

Optimizing the efficiency of port berth utilization under uncertainty: an application to the oil supply chain

Raphael Domingues de Lacerda Bittencourt, Silvio Hamacher, Davi Michel Valladão

Departamento de Engenharia Industrial, PUC-Rio

Rua Marquês de São Vicente, 225 – Gávea, Rio de Janeiro (RJ), Brasil

raphadombit@gmail.com, hamacher@puc-rio.br, davimv@puc-rio.br

ABSTRACT

Pipe Laying Support Vessels (PLSVs) are crucial resources in subsea interconnection operations, responsible for linking offshore wells to production platforms. To carry out these operations, the vessels must be loaded with various materials at designated berths located in different ports. This study proposes an exact solution method to support the PLSV loading plan for a Brazilian oil and gas company, aiming to minimize the maximum makespan across all berths and improve port efficiency, while accounting for operational uncertainties. The problem is modeled as a parallel machine problem, where the first stage assigns each load to a berth, and the second stage determines the planned start time for each load based on different scenarios. A data-driven instance built from real operational data demonstrates that increasing the number of scenarios enhances the robustness of the solution, reducing the likelihood of unfavorable outcomes in out-of-sample evaluations.

KEYWORDS. Pipe Laying Support Vessel. Optimization. Parallel Machine Problem.

EN&OG - OR in Energy, Oil and Gas, L&T - Logistics and Transport

1. Introduction

The oil and gas industry plays an important role in the world economy. The demand for oil is expected to continue to grow until at least 2040 [Lu et al., 2019]. In Brazil, a substantial proportion of oil production is derived from the pre-salt layers, situated at depths exceeding 2.000 meters. These deposits are characterized by a considerable degree of complexity and risk [Beltrao et al., 2009]. In the pre-salt, the oil is transported via an underwater pipeline network that extends from the well, situated on the seabed, to the production platforms.

The vessel responsible for carrying out the subsea interconnection between the wells and the production platforms is known as the Pipe Laying Support Vessel (PLSV). The PLSV is responsible for loading the subsea lines in the port, transporting them to the well site, connecting them to the oil producing platform, and thus allowing the production to begin [Abu-Marrul et al., 2020].

In order to perform subsea interconnection tasks, PLSVs must utilize a variety of materials, including rigid pipelines, flexible pipelines, or a combination of both [Ferreira, 2013]. To load these materials, the ships must proceed to a designated destination port and dock at a specific berth that is chosen in advance by the company. The loading stage of subsea lines represents a crucial stage in the oil production chain and has complex logistics.

The motivation for this project can be traced back to the planning of loadings of PLSVs by a Brazilian oil and gas company. Currently, the area responsible for managing the schedule does not utilize computational tools and does not consider the inherent uncertainties of the problem. The uncertainty in this problem lies in the duration of the subsea operations. Due to factors such as bad weather or technical issues, the PLSV may take longer than expected to complete these operations, potentially arriving at the port on an unplanned day and thus impacting port logistics. Given the critical nature of this stage in the oil and gas supply chain, it is essential to invest a significant amount of time in the planning process, which is currently conducted manually and therefore susceptible to errors. Consequently, in many instances, the loads are not distributed equitably among the company's berths, leading to prolonged periods of idleness at certain berths while others experience high utilization, resulting in inefficiencies in operational performance.

The main objective of this work is to propose a solution method to be used in the generation of a PLSV loading plan, seeking to minimize the largest makespan considering all berths and acknowledging the inherent uncertainty in the operation. The problem described can be modeled as a parallel machine problem in which we seek to program a set of jobs J on a set of machine K , such that each job is performed only once. Since the problem involves uncertainties, the developed model is classified as a two-stage model. The first stage indicates the machine k where the job j will be performed. The second stage of the model indicates the time t at which the job j begins in the scenario s .

The outline of this work is structured as follows. Section 2 describes the real problem that was used as a basis for the development of the project. Section 3 presents the methodology used for the development of this work, detailing the developed model to solve the problem. Computational experiments are presented and discussed in Section 4. Finally, Section 5 presents the final considerations.

2. Problem Description

The Pipe Laying Support Vessel (PLSV) is a maritime vessel with the capacity to perform a range of activities in support of subsea infrastructure, such as interconnecting subsea lines, pipeline maintenance, supporting submarines, and supplying materials to oil platforms, among others. Among these, one of the most important activities that this vessel performs is the connection of subsea oil wells to production units, which is the last step for the well to be operational [Mecler, 2020]. Thus, we can consider the PLSV a critical resource in offshore oil exploration [Abu-Marrul, 2017].

To perform subsea interconnection tasks, the PLSV needs to use several types of materials, of different sizes and functions. The main materials are: rigid pipelines (parts of subsea lines developed to transport oil and/or gas), umbilicals (subsea line composed of a set of hoses and electrical cables, used to remotely operate production and injection equipment and valves), floats (equipment equivalent to a buoy responsible for minimizing the vertical tension of the subsea lines at the platform connections) and coils (where the subsea lines are stored) [Ferreira, 2013]. The average duration of a given load is 65 hours, with the specific duration dependent on the quantity and origin of the materials to be loaded. These materials are stored in various onshore ports.

The length of a single pipeline or umbilical can reach 3.000 meters, and these can be stored in coils to be loaded by the PLSV. The average weight of an empty coil is 40 tons, while a full coil weighs approximately 240 tons. The considerable size and weight of the materials require complex logistics to be implemented to facilitate their transportation between various locations within the port and between the port and the ship. This process involves the use of various types of vehicles, personnel, and machinery. Consequently, the berth at which the PLSV will dock is selected in advance and remains unalterable due to the logistical impracticability of relocating the materials to alternative locations.

Berths are the places where the vessels will dock to carry out the loading and/or unloading operation. Because the number of berths is limited, if a PLSV arrives at a port and the berth where the loading is scheduled to take place is occupied, it must wait for the berth to become available in order to dock. Furthermore, it is interesting to note that, after the loading operation is completed, the PLSVs can move to another port or to the ocean to perform different subsea interconnection operations.

The duration of the subsea interconnection operations is influenced by the complexity of the operation, and only after they are completed does the PLSV sail back to a port. It is notable that, despite an expected duration, the actual time required to complete the submarine operation presents an element of uncertainty. The duration of the operation is influenced by two key factors: weather and logistics. In the event of inclement weather, the PLSV must await a favorable change in conditions before proceeding with the operation. Additionally, operational complications may arise unexpectedly, extending the interconnection duration beyond the anticipated timeframe. Furthermore, the operation may necessitate the assistance of other vessels, which may not always be available at the designated time, further contributing to the overall duration.

The PLSV loading schedule is prepared by a specialist and focuses on short-term operations over a three-month period. It plays a key role in coordinating different parts of the process. Port employees rely on it to organize and position materials correctly. PLSV crews use it to identify their docking berths and confirm what needs to be loaded. Manufacturers check it for any pending items in production. Managers use the schedule to track progress and determine if any actions are required.

As can be seen, the PLSV loading schedule is used by several areas; however, no computer tools are used to support decision-making, and different uncertainties related to the problem are not considered, which can result in the generation of a less than robust schedule. As a result, in some situations the schedule drawn up by the specialist may not be the best, resulting in extended periods during which some berths are unoccupied while others are at maximum capacity. In this context, it can be observed that the utilization of berths is ineffective, which ultimately impairs the company's operational efficacy. Accordingly, the proposed solution is designed to assist the specialist in developing quality solutions that are in accordance with the constraints of the problem, including:

- Each load must be performed only once;

- Once started, the load cannot be interrupted;
- It is not feasible for a berth to simultaneously accommodate two distinct loads;
- The duration of each load is known in advance and static.

3. Methodology

This work was developed in a series of stages, with the initial stage encompassing data collection and processing. The data, which was collected from a single database, will be used to generate parameters for the optimization of the mathematical model. However, it is crucial to emphasize the importance of processing the data to ensure its effective use in the optimization process and to accurately portray the complexity of the problem. Initially, the units of measurement for each parameter are verified and adjusted, if necessary, to ensure compatibility with the proposed formulation. Then, an analysis is conducted to correct any potential errors in the data.

Subsequently, an analysis of the data is conducted with the objective of generating values for the various parameters that will be utilized in the mathematical model. Given that the problem is rooted in a real-world case, it is worthwhile to utilize authentic data to produce synthetic values. In this stage, the parameter that was considered as uncertainty for the problem and how it would be generated were defined.

The following phase involves the mathematical model, which incorporates distinct parameters obtained from the synthetically generated data. Upon completion of the optimization process, the model produces a loading schedule, which specifies the date and berth allocated for a particular load. This schedule is designed to minimize the maximum makespan.

Sensitivity analyses are conducted to assess the impact of various factors on the outcomes. Initially, the number of scenarios is varied to verify the model's behavior and its performance. Afterward, an out-of-sample analysis is conducted for each number of scenarios tested, employing a range of scenario values. This approach is employed to verify the behavior of the result obtained and assess the robustness of the schedule found.

3.1. Data Analyses

The company in this study provided a historical record of PLSV loads, from which it was possible to obtain values for various parameters that can be used to generate instances. These included the duration of the loads and the setup time. While the first parameter was obtained directly from the extensive dataset, a series of calculations was required to determine the setup time. The start date, duration of each load, and the PLSV responsible for the operation were mapped. For each vessel, the loads were sorted in ascending order by their start dates. The setup time was then calculated as the difference between the end date of one load and the start date of the subsequent load.

An examination of the outliers for each parameter is conducted to prevent interference with the values of the generated instances and to achieve a closer approximation to the actual operation. For each parameter, data with discrepant values were subjected to manual analysis on a case-by-case basis. To enhance the interpretability of the data, an exploratory analysis was conducted to verify the data's behavior after removing outliers.

Measure	Loading	Setup
Quantity	275	275
Duration (Minimum) (hours)	14	24
Duration (Maximum) (hours)	210	1446
Duration (Average) (hours)	65.73	396.49
Duration (Median) (hours)	54	340

Table 1: Exploratory data analysis

As evidenced in Table 1, different characteristics of the problem in question have been validated. The duration of the loading process varies in accordance with the specific type of material being loaded. The loading of smaller materials tends to be completed relatively rapidly, whereas the loading of larger materials, such as coils and pipelines, typically requires a longer time to finish.

The wide variation in the duration of the setup is related to the type of operations the PLSV will carry out after loading has finished. In some situations, the PLSV carries out simple operations and is then directed to a port to load more materials. However, if the ship carries out a sequence of subsea interconnection operations, the time until the next loading increases, as these operations are large and complex.

In order to ascertain which distribution is the most appropriate for each variable, the log-likelihood was calculated. This estimate aims to identify the distribution and parameters that most closely resemble the observed behavior of the data [Myung, 2003]. Consequently, a higher value of the estimate indicates a better fit. Table 2 presents the results of this analysis. The first column lists the distributions that were tested, while the subsequent columns show the value of the estimate for each parameter. The value in bold represents the optimal result for that parameter.

Distribution	Loading	Setup
Discreet Uniform	-1447,60	-1982,12
Exponential	-1420,87	-1906,26
Log-Normal	-1347,99	-1920,11
Normal	-1397,75	-1953,50
Gamma	-1352,69	-1899,10
Geometric	-1422,95	-1906,61
Laplace	-1404,38	-1960,29
Pareto	-1458,59	-2032,00
Rayleigh	-1364,77	-1959,74
Inverse Gaussian	-1344,91	-1937,23
Uniform	-1446,20	-1981,93
Weibull	-1361,09	-1897,54

Table 2: Result of fitting the data to a probability distribution

The load duration data were best modeled using the Inverse Gaussian distribution, with parameters $\mu = 65.737$ and $\lambda = 155.598$, which was then used to estimate load durations. The setup duration, in turn, was best represented by the Weibull distribution, with parameters $\alpha = 1.236$ and $\theta = 424.022$, and was subsequently used to generate random setup times across different scenarios.

As the mathematical model developed uses the time-indexed formulation, it is necessary for the time periods to be whole numbers. It is therefore necessary to carry out data processing, where the random number generated is approximated to the nearest integer.

In the present study, a single instance was generated from the generator. The values selected for the deterministic parameters of the problem were based on actual operation and were as follows: 60 loads, 4 machines and 15 vessels. Although the data set is expressed in hours, an approximation to days was made in order to reduce the complexity of the model and to align it more closely with the actual phenomenon under study.

To clarify the concepts of setup time and precedence that will be used in the mathematical formulation, it is important to understand the structure of a voyage for a given PLSV. A voyage consists of loading different materials at the port, sailing to the location where subsea interconnection activities will be carried out, and finally returning to the port to load more materials and start a new voyage. It can be concluded that there is a precedence relationship between two loadings, as a subsequent loading can only occur after the vessel completes operations using the materials that were loaded previously. For the problem studied in this work, the sailing times and the duration of the activities are grouped together and considered as the setup time.

3.2. Mathematical Model

The problem described can be modeled as a parallel machine problem. This class of problem seeks to program a set of tasks J on a machine k , so that each task is performed only once [Mokotoff, 2001]. This is a class of problem used for decision-making in several industries, and it is possible to use different techniques to solve the problem in question [Pinedo, 2008].

The time-index formulation was chosen to model the problem because it is easy to interpret the results for real-world applications [Boland et al., 2016]. It is a solution method capable of achieving the optimal result in combinatorial problems based on the discretization of the time horizon T . In this way, we can define that this formulation seeks to program a set of jobs $J \in J$ (representing ship loads) on a set of machines $k \in K$ (representing berths) so that the time horizon $t \in T$ is respected.

To ensure that the start date of loading j respects the release date (r_j), an auxiliary parameter $ExistY_{j,k,t,s} \in \{0, 1\}$ was created, where $j \in J, k \in K, t \in T$ and $s \in S$. This parameter has a value of 1 when loading j can be started at time t in berth k , in scenario s if t is greater than r_j . The parameter $ExistY_{j,k,t,s}$ was used to create the decision variable $Y_{j,k,t,s}$. If the parameter is 1, the variable will be created. In this way, as well as being useful for guaranteeing that no loading will start before the release date, the parameter $ExistY_{j,k,t,s}$ is used to limit the creation of decision variables, resulting in a more efficient model.

The model developed is regarded as a two-stage model. The first stage indicates the machine k where the job j will be performed. The second stage of the model indicates the time t at which the job j begins in the scenario s . The uncertainty in the problem is reflected in the setup time ($s_{j,s}$), which denotes the duration of the subsea interconnection operations. With an understanding of the characteristics of the problem, the mathematical formulation developed to solve the problem will be detailed.

Sets:

J : set of jobs; $j \in J$

K : set of machines; $k \in K$

T : set of periods; $t \in T$

S : set of scenarios; $s \in S$

Parameters:

$r_j \in T$: release date (anchor date) of job j

$p_j \in T$: processing time of job j
 $s_{j,s} \in T$: setup time of job j in the scenario s
 $p1_j \in \{0, 1\}$: if the job j has precedence
 $p2_{j,j_2} \in \{0, 1\}$: if job j precedes job j_2

Decision variables:

$X_{j,k} \in \{0, 1\}$: indicates the machine k where the job j will be performed
 $Y_{j,k,t,s} \in \{0, 1\}$: indicates whether job j is allocated to berth k at time t in scenario s
 $Z_{j,s} \in T$: indicates the time at which job j can be started in scenario s
 θ : indicates the longest completion time

Thus, consider the following proposed model:

$$\min \quad \theta \quad (1)$$

$$\text{s.t. } \theta \geq Y_{j,k,t,s} \cdot (t + p_j), \quad \forall j \in J, \forall k \in K, \forall t \in T, \forall s \in S \quad (2)$$

$$\sum_{k \in K} X_{j,k} = 1, \quad \forall j \in J \quad (3)$$

$$Y_{j,k,t,s} \leq X_{j,k}, \quad \forall j \in J, \forall k \in K, \forall t \in T, \forall s \in S \quad (4)$$

$$\sum_{t \in T} \sum_{k \in K} Y_{j,k,t,s} = 1, \quad \forall j \in J, \forall s \in S \quad (5)$$

$$\sum_{j \in J} \sum_{t' \geq t - p_j + 1}^t Y_{j,k,t',s} \leq 1, \quad \forall k \in K, \forall t \in T, \forall s \in S \quad (6)$$

$$Z_{j_2,s} \geq \sum_{k \in K} \sum_{t \in T} Y_{j,k,t,s} \cdot (t + p_j) + s_{j_2,s}, \quad \forall j \in J, \forall j_2 \in J, \forall s \in S | p2_{j,j_2} = 1 \quad (7)$$

$$Z_{j,s} \geq r_j, \quad \forall j \in J, \forall s \in S \quad (8)$$

$$\sum_{k \in K} \sum_{t \in T} Y_{j,k,t,s} \cdot t \geq r_j \cdot (1 - p1_j) + Z_{j,s} \cdot p1_j, \quad \forall j \in J, \forall s \in S \quad (9)$$

$$X_{j,k} \in \{0, 1\}, \quad \forall j \in J, \forall k \in K \quad (10)$$

$$Y_{j,k,t,s} \in \{0, 1\}, \quad \forall j \in J, \forall k \in K, \forall t \in T, \forall s \in S | \text{Exist } Y_{j,k,t,s} = 1 \quad (11)$$

$$Z_{j,s} \geq 0, \quad \forall j \in J, \forall s \in S \quad (12)$$

The Objective Function (1) and Constraint (2) seek to minimize the time of the largest makespan of all the machines. Constraint (3) defines that each job will be allocated to only one machine, while Constraint (4) defines that the job can only be started on the machine chosen by Constraint (3). Constraint (5) ensures that each job will be allocated once in the schedule. Constraint

(6) ensures that there is no overlapping of jobs in the same machine. Constraints (7), (8) and (9) define when the job can begin in accordance with the established precedence. Constraint (7) defines that a job j_2 can only be started after the preceding job j has finished and its setup time has elapsed. Constraint (8) guarantees that the variable $Z_{j,s}$ must be greater than the release date of the job j . Constraint (9) guarantees that the job j can only start if the start date is greater than the release date (if the job has no precedence) or greater than the variable $Z_{j,s}$ (if the job has precedence). Constraints (10), (11) and (12) define the domain of the variables, emphasizing that the variable $Y_{j,k,t,s}$ is only created if $ExistY_{j,k,t,s} = 1$.

4. Computational Experiments

In this section, the results obtained by the developed mathematical model will be presented. The experiments were performed on a machine with an i7-1200K @ 3.6GHz and 64GB of RAM. All work was performed using the Julia language version 1.9.0 and solver Gurobi version 10.0.3. It was established that the number of scenarios that will be used for the experiments was $S = \{5, 10, 15, 20\}$. To avoid possible bias in the result due to the randomness of the data, 20 samples were generated for each scenario value. For a better understanding of the results obtained, a summary table was created with the main statistics considering different parameters. Table 3 presents the number of variables and constraints of the model, the average value of the objective function found, and the average optimization time to achieve such a result.

Parameter	5 scenarios	10 scenarios	15 scenarios	20 scenarios
# Variables	548301	1096361	1644421	2192481
# Constraints	1106705	2213350	3319995	4426640
Avg Objective Function (days)	161	163	172	174
Avg Optimization Time (s)	44,23	144,92	320,92	1143,80

Table 3: Summary of Results

As can be seen, the instances created are of different sizes due to the large variation found in the number of model variables and constraints. It can be observed that increasing the number of scenarios leads to a deterioration in the objective value of the model. This is a logical consequence of the fact that an increase in the number of scenarios will inevitably lead to an increase in the probability of generating unfavorable scenarios, thereby enhancing the robustness of the result. It is apparent that there was a considerable disparity in the optimization time. While the average time for 5 scenarios was 44,23 seconds, the average time for 20 scenarios was 1143,80 seconds.

This shows that there is a trade-off between the robustness of the model and the time needed to obtain such a result. However, it is interesting to perform a more detailed analysis of the results obtained to verify the impact of the variation of the scenarios on the value of the function and on the optimization time.

4.1. Effect of varying the number of scenarios on the value of the problem's objective function

The first analysis investigates the impact that varying the number of scenarios can have on the value of the algorithm's objective function. Figure 1 presents a box plot, where the x axis represents the number of scenarios and the y axis the value of the algorithm's objective function.

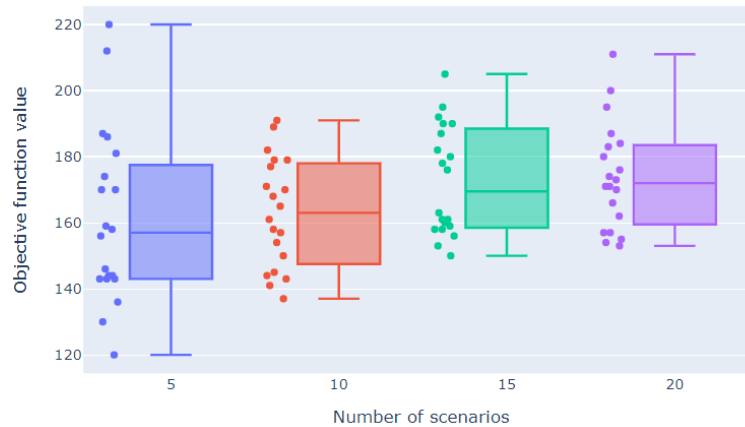


Figure 1: Effect of varying the number of scenarios on the value of the problem's objective function

It can be observed that the variation of the number of scenarios has an influence on the value of the objective function. The samples of 5 scenarios present a large dispersion of the values of the objective function, where it is possible to verify that the extreme values (smallest and largest value for the objective function) considered in all the data belong to this group. As the number of scenarios increases, the dispersion of the values of the objective function decreases, and the variability between the samples is reduced.

4.2. Effect of varying the number of scenarios on the algorithm's optimization time

The second analysis seeks to verify the impact that the variation in the number of scenarios can have on the execution time of the algorithm. In order not to affect the visualization of the analysis, 3 samples of 20 scenarios that had a high optimization time were removed from the figure. Figure 2 shows a box plot where the x-axis represents the number of scenarios and the y-axis represents the total optimization time of the algorithm.

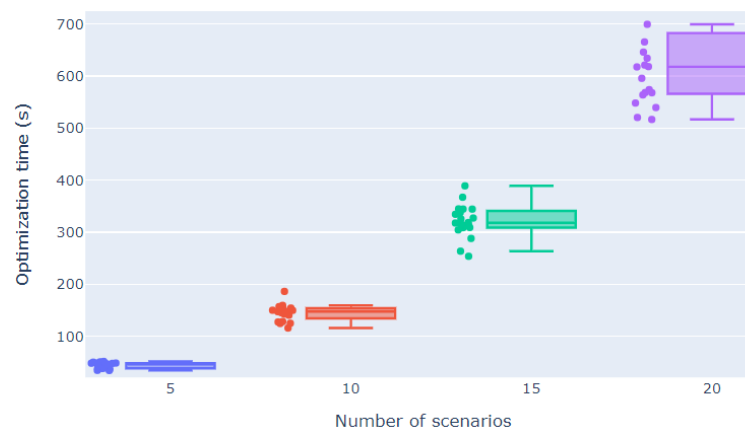


Figure 2: Effect of varying the number of scenarios on the algorithm's optimization time

It can be observed that the variation of the number of scenarios has an influence on the total execution time of the algorithm. The samples with 5 scenarios have a low computational cost of about 45 seconds. The samples with 10 scenarios need about 145 seconds to reach the optimal result, while the samples with 20 scenarios show a significant increase, with an average time of 1145 seconds. This result is consistent since the increase in the number of scenarios considered is

directly related to the size of the model. In other words, as the number of scenarios increases, so too does the size of the model. Additionally, it is evident that the optimization time exhibits a greater degree of dispersion between samples with an increase in the number of scenarios.

4.3. Probability of a worse outcome

An interesting topic for analysis would be the behavior of the results obtained from the model when out-of-sample scenarios are taken into account. The calculation is performed as follows: for each sample and each scenario, the result of the first stage is stored, that is, which machine will perform the job. With this information, the solution is tested in S out-of-sample scenarios to determine whether the value of the objective function is worse than that of the in-sample analysis. For example, if there are 100 out-of-sample scenarios and the result is worse in 10 of them, the probability of a worse result is 10%.

This is an interesting analysis that tries to analyze the robustness of the result found. For this work, it was defined that the values of the out-of-sample scenarios to be tested are $S = \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$. Figure 3 shows the consolidated result of this analysis, where it is possible to verify the behavior of the probability of a worse outcome for different in-sample and out-of-sample scenarios.

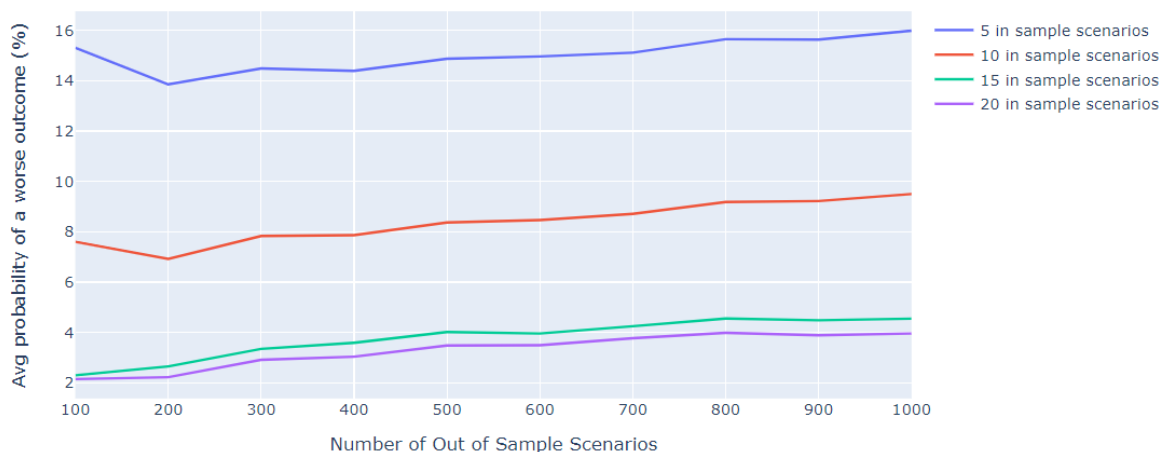


Figure 3: Probability of a worse outcome

As illustrated in the Figure 3, a slight yet notable upward trend in the probability of an unfavorable outcome is observed with an increase in the number of scenarios. In other words, the greater the number of out-of-sample scenarios considered, the greater the probability that the resulting value will be worse than that obtained from the in-sample scenarios. Moreover, as illustrated in the Figure 3 and Table 4, the increase in in-sample scenarios enhances the robustness of the result. This is evidenced by the inverse proportionality between the probability of a worse outcome and the number of in-sample scenarios across all out-of-sample scenarios tested.

It is also noteworthy that the discrepancy between the results obtained with 15 and 20 scenarios is relatively minor. This suggests that there is no significant disadvantage in utilizing 15 scenarios instead of 20. This is an intriguing observation, particularly when considering the model execution time, which is approximately 5 and 20 minutes, respectively. Consequently, if time is a critical factor and a rapid result is desired, 15 scenarios can be a viable option. Table 4 presents the results in more numerical detail. The first column shows the number of out-of-sample scenarios considered, while the following columns show the probability of a worse outcome (%) for different numbers of in-sample scenarios tested.

Out of Sample	5 scenarios	10 scenarios	15 scenarios	20 scenarios
100	15,30	7,60	2,30	2,15
200	13,85	6,92	2,60	2,22
300	14,48	7,83	3,35	2,91
400	14,38	7,86	3,58	3,04
500	14,87	8,37	4,02	3,48
600	14,95	8,46	3,96	3,49
700	15,10	8,70	4,25	3,77
800	15,64	9,18	4,56	3,99
900	15,63	9,21	4,48	3,89
1000	15,98	9,49	4,55	3,96
Average	15,02	8,37	3,77	3,29

Table 4: Probability of a worse outcome for different in sample and out of sample scenario configurations

4.4. Effect of varying the number of scenarios in sample on the probability of a worse outcome

The fourth analysis examines the influence of varying the number of in-sample scenarios on the probability of a worse outcome. Figure 4 presents a box plot, where the x-axis represents the number of scenarios and the y-axis represents the probability of a worse outcome. The case presented considers 1000 out-of-sample scenarios.

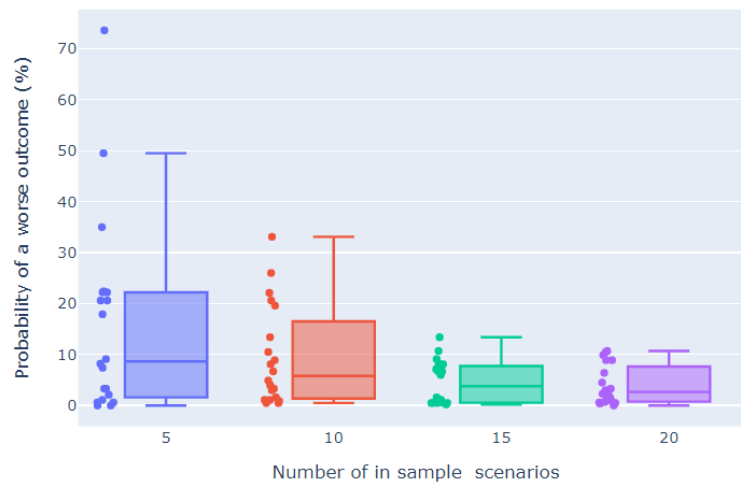


Figure 4: Effect of varying the number of scenarios in sample on the probability of a worse outcome considering 1000 out of sample scenarios

As previously stated in subsection 4.3, it can be verified that an increase in the number of scenarios results in a reduction in the probability of a less favorable outcome. However, the box plot graph demonstrates that the results of the samples for each scenario become more dispersed as the number of scenarios considered is reduced. This result highlights the importance of considering different samples to avoid potential bias in the results. Furthermore, this result indicates that considering 20 scenarios is a more robust approach than considering 5 scenarios because the probability of a worse outcome is lower.

5. Conclusions

The planning of the loading of PLSVs at berths represents a crucial phase within the broader context of the oil production chain. It delineates the specific time and location at which these vessels should berth in order to load the requisite materials for utilization in the subsea interconnection operation. However, despite their importance, the construction of the plan is done manually and does not consider the inherent uncertainties of the problem.

The main objective of this study was to propose a robust solution method to be used in the generation of a PLSV load plan. Initially, the concepts that are fundamental to understanding the problem were presented. Subsequently, the methodology that was employed to represent the actual problem that was studied was presented.

It was demonstrated that the increase in the number of in-sample scenarios enhances the robustness of the result when calculating the probability of a worse outcome, thus providing greater protection against the worst-case scenarios. However, this robustness is often associated with a significant computational cost as the number of scenarios considered increases.

Although an instance generator was developed, this study utilizes a single instance. Consequently, future work may benefit from analyzing instances of varying sizes. Additionally, as illustrated, an increase in the number of scenarios correlates with an increase in the optimization time of the model. Therefore, a second potential objective for future work would be the implementation of heuristic or decomposition method.

References

- Abu-Marrul, V. (2017). Reprogramação de embarcações de apoio à exploração de petróleo através de uma abordagem metaheurística. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- Abu-Marrul, V., Martinelli, R., Hamacher, S., and Gribkovskaia, I. (2020). Matheuristics for a parallel machine scheduling problem with non-anticipatory family setup times: Application in the offshore oil and gas industry. *Computers Operations Research*, 128:105162.
- Beltrao, R. L. C., Sombra, C. L., Lage, A. C., Netto, J. R. F., and Henriques, C. C. D. (2009). Challenges and new technologies for the development of the pre-salt cluster, Santos basin, Brazil.
- Boland, N., Clement, R., and Waterer, H. (2016). A bucket indexed formulation for nonpreemptive single machine scheduling problems. *INFORMS Journal on Computing*, 28:14–30.
- Ferreira, D. (2013). As principais operações das embarcações PLSV.
- Lu, H., Guo, L., Azimi, M., and Huang, K. (2019). Oil and gas 4.0 era: A systematic review and outlook. *Computers in Industry*, 111:68–90.
- Mecler, D. (2020). A metaheuristic for the pipe laying support vessels scheduling problem. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- Mokotoff, E. (2001). Parallel machine scheduling problems: A survey. *Asia-Pacific Journal of Operational Research*, 18.
- Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100.
- Pinedo, M. (2008). *Scheduling: Theory, Algorithms, And Systems*.