-cplex-optimization-studio-v1290
3 http://paradiseo.gforge.inria.fr/
4 https://www.swmath.org/software/4086
5 http://mauricio.resende.info/doc/brkgaAPI.pdf
6 https://www.gurobi.com/academia/academic-program-and-licenses/
7 https://www.gurobi.com/documentation/8.1/refman/cpp_grbmodel_optimize.html

5.1.2 Metaheuristic framework

Metaheuristic solvers are used to providing quality solutions for large instances where traditional and exact solvers tend not to find solutions in a reasonable computational time. In this work, the previously proposed Evolutionary Clustering Search (ECS) has been tested as an alternative for solving BAP instances.

5.1.3 The Greedy Heuristics

A greedy algorithm makes choices locally optimal hoping to reach near global optimal solution. In general, algorithms having linear running time growth are interesting when applied to large instances where exact methods, as those based on Enumeration or Dynamic Programming, tend to fail in finding quality solutions in a reasonable time (CORMEN et al., 2009).

The Greedy Heuristic (GH) proposed to solve the BAPTBI, described in Algorithm 1, receives as input the following parameters: S (Set of ships), B (Set of berths), λ defined criteria for building a priority queue and M (A set of Tidals).

The heuristic sorts the ships by a given greedy criteria, as arrival TTW, and for each element in the ordered set (O), it is defined the berth that offers the minimum time to complete the service (waiting and handling time). Then the Y set of remaining decisions is updated. The λ criteria is a rate, defined by the parameters related to the ship, is proposed as follows:

- A $= \frac{1}{ai}$
- B $= \frac{1}{\sum_{k \in K} qik}$
- C $= \frac{ai}{\sum_{k \in K} qik}$
- D $= \frac{1}{di}$
- E $= \frac{ai}{di}$
- F $= \frac{1}{di \times (\sum_{k \in K} qik)}$

Algorithm 1 Greedy Heuristic

```
Require: λ, S, B, M, α
Ensure: Y
  Q ← buidPriorityQueue(S, λ)
  Y ← ∅
  Σ_{i∈S} Σ_{j∈M} Σ_{l∈B} Y_{ijl} = 0
  for all i ∈ Q do
      best_fo ← ∞
      berth ← 0
      for all b ∈ B do
          delay ← waitingTime(i, b)
          handling ← handlingTime(i, b)
          fo ← calculateObjFunc(delay, handling, i)
          if best_fo ≥ fo then
              best_fo ← fo
              berth ← b
          end if
      end for
      j ← defineTidal(i, berth)
      l ← berth
      Y_{ijl} = 1
  end for
  if infeasible(Y) then
      repair(Y)
  end if
  return Y
```

In order to let the algorithm well explained, the functions used in its structure needs to be described. This will be done in the list bellow:

- buildPriorityQueue(S, λ) : The function receives the set of ships and a sort criteria and return an ordered set of ships to be served. Like a heap data structure.
- waitingTime(i,b) : Given a ship i and a berth b, the function indicates the number of tidals a ship will need to wait to be moored in the specified berth.
- handlingTime(i,b) : The function return the amount of tidals needed to serve the ship.
- calculateObjFunc(delay, handling, i) : Function that calculates the objective function value using the handling time and the calculated delay, once the objective functions considered use this for its calculus.
- defineTidal(i,berth) : The function takes a given ship i and a berth b and calculates the available tidal for berthing.
- infeasible(Y) : It is a binary function that verify the solution feasibility, testing it by the application of the model's constraints.
- repair(Y) : It is an abstract function that turn the solution into a viable answer to the problem.

5.1.4 GRASP

The GRASP uses the previous greedy heuristic to build the solutions. The solution are refined by a swap local search, storing the best solution found so far at the end of the search process.

The swap local searcher divides the ships by their arrival in ϕ sections, given by performance parameter. A section can be understood by an interval (σ) in the arrival set, having the length defined by the equation 5.1. For each section the ships are compared and if a swap in their allocated berth improves the solution, the change is accepted and the procedure goes for all sections.

$$\sigma = \frac{|M|}{(2 \times \phi)} \tag{5.1}$$

5.1.5 Evolutionary Clustering Search (ECS)

Evolutionary Clustering Search (ECS) is a generic framework that combines an evolutionary metaheuristic with a clustering algorithm to detect promising search areas for subsequently exploiting by problem-specific local search procedures. (OLIVEIRA; LORENA, 2007). ECS based approaches have been applied to several optimisation problems (FILHO; NAGANO; LORENA, 2007; CHAVES; CORREA; LORENA, 2007; CHAVES; LORENA; MIRALLES, 2009; OLIVEIRA; CHAVES; LORENA, 2013).

The current architecture of an ECS metaheuristic is shown in Fig.5. For sequencing problems(OLIVEIRA; LORENA, 2007), as Minimisation of Open Stacks Problem (MOSP), the Local Search component usually is a 2-Opt local search that evaluate all possible valid combinations in the neighbourhood.

The flexibility of ECS allows replacing the coupled evolutionary algorithm with any other population metaheuristic capable to feed the clustering process. In addition, the Local Search engine can also be replaced(OLIVEIRA; CHAVES; LORENA, 2013).

In this work, the 2-Opt has been replaced by a *controlled call* to the Gurobi's solver. It is understood by *controlled call* to call the solver limiting the number of iterations so that it assumes the function of a mere local search algorithm. This implementation decision was mainly motivated by the satisfactory performance reached by Gurobi's solver, even facing larger instances.

5.2 Model Instances

The mathematical model has been tested using synthetically generated instances to allow analysing the model's scalability as well as the impact of the model's parameters on the solvers' running time. The instances are generated by algorithmic means, not with random values. This procedure was adopted to make a smaller search space, keeping the feasibility and consistency.

In a port import terminal, the raw material needs to be available in yards to feed the manufacture daily. Negotiation with suppliers and shipowners is carried out well in advance by the logistics teams. The loaded trips are scheduled in a planning horizon and delivery lay-day windows are defined so that the operational teams can work in the short term to prepare the port lineup. Although the logistics teams work with a certain safety margin, the lineup represents a sequence of decisions that can impact the production of the manufacture.

An instance generator has been designed to create realistic situations, yet adding a little more risk of collapse to the manufacturing operation, based on hypothetical situations that a bulk port deals. Some parameters are considered in the generation process, some of them are just input parameters,independent set, differently of the dependent set.

The independent set is defined considering a standard bulk port, dealing with different goods (or raw material - K), equipped with L berths with different throughput (v_l). The number of goods in the port yard also affects the vessels' compartments (capacity of ship i of transportation of good k - q_{ik}) and the dimension of other input parameters, as stock initially available (e_k) and production rate (c_k).

The maximum number of tidal windows is calculated for each instance based on the total of cargo transported by all the vessels and the throughput capacity installed in the port terminal instanced. The maximum number of TTW is defined by Φ, in the formula is described bellow :

$$\Phi = \frac{\sum_{i \in N} \sum_{k \in K} q_{ik}}{|K|} \times \frac{\varphi}{|\sum_{l \in L} v_l - \sum_{k \in K} c_k|} \tag{5.2}$$

where:

- Φ is the maximum number of windows for berthing the ships.
- φ is a slack variable, empirically defined.

The density is a parameter related to the amount of goods transported by the ship given by a percentage, i.e. if the total amount of goods is $K = 4$ and the ships has 75% of density, each ship works with 3 goods with cargo randomly chosen.

The set of ships increases in a regular basis for each configuration of berths and goods, building a scalable instances set. The model parameters such as number of ships, berths, goods and density is given as input and others are defined randomly within a previously defined interval. An instance is presented as follows.

Table 4 – Instance example for bulk ports with heterogeneous berths

```
set N := 1 2 3 4 5 6 7 8 9 10;
set M := 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ··· 35;
set K := rawMatter1 rawMatter2 rawMatter3 rawMatter4;
set L := 1 2 3;
param v :=
1 5
2 4
3 2;
param a :=
1 1
2 2
⋮
10 11;
param e :=
rawMatter1 49
⋮
rawMatter4 57;
param ck :=
rawMatter1 3
⋮
rawMatter4 2;
param q : rawMatter1 rawMatter2 rawMatter3 rawMatter4 :=
1 6 0 0 0
2 3 0 0 0
⋮
10 10 0 0 0;
```

Table 4 shows the general structure of an instance. The first 4 lines are the sets, the K set is constructed with labels, not numbers, but it is built only for readability matters not having any visible important influence.

Each parameter is delimited by the ";" sign, representing the end of data for any given input. The defined structure could be changed to more common representations, i.e. CSV(Comma Separated Values), but this transformation will not make any impact in the problem solution, so this is kept.

5.3 Methodology Flow

An overview of the methodology used to run, test and analyse the experiments can be done by detailing the main steps taken.

5.3.1 Dataset Creation

Recalling that the model is generic, meaning that no specific port was modelled, but is flexible enough to be adapted and specified. The model has no real world dataset *per se* and an artificial version is required to be executed in the model.
We create the instances within a specified range, previously used in the preceding tests, of

ships, berths and goods[8]. The previous instances was handmade, not making the instance growth systematic and controlled. To contour this problem, we elaborate an algorithm considering the conjectures exposed in section 5.2.
The goal is to analyse the increasing in the difficult of finding a feasible solution, given by the time taken to achieve, before the threshold defined ($\approx$ 1h20min), The comparison of the actual created instances and the previous used will not be made because the methodology used in creating them are different and no objective and effective mean was found to make the comparison. The amount of instance generated by this method was 48.

5.3.2 Model Analysis

After running the instances, we analyse the results to verify which variables contributed to hardness the instances, considering the relations between them. The time taken to find the optimum or a feasible solution was the main criteria used to verify the instance growth.
The multivariate aspect of the problem leads to different approaches to study the variables correlation, or fixing some of them to reduce the dimensionality or using statistical tools. Considering the scenario of quantitative variables and the goal of reducing the amount of variables to be analysed and tested, the PCA was chosen.

5.3.3 Metaheuristic Framework Analysis

The metaheuristics former exposed will be executed in the same dataset to verify what of them can reach results near to the obtained by the solver. The metaheuristics are greedy algorithm, a single-solution based metaheuristic, a population-based and its hybrid version with the commercial solver. Each metaheuristic will run 10 times and have 10 min for a complete run. The solver solutions will be defined as the baseline to comparing with the smaller objective function value obtained by each metaheuristic.

5.4 Final Considerations

This chapter offers a general overview of the steps taken in the execution and analysis of the problem. Within the methodological plan, it has been presented issues concerning solvers, instances, and tools which have been employed to validate the mathematical model.

Some implementation details as well as the greedy criteria employed by the heuristic algorithms are detailed. It is important to highlight that greedy criteria are used to rank

[8] Available at: https://data.mendeley.com/datasets/58ph43s6h4/1

the queue of ships and are strongly related to the guidelines and policies for most of the berthing ports. Finally, the workflow is described considering the problem instance generation and the main analysis tools.

6 Computational Experiments

In this chapter, there are presented the computational experiments performed for validation of generated BAPTBI instances as well as algorithms for solving them.

6.1 Experiment 1

Initially, experiments were conducted on a subset of instances to obtain the optimal values for the two objective functions (handling time and demurrage) analysed in this work. Based on the mathematical model, a total of 48 instances has been generated by the algorithm described in Section 5.2. The instances are the same, but demurrage objective function is the only using data related to demurrage,

The result can be observed in Table 7, including the time taken by the Gurobi's solver. The solver was executed with default configurations, giving 5% of the run-time to the execution of pre-solve heuristics. The time limit for solver execution is defined as 4,800 seconds ($\approx$ 1h20min).

The results obtained by the application of the Gurobi's solver on these instances shows that the optimum was reached in 19 of them for handling time and 26 for demurrage. Only 9 instances remains with a gap above 10% for handling time and 6 for demurrage. The results can be seen in Table 6 in the Appendix.

6.2 Experiment 2

The second set of tests is concerned with the performance of the greedy algorithms and the criteria used. The greedy heuristic (5.1.3) was executed changing the criteria on the two objective functions and compared with the Gurobi's solver.

In tables 8 and 9 all the results for the greedy heuristic are presented. Figure 8 presents a subset of instances allowing a graphical analysis.

Figure 7 evidences the superiority of the 'Arrival' (a_i) and the $\frac{a_i}{q_{ik}}$ (criteria A and C, respectively) in comparison to the others. The criterion $\frac{1}{q_{ik} \times d_i}$ has the most unstable behaviour and the use of the demurrage parameter seemed to be no effective in the solutions' quality. Unexpectedly, it is not an excessive assumption to presume that the use of this parameter has the same effect of the 'Load' criterion, once they have an approximated behaviour.

Analysing Figure 8 it is clear that, against the baseline (Gurobi), the *load* criteria (B), defined in 5.1.3, presented low solution quality. The remaining criteria achieve similar

Figure 7 – GH for Demurrage

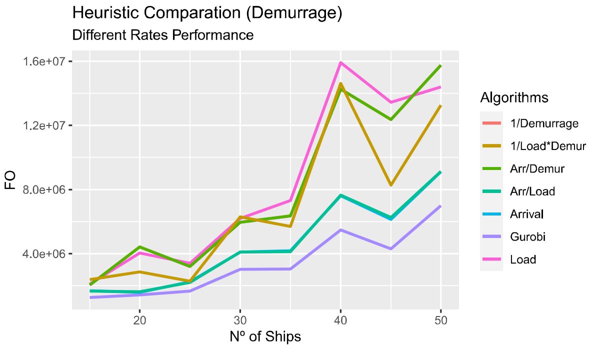
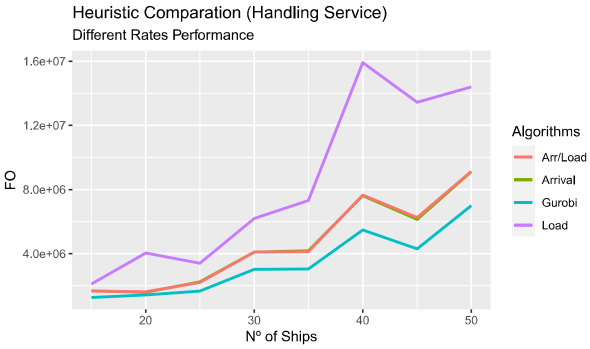


Figure 8 – GH for Handling Time



results and the preference for one of them is arbitrary, indicating that these criteria have the same impact in the generation of solutions, even after applying the local search to them.

6.3 Experiment 3

The third test set is devoted to evaluating the use of the proposed greedy heuristics in the pre-solver phase of Gurobi. Tables 9 and 8 show the results of the use or not of an heuristic to provide a feasible initial solution to evaluate the impact of the use of heuristics in the Gurobi's solver performance. As shown before, the Gurobi's solver running without pre-solve heuristic has an improvement in the speed for finding lesser gaps, in the small and medium size instances. Otherwise for larger instances, it does not reach even a feasible solution within the time limit of 4,800 seconds.

Employing the greedy heuristic proposed in chapter 5 using the "Arrival criteria (A)", the most stable and near to the baseline, the solver can reach the optimum in the large size instances, even if the gap is the same as the solver without heuristic initialisation, in the small and medium sized instances. Only one instance was not solvable, in both cases, the instance with 40 ships, 4 berths and 6 goods. This can be explained by a possible infeasibility of it, but the solver itself has not detected this situation and the instance indeed could have a particular difficult degree.

6.4 Experiment 4

Considering the heuristic execution and to evaluate the advantages reached by the use of different criteria, the GRASP used each version of the previous greedy heuristic to build the initial solution and apply a local search to them. The results in figures 9 and 10 are coherent with that told previously by the heuristic. The GRASP used 25% of the solution constituent elements for building the RCL, having in that configuration, a more "greedy" than stochastic behaviour.

Figure 9 – GRASP for Demurrage

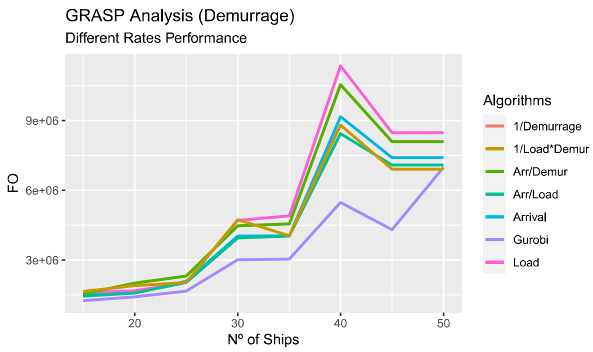


Figure 10 – GRASP for Handling Time

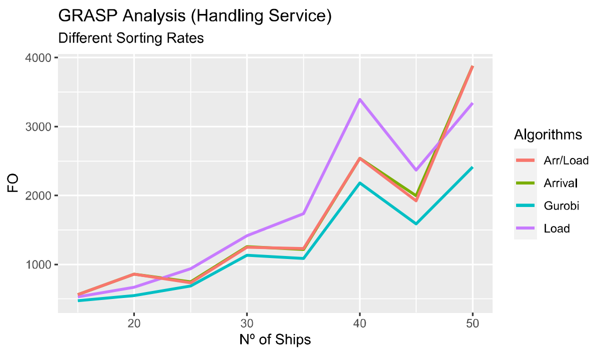


Figure 10 is related to the handling time objective function. In the general picture the "Arrival" and "Arrival/Load (Arr/Load)" reached results near the "Gurobi (Baseline)" in the majority of the instances, but the "Load" criterion had obtained interesting results in the small instances and having a poor performance in the bigger ones, except the instance with 50 ships where it wins the two others, different from the results obtained by the greedy heuristic, once the same criterion doesn't reached even near the baseline considered in any instance.
The demurrage objective function is represented in Figure 9. The "Load" criteria again loses in comparison to the others ones, but when mixed with demurrage in the criteria "1/Load*Demurrage" the results improved and compete with the others, winning in the instance with 50 ships obtaining a result equal to the baseline.
Here is important to make a note, there is no criteria that wins in all the instances in the both objective functions. The possible explanation to this result is that some criteria, even having a poor general performance in the general scenario, provides a good starting point to the local search used by the GRASP.

The overview of all the algorithms discussed until now can be analysed in figures 11 and 12. The overview shows a general good performance of the GRASP algorithm in the handling time objective function, represented in Figure 12, winning or reaching results near the others algorithms, but losing in the 50 ships instance. The ECS with the two different local search mechanisms has a stable behaviour through the instances, getting far away from the baseline with the increasing sizes of the instances. The greedy heuristic wins only in the instance with 45 ships, almost reaching the baseline, but loses or reaches the same results in the remain.

Figure 11 is related to the demurrage objective function. The GRASP only wins in the instance with 50 ships, reaching the value obtained by the baseline and winning all the others algorithms. The ECS performs better with the 2-Opt local search instead the

Figure 11 – Overview of Demurrage

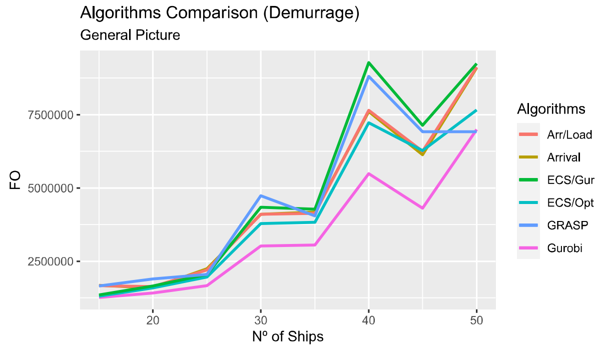
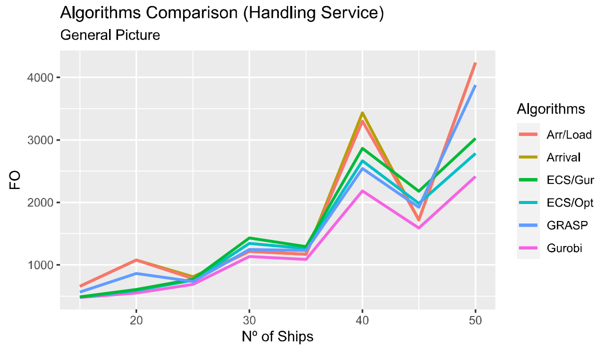


Figure 12 – Overview of Handling Time



gurobi's one, the results take us to consider that the use of a less sophisticated and more exhaustive local searcher can give better results, because the ECS using gurobi as a local search gets worse results in the increasing size of instances while the ECS using the 2-Opt remains stable and competitive in relation to the others algorithms considered.

6.5 Experiment 5

To analyse the possible correlations between the variables of the problem, the PCA was used. A subset was chosen to verify the linear combinations and define the most important variables. The output variable was the time taken by the solver to find the optimum or approximated solution, some measures was implicit defined in the dataset formulation: the total amount of cargo worked in the port, the sum of the berth speeds and the total time windows as also used.

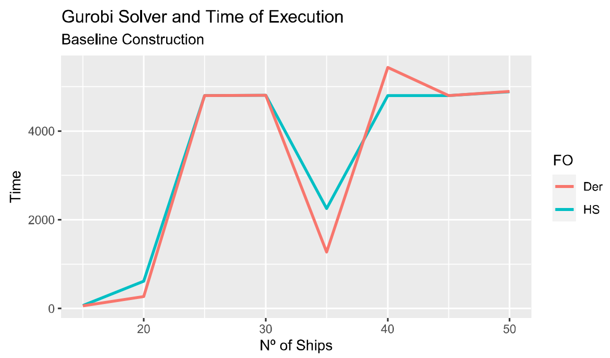
In Table 6 the results are summarised. The 'Time' column is related to the total amount of time taken by the solver to find a feasible or the optimum solution of a specific instance, indicated by the columns N, B and K, as mentioned before the port deals with ships having 100% of the operated goods, so the density is not specified.

Table 5 – Subset with fixed berths and goods

			HS		**Demurrage**	
N	**B**	**K**	**Time**	**FO**	**Time**	**FO**
15	4	5	68,27	475	58	1259000
20	4	5	616,64	547	269	1419000
25	4	5	4801,41	689	4801	1670000
30	4	5	4803,88	1134	4803	3018000
35	4	5	2249,75	1086	1273	3050000
40	4	5	4800,66	2181	5430	5484000
45	4	5	4800,47	1590	4800	4309000
50	4	5	4884,66	2413	5211,73	6994000

The graphical analysis of the results, to plot a 2-D graph first is necessary to reduce the number of considered variables. The graph in Figure 13 shows the time taken by the solver for each ship, fixing the number of berths and goods, evaluating the two objective functions to see a correlation or not. The same subset will be used to analyse the heuristic and metaheuristics considered in this work.
Figure shows that the addition of a weight, defined as demurrage(d_i), does not affect

Figure 13 – Subset of table 6 for graphical analysis



heavily the results obtained by the solver, even if in some points one function is easier or harder than the other, so a heavy correlation. It's clear that the increasing number of ships directly increases the time taken by the solver to find the optimum or a good solution candidate. But graphical analysis is an initial and empirical technique to analyse the results, once the dataset is multidimensional, what makes the visualisation for a robust analysis difficult. Being necessary the use of more sophisticated techniques to have a solid away to verify the hypothesis made previously, for this case, the PCA was selected as an interesting approach.

The PCA was applied considering the variables in table 7. The first hypothesis is concerned with the total amount of ships(N), berths(B) and goods(K). The PCA was used to verify if the set of considered variables have some effect in the output variable($Time$), considering the graphical analysis, the N set has an important effect and varying the total amount of each set was supposed to affect the output variable.

Some variables was implicit defined because they are spread in the dataset. Figures 14 and 15 show the result of PCA application. The figure 15 demonstrate that the three most important variables was between the implicit ones to explain the data, curiously the N variable is the fourth, showing that the relevance of this variable remains and it also explain the variations in the objective function value and time needed to solve it.
The PCA shows that the B and K sets can remain fixed because this variables has little impact in the time expended by the solver and varying the velocity of each berth and the amount of goods in each ships are a more interesting approach to build a model for creating "hard" instances.

Figure 14 – Relation of all variables

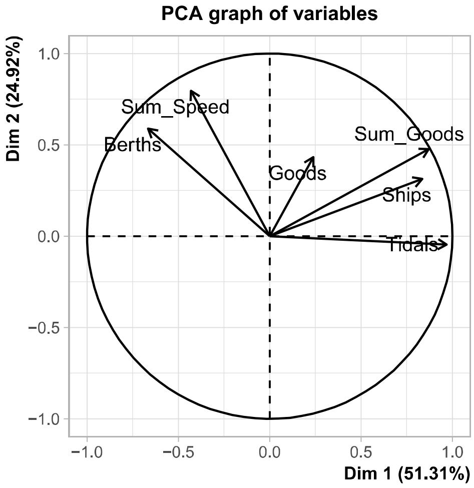
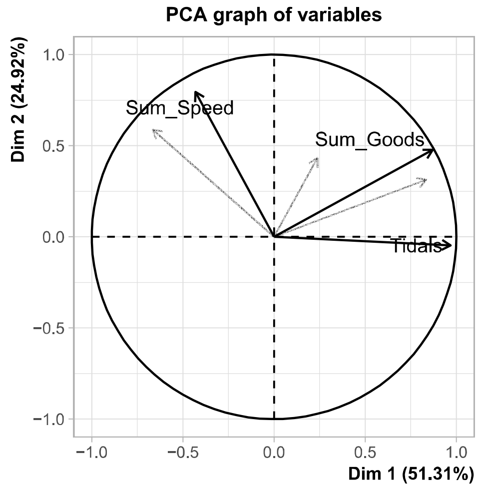


Figure 15 – Relation of the 3 most important variables



6.6 Final considerations

The main results of each experiment can be synthesised in the following manner. In the experiment 6.1 the solver was executed in the default configurations. Reaching the optimality in 37% of the handling time dataset and 54% of the demurrage. In the experiment 6.2 focused in applying greedy algorithms for the datasets. Evaluating a set of criteria while trying to reach results near to optimum. The results showed that from all the analysed criteria, only the load and the arrival had a relevant impact, the others reached the same results with very high gaps in comparison with the results obtained by the solver.

In the experiment 6.3 the use of the proposed greedy heuristic to replace the built-in version of the solver was considered. To analyse the impact of its use, the solver was run without the aid of heuristics. We observed that the solver obtained the same results as its execution in the default version and in some cases a subtle performance gain, even so a more careful inspection of the cause of this similarity is needed, considering that the initial guess of the proposed heuristic is much higher than that of the built-in heuristics.. In the experiment 6.4 address the performance of GRASP embedded with the previously proposed heuristic, as well as the comparison with ECS. We observed that GRASP performs better than heuristics, however not enough to beat the solver (Gurobi). ECS has stable performance, very close to the baseline provided by the solver, beating GRASP in almost all evaluated instances.

In the experiment 6.5 we apply PCA for multidimensional analysis. We observed that the variables considered in the construction of instances had little impact on the actual difficulty of the problem, considering the time taken by the solver as a metric. The results indicated other variables that can be used in the construction of new instances that do not vary the amount of berths and products operated by the port.

7 Conclusion

This work aims to contribute to the study of the dynamic Berth Allocation Problem in Tidal Bulk ports with Inventory level conditions (BAPTBS) heterogeneous case, proposed by (BARROS et al., 2011) in the homogeneous case. The mathematical model is inspired by operation scenarios that arises in the port terminals in São Luís. The model is discrete for time and berth, allowing a strict inventory control. Commercial solver, greedy heuristics and metaheuristics are employed to solver medium size problem instances.

The main contributions are:

1. Validation of the mathematical model for the heterogeneous case of Berth Allocation Problem in Tidal Bulk ports with Inventory level conditions (BAPTBS);
2. The design of an instance generator to create realistic situations based on hypothetical situations that a bulk port deals with;
3. A set of small and medium-sized instances representing a scenario of up to two weeks of operations in a large port terminal;
4. A suite of heuristics and metaheuristics capable of finding solutions compatible with the commercial solver, but in less computational time.

The model used is generic to the point of great flexibility and simplicity, but the actual decision variable formulation impacts the space complexity in reasonable meanings.

The algorithms proposed to solve the instances relied in the different sorting rates, using a heap data structure to build the queue of berthing. Between the 6 elaborated criteria, only 2 of them make significant difference, the remaining criteria reached similar results. The permutation approach of ECS was computational expensive when using the solver as a local searcher.

Future work intends to enrich the current study to contemplate export ports, in which constraints must consider the capacity of stockyards. Furthermore, other objective functions, such as makespan, can be validated in order to initiate a multi-objectivity study. In such study, it is possible to confront the objectives of shipowners, logistics and operation teams. Finally, it is still necessary to go deeper in terms of approximate algorithms hybridised with exact algorithms, such as Dynamic Programming.

8 Appendices

This chapter contains the complete results of the experiments executed in the Pantoja cluster. The decision of put them in a separated chapter was to maintain the aesthetics and an orderly presentation of the subject treated. Other cause is the size of the tables generated, each one taking a whole page.

Table 6 – Results of Gurobi Solver

			Handling time			**Demurrage**		
N	**B**	**K**	**Time**	**FO**	**GAP**	**Time**	**FO**	**GAP**
15	4	5	68.27	475	0.00%	58.04	1259000	0.00%
15	4	6	4800.59	458	1.75%	2368.09	1173000	0.00%
15	5	5	71.89	568	0.00%	64.28	1525000	0.00%
15	5	6	90.9	647	0.00%	71.24	1675000	0.00%
15	6	5	24.65	407	0.00%	17	1060000	0.00%
15	6	6	24.59	458	0.00%	28.64	1256000	0.00%
20	4	4	261.77	612	0.00%	169.58	1455000	0.00%
20	4	5	616.64	547	0.00%	269.26	1419000	0.00%
20	4	6	2910.88	594	0.00%	958.8	1626000	0.00%
20	5	4	69.13	505	0.00%	52.48	1338000	0.00%
20	5	5	138.25	609	0.00%	4800.49	1582000	4.80%
20	5	6	164.22	762	0.00%	170.96	2022000	0.00%
20	6	4	80.28	540	0.00%	60.88	1370000	0.00%
20	6	5	95.38	607	0.00%	76.83	1595000	0.00%
20	6	6	224.86	816	0.00%	2229.26	2152000	0.00%
25	4	4	289.33	721	0.00%	244.68	1708000	0.00%
25	4	5	4801.41	689	7.84%	4801.1	1670000	7.60%
25	4	6	4801.77	919	7.18%	4801.76	2441000	2.09%
25	5	4	237.94	727	0.00%	148.96	1796000	0.00%
25	5	5	674.3	725	0.00%	567.11	1926000	0.00%
25	5	6	4801.74	1012	1.48%	4801.43	2450000	0.53%
30	4	4	4808.14	1105	3.62%	4801.89	2938000	2.79%
30	4	5	4803.88	1134	0.79%	4803.44	3018000	0.56%
30	4	6	4806.71	1395	1.36%	2390.61	3906000	0.00%
30	5	4	4800.84	886	2.26%	312.71	2458000	0.00%
30	5	5	4801.41	930	1.51%	4801.53	2429000	1.81%
30	5	6	4802.41	1109	0.63%	4802.11	2916000	1.27%
35	4	4	983.72	896	0.00%	681.64	2502000	0.00%
35	4	5	2249.75	1086	0.00%	1273.31	3050000	0.00%
35	4	6	4803.54	1234	0.89%	2114.68	3284000	0.00%
35	5	4	4801.28	1210	0.41%	508.54	3415000	0.00%
35	5	6	4804.32	1597	4.88%	1555.22	4546000	0.00%
40	4	4	4800.82	1664	4.87%	4800.49	4813000	2.35%
40	4	5	4800.66	2181	8.44%	5430.46	5484000	3.34%
40	4	6	11928.1	2151	90.37%	11882	5977000	90.15%
40	5	4	4802.5	2015	4.12%	4800.4	5299000	1.02%
40	5	5	4801.25	2272	10.43%	4804.58	6072000	3.77%
40	5	6	4802.62	2661	24.16%	4808.67	7261000	17.57%
45	4	4	4992.29	1699	2.12%	1493.47	3972000	0.00%
45	4	5	4800.47	1590	5.97%	4800.78	4309000	1.93%
45	4	6	4800.7	2246	16.83%	4800.93	5271000	8.16%
45	5	5	4803.67	2101	7.28%	4914.78	5514000	3.61%
45	5	6	4801.56	2661	27.96%	4800.6	6503000	7.66%
50	4	4	4824.35	1987	5.74%	3023.95	5161000	0.00%
50	4	5	4884.66	2413	87.88%	4891.54	6160000	87.83%
50	4	6	4808.95	2755	28.60%	4812.62	7106000	17.35%
50	5	5	4803.8	2685	21.83%	4800.91	9425000	12.63%
50	5	6	4849.31	3526	25.52%	4800.89	6091000	13.99%

Table 7 – Instance Variables

N	B	K	M	$\sum_{l\in L} v_l$	$\sum_{n\in N}\sum_{k\in K} q_{ik}$	$\sum_{kinK} c_k$	**Time_TS**	**Time_DEM**
15	4	6	141	65	5865	13	68.27	58.04
15	5	5	88	75	5698	10	4800.59	2368.09
15	5	6	108	75	6603	14	71.89	64.28
15	6	5	41	105	4648	10	90.9	71.24
15	6	6	54	105	5869	14	24.65	17
20	4	4	119	75	6022	12	24.59	28.64
20	4	5	152	75	7192	16	261.77	169.58
20	4	6	185	75	8601	17	616.64	269.26
20	5	4	64	90	5228	8	2910.88	958.8
20	5	5	86	90	6796	11	69.13	52.48
20	5	6	107	90	7998	15	138.25	4800.49
20	6	4	55	100	5654	14	164.22	170.96
20	6	5	69	100	7034	15	80.28	60.88
20	6	6	88	100	8633	18	95.38	76.83
25	4	4	122	85	7007	13	224.86	2229.26
25	4	5	155	85	8794	14	289.33	244.68
25	4	6	183	85	10234	15	4801.41	4801.1
25	5	4	81	95	6761	12	4801.77	4801.76
25	5	5	105	95	8282	16	237.94	148.96
25	5	6	128	95	10007	17	674.3	567.11
30	4	4	188	65	8417	9	4801.74	4801.43
30	4	5	243	65	10489	11	4808.14	4801.89
30	4	6	311	65	12919	13	4803.88	4803.44
30	5	4	87	95	7519	9	4806.71	2390.61
30	5	5	113	95	9281	13	4800.84	312.71
30	5	6	143	95	11410	15	4801.41	4801.53
35	4	4	128	95	8734	10	4802.41	4802.11
35	4	5	178	95	11558	14	983.72	681.64
35	4	6	225	95	14035	17	2249.75	1273.31
35	5	4	101	110	10518	6	4803.54	2114.68
35	5	6	149	110	14713	11	4801.28	508.54
40	4	4	269	60	11192	8	4804.32	1555.22
40	4	5	351	60	14027	10	4800.82	4800.49
40	4	6	458	60	16853	14	4800.66	5430.46
40	5	4	178	75	11194	12	11928.1	11882
40	5	5	230	75	14032	14	4802.5	4800.4
40	5	6	290	75	16827	17	4801.25	4804.58
45	4	4	220	80	12170	11	4802.62	4808.67
45	4	5	282	80	14868	14	4992.29	1493.47
45	4	6	356	80	17965	17	4800.47	4800.78
45	5	5	187	95	15298	13	4800.7	4800.93
45	5	6	237	95	18450	17	4803.67	4914.78
50	4	4	227	90	13980	13	4801.56	4800.6
50	4	5	296	90	17522	16	4824.35	3023.95
50	4	6	384	90	21493	20	4884.66	4891.54
50	5	5	228	95	18473	14	4808.95	4812.62
50	5	6	237	110	22021	17	4803.8	4800.91

Table 8 – Gurobi's Heuristic Initialization - Handling time

Instances			Without Heuristic			GH Start		
Ships	**Berths**	**Goods**	**Time**	**FO**	**GAP**	**Time**	**FO**	**GAP**
15	4	5	66.42	470	0%	92.36	470	0%
15	4	6	64.77	453	0%	916.51	453	0%
15	5	5	58.06	568	0.18%	67.27	568	0%
15	5	6	77.13	643	0%	109.93	643	0%
15	6	5	16.41	407	0%	19.78	407	0%
15	6	6	25.65	458	0%	36.99	458	0%
20	4	4	138.98	612	0%	142.19	612	0%
20	4	5	146.80	540	0.18%	126.32	540	0%
20	4	6	3958.61	592	0.17%	3789.87	592	0.17%
20	5	4	81.53	505	0%	74.80	505	0%
20	5	5	4801.12	612	3.59%	4800.53	609	2.95%
20	5	6	397.71	755	0%	473.65	755	0.13%
20	6	4	61.72	540	0%	63.00	540	0%
20	6	5	66.93	607	0%	70.79	607	0%
20	6	6	164.40	816	0%	157.06	816	0%
25	4	4	212.92	721	0%	528.62	721	0%
25	4	5	282.05	686	0%	282.29	686	0%
25	4	6	419.20	911	0%	433.79	911	0%
25	5	4	216.61	727	0%	290.44	727	0%
25	5	5	296.56	721	0%	314.53	721	0%
25	5	6	4801.68	1011	0.099%	4801.52	1011	1.08%
30	4	4	4852.47	-	-	4858.08	1250	13.68%
30	4	5	1805.21	1134	1.30%	1119.42	1134	0.09%
30	4	6	4800.66	1391	2.013%	3496.46	1386	0%
30	5	4	4801.07	883	2.15%	4800.85	883	2.15%
30	5	5	1029.96	928	0%	1098.79	928	0%
30	5	6	922.65	1103	0%	1626.01	1103	0.72%
35	4	4	4801.19	897	3.45%	4801.45	915	5.35%
35	4	5	4802.24	1088	0.82%	2234.57	1086	0.09%
35	4	6	4803.8	1234	0.81%	4803.88	1234	0.24%
35	5	4	4801.41	1210	0.16%	1782.29	1210	0%
35	5	6	4802.86	1597	0.37%	4801.2	1599	0.62%
40	4	4	4800.71	-	-	4801.6	1697	6.36%
40	4	5	4802.8	-	-	4800.84	2254	7.94%
40	4	6	11748.9	-	-	11750	2172	72.01%
40	5	4	4804.61	-	-	4810.89	2053	6.23%
40	5	5	4800.34	-	-	4810.66	2359	11.44%
40	5	6	4803.46	-	-	4805.13	2606	19.45%
45	4	4	4801.21	1704	0.94%	4804.13	1691	0.24%
45	4	5	4804.73	-	-	4804.05	1675	8.42%
45	4	6	4803.26	-	-	4804.42	2245	12.60%
45	5	5	4801.04	-	-	4801.17	2116	7.94%
45	5	6	4802.94	-	-	4803.33	2603	14.60%
50	4	4	4800.05	-	-	4803.98	1996	7.86%
50	4	5	4815.76	-	-	5132.39	2286	72.88%
50	4	6	4802.77	-	-	4806.5	2891	30.23%
50	5	5	4801.76	-	-	4801.5	2752	21.62%
50	5	6	4802.04	-	-	4801.45	3508	19.84%

Table 9 – Gurobi's Heuristic Initialization - Demurrage

Instances			**Without Heuristic**			**GH Start**		
Ships	**Berths**	**Goods**	**Time**	**FO**	**GAP**	**Time**	**FO**	**GAP**
15	4	5	43.03	1242000	28.45%	44.96	1242000	28.45%
15	4	6	51.25	1162000	0%	57.87	1162000	0%
15	5	5	41.74	1525000	0%	42.98	1525000	0%
15	5	6	66.93	1660000	0%	68.13	1660000	0%
15	6	5	16.48	1060000	0%	19.81	1060000	0%
15	6	6	29.92	1256000	0%	44.58	1256000	6.96%
20	4	4	91.88	1451000	0%	94.75	1451000	0%
20	4	5	135.03	1404000	0%	148.39	1404000	0%
20	4	6	1770.71	1623000	0%	1594.42	1623000	0%
20	5	4	37.17	1338000	0%	38.53	1338000	0%
20	5	5	76.92	1579000	0%	77.49	1579000	0%
20	5	6	102.94	2011000	0.049%	109.26	2011000	0.049%
20	6	4	41.03	1370000	0%	42.65	1370000	0%
20	6	5	63.37	1591000	0%	63.70	1591000	0%
20	6	6	112.79	2152000	0%	113.31	2152000	0%
25	4	4	137.66	1708000	0%	146.12	1708000	0%
25	4	5	247.85	1643000	0.06%	257.07	1643000	0.06%
25	4	6	319.24	2436000	0.041%	325.42	2436000	0.041%
25	5	4	147.44	1796000	0%	147.59	1796000	0%
25	5	5	225.56	1914000	0.05%	222.05	1914000	0.05%
25	5	6	374.95	2450000	0%	382.91	2450000	0%
30	4	4	4313.76	2911000	0%	4318.55	2911000	0.17%
30	4	5	851.52	3008000	0%	858.96	3008000	0%
30	4	6	1672.47	3898000	0.15%	1824.5	3898000	0%
30	5	4	204.20	2450000	0.04%	206.41	2450000	0%
30	5	5	318.09	2427000	0%	320.30	2427000	0%
30	5	6	558.26	2895000	0%	558.67	2895000	0%
35	4	4	229.38	2502000	0%	232.29	2502000	0%
35	4	5	775.35	3050000	0%	790.76	3050000	0.42%
35	4	6	1595.83	3249000	0%	1424.7	3249000	0%
35	5	4	478.45	3415000	0%	431.979	3415000	0%
35	5	6	1033.42	4546000	0%	1364.94	4546000	0%
40	4	4	2858.62	4801000	0.12%	4805.64	4805000	0.94%
40	4	5	4805.12	-	-	4810.88	5484000	1.16%
40	4	6	11734.7	-	-	11734.9	-	-
40	5	4	2729.12	5299000	0%	2416.99	5299000	0%
40	5	5	4803.9	-	-	4800.72	6020000	0.83%
40	5	6	4802.12	-	-	4801.35	9521000	34.32%
45	4	4	2066.98	3972000	0%	1986.64	3972000	0%
45	4	5	4804.37	-	-	2485.14	4303000	0%
45	4	6	4801.39	-	-	4802.3	8238000	37.23%
45	5	5	3337.27	5487000	0%	2975.87	5487000	0.18%
45	5	6	4803.72	-	-	4802.7	8983000	31.33%
50	4	4	3780.63	5161000	0%	3469.45	5161000	0%
50	4	5	4876.97	-	-	4829.56	9000000	74.91%
50	4	6	4807.6	-	-	4801.62	10786000	44.54%
50	5	5	4802.05	-	-	4813.13	9431000	39.13%
50	5	6	4807.5	-	-	4806.63	12990000	34.93%

Table 10 – Standard ECS and Hybrid ECS for handling time

			ECS/Opt			ECS/Gur		
Ships	**Berths**	**Goods**	**Time**	**FO AVG**	**FO MIN**	**Time**	**FO AVG**	**FO MIN**
15	4	5	119,47	480,3	473	498,73	487,2	470
15	4	6	22,74	474,9	462	389,27	508,2	489
15	5	5	127,99	578,7	574	427,43	581,6	573
15	5	6	77,58	656,3	647	286,29	658,4	645
15	6	5	107,92	409,7	407	257,58	410,8	407
15	6	6	108,63	472,6	458	441,01	482	462
20	4	4	36,94	633,3	616	404,54	639,7	622
20	4	5	38,13	591,4	557	464,1	604,9	590
20	4	6	115,15	699,4	658	500,6	747	670
20	5	4	104,87	517,7	508	388,22	516,9	508
20	5	5	53,22	632,1	615	436,96	641,4	620
20	5	6	68,76	796,8	768	439,85	823,5	803
20	6	4	39,63	552	547	353,19	601	551
20	6	5	118,7	627,9	613	419,47	627,6	621
20	6	6	123,08	840,4	827	486,07	853,8	834
25	4	4	61,95	744,9	731	467,72	755,8	730
25	4	5	36,31	754,8	707	443,64	771,1	742
25	4	6	30,22	1007,9	958	401	1044,2	981
25	5	4	2,97	752,7	733	496,96	760,4	735
25	5	5	96,68	791,1	751	384,66	809,6	777
25	5	6	40,77	1070,2	1051	468,61	1102	1043
30	4	4	73,94	1256	1195	460,49	1374,5	1267
30	4	5	35,04	1345,6	1232	483,93	1430,5	1298
30	4	6	86,56	1756	1639	423,37	2654,12	1853
30	5	4	96,68	940,7	914	365,87	1000,8	934
30	5	5	93,47	1014,7	979	516,24	1080,3	1004
30	5	6	64,71	1238,9	1178	459,09	1326,7	1261
35	4	4	73,5	1012,8	976	495,27	1022,2	976
35	4	5	24,79	1262,4	1186	535,71	1291,9	1207
35	4	6	36,59	1434,5	1344	431,22	1517,3	1424
35	5	4	8,58	1289,1	1248	455,58	1333,9	1296
35	5	6	73,61	1730,2	1682	404,34	1785,9	1746
40	4	4	81,12	2173,8	2055	449,92	2145,9	1983
40	4	5	122,29	2664	2522	360,26	2863,2	2705
40	4	6	150,81	3260,5	2671	408,52	7824	2775
40	5	4	42,83	2189,8	2072	470,89	2289,4	2221
40	5	5	63,04	2605,8	2440	368,69	2818,3	2734
40	5	6	141,21	3004,2	2829	347,93	3307,7	3107
45	4	4	91,05	1926,3	1822	462,16	1963,2	1880
45	4	5	160,31	1983,7	1859	438,26	2179,3	1965
45	4	6	202,74	2610,3	2456	338,17	3121,4	2777
45	5	5	54,56	2426	2343	357,89	2496,3	2252
45	5	6	135,12	2907,6	2683	322,29	3454,8	2934
50	4	4	122,49	2328,5	2152	422,63	2345,7	2229
50	4	5	205,68	2783,5	2564	383,23	3025,7	2703
50	4	6	370,78	7420	3992	387,53	11890,8	4839
50	5	5	128,46	3130,9	2969	390,16	3287,2	3043
50	5	6	157,68	3825,4	3632	305,2	3871,7	3719

Table 11 – Standard ECS and Hybrid ECS for Demurrage

			ECS/Opt			ECS/Gur		
Ships	**Berths**	**Goods**	**Time**	**FO AVG**	**FO MIN**	**Time**	**FO AVG**	**FO MIN**
15	4	5	66,59	1310871	1267000	443,15	1348175	1293000
15	4	6	133,05	1257300	1207000	382,08	1342101	1233000
15	5	5	152,07	1562232	1534000	401,94	1608075	1552000
15	5	6	114,1	1768413	1701000	354,26	1823813	1782000
15	6	5	91,19	1042823	1033159	262,56	1072322	1054108
15	6	6	46,66	1250772	1239136	418,26	1296366	1267071
20	4	4	117,79	1565500	1489000	513,07	1628100	1583000
20	4	5	159,18	1587400	1457000	505,64	1652567	1522000
20	4	6	49,15	1878236	1769000	490,93	2100266	1838565
20	5	4	251,43	1396753	1355076	512,99	1495548	1440076
20	5	5	68,72	1673330	1589000	465,06	1714292	1667101
20	5	6	93,4	2134798	2047546	439,5	2260718	2112244
20	6	4	92,15	1429439	1395136	436,79	1546429	1477198
20	6	5	37,86	1708883	1658164	436,18	1814761	1721254
20	6	6	55,46	2347963	2278104	528,58	2455238	2335000
25	4	4	24,92	1929600	1826000	463,3	1932100	1848000
25	4	5	121,12	1965561	1851000	520,4	2047029	1922789
25	4	6	67,46	2862521	2737000	454,32	3139634	2915336
25	5	4	30,54	2004396	1957000	501,9	2078778	1974092
25	5	5	3,43	2192487	2103117	434,34	2350581	2247000
25	5	6	10,99	2925559	2677000	412,97	3112651	2876000
30	4	4	19,1	3556676	3359870	381,45	3713481	3396925
30	4	5	57,14	3786515	3531000	391,71	4343449	3808839
30	4	6	34,86	4728856	4470493	333,74	5741160	5433059
30	5	4	4,07	2802066	2728291	465,03	2999442	2843456
30	5	5	61,5	2851104	2725254	480,5	3079955	2918260
30	5	6	16,42	3649849	3152000	395,13	4217091	3866163
35	4	4	62,84	3047914	2771000	510,99	3177577	2894000
35	4	5	24,73	3828534	3624000	510,68	4280473	3930000
35	4	6	79,67	4255325	3890883	337,12	4877786	4493511
35	5	4	37,77	3927111	3802123	429,02	4324800	4147123
35	5	6	30,09	5264286	5039167	338,4	6117178	5657499
40	4	4	91,91	6076942	5428000	358,6	6938672	6323998
40	4	5	134,03	7223055	6940000	330,83	9275649	8346696
40	4	6	273,6	7805226	6811598	248,58	10344244	9465996
40	5	4	35,19	6779143	6348000	337,47	7602738	7104000
40	5	5	83,63	7617312	7193267	173,16	9153594	8444000
40	5	6	110,03	8468918	7825667	249,1	10343893	8942985
45	4	4	65,01	5251699	4593000	460,19	6125512	5330927
45	4	5	158,73	6274212	5573315	460,92	7142983	6756480
45	4	6	219,62	7609534	6526000	330,78	9207713	8039427
45	5	5	58,37	7074094	6683000	262,97	8380760	7936178
45	5	6	120,4	8391506	7715000	161,02	10690773	9575255
50	4	4	113,16	6741337	6322000	459,8	7874215	7146449
50	4	5	162,39	7661831	7244346	345,16	9248178	8621745
50	4	6	370,55	10736721	9046398	275,23	12028122	10574983
50	5	5	147,43	8378064	7683732	187,54	10027291	8793789
50	5	6	126,1	11190925	10798253	288,58	13745030	12715000

Table 12 – Greedy Heuristic - Handling time (All Criteria)

Criteria			A		B		C	
Ships	**Berths**	**Goods**	**Time**	**FO**	**Time**	**FO**	**Time**	**FO**
15	4	5	0.000114	655.00	0.000057	559.00	0.000056	655.00
15	4	6	0.000118	823.00	0.000061	10197.00	0.000062	823.00
15	5	5	0.000085	680.00	0.000071	786.00	0.000067	680.00
15	5	6	0.000088	781.00	0.000174	1030.00	0.000077	781.00
15	6	5	0.000066	442.00	0.000111	496.00	0.000055	439.00
15	6	6	0.000131	674.00	0.000148	532.00	0.000066	674.00
20	4	4	0.000088	713.00	0.000156	957.00	0.000168	714.00
20	4	5	0.000106	1077.00	0.000136	3311.00	0.000212	1077.00
20	4	6	0.000116	1221.00	0.000094	59990.00	0.000136	1217.00
20	5	4	0.000090	703.00	0.000077	671.00	0.000078	701.00
20	5	5	0.000103	690.00	0.000093	983.00	0.000091	693.00
20	5	6	0.000126	954.00	0.000111	1086.00	0.000104	955.00
20	6	4	0.000105	579.00	0.000094	845.00	0.000098	579.00
20	6	5	0.000123	658.00	0.000107	951.00	0.000105	671.00
20	6	6	0.000155	863.00	0.000129	1089.00	0.000132	864.00
25	4	4	0.000108	754.00	0.000113	1674.00	0.000093	753.00
25	4	5	0.000101	812.00	0.000123	5869.00	0.000101	777.00
25	4	6	0.000112	993.00	0.000128	17701.00	0.000120	993.00
25	5	4	0.000123	786.00	0.000139	1184.00	0.000113	782.00
25	5	5	0.000115	858.00	0.000132	4152.00	0.000124	857.00
25	5	6	0.000142	1092.00	0.000164	5882.00	0.000150	1095.00
30	4	4	0.000142	1216.00	0.000172	8669.00	0.000165	1219.00
30	4	5	0.000201	1216.00	0.000197	6830.00	0.000200	1213.00
30	4	6	0.000246	1652.00	0.000265	67013.00	0.000258	1652.00
30	5	4	0.000133	946.00	0.000152	1223.00	0.000140	946.00
30	5	5	0.000150	1027.00	0.000176	5023.00	0.000160	1018.00
30	5	6	0.000188	1294.00	0.000232	11259.00	0.000197	1289.00
35	4	4	0.000134	997.00	0.000163	1504.00	0.000142	997.00
35	4	5	0.000165	1170.00	0.000246	2770.00	0.000400	1167.00
35	4	6	0.000197	1396.00	0.000239	19479.00	0.000476	1397.00
35	5	4	0.000190	1254.00	0.000249	1818.00	0.000414	1255.00
35	5	6	0.000267	1646.00	0.000697	2892.00	0.000362	1640.00
40	4	4	0.000291	1957.00	0.000655	12707.00	0.000293	1953.00
40	4	5	0.000425	3433.00	0.000957	13730.00	0.000430	3298.00
40	4	6	0.000476	5315.00	0.000515	254482.00	0.001099	5317.00
40	5	4	0.000356	2366.00	0.000448	3605.00	0.000829	2364.00
40	5	5	0.000459	3569.00	0.000518	13560.00	0.000577	3571.00
40	5	6	0.001209	4013.00	0.000584	39079.00	0.000672	4021.00
45	4	4	0.000611	1724.00	0.000809	8065.00	0.000277	1716.00
45	4	5	0.000687	1716.00	0.000512	33089.00	0.000349	1721.00
45	4	6	0.000475	2261.00	0.001163	54958.00	0.000403	2265.00
45	5	5	0.000403	2115.00	0.000536	9308.00	0.000418	2121.00
45	5	6	0.000490	2607.00	0.000605	11061.00	0.000491	2611.00
50	4	4	0.000401	3505.00	0.000461	3594.00	0.000412	3520.00
50	4	5	0.000467	4237.00	0.000597	55470.00	0.000461	4237.00
50	4	6	0.000545	5244.00	0.000632	166038.00	0.000563	5085.00
50	5	5	0.000682	5328.00	0.000628	3455.00	0.000696	5328.00
50	5	6	0.000712	4510.00	0.000715	4354.00	0.000688	4511.00

Table 13 – Greedy Heuristic - Demurrage (All Criteria) - Part 1/2

Criteria			**A**		**B**		**C**	
Ships	**Berths**	**Goods**	**Time**	**FO**	**Time**	**FO**	**Time**	**FO**
15	4	5	0.000110	1684000.00	0.000060	2112976.00	0.000119	1672000.00
15	4	6	0.000087	1612000.00	0.000068	2129313.00	0.000134	1557000.00
15	5	5	0.000082	1828000.00	0.000075	2218000.00	0.000095	1828000.00
15	5	6	0.000100	2180000.00	0.000089	2663781.00	0.000101	2180000.00
15	6	5	0.000064	1230000.00	0.000056	1593165.00	0.000057	1218000.00
15	6	6	0.000079	1578000.00	0.000068	1534142.00	0.000069	1578000.00
20	4	4	0.000089	2016000.00	0.000087	2931000.00	0.000073	2029000.00
20	4	5	0.000094	1589000.00	0.000113	4043926.00	0.000082	1637000.00
20	4	6	0.000110	1967000.00	0.000115	4022843.00	0.000097	2096000.00
20	5	4	0.000088	1644000.00	0.000082	2054000.00	0.000076	1650000.00
20	5	5	0.000193	2109000.00	0.000208	2225262.00	0.000091	2106000.00
20	5	6	0.000170	2680000.00	0.000116	2849000.00	0.000108	2747000.00
20	6	4	0.000205	1705066.00	0.000119	2167131.00	0.000095	1705066.00
20	6	5	0.000194	2195000.00	0.000114	2961421.00	0.000107	2166000.00
20	6	6	0.000218	2755000.00	0.000149	2995103.00	0.000140	2787000.00
25	4	4	0.000124	2409000.00	0.000132	4457338.00	0.000094	2418000.00
25	4	5	0.000118	2250000.00	0.000122	3409087.00	0.000100	2200000.00
25	4	6	0.000140	2904000.00	0.000147	4707604.00	0.000131	2922000.00
25	5	4	0.000137	2735000.00	0.000127	3564108.00	0.000114	2680000.00
25	5	5	0.000144	2503000.00	0.000155	3232939.00	0.000124	2641000.00
25	5	6	0.000181	3374000.00	0.000176	4366651.00	0.000161	3599000.00
30	4	4	0.000176	3934000.00	0.000198	5758180.00	0.000151	3906000.00
30	4	5	0.000178	4103000.00	0.000260	6199556.00	0.000185	4108000.00
30	4	6	0.000334	5186000.00	0.000703	8355419.00	0.000228	5239000.00
30	5	4	0.000303	3357000.00	0.000375	3913226.00	0.000142	3264000.00
30	5	5	0.000267	3219000.00	0.000298	3898052.00	0.000187	3279000.00
30	5	6	0.000230	4029000.00	0.000278	6650537.00	0.000204	3959000.00
35	4	4	0.000161	3344000.00	0.000235	6196000.00	0.000140	3407000.00
35	4	5	0.000202	4184000.00	0.000274	7316707.00	0.000192	4130000.00
35	4	6	0.000237	4759000.00	0.000307	7632279.00	0.000234	4774000.00
35	5	4	0.000192	4665000.00	0.000524	6528488.00	0.000200	4709000.00
35	5	6	0.000251	6073000.00	0.000694	7857006.00	0.000272	6094000.00
40	4	4	0.000260	6132000.00	0.000401	11558776.00	0.000276	6173000.00
40	4	5	0.000795	7612000.00	0.000532	15910480.00	0.000364	7646000.00
40	4	6	0.000973	8326000.00	0.000625	22341744.00	0.000446	8291000.00
40	5	4	0.000343	7597000.00	0.000452	8537903.00	0.000348	7592000.00
40	5	5	0.000419	8705000.00	0.000589	11822509.00	0.000427	9001000.00
40	5	6	0.000496	9806000.00	0.000651	15493106.00	0.001137	9804000.00
45	4	4	0.000279	6385000.00	0.000774	8714197.00	0.000652	6324000.00
45	4	5	0.000298	6134000.00	0.000909	13448112.00	0.000315	6248000.00
45	4	6	0.000401	8357000.00	0.000510	14142594.00	0.000474	8258000.00
45	5	5	0.000453	7506000.00	0.000518	11059681.00	0.000508	7437000.00
45	5	6	0.000481	9141000.00	0.000580	14762702.00	0.000489	9095000.00
50	4	4	0.000364	6784000.00	0.000524	14207129.00	0.000337	6753000.00
50	4	5	0.000398	9105000.00	0.000574	14403856.00	0.000454	9128000.00
50	4	6	0.000488	10757780.00	0.000669	14720250.00	0.000508	10915780.00
50	5	5	0.000545	9427000.00	0.000722	14446710.00	0.000557	9476000.00
50	5	6	0.000611	12979000.00	0.000780	18458000.00	0.000626	13008000.00

Table 14 – Greedy Heuristic - Demurrage (All Criteria) - Part 2/2

Criteria			**D**		**E**		**F**	
Ships	**Berths**	**Goods**	**Time**	**FO**	**Time**	**FO**	**Time**	**FO**
15	4	5	0.000053	2050418.00	0.000118	2050418.00	0.000135	2382304.00
15	4	6	0.000068	2957767.00	0.000156	2957767.00	0.000109	2295863.00
15	5	5	0.000070	2319000.00	0.000093	2319000.00	0.000091	2557000.00
15	5	6	0.000107	2746759.00	0.000103	2746759.00	0.000105	2730451.00
15	6	5	0.000052	1613110.00	0.000057	1613110.00	0.000069	1323055.00
15	6	6	0.000150	1825142.00	0.000085	1825142.00	0.000083	1628142.00
20	4	4	0.000157	2612000.00	0.000076	2612000.00	0.000094	1896000.00
20	4	5	0.000200	4426197.00	0.000100	4426197.00	0.000118	2859057.00
20	4	6	0.000150	4246668.00	0.000117	4246668.00	0.000142	3781851.00
20	5	4	0.000082	2124074.00	0.000079	2124074.00	0.000100	1927000.00
20	5	5	0.000096	2668099.00	0.000096	2668099.00	0.000244	2719049.00
20	5	6	0.000149	2693854.00	0.000109	2693854.00	0.000252	2972201.00
20	6	4	0.000092	2501198.00	0.000105	2501198.00	0.000144	2130132.00
20	6	5	0.000111	2383164.00	0.000112	2383164.00	0.000163	2676082.00
20	6	6	0.000158	3352103.00	0.000149	3352103.00	0.000186	3331103.00
25	4	4	0.000108	3880660.00	0.000109	3880660.00	0.000128	2667400.00
25	4	5	0.000106	3200100.00	0.000109	3200100.00	0.000139	2287638.00
25	4	6	0.000131	5338549.00	0.000131	5338549.00	0.000172	3330694.00
25	5	4	0.000143	2832013.00	0.000119	2832013.00	0.000148	2580000.00
25	5	5	0.000152	3306159.00	0.000190	3306159.00	0.000170	3087255.00
25	5	6	0.000160	4632000.00	0.000358	4632000.00	0.000218	4444429.00
30	4	4	0.000205	5395616.00	0.000189	5395616.00	0.000217	4648390.00
30	4	5	0.000226	5960040.00	0.000197	5960040.00	0.000246	6296871.00
30	4	6	0.000349	8572290.00	0.000287	8572290.00	0.000287	7911987.00
30	5	4	0.000184	4458629.00	0.000155	4458629.00	0.000188	3453626.00
30	5	5	0.000205	4420244.00	0.000175	4420244.00	0.000191	3927166.00
30	5	6	0.000234	5435012.00	0.000223	5435012.00	0.000227	4422000.00
35	4	4	0.000184	6499657.00	0.000180	6499657.00	0.000197	5196000.00
35	4	5	0.000261	6347905.00	0.000201	6347905.00	0.000212	5693635.00
35	4	6	0.000266	8519360.00	0.000269	8519360.00	0.000266	7363184.00
35	5	4	0.000224	5927368.00	0.000244	5927368.00	0.000230	4674122.00
35	5	6	0.000294	7701714.00	0.000292	7701714.00	0.000319	6777896.00
40	4	4	0.000395	11457362.00	0.000396	11457362.00	0.000448	11144940.00
40	4	5	0.000485	14249341.00	0.000475	14249341.00	0.000517	14616997.00
40	4	6	0.000638	20847332.00	0.000653	20847332.00	0.000707	20529648.00
40	5	4	0.000442	9990599.00	0.000439	9990599.00	0.000423	8321629.00
40	5	5	0.000491	12976244.00	0.000490	12976244.00	0.000565	10418451.00
40	5	6	0.000692	14655344.00	0.000652	14655344.00	0.000639	13061085.00
45	4	4	0.000297	8339000.00	0.000317	8339000.00	0.000352	7971328.00
45	4	5	0.000432	12358609.00	0.000433	12358609.00	0.000411	8269895.00
45	4	6	0.000542	16544277.00	0.000522	16544277.00	0.000486	9533788.00
45	5	5	0.000534	11205952.00	0.001206	11205952.00	0.000542	10539036.00
45	5	6	0.000632	15083435.00	0.001423	15083435.00	0.000590	10556922.00
50	4	4	0.000479	15158170.00	0.000477	15158170.00	0.000444	10291162.00
50	4	5	0.000556	15759931.00	0.000556	15759931.00	0.000646	13252399.00
50	4	6	0.000674	16813922.00	0.000673	16813922.00	0.000668	15608989.00
50	5	5	0.000664	13264998.00	0.001513	13264998.00	0.000754	12586428.00
50	5	6	0.000836	16860500.00	0.001880	16860500.00	0.001801	16696321.00

Table 15 – GRASP for handling time (All criteria)

			A		B		C	
Ships	**Berths**	**Goods**	**Time**	**FO**	**Time**	**FO**	**Time**	**FO**
15	4	5	515.00	563.00	274.30	531.00	516.70	562.00
15	4	6	157.30	674.00	181.20	554.00	157.50	674.00
15	5	5	327.90	621.00	17.00	587.00	327.90	621.00
15	5	6	327.00	621.00	17.00	587.00	327.00	621.00
15	6	5	327.00	621.00	17.00	587.00	327.00	621.00
15	6	6	327.00	621.00	17.00	587.00	327.00	621.00
20	4	4	242.30	648.00	107.30	632.00	236.90	647.00
20	4	5	37.10	861.00	443.00	671.00	37.10	861.00
20	4	6	49.20	998.00	210.80	784.00	334.20	982.00
20	5	4	49.00	998.00	220.00	784.00	334.00	982.00
20	5	5	49.00	998.00	220.00	784.00	334.00	982.00
20	5	6	49.00	998.00	220.00	784.00	334.00	982.00
20	6	4	49.00	998.00	220.00	784.00	334.00	982.00
20	6	5	49.00	998.00	220.00	784.00	334.00	982.00
20	6	6	49.00	998.00	220.00	784.00	334.00	982.00
25	4	4	85.10	721.00	85.10	863.00	208.60	721.00
25	4	5	131.10	750.00	351.70	937.00	542.70	734.00
25	4	6	552.70	980.00	44.50	1290.00	550.20	979.00
25	5	4	571.00	980.00	46.00	1290.00	555.00	979.00
25	5	5	571.00	980.00	46.00	1290.00	555.00	979.00
25	5	6	107.50	1036.00	46.00	1290.00	596.70	1036.00
30	4	4	140.90	1202.00	208.20	1312.00	229.80	1189.00
30	4	5	434.90	1257.00	230.70	1415.00	438.80	1248.00
30	4	6	140.30	1625.00	201.10	2816.00	140.50	1631.00
30	5	4	140.00	1625.00	201.00	2816.00	140.00	1631.00
30	5	5	140.00	1625.00	201.00	2816.00	140.00	1631.00
30	5	6	140.00	1625.00	201.00	2816.00	140.00	1631.00
35	4	4	321.80	1001.00	570.40	1083.00	322.60	1005.00
35	4	5	481.00	1217.00	263.70	1738.00	133.30	1232.00
35	4	6	340.30	1325.00	80.00	1981.00	343.20	1323.00
35	5	4	340.00	1325.00	80.00	1981.00	357.00	1323.00
35	5	6	340.00	1325.00	80.00	1981.00	357.00	1323.00
40	4	4	93.50	1999.00	110.10	2429.00	414.30	2002.00
40	4	5	172.60	2540.00	250.10	3394.00	172.70	2540.00
40	4	6	560.70	4924.00	208.40	10569.00	561.40	4924.00
40	5	4	559.00	4924.00	208.00	10569.00	565.00	4924.00
40	5	5	559.00	4924.00	326.80	2781.00	565.00	4924.00
40	5	6	152.00	3413.00	105.60	3522.00	565.00	4924.00
45	4	4	152.00	3413.00	106.00	3522.00	565.00	4924.00
45	4	5	177.00	1999.00	465.90	2369.00	177.30	1924.00
45	4	6	58.10	2426.00	146.60	3206.00	103.50	2449.00
45	5	5	58.00	2426.00	148.00	3206.00	105.00	2449.00
45	5	6	13.10	2775.00	148.00	3206.00	419.00	2770.00
50	4	4	299.90	3254.00	241.50	2783.00	160.20	3219.00
50	4	5	206.50	3883.00	225.30	3344.00	206.60	3881.00
50	4	6	103.50	6370.00	393.30	6303.00	105.70	6179.00
50	5	5	106.00	6370.00	393.00	6303.00	103.00	6179.00
50	5	6	106.00	6370.00	98.30	3981.00	103.00	6179.00

Table 16 – GRASP for Demurrage (All criteria) - Part 1/2

			A		**B**		**C**	
Ships	**Berths**	**Goods**	**Time**	**FO**	**Time**	**FO**	**Time**	**FO**
15	4	5	178.30	1456000.00	96.10	1604516.00	178.80	1484000.00
15	4	6	93.10	1400000.00	491.00	1580445.00	287.10	1393000.00
15	5	5	532.00	1566000.00	492.00	1580445.00	530.60	1566000.00
15	5	6	529.00	1566000.00	492.00	1580445.00	530.00	1566000.00
15	6	5	529.00	1566000.00	492.00	1580445.00	530.00	1566000.00
15	6	6	529.00	1566000.00	492.00	1580445.00	530.00	1566000.00
20	4	4	486.30	1799000.00	175.40	2038000.00	521.40	1838900.00
20	4	5	503.30	1594000.00	39.00	1689000.00	504.00	1594000.00
20	4	6	579.80	1962000.00	305.90	2554252.75	208.30	2087000.00
20	5	4	580.00	1962000.00	305.00	2554252.75	208.00	2087000.00
20	5	5	580.00	1962000.00	305.00	2554252.75	208.00	2087000.00
20	5	6	580.00	1962000.00	305.00	2554252.75	208.00	2087000.00
20	6	4	580.00	1962000.00	305.00	2554252.75	208.00	2087000.00
20	6	5	580.00	1962000.00	305.00	2554252.75	208.00	2087000.00
20	6	6	580.00	1962000.00	305.00	2554252.75	208.00	2087000.00
25	4	4	196.50	1975000.00	228.60	2584000.00	151.00	1950000.00
25	4	5	408.30	2091000.00	8.00	2040636.75	431.20	2043498.00
25	4	6	294.90	2902000.00	323.00	3145762.00	342.20	2866411.25
25	5	4	295.00	2902000.00	326.00	3145762.00	341.00	2866411.25
25	5	5	295.00	2902000.00	326.00	3145762.00	341.00	2866411.25
25	5	6	295.00	2902000.00	326.00	3145762.00	341.00	2866411.25
30	4	4	295.00	2902000.00	326.00	3145762.00	341.00	2866411.25
30	4	5	279.70	4031000.00	20.10	4710113.50	279.50	3948000.00
30	4	6	91.20	5014000.00	364.90	6126144.00	93.80	5135000.00
30	5	4	91.00	5014000.00	362.00	6126144.00	92.00	5135000.00
30	5	5	91.00	5014000.00	362.00	6126144.00	92.00	5135000.00
30	5	6	91.00	5014000.00	362.00	6126144.00	92.00	5135000.00
35	4	4	91.00	5014000.00	362.00	6126144.00	92.00	5135000.00
35	4	5	321.00	4051000.00	181.00	4904059.50	187.50	4041000.00
35	4	6	423.00	4751000.00	524.50	6360153.50	480.00	4815000.00
35	5	4	422.00	4751000.00	524.00	6360153.50	478.00	4815000.00
35	5	6	422.00	4751000.00	524.00	6360153.50	478.00	4815000.00
40	4	4	480.70	7323000.00	512.30	8816906.00	254.50	7029000.00
40	4	5	384.80	9169682.00	161.10	11360577.00	382.70	8432680.00
40	4	6	42.20	9441043.00	576.50	13681706.00	42.60	9461335.00
40	5	4	43.00	9441043.00	584.00	13681706.00	43.00	9461335.00
40	5	5	43.00	9441043.00	584.00	13681706.00	43.00	9461335.00
40	5	6	579.60	10041000.00	530.00	10630841.00	172.60	9972000.00
45	4	4	584.00	10041000.00	528.00	10630841.00	172.00	9972000.00
45	4	5	511.60	7410000.00	86.70	8480322.00	236.50	7087088.00
45	4	6	302.20	9210000.00	23.00	9312000.00	210.90	9184000.00
45	5	5	304.00	9210000.00	23.00	9312000.00	210.00	9184000.00
45	5	6	418.50	9507000.00	23.00	9312000.00	210.00	9184000.00
50	4	4	342.20	6941176.00	593.50	9142815.00	286.80	7079051.00
50	4	5	341.00	6941176.00	596.00	9142815.00	289.00	7079051.00
50	4	6	154.70	11690150.00	102.40	11520055.00	298.40	12104351.00
50	5	5	151.00	11690150.00	102.00	11520055.00	297.00	12104351.00
50	5	6	385.00	13420000.00	102.00	11520055.00	27.10	13520000.00

Table 17 – GRASP for Demurrage (All criteria) - Part 2/2

Criteria			**D**		**E**		**F**	
Ships	**Berths**	**Goods**	**Time**	**FO**	**Time**	**FO**	**Time**	**FO**
15	4	5	241.70	1552140.00	242.50	1552140.00	532.00	1658273.00
15	4	6	22.10	1567903.00	22.00	1567903.00	536.00	1570238.00
15	5	5	-	-	-	-	531.00	1570238.00
15	5	6	-	-	-	-	531.00	1570238.00
15	6	5	-	-	-	-	531.00	1570238.00
15	6	6	-	-	-	-	531.00	1570238.00
20	4	4	288.70	1760000.00	291.40	1760000.00	465.00	1700000.00
20	4	5	425.60	2014000.00	427.00	2014000.00	561.80	1899000.00
20	4	6	447.40	2921795.25	446.20	2921795.25	243.20	2355816.75
20	5	4	-	-	-	-	241.00	2355816.75
20	5	5	-	-	-	-	241.00	2355816.75
20	5	6	-	-	-	-	241.00	2355816.75
20	6	4	-	-	-	-	241.00	2355816.75
20	6	5	-	-	-	-	241.00	2355816.75
20	6	6	-	-	-	-	241.00	2355816.75
25	4	4	57.10	2370368.75	57.30	2370368.75	160.50	1975000.00
25	4	5	120.60	2311872.75	121.50	2311872.75	565.30	2048000.00
25	4	6	530.80	3564424.00	536.70	3564424.00	404.60	2926604.00
25	5	4	-	-	-	-	407.00	2926604.00
25	5	5	-	-	-	-	407.00	2926604.00
25	5	6	-	-	-	-	407.00	2926604.00
30	4	4	-	-	-	-	407.00	2926604.00
30	4	5	154.20	4479639.50	154.10	4479639.50	330.50	4732664.50
30	4	6	271.40	6267598.50	271.00	6267598.50	381.70	5875328.00
30	5	4	-	-	-	-	381.00	5875328.00
30	5	5	-	-	-	-	381.00	5875328.00
30	5	6	-	-	-	-	381.00	5875328.00
35	4	4	-	-	-	-	381.00	5875328.00
35	4	5	418.00	4562384.00	424.30	4562384.00	42.20	4048859.50
35	4	6	428.90	5714752.00	429.70	5714752.00	448.40	5053184.00
35	5	4	-	-	-	-	445.00	5053184.00
35	5	6	-	-	-	-	445.00	5053184.00
40	4	4	590.10	8744277.00	597.00	8728648.00	268.40	8839591.00
40	4	5	400.30	10542872.00	400.30	10542872.00	510.70	8811041.00
40	4	6	257.80	12826294.00	257.40	12826294.00	555.40	11595009.00
40	5	4	-	-	-	-	555.00	11595009.00
40	5	5	-	-	-	-	555.00	11595009.00
40	5	6	278.60	11270289.00	278.30	11270289.00	62.10	10229185.00
45	4	4	-	-	-	-	62.00	10229185.00
45	4	5	83.50	8099031.00	83.80	8099031.00	309.90	6916076.00
45	4	6	461.40	9096000.00	461.00	9096000.00	273.80	9514000.00
45	5	5	-	-	-	-	272.00	9514000.00
45	5	6	-	-	-	-	36.00	9159016.00
50	4	4	125.20	9455166.00	124.60	9455166.00	374.60	7046383.00
50	4	5	-	-	-	-	373.00	7046383.00
50	4	6	556.10	11299875.00	558.20	11299875.00	379.00	11727415.00
50	5	5	-	-	-	-	382.00	11727415.00
50	5	6	-	-	-	-	382.00	11727415.00

Bibliography

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa operacional-Modelagem e algoritmos*. [S.l.]: Rio de Janeiro: Elsevier, 2006. Citado 2 vezes nas páginas 8 and 20.

ATENCIO, F. N.; CASSERES, D. M. A comparative analysis of metaheuristics for berth allocation in bulk ports: A real world application. *IFAC-PapersOnLine*, Elsevier, v. 51, n. 11, p. 1281–1286, 2018. Citado na página 17.

BARROS, V. H.; COSTA, T. S.; OLIVEIRA, A. C.; LORENA, L. A. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers & Industrial Engineering*, Elsevier, v. 60, n. 4, p. 606–613, 2011. Citado 4 vezes nas páginas 14, 18, 28, and 46.

BARROS, V. H. et al. Problema de alocação de berços em portos graneleiros com restrições de estoque e condições favoráveis de maré. Universidade Federal do Maranhão, 2010. Citado 2 vezes nas páginas 14 and 28.

BELFIORE, P.; FAVERO, L. P. *Pesquisa Operacional para cursos de Engenharia*. [S.l.]: Elsevier Brasil, 2013. v. 1. Citado na página 21.

BIERWIRTH, C.; MEISEL, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, Elsevier, v. 202, n. 3, p. 615–627, 2010. Citado na página 13.

BOUZEKRI, H.; ALPAN, G.; GIARD, V. A dynamic hybrid berth allocation problem with routing constraints in bulk ports. In: SPRINGER. *IFIP International Conference on Advances in Production Management Systems*. [S.l.], 2020. p. 250–258. Citado na página 16.

BROWNLEE, J. Clever algorithms. *Nature-Inspired Programming Recipes*, LuLu, v. 436, 2011. Citado 2 vezes nas páginas 8 and 27.

CARRER, N. L.; FERSON, S.; GREEN, P. L. Optimising cargo loading and ship scheduling in tidal areas. *European Journal of Operational Research*, Elsevier, v. 280, n. 3, p. 1082–1094, 2020. Citado na página 17.

CHAVES, A. A.; CORREA, F. de A.; LORENA, L. A. N. Clustering search heuristic for the capacitated pmedian problem. In: ______. *Innovations in Hybrid Intelligent Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 136–143. Citado na página 34.

CHAVES, A. A.; LORENA, L. A. N.; MIRALLES, C. Hybrid metaheuristic for the assembly line worker assignment and balancing problem. In: BLESA, M. J.; BLUM, C.; GASPERO, L. D.; ROLI, A.; SAMPELS, M.; SCHAERF, A. (Ed.). *Hybrid Metaheuristics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1–14. ISBN 978-3-642-04918-7. Citado na página 34.

CHEIMANOFF, N.; FONTANE, F.; KITRI, M. N.; TCHERNEV, N. A reduced VNS based approach for the dynamic continuous berth allocation problem in bulk terminals with tidal constraints. *Expert Systems with Applications*, n. January, p. 114215, 2020. ISSN 09574174. Citado na página 18.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to algorithms*. [S.l.]: MIT press, 2009. Citado na página 32.

CORRECHER, J. F.; Van den Bossche, T.; ALVAREZ-VALDES, R.; Vanden Berghe, G. The berth allocation problem in terminals with irregular layouts. *European Journal of Operational Research*, Elsevier B.V., v. 272, n. 3, p. 1096–1108, 2019. ISSN 03772217. Disponível em: <https://doi.org/10.1016/j.ejor.2018.07.019>. Citado na página 16.

DARWIN, C. *The origin of species by means of natural selection*. [S.l.]: Pub One Info, 1859. Citado na página 23.

ERNST, A. T.; OğUZ, C.; SINGH, G.; TAHERKHANI, G. Mathematical models for the berth allocation problem in dry bulk terminals. *Journal of Scheduling*, v. 20, n. 5, p. 459–473, 2017. ISSN 10946136. Citado na página 16.

FILHO, G. R.; NAGANO, M. S.; LORENA, L. A. N. Evolutionary clustering search for flowtime minimization in permutation flow shop. In: BARTZ-BEIELSTEIN, T.; AGUILERA, M. J. B.; BLUM, C.; NAUJOKS, B.; ROLI, A.; RUDOLPH, G.; SAMPELS, M. (Ed.). *Hybrid Metaheuristics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 69–81. ISBN 978-3-540-75514-2. Citado na página 34.

KITA, E. *Evolutionary algorithms*. [S.l.]: InTech, 2011. Citado 2 vezes nas páginas 8 and 23.

KOVAČ, N. Metaheuristic approaches for the berth allocation problem. *Yugoslav Journal of Operations Research*, v. 27, n. 3, p. 265–289, 2017. Citado 3 vezes nas páginas 9, 13, and 14.

KOVAČ, N.; DAVIDOVIĆ, T.; STANIMIROVIĆ, Z. Variable neighborhood search methods for the dynamic minimum cost hybrid berth allocation problem. *Information Technology and Control*, v. 47, n. 3, p. 471–488, 2018. Citado na página 18.

LIM, A. The berth planning problem. *Operations research letters*, Elsevier, v. 22, n. 2-3, p. 105–110, 1998. Citado na página 13.

LIN, S. W.; TING, C. J. Solving the dynamic berth allocation problem by simulated annealing. *Engineering Optimization*, v. 46, n. 3, p. 308–327, 2014. ISSN 0305215X. Citado na página 18.

LIN, S.-W.; YING, K.-C.; WAN, S.-Y. Minimizing the total service time of discrete dynamic berth allocation problem by an iterated greedy heuristic. *The Scientific World Journal*, Hindawi, v. 2014, 2014. Citado na página 18.

LIU, M.; LIU, R.; CHU, F.; CHU, C. Mathematical model and solution approach for berth allocation problem in tidal bulk ports with different vessel draft requirements. In: IEEE. *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*. [S.l.], 2018. p. 1–6. Citado na página 16.

MANEENGAM, A.; UDOMSAKDIGOOL, A. 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*, Multidisciplinary Digital Publishing Institute, v. 11, n. 11, p. 4840, 2021. Citado na página 17.

MENEZES, G. C.; MATEUS, G. R.; RAVETTI, M. G. A branch and price algorithm to solve the integrated production planning and scheduling in bulk ports. *European Journal of Operational Research*, Elsevier, v. 258, n. 3, p. 926–937, 2017. Citado na página 17.

OLIVEIRA, A. C.; LORENA, L. A. Detecting promising areas by evolutionary clustering search. In: SPRINGER. *Brazilian Symposium on Artificial Intelligence*. [S.l.], 2004. p. 385–394. Citado 2 vezes nas páginas 15 and 25.

OLIVEIRA, A. C. M.; LORENA, L. A. N. Hybrid evolutionary algorithms and clustering search. In: ______. *Hybrid Evolutionary Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 77–99. ISBN 978-3-540-73297-6. Disponível em: <https://doi.org/10.1007/978-3-540-73297-6_4>. Citado 2 vezes nas páginas 15 and 34.

OLIVEIRA, A. C. M. d.; CHAVES, A. A.; LORENA, L. A. N. Clustering search. *Pesquisa operacional*, SciELO Brasil, v. 33, n. 1, p. 105–121, 2013. Citado na página 34.

PAGÈS, J. *Multiple factor analysis by example using R*. [S.l.]: CRC Press, 2014. Citado 3 vezes nas páginas 8, 21, and 22.

RESENDE, M. G. C.; RIBEIRO, C. C. *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2016. ISBN 149396528X. Citado na página 15.

RUIZ, E. L.; IZQUIERDO, C. E.; BATISTA, B. M.; MORENO-VEGA, J. M. A metaheuristic approach for the seaside operations in maritime container terminals. In: SPRINGER. *International Work-Conference on Artificial Neural Networks*. [S.l.], 2013. p. 28–35. Citado na página 18.

STEFANELLO, F. et al. Hibridização de métodos exatos e heurísticos para resolução de problemas de otimização combinatoria. Universidade Federal de Santa Maria, 2011. Citado na página 14.

TALBI, E.-G. *Metaheuristics: from design to implementation*. [S.l.]: John Wiley & Sons, 2009. v. 74. Citado 6 vezes nas páginas 9, 14, 21, 22, 23, and 24.

YAN, S.; LU, C. C.; HSIEH, J. H.; LIN, H. C. A Dynamic and Flexible Berth Allocation Model with Stochastic Vessel Arrival Times. *Networks and Spatial Economics*, Networks and Spatial Economics, p. 903–927, 2019. ISSN 15729427. Citado na página 16.