

Innovative Cross-Border Risk Management for Hazardous Cargo Transportation: Enhancing Safety and Sustainability in the South Adriatic

Danilo Martino, Emilio Cimino, Nexhat Kapidani, Elizabeta Mrnjačević, Domenico Satalino, Cloridano Bellocchio

Abstract: The maritime transportation between Albania, Italy, and Montenegro is expanding, bringing economic benefits but also increasing the risk of disasters, particularly concerning the transportation of hazardous cargo. The Interreg IPA CBC Italy-Albania-Montenegro 2014-2020 CRISIS Project N.465 addresses these risks within the South Adriatic region and its ports, introducing an innovative Decision Support System (DSS) aimed at enhancing cross-border management of hazardous materials. This system focuses on preventing fatal accidents and promoting environmental sustainability by minimizing pollutant dispersion. It incorporates several factors, including weather forecasts (wind and wave conditions), port infrastructure, ship design, marine protected areas, and traffic flow to optimize Berth Allocation towards safer solutions. The solution developed within the project prioritizes safety by not only defining advanced algorithms but also ensuring that optimal solutions are applied throughout operations. The experience matured during the project execution led us to the formulation of a novel approach for solving Berth Allocation Problem, starting from evidence found in literature and evolving one of the most popular approaches towards the inclusion of sustainability-related parameters.

Keywords: Berth Allocation Algorithm (BAP), Hazardous Cargo Transportation, Risk Management, Cross-Border Cooperation & Innovative approach, Environmental Sustainability.

1. Introduction

1.1. The CRISIS Project

The Cross-border RISK management of hazardous material transportation (CRISIS) project is a European co-funded project involving partners from Italy, Albania and Montenegro. CRISIS aims to investigate these specific risks by analyzing data and evidence from the Italian, Albanian, and Montenegrin territories, along with road transportation in the surrounding regions. Its main objective was the development of an Information and Communication Technology (ICT) Platform to monitor hazardous materials transportation in the Adriatic Sea. This platform integrates two DSS Modules to enhance cross-border management of hazardous materials. The first module is focused on providing an enhanced berth allocation service, creating prioritized schedules based on ship features, transported cargo and weather condition; the second module focuses on open-sea transportation and aims to minimize navigation by including aspects related to Marine Protected Areas (MPA). The ICT platform, therefore, acts as a coordinator and monitoring tool for monitoring the transportation of such materials and support stakeholders in scheduling berth allocation plans. Based on the challenges identified during the project execution, the present article focuses on the development and implementation of an innovative algorithm for the second module with the aim of optimizing port operation under environmental and safety constraints.

1.2. Background and Motivations

Transportation by sea has been a cornerstone of global trade for centuries, facilitating the movement of raw materials, finished products, and commodities across vast distances [1]. As international trade has expanded and evolved, so too have the complexities and challenges associated with maritime transportation and cargo handling. One of the most pressing concerns is the safe and efficient management of ships carrying hazardous or dangerous cargo on board. Hazardous substances include gases (IMO Class 2), flammable liquids (IMO Class 3), toxic substances (IMO Class 6), and radioactive materials (IMO Class 7). These materials are vital to various industries, from manufacturing and energy production to agriculture and healthcare. While their transportation is crucial to sustaining modern economies, this also introduces unique risks that demand meticulous planning and execution.

The primary objectives when transporting dangerous materials via maritime routes are to ensure the safety of human life, protect vessels and cargo, and prevent environmental harm. The potential consequences of

accidents involving hazardous materials are severe, encompassing immediate threats to crew members, long-term health risks, and environmental degradation. With its delicate balance of flora and fauna, the marine ecosystem is particularly vulnerable to contamination from spills or leaks from ships.

Alongside transportation in open seas, an important topic is the management of ports and traffic near shores. This task, with its challenges and complexity, grows in importance when transportation of hazardous materials is involved. Effective berth allocation is crucial in any port management system, but when hazardous materials are in transit, the stakes are significantly higher [2].

Typical operational and service level indicators, such as berth allocation waiting time, are particularly important for port performance intensity of port asset utilization [3]. Usually, waiting times between arrival and the allocation of berths have been decreasing. The world's largest ports, like Antwerp and Hamburg, recorded a reduction in the port-to berth time. However, less positive performances were recorded elsewhere, while in some ports port-to-berth waiting times have increased like in India and some African countries [1].

Traditional algorithms for berth allocation focus on various operational objectives such as minimizing wait times, optimizing resource usage, and maximizing throughput. However, these objectives, although critical, are not sufficient to guarantee environmental and human safety, especially when hazardous materials are involved. Factors such as safety regulations, environmental risks, proximity to protected areas, and the potential for catastrophic events demand a more specialized approach to these problems.

The proposed algorithm evolution is designed to integrate multiple layers of complexity into a cohesive system that aligns with both operational goals and safety protocols. By leveraging a combination of technology and methodologies, the proposed algorithms aim to provide both time-optimized and safety-optimized planning solutions.

The article is structured as follows: Section 2 presents a review of the existing literature on berth allocation algorithms. Section 3 details the theoretical foundation and implementation of the proposed algorithmic modifications. Section 4 reports and discusses the results obtained from laboratory benchmarks, evaluating the effectiveness of the proposed approach.

2. Berth Allocation Algorithm

2.1. Literature Review

The Berth Allocation Problem (BAP) is a critical challenge in the domain of maritime logistics, particularly in the context of container port operations. The primary objective of BAP is to efficiently assign berths to incoming vessels based on various constraints and optimization criteria. This problem involves several considerations such as vessel characteristics, arrival times, berth availability, and compatibility with neighboring vessels. The complexity of BAP arises from the need to balance conflicting objectives, such as maximizing berth utilization while minimizing vessel waiting times and operational costs. In most cases, real-world factors like uncertainties in arrival schedules and dynamic berth availability further contribute to the intricacy of the problem. As a result, finding an optimal solution to the Berth Allocation Problem often requires advanced optimization algorithms and heuristics due to its NP-hard nature, making it a challenging and crucial area of research in maritime logistics.

The study of BAPs features a rich literature of methodologies and algorithms aimed at optimizing the allocation of berths in port terminals. Within this literature, researchers have explored various approaches tailored to different BAP classifications, such as static versus dynamic scenarios or continuous versus discrete layouts. Literature evidence shows that BAP is often solved with methodologies such as exact algorithms, heuristics, and metaheuristics, each offering unique benefits and trade-offs. The choice of methodology often depends on factors such as problem size, required accuracy, and available computational resources, with hybrid approaches often offering the most effective solutions by combining elements of different methodologies.

Among the solutions that have significantly contributed to BAP is the non-linear mixed integer programming model together with the stochastic beam search algorithm, proposed by [4] with the aim to minimize the costs of delay and asset reallocation on the terminal.

Efficient terminal management requires reducing the time of ships spent in the port on the loading/unloading and other services, and therefore, the “Port Collaborative Decision Making (PortCDM)” concept is introduced in [5]. The main contribution of this concept is the intelligent system that will improve port call data sharing and enable high-precision calculations of ships **Estimated Time of Arrival (ETA)** and **Expected Time of Departure (ETD)**, which is of great significance for berthing operations and reducing the ship time in port in waiting queues at anchorage, as well as other bottlenecks related to berthing/unberthing and servicing on the docks.

An interesting approach was proposed in [6], conceptualizing the “Bi-objective Robust Berth Allocation Model (BRBAM)”, which aims to determine a ship berthing program that minimizes operating costs and maximizes customer satisfaction. The focus is on economic performance and customer satisfaction, with the goal of optimizing the robustness of the berth assignment policy. In the field of metaheuristic algorithms, notable efforts have been made in [7] with the use of the Chemical Reaction Optimization (CRO) inspired by the thermodynamics law of chemical reaction and in [8] with the development of a novel evolutionary algorithm with the aim of assisting berthing scheduling at container terminals. In [9], one of the most recent studies in this field, the authors proposed a solution methodology involving the **Cuckoo Search Algorithm (CSA)** to minimize terminal costs, demonstrating its higher effectiveness compared to other metaheuristic algorithms [9,10].

The above-mentioned studies focused on minimizing costs, times or maximizing satisfaction, hence putting particular emphasis on the economic aspects of the BAP. In literature, fewer studies highlighted the sustainability aspects of such dangerous operation, especially when dealing with hazardous cargo or unsafe operational conditions. In [11] the authors studied BAP including tidal constraints. One of the most advanced literature evidence found was [12] where the authors focused their efforts on designing a “Risk Assessment for berthing of hazardous cargo vessels”, however the paper focused primarily on finding the causes of accidents in handling cargo vessels more than solving a BAP problem.

This article focuses on finding an optimal solution to improve with a more complex problem formulation that includes the above-mentioned risk factors in a solution which includes cargo risk levels, wharf structure and weather variables in the algorithmic solution. CSA was selected as the candidate algorithm based on the literature findings, due to its capability of finding optimal solutions in low computational times. Major effort was put into minimizing loading/unloading risks, hence increasing the complexity of the problem itself with respect to the classical formulation.

3. Risk-Aware Berth Allocation Algorithm

3.1. Modified Problem Formulation

This article is primarily focused on the hybrid berthing layout with dynamic vessel arrivals, hence it will be referred to as DH-BAP, which is more complex with respect to the scenario with static arrivals. The choice of hybrid berthing layout is taken due to the need to assign a safety score to pre-determined berthing point in the whole quay which is difficult if applied

with a high level of granularity. Hence, the hybrid layout comes from the division of the dock in a fixed number of berthing points, even if long ships are allowed to occupy more than one, if necessary.

For formulation simplicity, the Maritime Container Terminal (MCT) is considered to possess one berthing layout with known length to accommodate vessels arriving at various time points dynamically. The set of all potential berthing positions on the wharf is denoted as $B = \{1, 2, \dots, M\}$. This simple case is extendable for each berth, even with wharfs with particular berthing configuration, with minimal effort.

Typically, the BAP is tailored to a specific time frame for vessel arrivals, in this specific case a focus is placed on the upcoming 24 hours (next day). This period is hence divided into a set of 30-minute time intervals denoted as $T = \{1, 2, \dots, K\}$. Each interval is accompanied by a weather assessment, detailing both wind and sea conditions expected during that specific time segment.

The set $S = \{1, 2, \dots, N\}$ encompasses all ships scheduled to arrive at the terminal on the following day. Crucial information is available in advance for each ship, including ETA, PBP (Preferred Berthing Point), ship length, estimated (or required) ETD, and an estimate of cargo risk based on the pollution risk posed by the products transported and their potential impact on marine species. In addition, the estimated handling times for each ship were considered known in advance based on previous agreements between the MCT and the incoming ships, such as the number of quay cranes chartered by the ship or the number of containers to be loaded/unloaded during the handling period.

In an ideal scenario (free wharf and mild weather conditions), as soon as a ship arrives it is allocated at the safest spot in the quay, fastly handled and depart, respecting the handling times. If more ships arrive in the same interval, priority must be given to ships with higher cargo risk, reserving them the safest spots in wharf. Other ships are then allocated in less safe spots (if available) or have to wait for a safe spot to be available, based on weather conditions, wharf availability and cargo risk assessment. In the end, in case of severe weather conditions, the algorithm should be able to trade off handling speed and safety by delaying unsafe operations.

Total risk cost for a ship arriving at the MCT is split into three different terms, two of which have the most impact:

Waiting Costs (WC) influenced by the total time a ship has to wait before being served (Waiting Time or WT), the average wave risk assessment for the WT and the ship's cargo risk level. For waiting costs only wave risk is

considered due to waiting areas often exposed to higher marine currents. Equation for waiting costs, expressed as [risk/hour] is the following:

$$WC = W_w * [(CRS_s + 1)^2 * W_{WAS}] * WT_s \quad (1)$$

Where:

- W_w is the waiting weight, expressed as cost per unit time, indicating how waiting is considered high on cost impacts on the overall cost.
- CRS_s is the Cargo Risk Score of ship s.
- W_{WAS} is the average *Waiting WAve Score* for that wharf in the waiting times.
- $WT_s = BT_s - ETA_s$; BT_s is the berthing time for the ship.

Handling Costs (HC) are influenced by the time necessary for a ship to be served once docked (Handling Time, HT), the average wind risk assessment for the whole period in which the ship is served, the ship's cargo risk level and the berthing point safety assessment score. Since wharfs are usually protected from strong marine currents, only wind scores are considered for handling costs, being quay cranes operations riskier under severe wind conditions. Equation for handling costs is the following:

$$HC = H_w * [(CRS_s + 1)^{H_{WIS}/BSS}] * HT_s \quad (2)$$

Where:

- H_w is the handling weight, expressed as cost per unit time, the index on how handling costs impact on total cost.
 - H_{WIS} is the average *Handling WInd Score* for that wharf during the whole handling period of the ship.
 - BSS is the Berth Safety Score, assigned to a berthing point based on its positioning on wharf and its exposure to sea and winds.
- **Late Departure Costs (LDC)** are influenced only by the amount of time a ship exceeds its ETD. This difference is computable as: $LDT = ETD - (ETA + WT + HT)$ and it can assume also negative values, resulting in an incentive towards fast ship handling. The equation is the following one:

$$LDC = LD_w * LDT_s \quad (3)$$

Where LD_w is the late departure weight expressed as cost per unit time indicating how early or late departure impacts on the total cost.

Hence, the overall cost equation for a single ship can be expressed as following, considering a ship s , berthed at time BT_s and in berthing position BP_s , waiting under average wave conditions W_{WAS} and being served under average wind conditions H_{WIS} :

$$Cost(s, BP_s, BT_s, W_{WAS}, H_{WIS}) = WC + HC + LDC \quad (4)$$

The goal of the berth allocation problem is to find the optimal berthing position and times for all ships coming at the planning horizon such as the overall cost is minimized:

$$\begin{aligned} & \text{minimize} \sum_{s \in S} \sum_{b \in B} \sum_{t \in T} x_{sbt} * Cost(s, BP_s, BT_s, W_{WAS}, H_{WIS}) \\ & \text{subject to [9]:} \\ & \quad [C1] \ x_{sbt} \in 0,1 \quad \forall s \in S, b \in B, t \in T \\ & \quad [C2] \sum_{b \in B} \sum_{t \in T} x_{sbt} = 1 \quad \forall s \in S \\ & \quad [C3] \ BT_s \geq ETA_s \quad \forall s \in S \\ & \quad [C4] \ |BT_s - BT_{s'}| \geq SET \quad \forall s, s' \in S \\ & \quad [C5] \ BP_s + L_s \leq W \quad \forall s \in S \\ & \quad [C6] \sum_{s' \neq s \in S} \sum_{b=BP_s-L_{s'}+1}^{BP_s+L_s} \sum_{t=BT_s-HT_{s'}+1}^{BT_s+HT_s} x_{s'bt} = 0 \quad \forall s \in S \end{aligned} \quad (5)$$

- [1] x_{sbt} is a binary variable which takes value 1 if a ship s is assigned to berthing position b at berthing time t , 0 otherwise.
- [2] This constraint ensures that any ship is berthed only once during the planning horizon.
- [3] Third constraint ensures that ships cannot be served before their arrival.
- [4] **Safety Entrance Time (SET)** constraint ensures that two ships cannot be berthed simultaneously. Safety Entrance Time is included in the problem formulation and implementation since most ports welcome one ship at a time due to physical constraints at their entrance.
- [5] Length constraint is applied on the whole wharf, ensuring that all ships are allocated inside the physical dimension of the quay.

- [6] The last constraint ensures that, during planning, two ships cannot even partially overlap in both space and time: two ships cannot coexist in the same berthing point if they share the same handling time slots.

3.2. Chosen algorithm: CSA.

CSA is a powerful nature-inspired optimization technique that derives its inspiration from the unique reproductive behavior of cuckoo birds. The inspiration for CSA comes from the brood parasitism strategy employed by certain species of cuckoo birds. These birds lay their eggs in the nests of other bird species, shifting the responsibility for incubating and caring for their offspring onto unwitting host birds. To survive, the cuckoo chicks must outcompete the host birds' own chicks for food and care. This concept of laying eggs in other birds' nests, combined with the need for cuckoo chicks to thrive in a competitive environment, served as the foundation for the CSA. In optimization terms, the "eggs" represent potential solutions to a problem, while the "nests" are the solution spaces. The objective is to find the best-fit solution by continually improving and replacing eggs in suitable nests.

Introduced in 2009 by Xin-She Yang and Suash Deb, CSA has gained widespread recognition and adoption in the field of optimization. Its appeal lies in its ability to effectively address complex optimization problems [9,13], particularly those characterized by multi-modal and non-linear search spaces. CSA operates as a population-based optimization algorithm. It starts by initializing a population of "nests" or potential solutions to the optimization problem. Each nest represents a potential solution, and the quality of these solutions is evaluated based on an objective function. The algorithm then proceeds through a series of iterations, where cuckoos (representing new potential solutions) are introduced into the population. These cuckoos lay eggs (representing potential solutions) in nests, with the quality of the eggs determined by their fitness. If an egg is of higher quality than the nest it is placed in, it replaces the previous content of that nest. CSA also incorporates mechanisms to maintain diversity in the population. It identifies the "worst" nests and either replaces them with new random nests or abandons them altogether. Simultaneously, the "best" nests are retained to ensure that the algorithm does not lose promising solutions. The process continues for a predefined number of iterations or until a termination condition is met. Throughout these iterations, CSA explores the solution space, gradually improving the quality of solutions, and eventually converging to an optimal or near-optimal solution.

The time complexity of the CSA is a topic of interest, as it influences its practical applicability. CSA's time complexity depends on a range of factors,

including problem size, the choice of parameters, and the complexity of the objective function. In general, CSA exhibits a moderate time complexity, often comparable to other metaheuristic optimization algorithms such as genetic algorithms and particle swarm optimization [9]. The primary computational burden arises from the evaluation of the objective function for each nest (potential solution) and cuckoo (new potential solution). The algorithm's performance can vary significantly based on the problem's characteristics. In cases where the objective function is computationally expensive, CSA may require longer to converge. Additionally, the number of iterations and the size of the population influence the overall runtime. Efforts have been made to enhance CSA's efficiency, such as parallel implementations and hybridization with other optimization techniques. These adaptations aim to reduce the time complexity and accelerate convergence, especially for large-scale and computationally intensive problems.

CSA offers several notable advantages that make it a valuable tool in the realm of optimization:

- **Global Search Capability:** CSA's ability to explore extensive search spaces and locate global optima is one of its primary strengths. It excels in scenarios where the optimization landscape is complex and multi-modal, ensuring that it does not get trapped in local optima.
- **Simple Implementation:** The algorithm's simplicity is a significant advantage. CSA's minimal parameter requirements and straightforward structure make it accessible to both researchers and practitioners. It can be readily implemented and customized to address a wide range of optimization problems.
- **Diversity Maintenance:** CSA incorporates mechanisms for maintaining diversity within the population. By identifying and replacing the worst nests while preserving the best ones, the algorithm strikes a balance between exploration and exploitation. This feature reduces the risk of premature convergence and promotes the discovery of high-quality solutions.
- **Parallelization Potential:** CSA's population-based approach lends itself well to parallelization. This means that it can harness the computational power of modern hardware, making it suitable for addressing computationally intensive optimization problems efficiently.

While CSA offers several advantages, it is essential to consider its limitations:

- **Parameter Sensitivity:** CSA's performance is extremely sensitive to the choice of parameters, including the population size, the termination criteria and parameters related to random generation of new solutions or deletion of less important ones. Tuning these parameters to achieve optimal results can be a non-trivial task and may require extensive experimentation.
- **Limited Scalability:** CSA may encounter challenges when applied to very large-scale optimization problems. The population-based nature of the algorithm implies that it requires maintaining and updating a considerable number of nests, which can be computationally demanding and resource-intensive for massive problem instances.
- **Convergence Rate:** CSA, while effective at global exploration, may exhibit a slower convergence rate compared to some other optimization algorithms for certain problem instances. Achieving convergence to an optimal solution might require more iterations, making it less suitable for time-sensitive applications.

```

1: Objective function  $f(X), X = (f(x_1, x_2, \dots, x_d))^T$ 
2: Generate initial population of  $n$  host nests  $X_i$  ( $i=1, 2, \dots, n$ )
3: While  $t < Max\_iterations$  do
4:   Get a cuckoo randomly by Levy flights
5:   Evaluate its quality/ fitness  $F_i$ 
6:   Choose a nest among  $n$  (say,  $j$ ) randomly
7:   If  $F_i > F_j$  then
8:     replace  $j$  by the new solution;
9:   End If
10:  A fraction ( $Pa$ ) of worse nests are abandoned and
    new ones are built;
11:  Keep the best solutions
12:  Rank the solutions and find the current best
13: End While
14: Postprocess results and visualization

```

Fig. 1 – Example of pseudocode for CSA [14].

CSA proved to be a more effective algorithm compared to Mixed Integer Linear Programming (MILP) or Genetic Algorithms (GAs) in [8], giving both faster responses and converging to better optimal solutions than other metaheuristic algorithms. CSA implements a series of mechanisms to

improve exploration, such as the use of random walks or replacements of a portion of worst nests with the aim to generate new solutions. While random walks (Lévy Flights) help improve the solutions in the neighborhood of previous ones, nest replacement abandons worst solutions to explore new ones in the solution space.

CSA proved its effectiveness in searching acceptable local optima, often near the global ones, even with multi-objective functions, and a highly constrained search space like the one imposed by the berth allocation problem [9].

3.3. Modified CSA: design and implementation details

CSA was implemented using **python 3.10.4** and deployed as an independent and scalable module. It acts as a service on calls, accepting an input and returning the planning. The code was organized in classes, modeling both the inputs and the solution. While implementing it, several aspects were taken into account, such as:

- **Egg and Nest definitions:** it was important to define, pragmatically speaking, the characteristic of an egg, i.e., the shape of the solution. In the algorithm an egg was strictly related to a single ship, meaning that a nest is composed of N eggs, where N is the number of vessels taken into account in an execution. For each ship, both berthing time BT_s and berthing position BP_s were taken in account, as depicted by the cost function defined in Section 3.1. Hence, an egg is represented by a berthing point depending on the wharf and a time slot from those defined in the problem formulation. This adaptation to the specific discrete use case led to an egg structure similar to a hash map, basing the search space on the integer indexes of both the berthing points and the time slots.
- **Constraints definition:** two major types of constraints were identified while developing the solution, namely *egg-domain constraints* and *nest-domain constraints*. The former are related to the placing of a single ship in the wharf, so constraints C1, C2, C3 and C5; constraints C4 and C6, instead, involve more than one ship in a solution. Defining constraint types was useful to control operations while executing the planning algorithm, avoiding unfeasible solutions.
- **Starting conditions:** starting conditions are necessary for every evolutionary algorithm and adopting strategies allows them to converge as soon as possible. In the design of the CSA, it was impossible to set a fixed starting condition due to the variable nature of weather, ship arrivals and port structures. However, to avoid

unfeasibility, the starting population was forced to respect both nest and egg constraints.

- **Evolutionary strategy:** as depicted in [9], using levy flights led to a fast-convergence algorithm. The same strategy was adopted here, further details will be provided later in this chapter.
- **Replacement strategy:** two replacement strategies for CSA were designed for algorithm execution. The first one consisted of simply replacing the worst nests with new randomly generated ones. The second replacement strategy implements a *crossing over* mechanisms where the resulting new nests are bred from two random nests in the whole solution space.
- **Hyperparameters tuning:** once the problem definition was set, one of the most important parts for CSA execution is defining its working mechanisms by setting algorithm hyperparameters. CSA convergence speed is highly influenced by its settings and finding an optimal configuration is often a trial-and-error workflow. The following list is a comprehensive set of hyperparameters already tuned to provide a high convergence speed:
 - **N_nest = 100:** size of the solution space, namely the total number of nests generated as population sample. The higher the number of nests, the higher are the chances to find an optimal solution but also the execution times.
 - **N_iterations = 100:** max number of iterations of the algorithm. The higher the iterations, the higher the execution times but generally the lower the global fitness score reached. To avoid reaching the maximum number of iterations with no improvement, an early stopping mechanism was designed to stop the algorithm if it does not improve overall fitness after 10 iterations.
 - **pa = 0.65:** fraction of worst nests to be deleted. Usually, in this algorithm, this number fixes at 0.25. The higher the fraction, the higher the chances of finding an optimal solution and the execution times. A too high value, however, can lead to convergence problems depending on the strategy used to replace abandoned nests. The value was set so high due to the trade-off between execution times and constraint compliance.
 - **max_tries = 2:** maximum number tries for iteration to avoid generation operations to stuck in endless loops. This could happen if the nest is not able to produce a new solution due to constraints and the number of ships.

- **levy_beta = 1.5, sigma_u = 0.6966, sigma_v = 1, c_multiplier = 1**: set of hyperparameters for the levy flight operations from literature. Noteworthy the *c_multiplier* parameter which decides how much the levy flight step influences the new solution, usually set to a fraction, but being set to 1 in this use-case due to the particular solution structure.

Here, the following pseudo-code to document the most important modifications apported to CSA for solving the DH-SBAP. Operation on eggs were mainly performed on two-element arrays, containing the indexes of berthing points and time slot of the current solution. When the egg indices are modified by the algorithm, the resulting object field for berthing point and time slot is also filled. Each egg has the responsibility to compute its fitness score, based on the BAP environment (weather variables included in the time slots list). Nests' fitness and all constraints, instead, are handled by the berth allocation solver.

Start

Given the following objective function:

$$\min \sum_{s \in S} \sum_{b \in B} \sum_{t \in T} x_{sbt} * Cost(s, BP_s, BT_s, W_{WAS}, H_{WIS})$$

Generate Random Population

While $t < n_{iterations}$ **do**:

 Get Best Nest

for each nest **do**:

 Perform Nest Levy Flight (nest, bestnest)

 Sort Nests by Fitness

for each nest **do**:

 Perform Egg Elimination(nest)

 Reduce current *c_multiplier* by $\frac{1}{100}$ its current value

End

Fig. 2 – CSA, CRISIS version.

```

Start

 $levy_{\beta} = 1.5$ 
 $\sigma_u = 0.6966$ 
 $\sigma_v = 1$ 
Get current nest fitness score
while n < max tries do:
    for each egg in nest do:
        Get the same ship egg in the best nest
        Compute levy flight step toward a better solution using indices:
            u = array of 2 values normally distributed with mean 0 and std  $\sigma_v$ 
             $s = \frac{u}{|v|^{levy_{\beta}}}$ 
            new egg =  $c_{multiplier} * s * (egg - bestegg)$ 
        if new egg respects egg-constraints then:
            Compute new egg fitness score
            if new egg is better than the previous one then:
                Swap eggs
        if new nest respects nest-constraints then:
            return new nest
End
    
```

Fig. 3 – CRISIS CSA's modified levy flight step.

```

Start

Get the  $p_a$  fraction of worst nests
for each nest in worst nest list do:
    while t < max tries do:
        Generate two different random number between 0 and  $n_{nests} - 1$ 
        Pick  $nest_1, nest_2$  in solution space using these two numbers
        for each egg in current selected nest do:
            select  $egg_1, egg_2$  from  $nest_1, nest_2$  using the related egg indices
            new egg =  $egg + N[0, 1] * (egg_1 - egg_2)$ 
            if new egg respects egg-constraints then:
                Compute new egg fitness score
                if new egg is better than the previous one then:
                    Swap new egg with old egg
            if new nest do not respects nest-constraints then repeat attempt
        Swap worst nests with new nests
End
    
```

Fig. 4 – CRISIS Mixing Replacement Strategy. Egg operations are treated as vector elementwise operations.

Start

```
Get the  $p_a$  fraction of worst nests
for each nest in worst nest list do:
  while  $t < \text{max tries}$  do:
    for each egg in current selected nest do:
      Generate random egg
      if new egg respects egg-constraints then:
        Compute new egg fitness score
        if new egg is better than the previous one then:
          Swap new egg with old egg
      if new nest do not respects nest-constraints then repeat attempt
Swap worst nests with new nests
End
```

Fig. 5 – *CSA Replacement Strategy.*

Among the two above-mentioned replacement strategies, the first one was kept, since it resulted in a better convergence rate and slightly lower execution times. Each time a new egg is generated or evolved from other ones, its fitness is evaluated against that of the whole nest: if the nest with the new egg has an overall fitness score lower than the previous one, the new egg is kept. The use of more nests ensures the algorithm to check for different optima in the solution space, trying different combinations.

```

computeFitness(egg, ship)
Start

    Get the ship related to the current egg
    Get all time slots included in waiting time period of the related ship
    Compute mean wave risk score for the selected waiting time slots
    Get all time slots included in handling time period of the related ship
    Compute mean wind risk score for the selected handling time slots
    Compute waiting times, handling times and late departure times in seconds
    Compute waiting costs using the average wave risk
    Compute handling costs based on the average wind risk
    Compute late departure costs
    Sum all costs
End

```

Fig. 6 – *Egg fitness score computing pseudocode, based on the fitness function formula above.*

4. Results and Discussion

The implemented BAP algorithm described in Section 3.1 was evaluated against synthetic data. These synthetic data represent a set of realistic values from weather variables, cargo risk score and berth safety scores, each included in an integer variable in interval [1,10]. The idea behind these tests was to assess the behavior of our approach against simulated environments, evaluating if CSA can perform berth allocation following the cost function. During these experiments, both wind and wave indicators were simulated through a constrained random walk, while CRS scores and BSS scores were randomly sampled within the given set of values. Other variables were randomly sampled from a random distribution, like the ship ETA and ETD, handling time for each ship and ship length.

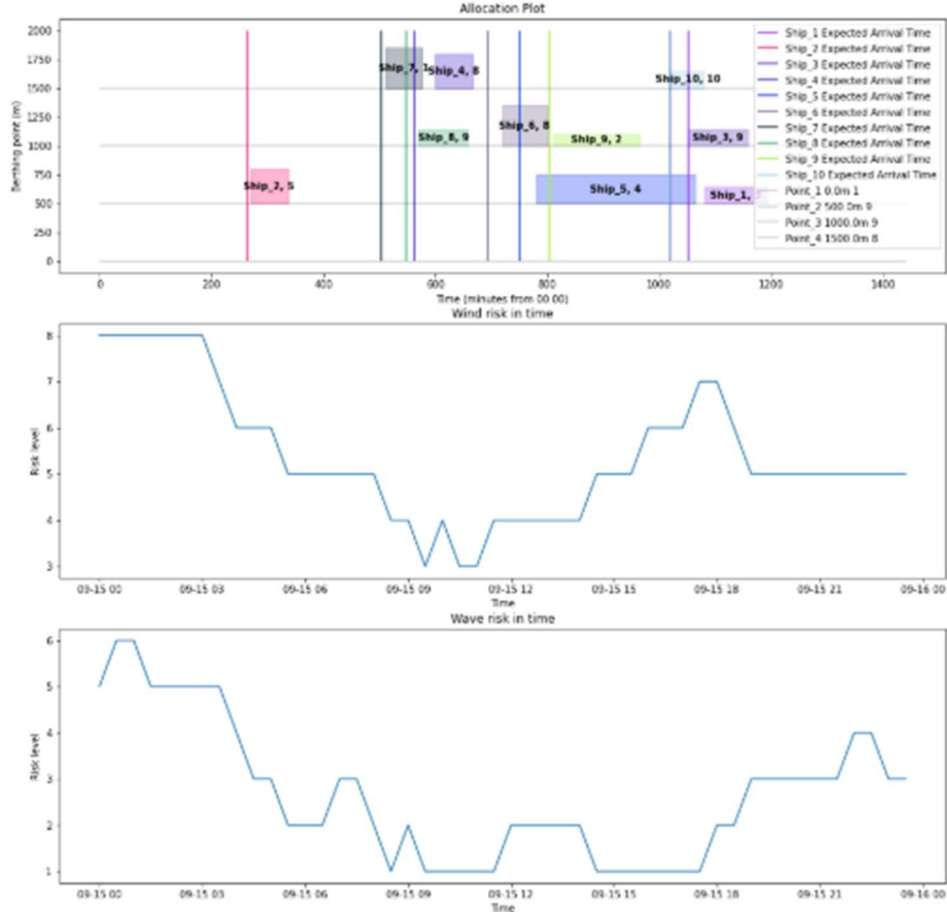


Fig. 7 – Example BAP testing environment. Reading from above to below: **A)** Ship placing plot. **B)** Wind intensity simulation plot **C)** Wave intensity simulation plot.

In Fig. 7 there is a plot depicting the content of the typically used BAP testing environment, comprehending information about ship placement and weather simulation.

The upper graph represents the positioning of the ship in a spatial and temporal graph: the x-axis represents time (in minutes) while the y-axis represents the length of the wharf. The horizontal gray lines starting from the vertical axis represent the berthing points that are the starting point for the ship's position. Berthing points are also reported in plot legend for clarity. Ships are represented as colored boxes, which show the ship's name and cargo risk score. The length of the box (x-axis) represents the ship's handling time, while its height (y-axis) represents the length of the ship. The

colored vertical lines, on the other hand, represent the ETA of each ship, and the color of the lines corresponds to the color of the ship. A ship can only be placed on berthing points, but can occupy more than one if necessary, provided it does not exceed the length of the dock. Moreover, x-axis is divided into timeslots of pre-defined length.

The middle and lower plots, instead, represent the random walk simulation of wind and wave variables, respectively. The x-axis is aligned with the upper plot while the y-axis represents the intensity of the specific weather phenomena.

4.1. Laboratory Benchmark and Sustainability Evaluation

Given the experiment setup, it is necessary to assess if berthing schedules produced by the algorithm effectively contribute to improving the sustainability and safety of operations. The following statements represent the desired behavioral outcomes:

1. In the case of severe sea conditions, ships should be served promptly, giving priority to high-risk cargoes, regardless of the berthing point. This is based on the simplified assumption that docks are protected from sea currents and waves, so loading/unloading operations can be conducted even under the presence of poor outer sea conditions.
2. In heavy winds, high-risk cargo ships have two options:
 - a. If berthing points with high safety scores are available, handling operations are allowed and ships can be served as soon as possible, reserving the highest safety points for the highest risk cargoes.
 - b. If safe mooring points are not available and the wind is expected to decrease in intensity, it is preferable to wait.
3. In the event of both adverse sea and wind conditions, a trade-off must be made. High-risk cargo should be handled promptly if safe docking points are available, otherwise it is preferable to wait until conditions are better.
4. In case of mild weather data, the problem turns into a simple schedule optimization problem.

These conditions are encapsulated in the mathematical formulation proposed in the previous sections. However, by exploiting only the cost function, it is difficult to estimate the effectiveness of the results themselves. Therefore, an evaluation approach based on a sustainability index is formulated.

Sustainability considers different metrics and KPIs based on the case study: in context of logistics and transportation, a recent study [15] identified a set of useful sustainability-related KPIs, assessing both economic, environmental and social dimensions. This article's focus on sustainability is based on preservation of marine ecosystem. For these reasons, the modified CSA is built on a hazard-mitigation cost function, considering several risk factors and penalizing unsafe cargo operations. However, quantifying the risk of pollution based on the BAP-produced schedule is a challenging task to perform without a rich set of field data.

Therefore, a custom comparative evaluation approach was proposed to validate results obtained from these algorithms. The comparison involves comparing two versions for each algorithm with their respective results:

- **Sustainable version:** considers risk factors such as weather assessment, cargo risk assessment, wharf security assessment and marine protected areas, producing a **sustainable-optimal result**. In this context, sustainability is measured by the respective cost functions of each modified version of the algorithm.
- **Base version:** is the base version of each algorithm, explicitly formulated to exclude weather data, wharf security assessment and cargo risk assessment, optimizing only service times. The base version produces a **time-optimal result** or **unsustainable result**.

The following values are computing by crossing algorithms and results obtained:

- **Sustainable Result over Sustainable Cost (SRSC):** The total cost obtained with a normal execution of the sustainable version of the algorithm.
- **Unsustainable Result over Unsustainable Cost (URUC):** The total cost obtained with a normal execution of the base version of the algorithm.
- **Unsustainable Result over Sustainable Cost (URSC):** Total cost obtained by applying the sustainable cost function over the results obtained by the execution of the unsustainable algorithm. It expresses how much the time-optimal solution costs in terms of sustainability.
- **Sustainable Result over Unsustainable Cost (SRUC):** Total cost obtained by applying the cost-optimal function over the results obtained by the execution of the sustainable algorithm. It expresses how much the sustainable-optimal solution costs in terms of efficiency.

In the end, from these three values, three indices are provided:

- **Sustainability Index (SI):** $\frac{URSC}{SRSC}$, where $URSC \geq SRSC$, has a minimum value of 1 and no upper values. It indicates the improvement ratio in terms of sustainability brought by the sustainable solution against the cost-optimal one.
- **Effectiveness Index (EI):** $\frac{URUC}{SRUC} * 100$ where $URUC \leq SRUC$, has a maximum value of 100% and asymptotes towards 0%. It defines the percentage of effectiveness of the sustainable solution against the cost-optimal one.
- **Sustainability to Effectiveness Ratio:** $SI * EI$, asymptotically touches 0 and grows to infinity. It defines how much improvements in terms of sustainability there are against the loss in terms of efficiency.

4.2. Test Results

BAP experiments were conducted in two stages to ensure thorough testing and evaluation of the system's performance. The first stage, conducted in a development environment, aimed to develop a correct algorithm, and visualize how the algorithm reacts to objective function minimization. In this environment, weather variables, wharf structure, and ships were simulated and/or forced to specific values.

The first experiment was conducted with two ships, with the aim of testing scenarios involving simple concurrency. These ships had similar ETA but different cargo risk levels, so they must “compete” for the safest spot. The objective of the experiment was to test the algorithm's capability to allocate ships to the safest spot based on the cargo risk level they exhibit.

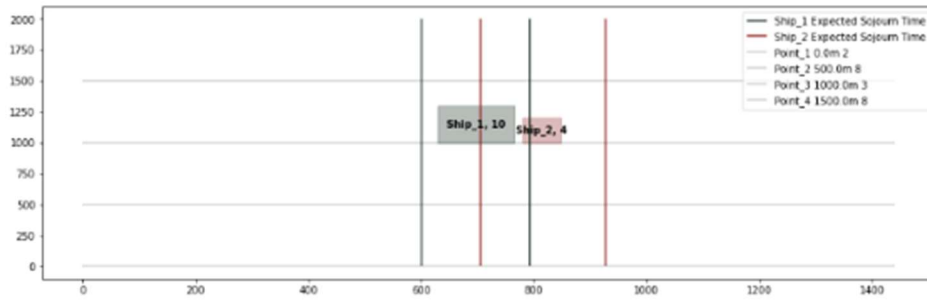


Fig. 8 – Experiment 1 – Initial random placement.

In this experiment, the wave and wind conditions were set to cross sharply. In the first half of the time horizon, the wind score was set high (9-

10) and then decreased to an average of 6 after noon. On the other hand, the waves were set to increase in intensity, starting from a low level and reaching severe conditions at noon (Fig. 9). This experimental setup should test the harsh weather trade-off, expecting ships to be served on safest berths and promptly. Time slots are set to be 30 minutes wide.

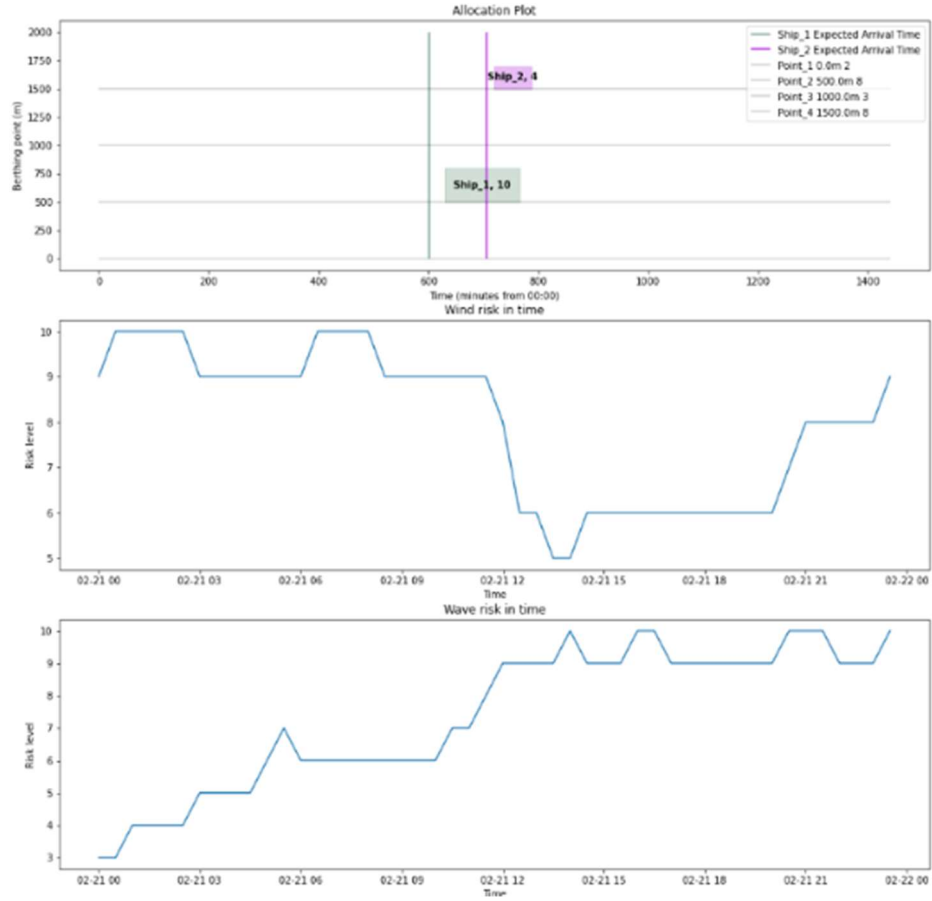


Fig. 9 – Experiment 1 - Berthing schedule and weather variables.

The test produced correct and valid results: since **Ship_1** had the highest risk (**10**), it was allocated to one of the two berthing points with the highest safety score (**Point_2, 8**). Moreover, it waited a time slot (30 minutes) before starting handling operations due to harsh wind conditions and since sea conditions were more permissive in the first half of the day. **Ship_2** with a medium cargo risk (**4**) was immediately allocated to the second safest spot

available (**Point_4, 8**) instead of waiting for the first ship to finish. This was due to the increasing wave risk level, which impacts waiting times and to the lower cargo risk.

The second experiment focused on testing the algorithm in a busy environment and verifying its convergence to an optimal solution. In this experiment time slots are set to 60 minutes to decrease available slots and 8 different ships with different cargo risk level were placed randomly. Moreover, wind conditions were set to be severe, while sea conditions worsened during the time horizon, reaching their maximum intensity at the end of the day.

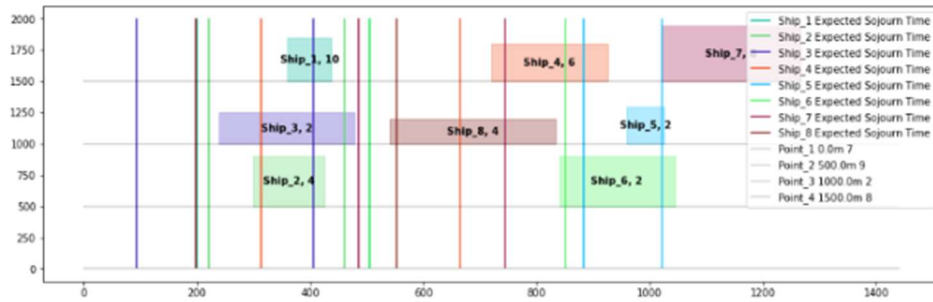


Fig. 10 – Experiment 2 – Initial random placement.

Based on the results shown in Fig. 11, it is possible to observe how the algorithm, starting from a random placement, converged to a better solution. It is also possible to see how ships with lower risk levels usually have more flexible scheduling. This test also showed improvement windows, for example, when there is the possibility to anticipate the allocation of low-risk cargo. For instance:

- **Point_3** which is the least safe with a safety score of **2** is not considered by the algorithm. This is a desirable outcome, since even low-risk cargo are untreatable under severe conditions and need to be handled in safer conditions.
- **Ship_1** has the highest risk score (**10**) was immediately handled in **Point_2** with the highest safety score (**9**).
- **Ship_7** with a risk score of **8** was served at **Point_1** with a berth safety score of **7**. The plot exhibits a delay between the ETA line and the berthing point: this delay is caused by the time slot width of 60 minutes. The ship could be served at other berths by moving other ships, but other configurations are suboptimal. For example, moving this ship to **Point_2** would force both **Ship_6** and **Ship_4** to move to

other berthing points. For example, both could swap points with **Ship_7**, but they would also have to delay their operations by one time slot.

- **Ship_4** with a risk score of **6** was slightly delayed before being served in **Point_2**. This allowed for operations under slightly better wind condition, well supported by berthing point safety score.
- Low-risk ships like **Ship_8** or **Ship_6** had a more flexible scheduling.
- **Ship_5** had no concurrence, so it is immediately served at the safest berth.

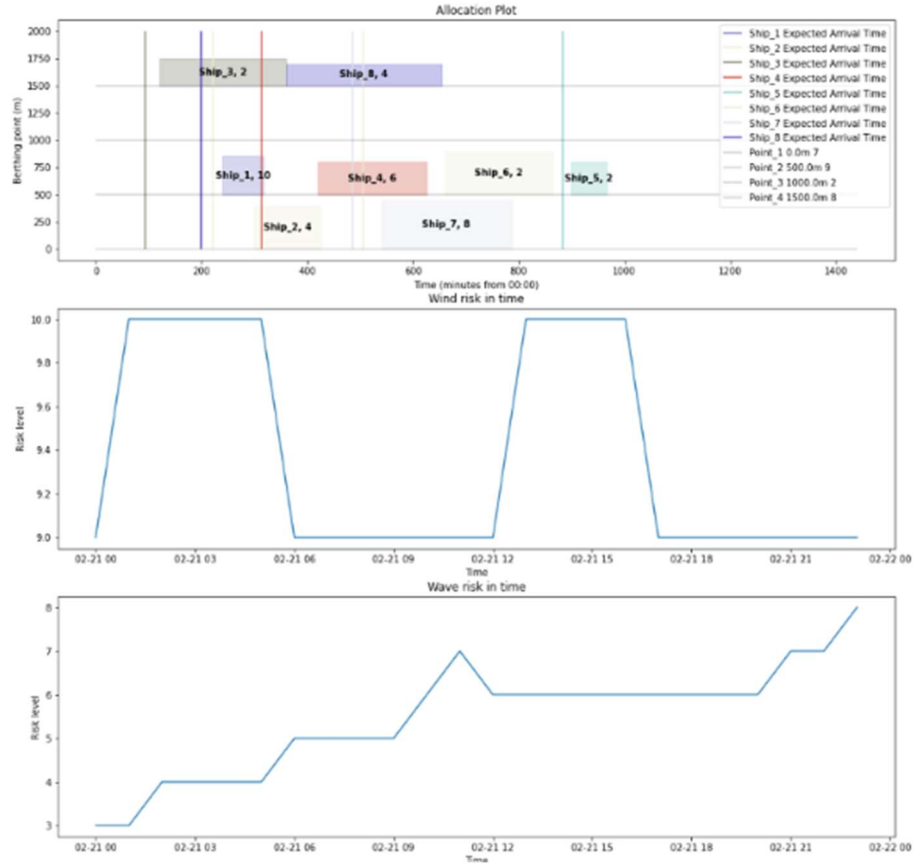


Fig. 11 – Experiment 2 - Berthing schedule and weather variables.

Concerning sustainability and efficiency preservation scoring, modified BAP showed appreciable performances. In the first test, for instance, our CSA approach correctly increased sustainability by reducing exposure to adverse weather conditions, still maintaining a prominent percentage of efficiency: ships were handled as soon as possible. In the second case, instead, it is

possible to observe how some ships were delayed with respect to the time-optimal solution (it is sufficient to consider the third berthing point to imagine a better schedule). On the other hand, however, the choice of excluding the least safe part of the wharf increased safety in operations by several units, keeping the Sustainability to Effectiveness Ratio positive.

The sustainability scoring system, however, raised two major problems, due to the nature of the formulation itself, that are to be considered while reading the sustainability index:

- It is **sensitive to initial conditions**: when repeatedly tested on a small number of ships and berths under varying weather conditions, especially when handling time slots do not overlap, the results show high variance. In some runs, the cost-optimal and sustainability-optimal solutions may coincide, while in others, the sustainability index can be several orders of magnitude higher. This phenomenon is pronounced when the cost-optimal solution assigns high-risk cargo ships to one of the available low-safety berths under severe winds, since it prioritizes minimal service times.
- The sustainable version of the algorithm is based on exponential functions, which means that **sustainability scores can vary abruptly**, especially in high-risk situations. For this reason, a qualitative scale was provided to scale down the sustainability score and avoid basing the assessment on extremely large values.

5. Conclusion

This article presented a preliminary study aimed at proposing alternative versions of commonly used algorithms to address Berth Allocation. This algorithmic solution was developed during the execution of the CRISIS project, alongside a second solution for identifying the Shortest Safe Path, together addressing both open sea transportation and berthing operations. Overall, the proposed approach opened new roads for the exploration of sustainable algorithm design, yielding promising results in laboratory testing. Following initial tests, the solution was successfully deployed in a second stage testing environment: a fully functional platform where real-time weather data is collected from external services and applied to realistic scenarios input by operators. The tests produced positive outcomes, enhancing safety during cargo loading and unloading by adjusting berthing points and/or times to reduce overall operational risks.

Accordingly, the authors acknowledge the study's limitations and identify directions for future research.

A primary limitation is the need to explore a broader range of algorithms. This article primarily focused on evolving the objective functions of CSA but did not investigate the possibility of applying the same cost functions to alternative algorithms.

Another limitation is the validation methodology, which is currently applicable only in a laboratory environment. The methodology is, in fact, purely theoretical and closely tied to the defined objective function. A suggested approach could be to create specific simulations that include ship features, transported goods, and weather conditions, allowing for the measurement of accident probability and related severity. These results could then be compared with classical approaches, enabling the definition of well-structured KPIs to assess how effectively the proposed modifications improve sustainability.

Finally, there is awareness of potential improvements to the proposed approach, which is currently based on value estimation. These improvements could be introduced by incorporating more accurate situational factors, including, but not limited to, a wider range of weather variables, operational efficiency factors, human risk factors, and the evaluation of possible violations of Collision Regulations, and defining a rigorous scoring method to correctly score cargo risk and berth safety.

6. Acknowledgments

This article has received funding from the European Union's INTERREG IPA CBC Italy – Albania – Montenegro Programme 2014-2020, CRISIS Project N. 465. This paper reflects only the authors' views and the European Union and/or IPA CBC Italy – Albania – Montenegro programme authorities can under no circumstances be regarded as responsible for any use that may be made of the information it contains.

References

- [1] UNCTAD, *Review of Maritime Transport 2018*. UNCTAD, 2018. [Online]. Available: https://unctad.org/system/files/official-document/rmt2018ch4_en.pdf. [Accessed: Apr. 24, 2024].
- [2] R. U. Khan, J. Yin, F. S. Mustafa, and N. Anning, "Risk assessment for berthing of hazardous cargo vessels using Bayesian Networks," *Ocean & Coastal Management*, vol. 210, p. 105673, 2021. doi: 10.1016/j.ocecoaman.2021.105673.
- [3] D. G. Siddaramaiah, D. S. Karnoji, and V. Gurudev, "Factors affecting the Vessel Turnaround time in a Seaport," in *Proc. HYDRO-2020 Int. Conf.*, NIT Rourkela, India, Mar. 2021.

- [4] F. Wang and A. Lim, "A stochastic beam search for the berth allocation problem," *Decision Support Systems*, vol. 42, no. 4, pp. 2186–2196, 2007.
- [5] M. Lind, M. Michaelides, R. Ward, H. Herodotou, and R. Watson, "Boosting data-sharing to improve short sea shipping performance: Evidence from Limassol port calls analysis," *UNCTAD Transport and Trade Facilitation Newsletter*, no. 82, Tech. Rep. 35, 2019.
- [6] X. Xiang, C. Liu, and L. Miao, "A bi-objective robust model for berth allocation scheduling under uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, pp. 294–319, 2017.
- [7] A. De, S. Pratap, A. Kumar, and M. Tiwari, "A hybrid dynamic berth allocation planning problem with fuel costs considerations for container terminal port using chemical reaction optimization approach," *Annals of Operations Research*, vol. 290, no. 1, pp. 783–811, 2020.
- [8] M. A. Dulebenets, "Application of evolutionary computation for berth scheduling at marine container terminals: Parameter tuning versus parameter control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 25–37, 2017.
- [9] S. Aslam, M. P. Michaelides, and H. Herodotou, "Enhanced berth allocation using the cuckoo search algorithm," *SN Computer Science*, vol. 3, no. 4, 2022. doi: 10.1007/s42979-022-01211-z.
- [10] S. Aslam, M. Michaelides, and H. Herodotou, "Optimizing multi-quay berth allocation using the cuckoo search algorithm," in *Proc. 8th Int. Conf. Vehicle Technology and Intelligent Transport Systems*, 2022, pp. 124–133. doi: 10.5220/0011081200003191.
- [11] E. Shadkam and M. Bijari, "Evaluation the efficiency of cuckoo optimization algorithm," *International Journal on Computational Science & Applications*, vol. 4, no. 2, pp. 39–47, 2014. doi: 10.5121/ijcsa.2014.4205.
- [12] N. Cheimanoff, F. Fontane, M. N. Kitri, and N. Tchernev, "A reduced VNS based approach for the dynamic continuous berth allocation problem in bulk terminals with tidal constraints," *Expert Systems with Applications*, vol. 168, p. 114215, 2021. doi: 10.1016/j.eswa.2020.114215.
- [13] M. Shehab, A. T. Khader, and M. Laouchedi, "A hybrid method based on Cuckoo search algorithm for global optimization problems," *Journal of Information and Communication Technology*, vol. 17, 2018.

- [14] M. J. Mihanović, S. Aksentijević, E. Tijan, and N. Kapidani, "Measuring the efficiency of Innovative Logistics Projects' Resilience: Epicenter Project Experience," in *Proc. 47th MIPRO ICT and Electronics Convention (MIPRO)*, 2024, pp. 962–967, doi: 10.1109/mipro60963.2024.10569218.

Submitted: 08/04/2025
Accepted: 14/05/2025

Danilo Martino
Advantech s.r.l., Italy,
Email: danilo.martino@advantechsrl.com

Emilio Cimino
Computer Science Engineer, Italy,
Email: emilio.cimino@outlook.it

Nexhat Kapidani
University of Montenegro, Faculty of
Maritime Studies Kotor, Montenegro,
Email: kapidani@t-com.me

Elizabeta Mrnjačević
Municipality of Ulcinj, Montenegro,
Email: elizabeta.mrnjacevic@ul-gov.me

Domenico Satalino
Municipality of Molfetta, Italy,
Email:
domenico.satalino@comune.molfetta.ba.it

Cloridano Bellocchio
Coastal Molise Flag, Italy,
Email: clobellocchio@gmail.com