ORIGINAL RESEARCH (AR)
DOI: 10.4274/jems.2025.15428

Journal of ETA Maritime Science

A Hybrid Harmony Search and Rule-Based Approach for Dynamic Crane Scheduling Problem

Hizia Amani, Linda Bouyaya, Rachid Chaib

University of Constantine 01, LITE Laboratory, Department of Transport Engineering, Constantine, Algeria

Abstract

The service time of container ships has become a central indicator of port productivity. In operational contexts, this time is mainly influenced by the number of quay cranes and the way they are scheduled. This study proposes a novel hybrid approach combining harmony search (HS) algorithm and rule-based dynamic simulation to solve the Quay Crane Scheduling Problem under non-crossing constraints. The problem is addressed in two consecutive stages, the first aims to identify efficient initial crane positions using HS, while the second simulates crane movements dynamically, relying on a set of predefined rules that guide real-time decisions. Additionally, three analytical features are defined to characterize the behavior of HS during the exploitation phase. Statistical tests confirmed that the proposed hybrid approach achieves significant improvement over the unidirectional heuristic while showing statistical equivalence with the genetic algorithm, thus validating its effectiveness and competitiveness.

Keywords: Maritime logistics, port productivity, Harmony search algorithm, dynamic crane scheduling

1. Introduction

Over the past decades, maritime transport has played an increasingly vital role in facilitating global trade. Containerization, in particular, has revolutionized the way goods are handled, allowing for standardized, secure, and efficient movement across vast supply chains [1,2]. Ports serve as critical nodes in this network, acting as gateways between sea and land transport [3]. As container traffic continues to grow, improving the performance of port operations has become a strategic necessity for economies relying on seamless and timely cargo flow [4].

Among the various operational indicators used to evaluate port efficiency, the service time of container ships remains one of the most crucial [5]. This metric directly reflects how effectively terminal resources are managed, especially the allocation and scheduling of quay cranes [6,7]. The time a ship spends at berth is not only a matter of cost but also a determinant of terminal throughput and customer satisfaction. Optimizing crane schedules, therefore, plays a central role in enhancing the overall performance and competitiveness of container terminals [8-10]. Figure 1 illustrates a typical container terminal environment, where quay cranes operate alongside berthed ships and coordinate with other handling equipment to ensure seamless cargo flow.

In this context, quay crane scheduling refers to the process of assigning unloading or loading tasks to cranes in an efficient and coordinated manner. The problem has been extensively studied in the literature, with most research focusing intensely on unidirectional (UDS) and bidirectional movement strategies. However, the internal distribution of containers across the ship bays also plays a critical role, particularly in studies where a bay is treated as a complete task [11-13]. Some works have restricted crane movement to a single direction to avoid potential interference [14-16], but such limitations can increase ship service time and lead to workload imbalance. Other studies have explored bidirectional movements and the possibility of assigning the same bay to multiple cranes, thereby enhancing solution flexibility [17]. Yet, unless these shared bays are processed in a tightly synchronized manner, they can cause weight imbalance, especially lateral stability between the seaside and land side of the ship. Early studies in this domain relied

Address for Correspondence: Hizia Amani, University of Constantine 01, LITE Laboratory, Department of Transport Engineering, Constantine, Algeria
E-mail: Amanihizia95@gmail.com
ORCID iD: orcid.org/0009-0007-9080-8191

Received: 11.07.2025
Last Revision Received: 26.09.2025
Accepted: 16.10.2025
Epub: 14.11.2025

To cite this article: H. Amani, L. Bouyaya, and R. Chaib. "A hybrid harmony search and rule-based approach for dynamic crane scheduling problem," *Journal of ETA Maritime Science,* [Epub Ahead of Print]

heavily on exact algorithms and commercial solvers, which tend to become inefficient as problem size increases and constraint verification becomes more complex.

Based on the limitations discussed above, this study emphasizes the adaptability of metaheuristics; unlike exact methods, which face scalability barriers as problem size grows, metaheuristics can be seamlessly hybridized with operational rules. This property enables the proposed approach to embed practical constraints and enhance realism, thereby directly overcoming the issue of insufficient dynamic adaptability.

Given the hybrid nature of this work, harmony search (HS) offers an appropriate balance between simplicity and effectiveness. While genetic algorithm (GA) and particle swarm optimization (PSO) are recognized for their strong performance in purely metaheuristic settings, the lightweight structure of HS makes it more suitable when combined with a rule-based approach. Unlike single-trajectory approaches such as Tabu Search or Simulated Annealing, HS evolves a population of candidate solutions through its harmony memory, thereby enhancing exploration while preserving computational efficiency. This combination of diversity and simplicity makes it particularly well aligned with the requirements of the proposed framework.

The objective of this study is to propose a novel solution to the quay crane scheduling problem (QCSP) under the non-crossing constraint, offering a perspective that departs from conventional approaches in the literature. In the following section, we review related studies, highlight the limitations inherent in each, and clarify the specific gaps that the present work addresses.

2. Literature Review

In this part, we divide the existing literature into three main categories based on how the QCSP has been approached. The first category encompasses foundational works that gradually shaped the problem's structure by addressing key operational constraints. The second category includes models that integrate crane scheduling with broader port operations.

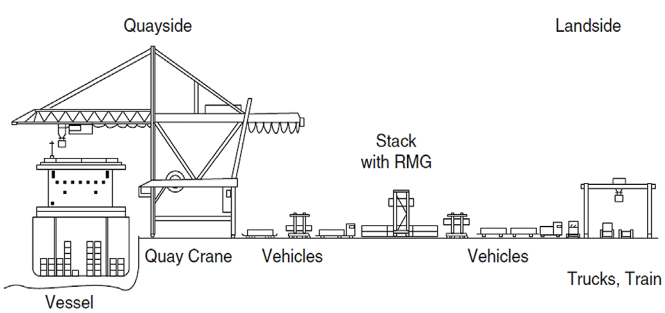

Figure 1. *Configuration of container terminal [18]*

The third category focuses on QCSP under uncertainty. While these models offer adaptive strategies, we also point out certain limitations related to movement flexibility and directional constraints that remain unaddressed.

2.1. QCSP

The QCSP was initially introduced by the author in [19], who formulated a mixed-integer programming model aiming to minimize ship turnaround time. The study [20] extended this formulation through a branch-and-bound approach, though both studies did not consider crane interference. The work [21] proposed a foundational model that treats tasks as complete bays and introduces non-crossing constraints. The study [22] expanded this modeling by presenting a more detailed framework that included non-crossing, minimum distance, and job-separation constraints. The proposed approach combined exact and heuristic methods. The research [23] improved the model of [21] by incorporating safety distances to prevent interference and solved it using a branch-and-cut method. The study [24] developed this line of research by employing a Tabu Search based on a disjunctive graph structure for faster solution generation. The work [11] presented a scheduling model based on individual jobs with non-crossing constraints, which implicitly enforced unidirectional crane movements. The proposed formulation introduced a detailed integer programming model and a graph-based simulated annealing method to solve large instances. The study [12] extended the model beyond single-ship scheduling by addressing multiple ships and proposed a two-level heuristic for minimizing ship tardiness. This study integrated berth-level QC allocation with ship-level workload scheduling, incorporating non-crossing, safety margins, and crane traveling times. It also allowed bay-level workload splitting, which improved flexibility and performance. The research [25] proposed the UDS heuristic for QCSP with container groups. This work outperformed previous approaches in both solution quality and computational speed, especially on benchmark instances. The study [26] introduced crane time windows, enabling dynamic task assignment based on crane availability over time. The work [27] incorporated unidirectional movement, time windows, and spatial limitations, and proposed a mixed-integer programming model combined with a tailored tabu search algorithm. The study [28] used a GA that permitted bidirectional crane movement while maintaining non-crossing constraints and balancing workloads. The research [29] presented hybrid evolutionary computation methods capable of generating diverse and efficient schedules. The study [17] developed a bidirectional model that allowed multiple cranes to serve the same bay, focusing on minimizing idle times and workload imbalance. The research [1] compared two metaheuristics (ACO and

GRASP–GA) for QCSP in non-automated terminals, and the best approach was embedded in a DSS to enhance crane utilization and terminal productivity.

These studies collectively illustrate the progressive evolution in how researchers have approached QCSP. Early works relying on exact optimization delivered precise solutions but quickly became impractical as problem size and constraint complexity increased, underscoring a fundamental scalability limitation. Later approaches based on pure metaheuristics alleviated this issue, yet they often left open the question of how close the obtained schedules were to the optimal, since solution quality depended heavily on stochastic search without embedded operational logic. In contrast, the present work integrates metaheuristic search with rule-based coordination, allowing operational constraints to directly guide the search process. This hybrid design enhances the reliability of the outcomes and narrows the gap to optimal solutions. A further distinction in prior studies lies in how constraints have been modeled-some focusing on entire bays, others on container groups, some including safety margins, others omitting them, making direct comparisons difficult. By adopting a consistent bay-level model enriched with practical rules such as non-crossing and workload balance, our approach provides a more unified and operationally realistic framework.

2.2. Integrated QCSP

In berth scheduling, the service time of a ship is not a fixed input; it directly depends on the number of quay cranes assigned and the efficiency of their coordination. This dependency has led several studies to embed quay crane scheduling within berth allocation models [30,31]. The study [32] developed a simulation-optimization framework that integrates berth allocation and quay crane scheduling under stochastic handling times and dynamic ship arrivals. The proposed model was solved using simulated annealing to minimize makespan. The research [33] addressed this integration by proposing a joint model where berthing order, berthing position, and the number of cranes per ship are optimized simultaneously. This work relied on an enhanced genetic algorithm with population partitioning that explicitly incorporates crane assignment. The study [34] moved further toward realism by introducing time-variant crane scheduling under tidal access constraints. In this model, ships are assigned cranes dynamically throughout their stay, depending on tidal windows and operational feasibility. The authors proposed three hybrid heuristics to handle the combined berth and crane decisions efficiently. The research [35] proposed a deeply integrated framework in which berth allocation, crane assignment, and crane scheduling are handled together while enforcing non-crossing constraints and safe distances between cranes. The model was solved using a random-topology PSO designed for large-scale instances. The work [36] adopted a discretization strategy that converts the continuous berth space into manageable segments. The proposed approach included a large neighborhood search mechanism with dedicated procedures to preserve feasibility throughout the optimization process.

Some recent contributions have focused on the joint scheduling of quay cranes and yard trucks to enhance coordination and minimize operational bottlenecks. The study [37] proposed a mixed-integer linear programming model for jointly scheduling quay cranes and trucks, addressing precedence, crane interference, and blocking constraints. The authors used an improved PSO algorithm and considered both unidirectional and bidirectional container flows. The research [38] formulated a mixed-integer linear programming model for the joint scheduling of quay cranes, yard trucks, and yard cranes. A genetic algorithm with a 3D chromosome design was introduced to improve efficiency and to support low-carbon terminal operations. The model in [37] was extended by the study [39], which corrected previous assumptions such as allowing two-container handling per crane or truck. The authors proposed an adaptive PSO that dynamically adjusts parameters and demonstrated that it outperformed earlier methods on larger problem instances.

While these integrated studies provide valuable insights into berth-quay and quay-yard interactions, they generally target a broader scope of terminal operations. In contrast, the present work focuses specifically on intra-vessel quay crane coordination under non-crossing constraints, providing a complementary but distinct contribution to the literature.

2.3. QCSP Under Uncertainty

Several researchers have addressed QCSP under uncertainty, with growing attention to realistic parameters such as stochastic processing times, uncertain ship arrivals, energy fluctuations, and even fuzzy handling conditions. Focusing on processing time as a primary stochastic element, the author in [40] employed a two-stage stochastic programming model complemented by metaheuristics for scalability, while the study in [41] introduced a unified distributionally robust framework that balances between stochastic and robust extremes through a tunable parameter. In a similar approach, the study in [15] addressed uncertainty from a dual-objective perspective, minimizing makespan and energy consumption using exact techniques under deterministic but fixed-direction conditions. The study in [42] tackled integrated berth and crane scheduling in multi-terminal tidal ports under uncertainty, using a nonlinear formulation and an adaptive genetic algorithm enhanced with simulated annealing. The author in [43] considered berth and crane assignment jointly, formulating a robust model that also incorporates fluctuating energy prices. Meanwhile, the study in [16] extended the discussion to dynamic rescheduling, reacting to ship delays and unscheduled arrivals with a rolling-horizon heuristic. Additionally, the study in [44] adopted a fuzzy logic approach to handle the uncertainty of ship arrival times, offering an alternative to probabilistic representations. From an algorithmic standpoint, the author in [14] proposed a hybrid estimation of distribution algorithm enhancing traditional probabilistic models through bio-inspired local search. Despite the diversity and depth of these contributions, most of the above works, with the exception of the study in [42], adopt the unidirectional crane movement assumption, which is often presented as the optimal strategy in the literature. While this simplification helps avoid interference and reduces model complexity, it does not fully reflect real operational conditions. In practice, the distribution of containers along the containership can significantly affect the completion time of tasks; enforcing a single movement direction may prevent the achievement of optimal makespan. In our approach, cranes are allowed to move in both directions, which provides greater flexibility and enables more efficient scheduling. We summarize the main contributions of this research as follows:

- The study introduces a novel hybrid formulation of the QCSP, combining HS with rule-based coordination to ensure realistic scheduling. This directly responds to the scalability limits of exact methods and the lack of optimality assurance in pure metaheuristics.

- Explicit integration of initial crane positioning as a decision variable, overcoming the fixed-starting-point limitation of earlier QCSP models.

- Introduction of empty and full path concepts to guide dynamic crane assignment, providing flexibility beyond the rigid unidirectional assumptions common in previous studies under uncertainty.

- Feature-based analysis to explore the behavior of the Harmony algorithm, highlighting the emergence of a dominant feature across multiple scenarios.

- Comprehensive performance assessment of the proposed approach through comparative experiments against state-of-the-art methods, including a benchmark GA and the UDS heuristic, as well as a relaxed non-crossing scenario used to estimate the solution space limits.

The remainder of this paper is organized as follows. Section 3 introduces the problem and outlines the operational assumptions adopted in this study. Section 4 details the proposed hybrid methodology. In Section 5, a comprehensive experimental analysis is presented, highlighting the behavioral features of the proposed method, benchmark comparisons, and sensitivity tests. Finally, Section 6 concludes the paper with key findings and future research perspectives.

3. Problem Description

This study focuses on the scheduling of multiple quay cranes assigned to unload containers from a single ship. The ship is divided into a sequence of bays, each containing a known number of containers as shown in Figure 2. Each bay is treated as an indivisible task that must be fully handled by one crane. A set of quay cranes is available and allowed to move bidirectionally along the ship. Bays are indexed from left to right, with Bay 1 located at the far left and the last bay at the far right. Throughout the operation, a strict non-crossing constraint is enforced, ensuring that cranes never pass over or interfere with one another. The time required to move between adjacent bays is assumed to be negligible. All cranes are assumed to operate at the same speed, and the handling time of a bay is calculated by the number of containers in the bay divided by the crane's handling speed. In contrast to many previous studies where the initial crane positions are fixed [11,21,37,39,41], this work considers these positions as part of the decision process. The aim is to explore how different initial configurations can influence the efficiency of the resulting schedule. To support this investigation, three positioning strategies are proposed and evaluated. The objective is to minimize the total completion time of the unloading process, also known as the makespan, which corresponds to the latest finishing time among all cranes. This objective also expresses the balance of workload among cranes, as minimizing the makespan requires distributing the unloading tasks as evenly as possible.

The problem setting and constraints described above form the foundation upon which our hybrid resolution method is developed. The next section presents the methodological framework combining HS and rule-based simulation to solve the QCSP efficiently.

4. Methodology

This section presents the proposed methodology for solving the QCSP, focusing on the intelligent initialization of crane

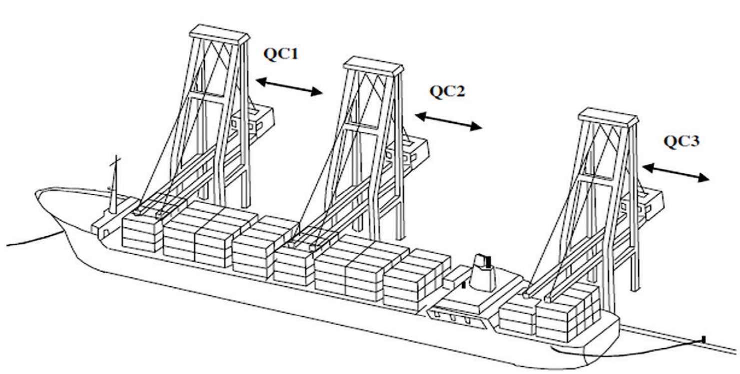

Figure 2. *QCs working on a ship [21]*

QCs: Quay Cranes

positions and the rule-based simulation of their movements. The entire implementation was developed using Python 3.9. This programming language was chosen for its simplicity and ease of integration with algorithmic development, which makes it particularly suitable for prototyping optimization models.

4.1. Rationale for Initial Crane Positioning

In practice, the uneven distribution of containers across the bays is a key factor contributing to workload imbalance. Ideally, if the number of containers was evenly distributed, the scheduling process would be more predictable and straightforward. Based on this observation, we argue that determining the initial crane positions should not be left to random selection or simple criteria. Instead, it must be approached with careful analysis.

4.2. HS Algorithm

Originally introduced in 2001, the HS algorithm is a population-based metaheuristic inspired by the improvisation process of musicians seeking the best harmony. Each musician contributes a note, and together they strive to achieve the most pleasing combination [45,46].

The HS algorithm has shown remarkable adaptability in a variety of optimization problems [47-49]. However, despite its potential, it remains underexplored in maritime scheduling applications, especially compared to more popular methods such as GAs or PSO [50]. This relative scarcity provides an opportunity to investigate its performance in a novel context.

4.3. Parameters

4.3.1. Harmony memory size (HMS)

This represents the number of solutions stored per iteration. Each solution corresponds to a unique set of initial crane positions. HMS was set to 7 based on preliminary tests, as higher values did not improve results and lower values sometimes degraded performance.

4.3.2. Harmony memory consideration rate (HMCR)

This parameter governs the probability of choosing a value from the memory rather than generating it randomly.

4.3.3. Pitch adjustment rate (PAR)

It controls the fine-tuning of selected variables. Pitch adjustment enables slight modifications to crane positions, allowing the algorithm to explore nearby configurations and escape potential local optima.

HMCR and PAR were fixed at 0.95 and 0.3, respectively, following commonly used values in the literature [45].

4.3.4. Bandwidth (BW)

BW defines the maximum allowable shift applied to a crane's bay assignment during the pitch adjustment phase. When a bay position is selected from memory and the pitch adjustment condition is met (governed by PAR), the selected value is perturbed by a random amount within $\pm$bw. BW was set to 1, meaning that each adjustment modifies the initial position of a crane by one bay.

4.3.5. Number of improvisations (NI)

It defines the number of iterations performed. NI depended on the problem size and the stability of the results; it was set between 20 and 100.

4.4. Rules-Based Approach to Define the Movement Path of Cranes

Once the initial positions of each crane are determined using the HS algorithm, the unloading time of each crane is calculated based on the number of containers in its assigned bay; and is then divided by the crane's speed.

These unloading times are recorded, and the remaining bays are updated accordingly. Then, the crane that completes its task first is selected to move, reflecting a realistic scenario aimed at minimizing idle and waiting times. The selection of the next bay depends on the available path, following a set of prioritized rules.

4.4.1. First priority-moving along the empty path

Cranes are initially ranked according to the bay from which they start. Since cranes are not allowed to cross each other, this order must be respected until the end of the unloading process. The ranking increases in the same order as the bays (from the far left to the far right). The first priority is to select an empty path, which means a path that does not contain any crane either with a lower or a higher rank.

For example, if the crane to be moved is at bay 5- and other cranes are at bays 7 and 10, then the empty path is in the left direction, covering bays 4, 3, 2, and 1.

Bay Selection Within the Empty Path:

If an empty path is available, the crane must select the farthest bay in that path, meaning the bay with either the lowest or highest numerical rank among the remaining bays, depending on the direction of the path. In the given example, the farthest bay in the empty path is bay 1.

4.4.2. Second priority-moving along the full path

This priority applies when the empty path is not available and a full path is present only in one direction. In such a case, the crane is guided along that direction. This situation typically arises in two cases:

- The crane is initially positioned at the edge of the bay layout, (i.e., the first or the last bay), leaving only one possible path.

- During the unloading process, all remaining bays in the designated path have been completed, leaving only bays the direction of other operating cranes.

Bay Selection Within the Full Path:

The crane must select the closest bay in that path.

- If this occurs immediately following the initial position, the selected bay is the bay ranked immediately before or after the initial bay, depending on the direction of the path (left or right).

- If the situation occurs during the process, the selected bay is the one with the minimum distance to the crane's current position, among the remaining bays.

4.4.3. Third priority-crane between two full paths

This case applies when the crane's current rank is located between two full paths, meaning there are cranes operating in both directions (left and right).

Bay Selection in this Situation:

The crane must select the closest bay from one of the two paths, based on one of the following criteria:

- Prefer the path that has the greater number of remaining bays.

If this number is equal in two directions:

- Prefer the path in which the total number of containers in the remaining bays is higher.

4.4.4. Blocked crane

This concept is introduced to represent the situation where a crane becomes restricted by the non-crossing constraint, and no path is available for further movement.

During the path selection process, it is possible for a crane to find neither an empty nor a full path, either due to no remaining bays in those paths, or ranking constraints imposed to prevent crossing.

In such cases, the crane is identified as a blocked crane, and its operation is terminated by recording its final unloading time, which corresponds to the last bay it was assigned to.

The three detailed priorities for selecting paths and bays are applied sequentially for each crane. The crane that finishes its current task first is assigned to the next bay, and this order is respected continuously until all bays are completely unloaded.

4.4.5. Interaction of crane movement rules with the HS

The HS generates the initial crane positions, by considering a number of bays equal to that of the cranes. From these initial positions, the rule-based procedure simulates the movement of each crane and constructs a complete unloading schedule. The resulting makespan value represents the quality of the candidate solution and determines its inclusion in the harmony memory. To enhance realism, the concept of a blocked crane was explicitly incorporated and treated as a natural operational scenario. Such cases are efficiently handled within the simulation, where the schedule automatically results in an increased makespan, reflecting the reduced efficiency of the configuration. This mechanism ensures that blocked-crane situations are realistically captured without disrupting the process, while simultaneously reducing the probability of such solutions being retained in the harmony memory.

Example:

We illustrate an example with 3 cranes and 10 bays, as shown in Figure 3.

The number of containers in each bay is (16, 18, 22, 14, 17, 11, 20, 13, 15, and 19), respectively, with a fixed unloading time of 1 minute per container.

We select random initial positions: bays 2, 5, and 8 for the three cranes.

- The crane at bay 8, which finishes first with a minimum time of 13 minutes, moves along the empty path in the right direction to the farthest bay, bay 10. It concludes at 32 minutes, calculated as 13+19.

Remaining bays: [1,3,4,6,7,9].

- The crane at bay 5 can move in two directions, finishing at 17 minutes.

In both directions, the number of remaining bays is 3, so we apply the second rule:

For bays 1, 3, 4 → total containers = 16+22+14=52

For bays 6, 7, 9 → total containers = 11+20+15=46

Therefore, the crane moves along the full path to the left to the closest bay, (bay 4). The duration finishes at 31 minutes, calculated as starting at 17 minutes and adding 14 minutes.

Remaining bays: [1,3,6,7,9].

- The crane at bay 2, with a finish time of 18 minutes, moves along the empty path in the left direction to the farthest bay, bay 1. It finishes at 18+16=34 minutes.

Remaining bays: [3,6,7,9].

- The crane at bay 4 moves to the closest bay, bay 6, along the full path in the right direction, passing one bay in the left direction, bay 3, and two bays in the right direction, bays 7 and 9. It finishes at 31+11=42 minutes.

Remaining bays: [3,7,9].

The crane at bay 10 moves along the full path available to the left, stopping at the closest bay, bay 9. It finishes at 32+15=47 minutes.

Remaining bays: [3,7].

- The crane at bay 1 moves along the full path to the closest bay, bay 3. It finishes at 34+22=56 minutes.

Remaining bay: [7].

- The crane at bay 6 moves along the full path in the direction of the closest bay, bay 7. It finishes at 42+20=62 minutes.

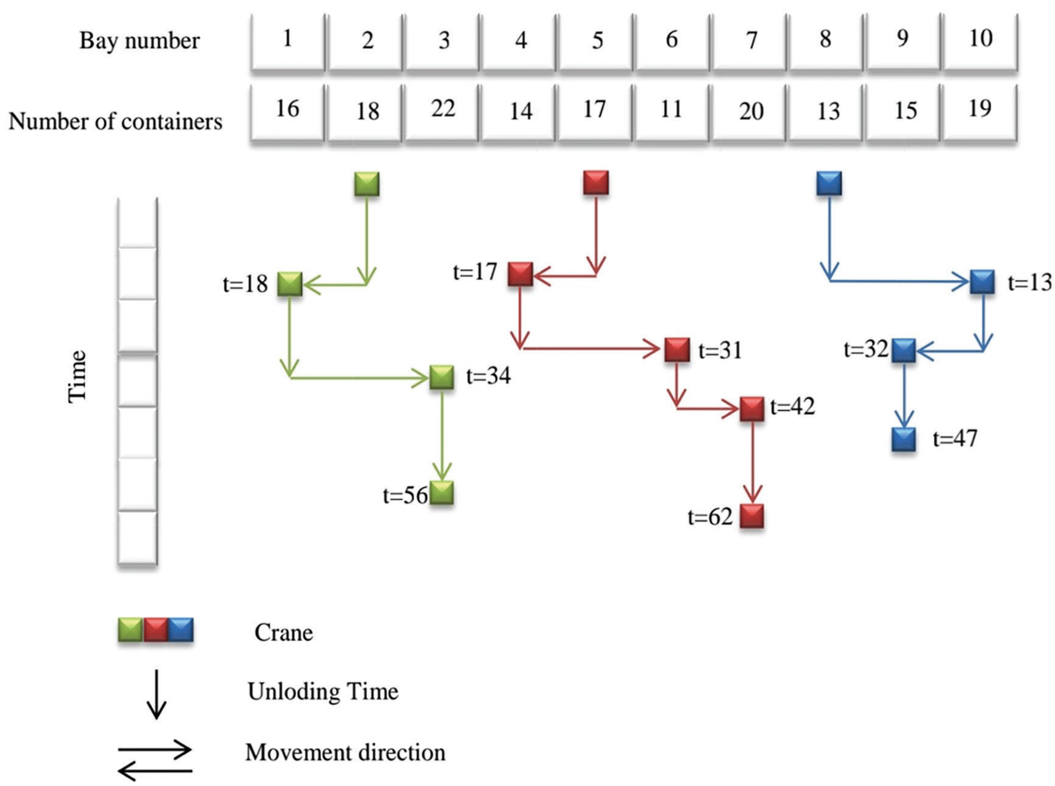

Figure 3. *Quay crane movements*

Thus, C_{max} = 62 minutes.

However, the near-optimal solution for this example, based on the proposed method, is 55 minutes, with initial positions [3,5,7] for the three cranes.

Figure 4 presents the flowchart providing a comprehensive overview of the proposed hybrid method.

5. Results and Analysis

5.1. Behavioral Analysis of the Harmony-Based Approach

To understand how the proposed harmony-based approach structurally behaves, we analyzed three key features extracted from each solution generated during the optimization process. These features were observed over 30 independent test cases, each involving a distinct bay configuration and random container distribution. The goal was to determine which structural tendencies consistently appear in high-performing solutions.

5.1.1. Total containers in assigned bays (TCA)

This feature reflects the total number of containers handled by the selected crane positions. A higher TCA indicates that choosing fuller bays helps to accelerate the unloading process and reduce the makespan.

5.1.2. Container count range (CCR)

The CCR is defined as the difference between the maximum and minimum number of containers among the selected bays.

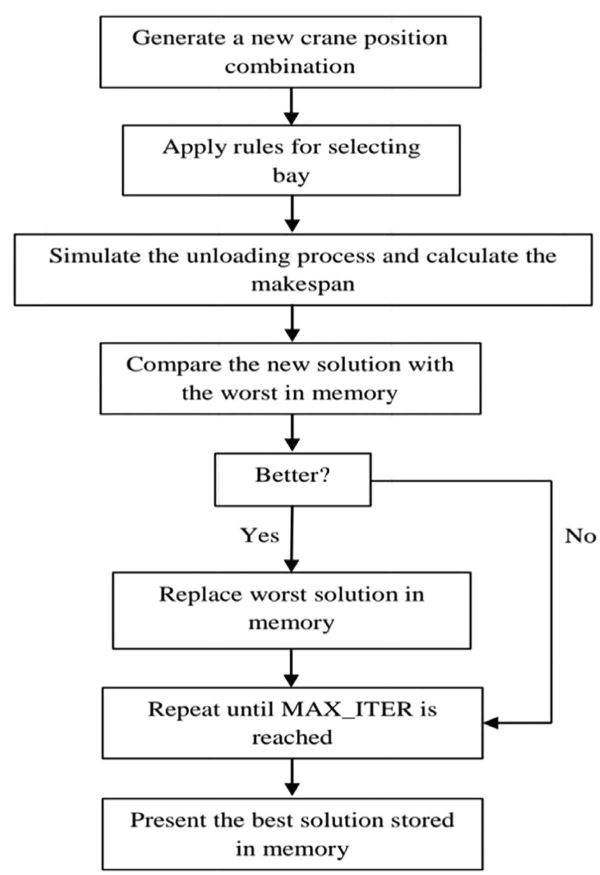

Figure 4. *The flowchart*

Smaller CCRs indicate more balanced workloads between cranes, helping to avoid bottlenecks.

5.1.3. Minimum initial distance between cranes (MID)

This is the minimum distance between any two cranes in the initial layout. Larger spacing reduces crane interference and allows more flexible movement in the dynamic phase.

5.1.4. Application

For each of the 30 test cases, the HS algorithm generated a memory of candidate solutions (HMS=7). The best-performing solution (with the lowest C_{max}) was then compared feature-by-feature against the other solutions in the memory. A feature (TCA, CCR, or MID) was considered dominant if the best solution performed better than at least half of the other candidates in that particular metric. If the best solution had a TCA that is higher than or equal to at least 4 out of the 6 other candidates, TCA was counted as dominant for that test case. This comparison was repeated for CCR and MID independently. To evaluate the consistency of feature dominance patterns, the experiment was repeated 10 times using different random seeds, with the results as shown in Table 1.

- Although the HS algorithm involves randomness in candidate generation and memory updates, the overall dominance trends remained relatively stable. Across the 10 runs, the MID feature showed the highest average dominance, frequently appearing in the range of 50% to 70%. For instance, it reached 67% in run 3, 70% in run 4, and 67% again in run 5. Even in the lowest cases (run 6, 8 and run 9), it still maintained a value of 53% and 50%, which is higher than or equal to the other features in those runs. This confirms its importance in maintaining spatial separation between cranes and minimizing interference during operation.

- The CCR and TCA features showed more fluctuation, occasionally reaching dominance in individual runs, but without the same consistency. For example, CCR reached 57% in run 4, while TCA reached 53% in runs 2 and 8. Their dominance values typically ranged between 37% and 57% depending on the run.

- These findings suggest that while feature dominance can vary slightly between executions, MID tends to be the most influential structural characteristic, reinforcing the importance of crane spacing in effective scheduling.

Table 1. *Summary of feature dominance across 10 runs*

Run	TCA (%)	CCR (%)	MID (%)
1	40%	37%	63%
2	53%	43%	63%
3	43%	53%	67%
4	43%	57%	70%
5	47%	50%	67%
6	40%	37%	53%
7	43%	53%	63%
8	53%	43%	53%
9	40%	50%	50%
10	50%	50%	63%

TCA: Total containers in assigned bays, CCR: Container count range, MID: Minimum initial distance

5.2. Benchmark Comparison

To validate the performance of our proposed method, we compare its results against established approaches. This includes two benchmarks from the work of Skaf et al. [13]: a GA, and an exact method based on dynamic programming algorithm (DPA), both developed for the same crane scheduling problem (Table 2).

The 26 instances considered in Table 2 were designed to cover a range of problem sizes and complexities. For scenarios with two quay cranes, the number of bays varied between 5 and 14, while for three cranes it ranged from 7 to 22 bays. This corresponds approximately to the handling of small- to medium-sized ships. Increasing the number of bays and cranes naturally increases the computational complexity of the problem, thereby providing a representative set of instances for evaluation.

Additionally, we include results from a classical heuristic (UDS - unidirectional scheduling), which provides a structured but simpler baseline for comparison (Table 3). To estimate the quality of the proposed schedules under real operational constraints, a relaxed version of the problem was solved using a multi-start greedy algorithm (GEA). This approach ignores crane interference and balance the workload among cranes by testing multiple randomized bay allocations and selecting the most balanced configuration. Although this solution is not applicable in practice due to the absence of non-crossing constraints, it provides a useful lower bound for evaluating the performance of the implemented scheduling strategies (Table 3). The results of the GEA reported in Table 3 correspond to the stable outcome that consistently reappeared across multiple runs, rather than a single random result.

To ensure an objective comparison, the experimental data published in Skaf's thesis 2020 [51] was used. Table 2 includes his original results, as well as those produced by our proposed method.

For the comparison with UDS, synthetic data was used, specifically designed for this problem to test our approach in various scenarios.

- To statistically validate the comparative results reported in Table 2, we performed paired analyses on the 26 benchmark instances. For the comparison between hybrid harmony search (HHS) and GA (n=26), a paired t-test revealed no significant difference [t (25)=0.254, p=0.802], with a negligible effect size (Cohen's d=0.05). This indicates that both metaheuristics perform equivalently in terms of makespan. Since the exact DPA failed to return a solution in instance 26, comparisons involving DPA were conducted on the remaining 25 instances. Repeated- measures ANOVA [F (2.48)=8.983, p=0.000485] and Friedman's non-parametric test [χ^2 (2)=21.06, p≈2.710-5] confirmed significant overall differences among the three methods. The post-hoc test showed that DPA, as expected from an exact method, achieved a significantly lower makespan than both GA (p<1e-6, large effect) and HHS (p≈0.003, medium effect). Importantly, the performance gap between the proposed HHS and the exact DPA remains limited, highlighting that the metaheuristic produces near-optimal solutions with a fraction of the computational effort. These findings confirm the competitiveness and practical value of HHS. While DPA represents the theoretical benchmark, HHS achieves results statistically indistinguishable from the GA and only marginally higher than the exact solutions.

- However, while the proposed method outperformed the GA in most cases, the latter achieved better results in some specific instances. These exceptions highlight that although HHS is generally robust, it could benefit from further refinement to increase adaptability without losing its structured coordination rules.

- The proposed hybrid approach consistently achieved execution times below one second across all problem sizes, even where the GA required significantly longer central

Table 2. *Results of DPA, GA, and HHS*

Instances	NQ	NB	Makespan			CPU time	
			DPA (min)	GA (min)	HHS (min)	GA (s)	HHS (s)
1	2	5	39.78	39.78	39.78	<1	<1
2	2	6	51.48	51.48	52.65	<1	<1
3	2	7	58.5	59.67	58.5	<1	<1
4	2	8	73.71	75.71	73.71	<1	<1
5	2	9	90.09	91.26	90.09	<1	<1
6	2	10	99.45	100.62	100.62	<1	<1
7	2	11	104.13	107.6	106.64	<1	<1
8	2	12	107.64	109.98	107.64	<1	<1
9	2	13	114.66	115.83	114.66	<1	<1
10	2	14	119.34	122.85	122.85	<1	<1
11	3	7	39.78	39.78	39.78	<1	<1
12	3	8	51.48	52.8	52.65	<1	<1
13	3	9	60.84	63.35	66.69	<1	<1
14	3	10	70.2	72.71	73.71	<1	<1
15	3	11	73.71	76.02	73.71	<1	<1
16	3	12	73.71	74.88	73.71	<1	<1
17	3	13	76.05	76.05	81.9	<1	<1
18	3	14	81.9	83.07	91.26	<1	<1
19	3	15	91.26	92.43	93.6	<1	<1
20	3	16	95.94	98.28	100.62	1.59	<1
21	3	17	101.79	105.3	104.13	3.12	<1
22	3	18	105.3	106.47	108.81	7.39	<1
23	3	19	113.49	113.49	112.32	10.62	<1
24	3	20	118.17	119.34	114.66	28.64	<1
25	3	21	125.19	127.53	127.53	34.72	<1
26	3	22	-	134.69	132.21	60.12	<1

NB: Number of bays, NQ: Number of cranes, DPA: Dynamic programming algorithm, GA: Genetic algorithm, HHS: Hybrid harmony search

Table 3. *Results of UDS, HHS, and GEA*

Instances	NQ	NB	UDS (min)	HHS (min)	GEA (min)
1	2	10	98.28	85.41	85.41
2	3	10	62.01	59.67	58.5
3	2	10	115.15	104.13	101/79
4	3	10	76.05	72.54	67.86
5	2	10	101.79	101.79	100.62
6	3	10	81.9	71.37	67.86
7	2	10	112.32	112.32	111.15
8	3	10	84.24	77.22	74.88
9	3	20	145.08	143.91	142.74
10	4	20	113.49	107.64	107.64
11	3	20	143.91	140.4	134.55
12	4	20	111.15	106.47	101.79
13	3	20	150.93	139.23	133.38
14	4	20	106.47	104.13	100.62
15	3	20	148.59	135.72	134.55
16	4	20	108.81	101.79	101.79

NB: Number of bays, NQ: Number of cranes, HHS: Hybrid harmony search, UDS: Unidirectional scheduling, GEA: Greedy algorithm

processing unit times. This confirms that the integration of rule-based movement with HS not only improves solution quality but also ensures computational efficiency under realistic operational constraints.

- The statistical analysis of Table 3 confirms the robustness of the proposed approach. A paired t-test between UDS and HHS revealed a significant difference ($t=5.29$, $p<0.001$), indicating that the hybrid HS consistently outperforms the unidirectional heuristic in terms of makespan reduction. The comparison of the three methods was extended using the Friedman test, and the results showed a highly significant overall difference ($\chi^2=29.53$, <0.001), which confirms that the performance of the algorithms is not equivalent across scenarios. To further investigate these differences, a post-hoc pairwise Wilcoxon signed-rank test was conducted. The results showed that both HHS and GEA significantly outperformed UDS, and the difference between HHS and GEA was also statistically significant, favoring GEA. Nevertheless, it is important to note that the apparent advantage of GEA stems from its relaxed assumptions, since it ignores crane interference constraints, whereas HHS achieves competitive results while fully respecting operational realities. These findings highlight the practical value of the hybrid approach: it not only yields significantly better performance than traditional heuristics but also approaches the efficiency of lower-bound methods while remaining realistic and applicable.

5.3. Effect of bay layouts on unloading performance

To examine the influence of container distribution on unloading performance, we designed a comparative experiment involving six different layouts for each scenario. Two cases were considered: one where the number of bays is divisible by the number of cranes, and another where it is not. For each case, five randomly generated unbalanced layouts were compared to a perfectly balanced one, in which each bay contains an equal number of containers. The goal was to assess how irregular distributions affect the makespan, even when the total workload remains constant. The layouts and corresponding C_{max} values are presented in Tables 4 and 5.

5.3.1. Scenario 1

This scenario involves 90 containers distributed across 9 bays, to be unloaded by 3 cranes. The number of bays is divisible by the number of cranes, and each crane can handle an equal number of bays, resulting in a minimum theoretical C_{max} of 30 in the balanced layout.

- The results show that in the case of 9 bays and 3 cranes, the container layout significantly affects unloading efficiency. The balanced layout achieved the theoretical minimum makespan of 30, representing perfect workload distribution. However, layout 1 resulted in the highest C_{max} of 40, highlighting how imbalance can severely hinder performance. In contrast, layouts 2 and 3 achieved a significantly lower C_{max} of 33, demonstrating that some asymmetrical configurations can reduce total unloading time. Layouts 4 and 5 performed

Table 4. *Case of 9 bays and 3 cranes*

Layout	Number of containers	C_{max}
Balanced	[10, 10, 10, 10, 10, 10, 10, 10, 10]	30
1	[6, 8, 12, 14, 20, 14, 12, 8, 6]	40
2	[5, 6, 7, 8, 9, 10, 11, 12, 22]	33
3	[7, 14, 8, 11, 9, 13, 6, 10, 12]	33
4	[4, 5, 6, 7, 8, 9, 15, 16, 20]	36
5	[20, 16, 15, 9, 8, 7, 6, 5, 4]	36

Table 5. *Case of 10 bays and 3 cranes*

Layout	Number of containers	C_{max}
Balanced	[10, 10, 10, 10, 10, 10, 10, 10, 10, 10]	40
1	[6, 8, 12, 14, 15, 14, 12, 8, 6, 5]	40
2	[5, 6, 7, 8, 9, 10, 11, 12, 13, 19]	35
3	[7, 14, 8, 11, 9, 13, 6, 10, 12, 10]	36
4	[4, 5, 6, 7, 8, 9, 15, 16, 20, 10]	40
5	[20, 16, 15, 9, 8, 7, 6, 5, 4, 10]	38

moderately, with a C_{max} of 36 each. These outcomes suggest that although balanced layouts are theoretically optimal, certain unbalanced distributions, if strategically designed, can also achieve competitive or even improved efficiency depending on the dynamic movement of cranes and bay assignments.

5.3.2. Scenario 2

This scenario involves 100 containers distributed across 10 bays, to be unloaded by 3 cranes. Since the number of bays is not divisible by the number of cranes, at least one crane must handle more bays, resulting in a minimum theoretical C_{max} of 40 in the balanced layout.

- The results confirm that in cases where the number of bays is not divisible by the number of cranes, the container layout continues to influence unloading efficiency. Although the balanced layout yielded a C_{max} of 40, it was matched by layouts 1 and 4, and slightly outperformed by layout 5 (C_{max}=38). Most notably, layout 2 achieved the best performance with a C_{max} of 35, demonstrating that strategic imbalances can reduce overall makespan. Layout 3 also slightly improved upon the balanced case. These observations highlight that even in structurally imperfect divisions, thoughtful container distribution can mitigate workload disparities among cranes and enhance system efficiency.

6. Conclusion

This study introduced an innovative hybrid approach combining the HS algorithm with a rule-based dynamic simulation to solve the QCSP under strict non-crossing constraints. Our approach ensures logical decision-making and preserves spatial order without relying on exhaustive search. Experimental results demonstrated that the proposed approach delivers competitive outcomes with a computation time of less than one second, even in complex scenarios.

HS was employed in this study due to its balance between simplicity and efficiency in exploring large discrete solution spaces, making it well-suited for the QCSP. Nevertheless, other metaheuristics such as ACO and the Firefly Algorithm; though less frequently applied in this domain, may provide valuable perspectives and represent interesting directions for future research.

The proposed method could be further integrated into terminal operating systems as a decision-support tool, providing real-time guidance for crane allocation and coordination.

Since travel time between bays was not considered, addressing this factor becomes particularly meaningful and impactful as ship size increases.

While the current work focused on a single ship, extending simulation-based approaches to multiple ships and integration of berth allocation would represent an important step toward enhancing practical applicability in large container terminals.

Future work could also address uncertainty in crane scheduling, for instance, by considering variations in handling times, random equipment failures, or unexpected delays.

Footnotes

Authorship Contributions

Concept design: H. Amani, Data Collection or Processing: H. Amani, Analysis or Interpretation: H. Amani, L. Bouaya, and R. Chaib, Literature Review: H. Amani, Writing, Reviewing and Editing: H. Amani, L. Bouaya, and R. Chaib.

Funding: This research received no external funding.

References

[1] F. Eldemir, and M. E. Taner, “A hybrid meta-heuristic approach for solving single-vessel quay crane scheduling with double-cycling,” *Journal of Marine Science and Engineering*, vol. 13, pp. 371, Jun 2025.

[2] Ü. Özdemir, “Analysis of empty container accumulation problem of container ports,” *Journal of ETA Maritime Science,* vol. 6, pp. 319-332, Dec 2018.

[3] F. Rodrigues and A. Agra, “Berth allocation and quay crane assignment/scheduling problem under uncertainty: a survey,” *European Journal of Operational Research,* vol. 303, pp. 501-524, Dec 2022.

[4] M. Safaeian, F. Etebari, and B. Vahdani, “An integrated quay crane assignment and scheduling problem with several contractors in container terminals,” *Scientia Iranica*, vol. 28, pp. 1030-1048, 2021.

[5] M. Saini and T. Lerher, “Assessing the factors impacting shipping container dwell time: a multi-port optimization study,” *Business: Theory and Practice*, vol. 25, pp. 51-60, Feb 2024.

[6] C. T. Ngoc, X. Xu, H. S. Kim, D. A. Nguyen, and S. S. You, “*Container port throughput analysis and active management using control theory*,” Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment, vol. 236, pp. 185-195, 2022.

[7] M. O. Herup, G. C. W. Thiesgaard, J. van Twiller, and R. M. Jensen, “*A linear time algorithm for optimal quay crane scheduling,*” In: de Armas, J., Ramalhinho, H., Voß, S. (eds) Computational Logistics. ICCL 2022. Lecture Notes in Computer Science, vol 13557. Springer, Cham.

[8] A. Al-Refaie and H. Abedalqader, “Optimal quay crane assignment and scheduling in port’s container terminals,” *Jordan Journal of Mechanical & Industrial Engineering*, vol. 15, pp. 153-167, Jun 2021.

[9] G. Tuncel, Ö. Yalçınkaya, E. Deniz, and S. Esmer, “Simulation modeling frameworks for single-cycling and double-cycling strategies in container terminals,” *Journal of ETA Maritime Science*, vol. 12, pp. 319-331, Aug 2024.

[10] E. T. Yassen, M. Ayob, A. A. Jihad, and M. Z. A. Nazri, “A self-adaptation algorithm for quay crane scheduling at a container terminal,” *IAES International Journal of Artificial Intelligence*, vol. 10, pp. 919-929, Dec 2021.

[11] Y. Zhu and A. Lim, “Crane scheduling with non-crossing constraint,” *Journal of the Operational Research Society*, vol. 57, pp. 1464-1471, Dec 2006.

[12] J. Liu, Y. W. Wan, and L. Wang, “Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures,” *Naval Research Logistics (NRL)*, vol. 53, pp. 60-74, Feb 2006.

[13] A. Skaf, S. Lamrous, Z. Hammoudan, and M. A. Manier, “Solving methods for the quay crane scheduling problem at port of Tripoli-Lebanon,” *RAIRO-Operations Research*, vol. 55, pp. 115-133, Mar 2021.

[14] R. Pérez-Rodríguez, “A hybrid estimation of distribution algorithm for the quay crane scheduling problem,” *Mathematical and Computational Applications*, vol. 26, pp. 64, Sep 2021.

[15] H. Li and X. Li, “A branch-and-bound algorithm for the bi-objective quay crane scheduling problem based on efficiency and energy,” *Mathematics*, vol. 10, pp. 4705, Dec 2022.

[16] H. Zheng, Z. Wang, and H. Liu, “The integrated rescheduling problem of berth allocation and quay crane assignment with uncertainty,” *Processes*, vol. 11, pp. 522, Feb 2023.

[17] N. Al-Dhaheri and A. Diabat, “The quay crane scheduling problem,” *Journal of Manufacturing Systems*, vol. 36, pp. 87-94, Aug 2015.

[18] D. Steenken, S. Voß, and R. Stahlbock, “Container terminal operation and operations research - a classification and literature review,” *OR Spectrum*, vol. 26, pp. 3-49, Jan 2004.

[19] C. F. Daganzo, “The crane scheduling problem,” *Transportation Research Part B: Methodological*, vol. 23, pp. 159-175, 1989.

[20] R. I. Peterkofsky and C. F. Daganzo, “A branch and bound solution method for the crane scheduling problem,” *Transportation Research Part B: Methodological*, vol. 24, pp. 159-172, Jun 1990.

[21] K. H. Kim and Y. M. Park, “A crane scheduling method for port container terminals,” *European Journal of Operational Research*, vol. 156, pp. 752-768, Feb 2004.

[22] A. Lim, B. Rodrigues, F. Xiao, and Y. Zhu, “Crane scheduling with spatial constraints,” *Naval Research Logistics (NRL),* vol. 51, pp. 386-406, 2004.

[23] L. Moccia, J. F. Cordeau, M. Gaudioso, and G. Laporte, “A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal,” *Naval Research Logistics (NRL),* vol. 53, pp. 45-59, 2006.

[24] M. Sammarra, J. F. Cordeau, G. Laporte, and M. F. Monaco, “A tabu search heuristic for the quay crane scheduling problem,” *Journal of Scheduling*, vol. 10, pp. 327-336, Aug 2007.

[25] C. Bierwirth, and F. Meisel, “A fast heuristic for quay crane scheduling with interference constraints,” *Journal of Scheduling,* vol. 12, pp. 345-360, Feb 2009.

[26] F. Meisel, and C. Bierwirth, “A unified approach for the evaluation of quay crane scheduling models and algorithms,” *Computers & Operations Research*, vol. 38, pp. 683-693, Mar 2011.

[27] M. F. Monaco and M. Sammarra, “Quay crane scheduling with time windows, one-way and spatial constraints,” *International Journal of Shipping and Transport Logistics,* vol. 3, pp. 454-474, Jul 2011.

[28] S. H. Chung and F. T. Chan, “A workload balancing genetic algorithm for the quay crane scheduling problem,” *International Journal of Production Research*, vol. 51, pp. 4820-4834, 2013.

[29] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, “Hybrid evolutionary computation methods for quay crane scheduling problems,” *Computers & Operations Research*, vol. 40, pp. 2083-2093, Aug 2013.

[30] T. Wang, M. Li, and H. Hu, “Berth allocation and quay crane-yard truck assignment considering carbon emissions in port area,” *International Journal of Shipping and Transport Logistics*, vol. 11, pp. 216-242, Apr 2019.

[31] A. Nourmohammadzadeh and S. Voß, “A robust multi objective model for the integrated berth and quay crane scheduling problem at seaside container terminals,” *Annals of Mathematics and Artificial Intelligence*, vol. 90, pp. 831-853, 2022.

[32] G. Tasoglu and G. Yildiz, “Simulated annealing based simulation optimization method for solving integrated berth allocation and quay crane scheduling problems,” *Simulation Modelling Practice and Theory,* vol. 97, Jul 2019.

[33] Q. Fu and C. Cai, “*Berth and quay crane scheduling optimization based on improved genetic algorithm,*” in 2021 China Automation Congress (CAC), pp. 2138-2142.

[34] X. Jiao, F. Zheng, M. Liu, and Y. Xu, “Integrated berth allocation and time-variant quay crane scheduling with tidal impact in approach channel,” *Discrete Dynamics in Nature and Society*, vol. 2018, no. 1, 9097047, 2018.

[35] A. Malekahmadi, M. Alinaghian, S. R. Hejazi, and M. A. A. Saidipour, “Integrated continuous berth allocation and quay crane assignment and scheduling problem with time-dependent physical constraints in container terminals,” *Computers & Industrial Engineering,* vol. 147, 106672, Jul 2020.

[36] M. Tang, B. Ji, X. Fang, and S. S. Yu, “Discretization-strategy-based solution for berth allocation and quay crane assignment problem,” *Journal of Marine Science and Engineering*, vol. 10, 495, Apr 2022.

[37] L. Tang, J. Zhao, and J. Liu, "Modeling and solution of the joint quay crane and truck scheduling problem," *European Journal of Operational Research*, vol. 236, pp. 978-990, Aug 2014.

[38] Y. Zheng, M. Xu, Z. Wang, and Y. Xiao, "A genetic algorithm for integrated scheduling of container handling systems at container terminals from a low-carbon operations perspective," *Sustainability,* vol. 15, Mar 2023.

[39] D. C. Hop, N. Van Hop, and T. T. M. Anh, "Adaptive particle swarm optimization for integrated quay crane and yard truck scheduling problem," *Computers & Industrial Engineering*, vol. 153, pp. 107075, Mar 2021.

[40] S. Ma, H. Li, N. Zhu, and C. Fu, "Stochastic programming approach for unidirectional quay crane scheduling problem with uncertainty," *Journal of Scheduling*, vol. 24, pp. 137-174, Apr 2021.

[41] F. Rodrigues and A. Agra, "Handling uncertainty in the quay crane scheduling problem: a unified distributionally robust decision model," *International Transactions in Operational Research*, vol. 31, pp. 721-748, Mar 2024.

[42] M. Jiang, J. Zhou, J. Feng, L. Zhou, F. Ma, and G. Wu, "Integrated berth and crane scheduling problem considering crane coverage in multi-terminal tidal ports under uncertainty," *Journal of Marine Science and Engineering*, vol. 10, pp. 506, Apr 2022.

[43] K. Chargui, T. Zouadi, V. R. Sreedharan, A. El Fallahi, and M. Reghioui, "A novel robust exact decomposition algorithm for berth and quay crane allocation and scheduling problem considering uncertainty and energy efficiency," *Omega*, vol. 118, pp. 102868, Jul 2023.

[44] E. Lujan, E. Vergara, J. Rodriguez-Melquiades, M. Jiménez-Carrión, C. Sabino-Escobar, and F. Gutierrez, "A fuzzy optimization model for the berth allocation problem and quay crane allocation problem (BAP + QCAP) with n quays," *Journal of Marine Science and Engineering,* vol. 9, pp. 152, Feb 2021.

[45] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation,* vol. 76, pp. 60-68, 2001.

[46] F. Qin, et al. "Hybrid harmony search algorithm integrating differential evolution and lévy flight for engineering optimization," *IEEE Access,* 2025.

[47] A. Ala'a, A. A. Alsewari, H. S. Alamri, and K. Z. Zamli, "Comprehensive review of the development of the harmony search algorithm and its applications," *IEEE Access,* vol. 7, pp. 14233-14245, Jan 2019.

[48] M. Dubey, V. Kumar, M. Kaur, and T. P. Dao, "A systematic review on harmony search algorithm: theory, literature, and applications," *Mathematical Problems in Engineering,* vol. 2021, 5594267, 2021.

[49] F. Qin, A. M. Zain, and K. Q. Zhou, "Harmony search algorithm and related variants: a systematic review," *Swarm and Evolutionary Computation,* vol. 74, 101126, Oct 2022.

[50] D. Kizilay and D. T. Eliiyi, "A comprehensive review of quay crane scheduling, yard operations and integrations thereof in container terminals," *Flexible Services and Manufacturing Journal,* vol. 33, pp. 1-42, Mar 2021.

[51] A. Skaf, "*Planification of logistics activities in ports*," 2020.