Journal of
Marine Science and Engineering

MDPI

Article

A Hybrid Meta-Heuristic Approach for Solving Single-Vessel Quay Crane Scheduling with Double-Cycling

Fahrettin Eldemir [1,*] and Mustafa Egemen Taner [2]

[1] Department of Industrial and Systems Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia
[2] Department of Industrial Engineering, Tarsus University, Mersin 33402, Turkey; metaner@tarsus.edu.tr
* Correspondence: eldemir@uj.edu.sa

Abstract: The escalating global demand for containerized cargo has intensified pressure on container terminals, which serve as vital nodes in maritime logistics. This study aims to enhance operational efficiency in non-automated container terminals by examining two meta-heuristic approaches—Ant Colony Optimization (ACO) and a hybrid Greedy Randomized Adaptive Search Procedure (GRASP)—Genetic Algorithm (GA)—for quay crane scheduling. Their performance is benchmarked across various problem scales, with process completion time serving as the primary metric. Based on these findings, the most effective approach is integrated into a newly developed Decision Support System (DSS) to streamline practical implementation. Statistical analyses confirm the robustness of both methods, underscoring how meta-heuristics combined with a DSS can optimize quay crane utilization, bolster maritime logistics, and ultimately boost terminal productivity.

Keywords: port management; maritime transport; container terminals; quay crane scheduling; meta-heuristics; GRASP-GA; ant colony optimization; decision support system

check for updates

Academic Editors: Livia Maglić, Lovro Maglić, Ana Peric Hadzic, Marina Zanne and Nam Kyu Park

Received: 1 January 2025
Revised: 30 January 2025
Accepted: 5 February 2025
Published: 17 February 2025

Citation: Eldemir, F.; Taner, M.E. A Hybrid Meta-Heuristic Approach for Solving Single-Vessel Quay Crane Scheduling with Double-Cycling. *J. Mar. Sci. Eng.* **2025**, *13*, 371. https://doi.org/10.3390/jmse13020371

1. Introduction

Container transport has become one of the most prominent methods in maritime shipping, a sector that accounts for approximately 90% of global trade. Standardized containers enable reliable, cost-effective, and efficient solutions by expediting the intercontinental flow of goods. Recent data from the Review of Maritime Transport 2022 published by the United Nations Conference on Trade and Development (UNCTAD) indicate that containerized cargo volume has increased by more than 50% over the last decade, reflecting the ongoing expansion in global trade [1]. In 2023, worldwide container traffic grew by an additional 3.5%, pushing the total handled volume to nearly 865 million TEUs (twenty-foot equivalent units). This momentum underscores the effectiveness of port operations and highlights the critical role of capacity expansion. Projections for 2024 suggest that this upward trend will continue with an estimated 4% growth rate.

Ongoing globalization has transformed logistics networks by elevating ports to strategic hubs within the global supply chain. Container transport, in particular, has facilitated the rapid integration of economies into worldwide markets. However, surging trade volumes place considerable strain on existing port infrastructures, and it often leads to operational bottlenecks. Over the past decade, developing economies have emerged as major participants in global trade; notably, ports in China, Singapore, and the Republic of Korea stand out due to their high annual container throughput. This economic dynamism has propelled the prominence of Asian ports, which now handle approximately 60% of the world's container traffic.

Container shipping is indispensable for the sustainable growth of global trade. The efficiency and optimization of port operations are vital for meeting the increasing transportation demand. It is anticipated that port automation systems, artificial intelligence-supported scheduling algorithms, and infrastructure investments—particularly those focused on green transformation—will further drive sectoral growth.

Port efficiency directly affects the operational success of container transportation. Quay cranes (QCs) have played a critical role in increasing operational efficiency by shortening the loading and unloading times of vessels in ports. Quay Crane Scheduling Problems (QCSP) are one of the most important optimization problems that determine port performance. The effectiveness of these operations in ports helps minimize waiting times and reduce cargo handling costs.

Since optimization algorithms and decision support system (DSS) applications are rapidly evolving, recent advancements in these areas are particularly noteworthy. developments, extensively explored in scientific literature, are examined in two separate groups to highlight their innovative aspects. Therefore, the literature review is structured into two main sections: Sections 1.1 and 1.2.

1.1. Previous Studies (Pre-2018)

In Previous Studies, the literature on container terminal operations primarily focuses on three main areas: quay operations, yard operations, and transport operations. Within quay operations, two key decision problems have been studied extensively: QCSP and the integrated berth allocation and crane scheduling problem.

Research on QC scheduling has advanced significantly with a focus on optimizing crane operations under realistic constraints. Kim and Park [2] introduced a mixed-integer programming model integrated with the Greedy Randomized Adaptive Search Procedure (GRASP) heuristic which effectively handled challenges like non-crossing cranes and initial crane positions. This approach demonstrated superior performance, particularly for large-scale problems. Similarly, Lee et al. [3] proposed a genetic algorithm (GA) for QCSP with prioritizing computational efficiency by outperforming CPLEX, though inter-bay travel times were not considered. Bierwirth and Meisel [4] introduced a unidirectional scheduling heuristic that made the search process in the solution space effective in terms of solution quality and computation time. Building on practical constraints, Legato et al. [5] developed a hybrid approach accommodating factors such as safety gaps and setup times and obtained superior results by comparing them with the unidirectional schedule (UDS) method in the literature. Subsequently, Meisel [6] extended the UDS by incorporating time-windowed scheduling for multiple vessels and cranes, while Chung and Choy [7] refined GA methods through minimizing crane travel times and balancing workloads. Another significant research focus is the joint consideration of berth allocation and QC scheduling, and these operations are inherent interdependence. In this context, Park and Kim [8] pioneered a mixed-integer programming model enhanced by Lagrangian relaxation. Following this, Moccia et al. [9] proposed a methodology based on a hierarchical branch-and-cut algorithm wherein berth allocation preceded crane scheduling to improve complex terminal decision-making. Liang et al. [10] then put forward a GA-based approach that underscored the benefits of vessel-based crane allocation across diverse scenarios. In order to tackle the uncertainties and disruptions, Zeng et al. [11] merged simulation-optimization techniques with a mixed-integer programming framework enriched by tabu search and localized rescheduling. These studies have laid a strong foundation for solving QCSP and integrated berth-crane scheduling problems through heuristic and meta-heuristic methods, including GA, tabu search, and Lagrangian relaxation. However, limitations such as computational efficiency for large-scale problems, real-time applicability, and integration into decision support systems continue to pose challenges for improvement. More recent studies by Iris et al. [12,13] proposed advanced formulations and solution techniques for integrated berth allocation and quay crane assignment, incorporating novel set partitioning approaches, variable fixing, and adaptive large neighborhood search. These studies achieved improved bounds and solution quality across a variety of instances, reflecting the growing trend toward holistic port operation planning.

The literature on yard operations in container terminals addresses four primary decision problems: yard layout and equipment selection, block allocation, equipment assignment and scheduling, and reshuffling operations. Each of these solutions has contributed significantly to terminal efficiency and productivity. Regarding yard operations, the design of yard layouts and the equipment choice profoundly affect the investment costs and container capacity. Carlo et al. [14] and Taner et al. [15] emphasized the importance of integrating block configuration with equipment performance in this context. Wiese et al. [16] and Lee and Kim [17] investigated how block dimensions, stacking heights, and equipment types affect yard performance with the help of simulation and mathematical models. As automated stacking cranes became prevalent in some European terminals, the researchers' attention shifted to optimizing block configurations and reducing reshuffling. Focusing on efficient block allocation, Ng et al. [18] and Park et al. [19] proposed hierarchical and integrated approaches to minimize cycle times and improve the container flow. Tailored allocation strategies for specialized containers, as explored by Fu et al. [20] and Guldogan [21], further boosted yard utilization. Similarly, in equipment assignment and scheduling studies, Petering et al. [22] showcased the productivity gained from prioritizing export containers. Stahlbock and Voss [23] demonstrated how automated European terminals can improve crane scheduling and minimize delays. Reshuffling operations, a critical cost factor, were addressed by Kim and Hong [24] and Yang and Kim [25] through optimization models, while Hirashima et al. [26] introduced Q-learning-based heuristics to reduce reshuffling through pre-marshaling strategies. Advanced techniques like remarshaling and housekeeping, as analyzed by Choe et al. [27], demonstrated significant efficiency gains in twin-crane systems. Despite such substantial progress in yard operations, early studies often lacked integration across terminal functions. Although the automated systems excelled in European contexts, the cost and adaptability challenges persisted in many Asian terminals. Recent advancements in hybrid and meta-heuristic methods, such as GA and tabu search, have offered promising solutions to complex operational challenges, paving the way for comprehensive decision-support systems. However, holistic approaches that integrate yard and quay operations remain scarce, indicating an opportunity for more comprehensive DSS frameworks.

Beyond yard operations, transportation between the quay and yard in container terminals relies heavily on the type of equipment available, with automation levels shaping the choice of vehicles. While automated guided vehicles (AGVs) are prevalent in highly automated terminals, truck-based systems are more common in fill-type terminals, such as those in Turkey. Key research contributions include Ng et al.'s [28] GA for truck fleet scheduling, which improved performance by accommodating varying preparation and processing times. Cao et al. [29] advanced this field by integrating truck and yard crane scheduling through Bender's decomposition methods, which yielded effective results for large-scale scenarios in Keppel Port. Petering [30] focused on real-time truck assignments in twin-loading terminals, exploring strategies like prioritizing empty trucks and cross-docking to optimize QC usage. In the context of Turkish terminals, Kulak et al. [31] utilized simulation techniques to identify efficiency gains through better resource allocation. Collectively, these studies underscore the significance of effective truck scheduling and resource integration, yet challenges persist in addressing dynamic conditions—particularly under high traffic variability and short lead times—and synchronizing operations across multiple resources to achieve optimal terminal performance.

Integrated approaches to terminal operations strive to optimize the interplay between QCs, yard cranes, and transport vehicles, recognizing their interdependent roles in overall terminal efficiency. Among notable contributions, Chen et al. [32] proposed a hybrid flow shop scheduling model that effectively managed setup times and precedence constraints, leveraging tabu search-based heuristics for robust outcomes. Zeng and Yang [33] integrated GA with neural networks, enhancing QC sequence optimization and showcasing the promise of hybrid methodologies. Homayouni and Tang [34] developed a multi-objective model aimed at minimizing vehicle travel times and crane delays, achieving superior results through GA-based solutions. Lu and Le [35] investigated uncertainties in scheduling using particle swarm optimization in order to understand adaptability and resilience in complex environments. These findings highlight the potential of integrated DSS to coordinate the intricate relationships within terminal operations. Nevertheless, scalability and real-time feasibility remain pressing issues, and they imply a need for further innovation in this domain. As a result, real-time DSS solutions underpinned by robust scheduling algorithms continue to be an active area of study, especially for large-scale or time-sensitive scenarios.

1.2. Recent Advances (2018–2024)

In line with the aforementioned integrated approaches, recent advances in both optimization algorithms and DSS design have significantly impacted container terminal research, especially with respect to meta-heuristic strategies and QC scheduling. Between 2018 and 2024, researchers have paid more attention to improving algorithms to meet the challenges of dynamic and real-time scheduling contexts. In addition, hybrid heuristic solutions and new real-time scheduling methods were developed to increase operational efficiency. At the same time, optimization modules were integrated into DSS to provide intelligent, user-oriented systems that support terminal operators. Therefore, the expanded consideration of quay, yard, and transport operations has contributed greatly to a more holistic understanding of container terminal processes. In this context, the review of post-2018 studies will highlight these recent contributions, which reflect the critical importance of innovation in meeting the evolving needs of container terminals.

The increasingly complex challenges in container terminal operations are addressed by recent advances in QC scheduling and meta-heuristic algorithms. For instance, Iris et al. [36] expanded integrated QC assignment to include container loading operations and transfer vehicle assignments, illustrating the cost-reduction potential of a truly holistic planning framework. These approaches highlight the growing interest in concurrent decision-making for multiple terminal resources—a trend our work follows by incorporating double-cycling and real-time crane travel times. Additionally, Safaeian et al. [37] integrated berth and QC scheduling using a teaching-learning-based optimization algorithm, demonstrating its effectiveness for large-scale problems in Shahid Rajaee Port, Iran. Regarding multi-objective scheduling under uncertainty, Nourmohammadzadeh and Voss [38] developed a multi-objective model for berth and QC scheduling, incorporating uncertainties in vessel arrivals and crane availability. Their Pareto-simulated annealing approach outperformed alternative methods, offering robust multi-objective solutions.

In a broader context, hybrid and simulation-based frameworks have become pivotal. Behjat and Nahavandi [39] proposed a grouping-based imperialist competitive algorithm for integrated QC and yard truck scheduling. The results showed the superiority of their proposed algorithm over conventional algorithms. On the other hand, Hsu et al. [40] developed an integrated scheduling framework that includes vessel stowage plans. This framework included methods such as GA, particle swarm optimization (PSO), and multiple-group PSO (MGPSO). The findings showed the role of MGPSO in minimizing the makespan.

Meanwhile, Hsu and Wang [41] focused on berth allocation and QC scheduling using multiple particle swarm optimization. Their results further demonstrated the potential for cost reduction and resolution of vessel overlaps. Xin et al. [42] explored the energy-efficient scheduling for automated container terminals. A customized GA was used to achieve substantial energy savings without compromising the operational efficiency. Jiang et al. [43] studied combined berth and QC scheduling under uncertainty and proposed a carbon-tax-inclusive optimization model. An adaptive spiral flying dung beetle optimization algorithm was used to solve the problem. Their work did not only reduce operational costs but also contributed to a significant reduction in carbon emissions. In parallel, hybrid approaches utilizing multiple optimization strategies have garnered more attention. For example, Hsu et al. [44] developed a hybrid whale optimization algorithm combined with PSO to schedule automated QCs, automated lift vehicles, and automated stacking cranes. The proposed framework offered a balanced solution for efficiency and energy use.

Ant colony optimization (ACO) and its variants have also proven valuable in integrating various resource scheduling jobs at container terminals. Yang and Jiang [45] combined GA and ACO with nested partition frameworks to coordinate the yard crane activities in ground trolley-based automated container terminals. The methodology produced robust results for intricate coordination jobs. Wang et al. [46] introduced a pre-selection-based ant colony system for managing resources such as berth allocations and handling speeds and demonstrated advantages over existing state-of-the-art techniques.

Beyond these developments, hybrid ACO solutions have emerged. Rouky et al. [47] devised a hybrid ACO algorithm paired with variable neighborhood descent to handle uncertain QC scheduling and effectively manage stochastic conditions. This highlighted an ongoing trend toward applying ACO-based methods to ever more complex and unpredictable terminal contexts. Similarly, Li et al. [48] advanced the field by proposing quantum ant colony optimization for AGV path planning, exploiting quantum mechanics to enhance search performance and avoid conflicts.

The broader incorporation of DSS in container terminal operations has also shown considerable promise. Castilla-Rodríguez et al. [49] applied hybrid estimation of distribution algorithms within simulation to bolster QC scheduling under uncertain scenarios, enabling more solid evaluations of alternative solutions. Klar et al. [50] and Ding et al. [51] investigated digital twin (DT)-based frameworks for real-time operational decision-making and resource management, and the transformative role of DTs in promoting autonomous and efficient port operations was emphasized. Wang et al. [52] used DT models in conjunction with neural networks to optimize the handling of hazardous cargo and offered improvements in predictive accuracy and emergency response. Meanwhile, Abril et al. [53] developed an event-driven DSS for smart port operations combining real-time optimization capabilities with heightened adaptability and planning. Most recently, Bilican et al. [54] introduced a bi-level programming-based DSS that integrates sustainability and revenue maximization in container shipping focusing on cargo composition and stowage planning. This multi-stage approach further underscored how advanced decision-making can enhance maritime logistics.

Across the scientific literature, GA and ACO have attracted attention for their ability to tackle domain-specific challenges and craft adapted solutions in quay-side operations. Their successful use in areas like energy efficiency and sustainability underlines their versatility and value. However, their limitations are evident in addressing dynamic conditions, real-time scenarios, and large-scale problems. To address these gaps, this study develops hybrid meta-heuristic approaches, further integrating them into terminal operations to push the boundaries of current literature. In particular, the study differs from existing research by modeling the QC operations with realistic considerations, including constraints such as non-crossing cranes, safety gaps, job precedence, and horizontal movement velocities. Additionally, it aims to increase intra-vessel reshuffling rates and implement a double-cycling strategy to offer a unique perspective compared to prior studies. By proposing a DSS that ensures integrated coordination of terminal resources (QCs, yard cranes, transport vehicles, and gates), our approach seeks to minimize waiting times and optimize resource utilization, thus offering a comprehensive framework for real-world terminal operations.

This study is organized as follows: Section 2 defines the QCSP and provides the conceptual framework for the constraints and operational assumptions. Section 3 outlines the methodology by detailing the application of GA and ACO for solving the problem. Section 4 includes numerical experiments followed by a discussion on the performance of the proposed methods through benchmark problems. Section 5 introduces the proposed DSS structure and highlights its architecture, user interface, and practical contributions. Finally, Section 6 concludes the study with a summary of findings, originality, limitations, and suggestions for future research.

2. Problem Definition

This section outlines the conceptual framework devised for addressing the QCSP in container terminals. To begin, we characterize the QCSP and underscore its principal parameters, constraints, and operational hurdles specific to terminal operations. Rather than providing a full mathematical model, how these elements shape the scheduling decisions and prepare the ground for our methodological and experimental discussions that follow is emphasized. Given these complexities, the subsequent sections will detail our meta-heuristic approach and experimental design which bridges the gap between conceptual insights and practical solution strategies.

Container terminals are complicated systems intended to facilitate the efficient transfer of standardized container units between vessels and storage zones. Generally, these terminals comprise three main segments: the quay area, the yard area, and the transport area (see Figure 1). The present focus is on the QCSP, which deals with pivotal operational constraints, such as crane interference and safety clearances, to optimize loading and unloading processes across these zones.

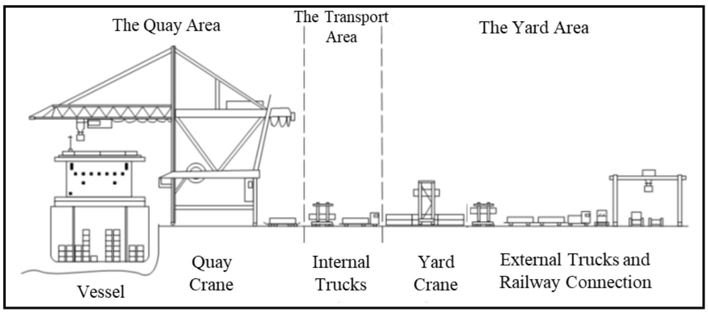


Figure 1. Container terminal operations and operation areas.

Within the quay area, loading and unloading activities take place via QCs, which shuttle containers between vessels and transport vehicles. These cranes, as illustrated in Figure 2, operate on fixed rails parallel to the vessel, sharing a common track. This configuration imposes critical constraints, such as non-crossing cranes and safety gaps, which must be accounted for in the QCSP. The yard area comprises stacking blocks classified for export, import, and empty containers. Yard cranes facilitate stacking operations, enabling up to seven tiers of containers. The transport area between the quay and the yard involves equipment such as AGVs and large-wheel trucks, which manage horizontal and vertical container movements. Efficient transport operations significantly impact vessel turnaround time and terminal productivity.

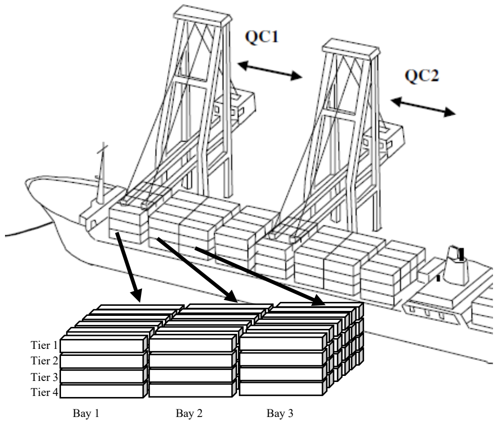
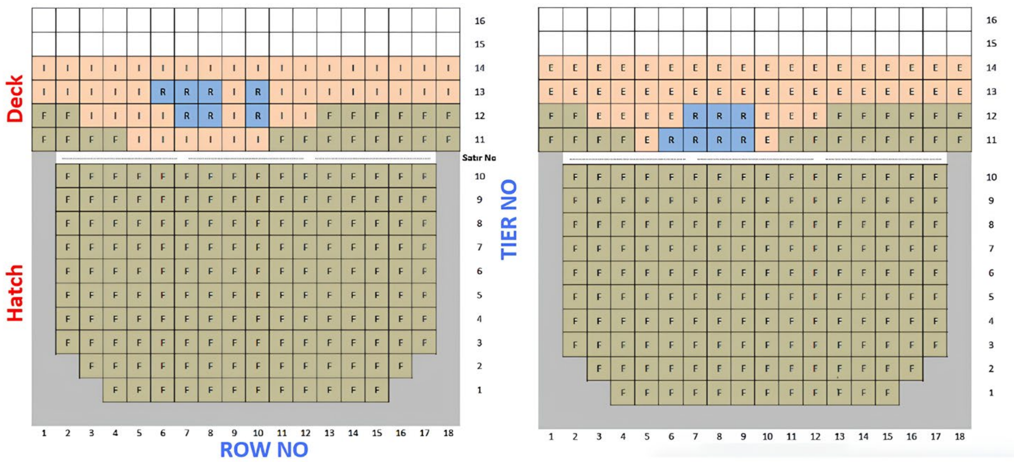


Figure 2. Vessel loading/unloading and QCs.

Containers on a vessel are organized into bays, with each bay further divided into rows and tiers, as depicted in Figure 3. The position of each container is determined by its bay, row, and tier, and is uniquely coded for precise identification. For example, a container labeled "100110" represents its placement in Bay 10, Row 01, and Tier 10. This arrangement facilitates efficient handling during both loading and unloading operations. The incoming load plan identifies containers to be offloaded at the current port, while the outgoing load plan specifies the placement of new containers to optimize subsequent operations.



(**a**) (**b**)

Figure 3. Stowage plans: (**a**) arrival stowage plan of a bay; (**b**) departure stowage plan of a bay.

Figure 3 also illustrates the dynamic nature of these stowage plans, where containers are categorized into three groups during the discharge phase: containers to be unloaded at the current port (I), containers remaining in place (F), and containers requiring reshuffling (R) to access others beneath them. During the loading phase, new containers (E) are added while existing containers are reorganized to align with the next port's requirements. This intricate coordination of loading and unloading is a central aspect of QCSP, as it directly affects the operational efficiency of QCs. Hence, this requirement constitutes a core challenge in the QCSP, as it directly affects the QC efficiency.

In the literature, QCSPs were often categorized based on criteria such as single vs. multiple vessels, single vs. multiple cranes, the presence of double-cycling, non-crossing constraints, or time-windowed operations Kim and Park [2]; Legato et al. [5]; etc. Following these established taxonomies, the problem in this study can be described as a single-vessel QCSP with multiple quay cranes operating under non-crossing constraints, incorporating double-cycling (simultaneous unloading and loading within one cycle). Additionally, a real-time (distance/velocity) perspective is adopted for crane travel which places our variant in the category of dynamic or real-time QCSP. This ensures that any crane's travel duration dynamically reflects the operational environment by leading to a scheduling plan that accounts for realistic vessel configurations and crane actions. Consequently, the objectives aim to minimize the makespan (i.e., overall completion time) by focusing on whichever QC finishes last and to minimize extra handling times or re-handlings by ensuring efficient crane usage across the vessel's bays. These two goals can be combined or balanced in a multi-objective context, capturing both operational throughput and resource utilization.

Specifically, the first objective (Z_1) seeks to minimize the finishing time of the slowest or last-finishing quay crane. Let S_q represent the initial time of crane q for its first job, and F_q the completion time for its final job; thus, the relevant processing duration (PTQ_q) indicates how long crane q remains active. The overall objective is to reduce max(PTQ_q). The second objective (Z_2) aims to minimize the extra handling time (QR) across all quay cranes, which accounts for distance/velocity ratios in different axes and includes additional moves or re-handlings performed on the vessel.

Prior research on the QCSP includes the foundational mathematical model introduced by Meissel and Wichmann [55]. Building on that framework, Liu et al. [56] extended the objective function to accommodate alternative processing and setup times under a single assumption. In this study, we have further adapted the objective function to achieve real-time solutions by deriving crane operation times from a distance/velocity relationship. This modification ensures that processing durations accurately reflect the dynamic nature of crane travel and handling activities. Though previous studies have proposed formal MIP models, we opt here for a more conceptual definition of the problem to better align with our heuristic/meta-heuristic solution strategy, which emphasizes practical deployment within a DSS.

The proposed approach draws inspiration from the mathematical modeling tradition while adapting certain aspects to real-world constraints. By deriving crane operation times from a distance/velocity relationship, each job's handling duration is refined beyond the static or fixed-time assumptions found in previous studies. Meanwhile, double-cycling is integrated so that, whenever feasible, the same crane can handle both unloading and loading within a single cycle, thereby potentially reducing the total makespan. Lastly, focusing on a single vessel per scheduling horizon removes the complexities of berth allocation or multiple vessels and allows us to emphasize the crane scheduling dimension.

In this study, we examine the scheduling and assignment of the QCs—taking into account the loading and unloading plans of arriving and departing vessels—as part of the QCSP. In addition, the QCSP is tailored to smaller and medium-sized fill-type terminals common in Turkey. Unlike larger ports, these terminals typically feature fixed berth-to-crane assignments, eliminating berth allocation issues. Therefore, integrated loading and unloading processes are handled by optimizing crane movements to minimize conflicts and improve operational efficiency. The problem setup includes a mathematical representation of crane operations and their interaction with container handling plans, as illustrated by vessel-specific load and discharge configurations.

3. Methodology

This section describes the methodologies applied to solve the QCSP which is a combinatorial optimization problem categorized as NP-hard. The study explores three approaches: a classical scheduling method, Ant Colony Optimization (ACO), and a hybrid method combining Greedy Randomized Adaptive Search Procedure (GRASP) and Genetic Algorithm (GA). The first method, referred to as Method 1, applies a classical scheduling approach by organizing jobs on a bay basis using parallel machine scheduling principles. Method 2 employs ACO to address the sequencing and scheduling problem by leveraging its similarity to the vehicle routing problem, where ACO has proven to be highly effective. Lastly, Method 3 combines GRASP for solving the sequencing problem within bays and GA for scheduling the order of bays assigned to QCs by utilizing the strengths of GA as one of the most effective algorithms for scheduling problems. Each method is detailed with example problem instances in Section 3.1, Section 3.2, and Section 3.3, respectively.

3.1. Method 1: Classical Scheduling Approach

This method treats the QCSP as a parallel machine scheduling problem building on the notion that each bay can be seen as a discrete job. Each bay on the vessel is considered an independent and indivisible job, while QCs are modeled as machines. The approach incorporates critical constraints, such as a one-bay safety distance between cranes and restrictions preventing cranes from crossing over each other. Additionally, the initial positions of the cranes are assumed to be at opposite ends of the vessel.

The steps of this classical scheduling method are as follows:

1. Job Sorting: All jobs (i) are sorted by their corresponding bay numbers (j) in ascending order;
2. Bay Time Calculation: The total operation time for each bay (C_j) is calculated by summing the operation times (C_{ij}) of individual jobs within the bay item;

$$C_j = \sum_{i=1}^{n} C_{ij}, \quad \forall j \in [1, m] \tag{1}$$

3. Initial Scheduling: Jobs are assigned to QC_1 starting from the first bay (j_1 = 1) and to QC_2 starting from the last bay (j_2 = m) adhering to the defined constraints;
4. Conflict Resolution: When the cranes are one bay apart ($|j_1 - j_2|$ = 1), the crane with the higher remaining total bay operation time (C_j) is prioritized for the next job;
5. Makespan Calculation: The makespan, or the total operation time, is determined as the maximum of the individual crane completion times.

To illustrate this method, we consider a vessel with 25 bays (each containing import, export, and reshuffling operations), served by two QCs. A total of 40 jobs are distributed across these bays, and the cranes start from opposite ends of the vessel. The scheduling process produced the results shown in Table 1. For instance, at time unit 727, QC_1 completed operations in bay 13, while QC_2 finished bay 15. Because cranes cannot operate adjacent to each other (safety gap constraint), QC_2 then moved to bay 16. Meanwhile, QC_1 proceeded with bay 14. The final makespan was 733 time units, corresponding to the completion time of QC_1, as it is the longest operation. Hence, any improvement in QC_1's schedule would potentially lower the overall makespan.

In summary, Method 1 effectively schedules bays as independent jobs, although crane adjacency constraints can lead to idle times when switching between distant bays. Next, we explore ACO in Section 3.2, aiming to handle the problem's combinatorial complexity more flexibly.

Table 1. Scheduling results.

Bay	Operation Time	Completion Time of QC_1	Completion Time of QC_2	Bay	Operation Time	Completion Time of QC_1	Completion Time of QC_2
1	14	14	1460	14	5	733	732
2	31	46	1445	15	22	756	726
3	138	185	1413	16	7	764	703
4	11	197	1274	17	117	882	695
5	11	209	1262	18	146	1029	577
6	64	274	1250	19	325	1355	430
7	108	383	1185	20	22	1378	104
8	14	398	1076	21	5	1384	81
9	13	412	1061	22	42	1427	75
10	24	437	1047	23	15	1443	32
11	272	710	1022	24	-	-	-
12	10	721	749	25	15	1460	15
13	5	727	738				

3.2. Method 2: Ant Colony Optimization (ACO)

This method utilizes ACO to address QCSP. ACO is a swarm intelligence algorithm inspired by the foraging behavior of ants. It has been widely applied to various combinatorial optimization problems including scheduling, routing, and job assignment by exploiting how ants deposit and follow pheromone trails [57]. ACO mimics the cooperative behavior of ants in their natural environment, particularly their ability to find the shortest paths between their nests and food sources. Ants communicate indirectly by depositing a chemical substance called pheromone along their paths. The intensity of the pheromone influences the probability of other ants following the same path, effectively reinforcing optimal solutions. Over time, paths with higher pheromone concentrations are more likely to be selected, allowing the algorithm to converge towards the shortest route.

Key features of ACO involve initializing a pheromone matrix, which represents the attractiveness of paths between jobs. During the solution construction phase, artificial ants iteratively build solutions by utilizing pheromone levels and heuristic information to guide their choices. Finally, the pheromone trails are updated through a two-step process: evaporation, which prevents premature convergence, and reinforcement, which strengthens paths associated with high-quality solutions.

The algorithm is implemented to minimize the makespan by utilizing the flexibility of pheromone-based optimization and heuristic guidance by following the specific parameters defined for the QCSP. It has been applied to the previously described QCSP scenario involving 40 jobs across 25 bays and two QCs. The key parameters are:

- Number of ants: 50;
- $\Delta = 1$ (pheromone reinforcement);
- $\rho = 0.2$ (pheromone evaporation rate).

The algorithm follows these steps:

1. Initialization
   a. Create a 42×42 pheromone matrix where each element is initialized to 1, representing equal attractiveness for all job paths.
2. Solution Construction (For each ant a from 1 to 50)
   a. Generate a random number u from a discrete uniform distribution $u \sim U$ [1,39].
   b. Partition the jobs into two sets: $n_1 = u$ jobs assigned to QC_1 and $n_2 = 40 - u$ jobs assigned to QC_2.

c. Form the unassigned job set S = [1,. . .40].
d. For QC_1 select n_1 jobs using pheromone-based probabilities to construct the job list W_1, and update S by removing selected jobs.
e. Similarly, construct W_2 for QC_2 using the remaining jobs in S.
f. Combine W_1 and W_2 to form $W_a = W_1 \cup W_2$.
g. Compute the completion time TC_a for the constructed solution.

3. Pheromone Update
   a. If $TC_a < TC_{min}$, update $W_{min} = W_a$ and $TC_{min} = TC_a$.
   b. Update the pheromone matrix. For paths in the best solution, increase pheromone levels by $\Delta = 1$. For all other paths, reduce pheromone levels by a factor of $(1 - \rho)$ where $\rho = 0.2$.
4. Termination
   a. Check if the stopping criterion is met (a predefined number of iterations or convergence of solutions).
   b. If the criterion is met, terminate the algorithm. Otherwise, return to Step 2.

The pheromone matrix reflects the desirability of paths based on cumulative results. At initialization, all matrix elements are equal, indicating no preference. After each iteration, the pheromone values associated with better solutions increase by a fixed amount (Δ) and other values decrease due to evaporation, promoting the exploration of new solutions and avoiding premature convergence. For instance, with Δ = 1 and ρ = 0.2, if the sequence 3-6-5-10 is part of the best solution, the corresponding matrix elements are doubled, while others are reduced to 80% of their previous values. Therefore, this process dynamically adjusts the likelihood of selecting specific paths in subsequent iterations. After 1816 iterations, the algorithm converged successfully, yielding a best solution with a makespan of 612. This underscores ACO's capacity to efficiently navigate the QCSP search space under the given constraints.

3.3. Method 3: Hybrid Greedy Randomized Adaptive Search Procedure (GRASP)—Genetic Algorithm (GA)

The QCSP is approached in two stages to use this method. The first stage has involved sequencing the QC operations within each bay of the vessel, determining the order of unloading containers, reshuffling operations for containers destined for other ports, and loading operations. This stage also calculates the total processing time for each bay. In the second stage, we use these total processing times to schedule the cranes and determine the vessel's makespan.

GRASP is a multi-start meta-heuristic algorithm developed in the late 1980s for solving complex combinatorial problems [58]. Each iteration consists of two main phases: construction and local search. GRASP has been widely applied to scheduling, quadratic assignment problems, personnel scheduling, machine and vehicle assignments, and layout problems.

To conduct loading and unloading operations with maximal efficiency, it is essential to increase the proportion of double cycling while simultaneously minimizing re-handling operations. By reducing vessel-based reshuffling as much as possible, this approach lowers both cost and time overhead. To achieve this goal, a GRASP strategy has been employed, one that integrates double cycling and endeavors to perform re-handling tasks onboard the vessel whenever feasible.

Double cycling refers to a quay crane strategy that ensures the crane remains loaded during each movement, whether it is engaged in loading or unloading. In other words, after finishing an unloading task, the crane does not travel empty to retrieve the next container; instead, it uses that transit to place a container scheduled for loading into its designated location on the vessel. The same principle applies in reverse for loading operations. The main steps of the double-cycling procedure can be summarized as follows:

1. Unload all designated import containers from the initial tier.
2. Simultaneously load export containers to their target positions, if feasible.
3. If an appropriate container is available during the return journey, avoid traveling empty and bring the container to its correct placement.

GRASP uses the vessel's load balance and container reshuffling within the vessel to achieve an effective sequence of operations, considering real-world constraints such as deck closures during hatch cover openings. The bay operation times from GRASP serve as inputs for the GA's scheduling stage. The algorithm proceeds as follows until the containers to be handled run out:

1. Check Hatch-Cover Status: If a hatch cover is about to be opened, skip to Step 3; otherwise, go to Step 2.
2. Determine Re-Handled Containers: Examine whether a re-handled container (labeled R) is present. If yes, proceed to Step 9; if no, advance to Step 3.
3. Identify Candidate Container List: Construct the candidate list (l_i^c) filtering out tasks that do not require further re-handling. These tasks form the set F (relevant to job j).
4. Apply Feasibility Equation: For each task, check compliance with Equation (2). Tasks meeting this criterion are excluded from set F.

$$1/\left|l_i^c - l_j\right| < r \cdot \max_{\mathrm{j} \in \mathrm{F}}\left\{1/\left|l_i^c - l_j\right|\right\} \tag{2}$$

5. Compute Probabilities: For tasks still in F, determine the probability of selecting each container for the next operation using Equation (3).

$$\frac{1}{\left|l_i^c - l_j\right|} / \sum\left(1/\left|l_i^c - l_j\right|\right) \tag{3}$$

6. Execute Selection: Generate a random number and apply a roulette-wheel procedure using probabilities from Step 4 to choose which container's tier to unload.
7. Unloading Process: Continue unloading the tier chosen in Step 6. If a hatch cover must be opened, proceed to Step 14; otherwise, go to Step 8.
8. Loading Export Containers (E): Once the chosen tier is emptied of import containers (I), begin loading any export containers (E) into that tier. Then return to Step 2 to identify another tier for unloading.
9. Obtain Current and Target Positions for (R): Gather the positions of all re-handled containers (R) and their intended destinations.
10. Check Target Position Below Current: If the R container's target is below its current location, select one tier to the left and right plus all upper cells of the current and target positions—omitting operations in other tiers—then proceed to Step 11.
11. Unload Import Containers (I): For the chosen region, remove all import containers (I). If the re-handled container's (R) target position lies below its current location, temporarily place it in the nearest empty tier and move to Step 12; otherwise, move it directly to the target position.
12. Clear and Transfer (R): Unload any import container (I) accessible within the same region. Then transfer the re-handled container (R) from the cleared area to its target location. Continue to Step 13.
13. Load Export Containers (E): Once all R containers are placed in their final positions, load any remaining export containers (E).

14. Unload t_k Tiers Adjacent to the Last Tier: Remove containers from t_k tiers adjacent to the tier just emptied (where t_k is the number of tiers to be cleared). Continue to Step 15.
15. Unload Symmetrical Region: After unloading t_k tiers, unload the symmetrical t_k tiers on the vessel.
16. Complete Deck Unloading and Open Covers: Clear all containers from the deck and open hatch covers as needed for re-handling beneath. Return to Step 3.

Genetic Algorithm (GA) is a heuristic optimization technique inspired by the principles of natural selection, introduced by John Holland in 1975. GA operates on a population of solutions, iteratively improving them through genetic operators like selection, crossover, and mutation.

Following the GRASP stage, we employ GA as a population-based method. For the QCSP, the GA encodes solutions via a two-layer chromosome structure, in which each gene corresponds to a specific bay. The upper layer determines the order in which bays are handled, and the lower layer specifies which quay crane (QC_1 or QC_2) is assigned to each bay. This dual-layer representation effectively models the complexity of the QCSP. When GA is applied to solve the QCSP, the process starts by generating an initial population which consists of a sizable set of randomly generated chromosomes. This initial population is pivotal in accelerating convergence toward an optimal or near-optimal outcome.

Each bay on the vessel is treated as an individual job, characterized by its processing time and location, as shown in Table 2. The table's rows list job identifiers, processing durations, and the corresponding bay numbers. Two QCs are available to handle the container operations linked to these jobs.

Table 2. Job information.

Jobs	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14
Processing Time	86	90	88	85	61	63	68	78	81	71	59	95	66	69
Bay	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Constructing a chromosome involves the random assignment of bays in the upper layer to define the job sequence, while in the lower layer the quay cranes (QC_1 or QC_2) are randomly allocated for executing each bay. This approach ensures a diverse initial population, as shown in Figure 4, which provides the foundation for iterative optimization within the GA framework.

Bay	4	1	10	7	2	6	8	12	3	5	9	11	14	13	4	1	10
QC	1	1	2	1	1	2	2	2	1	1	2	1	2	2	1	2	1

Figure 4. Job sequence and crane assignment on the chromosome.

In each generation, all chromosomes in the population are evaluated using a fitness function. Chromosomes with higher fitness values are selected for the mating pool to generate offspring. The pairing of chromosomes with superior fitness values accelerates convergence toward an optimal solution. In the developed GA, the fitness function is defined as the total time required to complete all bays in the schedule.

The GA operators and strategies employed in the application process of this study include elitism and family selection methodologies such as roulette wheel selection, rank-based weighting, and tournament selection. During the new generation production, elitism ensures that top-performing solutions advance directly to the next generation. By preserving a subset of the highest-quality solutions, known as elites, the GA retains valuable genetic information and moves closer to convergence more quickly. These elites are chosen based on their fitness values, alongside new offspring arising from crossover and mutation. In certain instances, mutation is also applied directly to elite chromosomes to improve them, ensuring enhanced solutions are passed on. To build the mating pool each generation, a family selection procedure produces one or more offspring from chosen parent chromosomes. This pool-formation step is critical to the GA's effectiveness, as it dictates how well the algorithm explores and exploits the search space. In this study, the sigma-truncation method is used to determine which chromosomes are included in the mating pool. Here, σ represents the standard deviation of the fitness values within the population. The adjusted fitness function guides the selection process to prioritize high-quality solutions. In the adjusted population, the determination of which individuals are included in the mating pool is governed by the coefficient c, which specifies how far below the population mean fitness value a chromosome can fall. Using this formulation, solutions with fitness values below the defined threshold are assigned a new fitness value below zero, effectively excluding them from the solution set. Only chromosomes included in the mating pool are subjected to crossover operations, ensuring that high-quality solutions are paired to generate superior offspring. For selecting parent chromosome pairs from the mating pool, three methods are utilized: the roulette wheel, the rank-based weighting, and the tournament. These approaches ensure diversity and maintain competitive selection among chromosomes in the population.

Additionally, genetic operators such as the uniform crossover and the replacement mutation are utilized. In the proposed GA approach, the uniform crossover is utilized for leveraging a binary sequence to decide which parent contributes each gene. A '1' denotes inheritance from the first parent, while a '0' indicates inheritance from the second. For mutation, the algorithm uses a replacement operator that swaps the positions of certain genes within a chromosome introducing needed diversity and helping stave off premature convergence. Additionally, a combination of a fixed mutation strategy and one that decreases exponentially across iterations allows the GA to strike a balance between exploration and exploitation.

A repair function and termination criteria are also incorporated to ensure both feasibility and effective completion of the algorithm. Sometimes, specific technical constraints may prevent a bay from being assigned to a crane. Following each chromosome's creation, the algorithm confirms compliance with these constraints, especially after crossover and mutation, which can generate unworkable chromosomes. A repair strategy amends any infeasible chromosome, so it meets the necessary conditions. The GA terminates based on two conditions: reaching a threshold fitness value and hitting a predefined time limit. This enables the method to conclude once acceptable performance or computational bounds have been met. If the average and best fitness values within the population remain unchanged over a specified number of iterations, the search process is terminated by indicating convergence. Additionally, a time limit is implemented for each experiment to ensure that the computational duration remains under control.

To illustrate the application of this method, an example involving a vessel with 25 bays, 12 rows, and 6 tiers requires the handling of 40 jobs (comprising 25 import containers and 15 export containers) by two QCs, is used. Table 3 provides detailed information about these jobs, including container positions, the number of containers above each target container, the last position of the QCs, operation times, and operation types.

Table 3. Example features.

No	Bay	Row	Tier	The Number of Containers Above	The Last Position of the QCs	Operation Times	Operation Types
1	7	1	6	3	1	78	Unloading
2	18	9	4	3	9	146	Unloading
3	17	5	2	1	5	30	Unloading
4	10	4	2	0	0	8	Unloading
5	2	7	1	0	0	9	Unloading
6	6	2	1	0	0	4	Unloading
7	15	11	1	0	0	13	Unloading
8	11	8	6	2	8	116	Unloading
9	17	5	3	2	5	62	Unloading
10	16	3	2	0	0	7	Unloading
11	9	5	1	0	0	7	Unloading
12	11	5	6	4	5	156	Unloading
13	3	11	2	1	11	54	Unloading
14	3	12	4	1	12	70	Unloading
15	17	3	2	1	3	22	Unloading
16	7	1	3	2	1	30	Unloading
17	2	3	2	1	3	22	Unloading
18	19	4	6	4	4	140	Unloading
19	1	2	6	0	0	14	Unloading
20	6	11	3	1	11	60	Unloading
21	8	12	1	0	0	14	Unloading
22	22	8	2	1	8	42	Unloading
23	5	1	1	0	0	3	Unloading
24	21	3	1	0	0	5	Unloading
25	19	6	6	4	6	172	Unloading
26	17	2	1	-	2	3	Loading
27	13	2	3	-	2	5	Loading
28	3	11	3	-	11	14	Loading
29	10	11	5	-	11	16	Loading
30	25	9	6	-	9	15	Loading
31	9	2	4	-	2	6	Loading
32	20	6	6	-	6	12	Loading
33	15	5	4	-	5	9	Loading
34	23	12	3	-	12	15	Loading
35	12	7	3	-	7	10	Loading
36	14	1	4	-	1	5	Loading
37	4	9	2	-	9	11	Loading
38	19	8	5	-	8	13	Loading
39	20	7	3	-	7	10	Loading
40	5	2	6	-	2	8	Loading

At the start of the scheduling process, it is assumed that the QCs begin their operations from opposite ends of the vessel. During the handling process, the travel time between adjacent bays (t) is set to 1 unit, and the QCs' velocity across all three axes is considered to be 1 unit per time unit. To minimize the total handling time, the 40 jobs are distributed between the two QCs, and their execution sequence is determined.

The sequencing problem has been initially solved using the GRASP method. The results of this sequencing process are presented in Table 4.

These results highlight the effectiveness of the GRASP method in optimizing operations, particularly with a high percentage of reshuffling performed onboard and a significant proportion of double-cycling operations. The handling times for each bay obtained using the GRASP method were scheduled for two QCs using the GA. The results, along with the GA parameters, are presented in Table 5.

Table 4. Results of the GRASP for the example.

Operation Method	GRASP
Number of Import Containers	25
Number of Export Containers	15
Number of Reshuffled Containers	30
Reshuffled Containers Onboard	29
Reshuffled Containers Offloaded	1
Total Operations (QC Steps)	150
Double-Cycling Operations	7
Computation Time (sec)	33.67
Double-Cycling Rate (%)	46.67
Onboard Reshuffling Rate (%)	96.67
Offloaded Reshuffling Rate (%)	3.33

Table 5. Results and parameters of the GA for the example.

Computation Method	Free Population
The Number of Replication	5
Fitness Value (min)	564.8
Computation Time (sec)	22.088
Population Size	50
Elitism Rate	1
Crossover Type	Uniform
Crossover Rate (%)	60
Chromosome Selection Method	Rank-Based Weighting
Mutation Type	Replacement
Mutation Rate (%)	60
Mutation Decay Rate (%)	10
Number of Replacements	15
C	3

As indicated in the table, the handling time for the example is calculated to be 564.8 min, demonstrating the effectiveness of the GA in scheduling operations efficiently.

4. Numerical Evaluation of the Methods

This section analyzes three proposed methods for solving the QCSP via a series of test instances. Section 4.1 presents the details of the problem sets created for numerical experiments which include varying problem sizes and complexities. Additionally, parameter analysis for the meta-heuristic methods is provided to ensure their optimal performance. In Section 4.2, we discuss the results obtained from each scenario, highlighting the performance differences among the proposed methods.

4.1. Test Sets and Parameter Analysis

To evaluate the effectiveness of the proposed methods, four distinct problem sets of varying sizes have been designed. The first set, which includes a small-scale job list presented in Section 3, is referred to as Scenario-1. The other sets reflect the operational complexities of modern container vessels, with three of them corresponding to common vessel types, namely Panamax, Post-Panamax, and New Panamax. These scenarios are designated as Scenario-2, Scenario-3, and Scenario-4, respectively. The characteristics of these problem sets are summarized in Table 6, while the 34-bay vessel layout plan for Scenario-3 is provided in Appendix A.

Table 6. Characteristics of the problem sets for all the scenarios.

	Number of Bays	Maximum Number of Rows	Maximum Number of Tiers	Total Number of Import Containers (I)	Total Number of Export Containers (E)	Total Number of Reshuffled Containers (R)	Total Number of Fixed Containers (F)
Scenario-1	25	12	10	25	15	30	407
Scenario-2	34	13	15	3352	2920	496	562
Scenario-3	34	15	15	2972	4107	695	659
Scenario-4	44	20	20	10,277	9806	2110	1531

In all scenarios, the QC operations have been carried out using two rail-mounted QCs. These cranes possess technical specifications that include a lifting velocity of 90 m/min under load, 180 m/min when empty, a trolley velocity of 240 m/min, and a horizontal movement velocity of 45 m/min. Additionally, the travel time between bays for the cranes has been standardized as 1 min. For analytical purposes, all these parameters have been consistently applied across all scenarios to ensure uniformity in the evaluation process.

Since the methods employed in this study utilize parameter-sensitive meta-heuristic algorithms such as ACO and GA, extensive parameter analyses have been conducted to determine the optimal settings. For these analyses, it is determined that over 20,000 experiments would be required to fully explore the parameter space. To streamline this process, the Taguchi Orthogonal Array Design method was employed instead of a full factorial experiment design. This approach enabled the efficient identification of optimal parameter combinations while reducing computational effort. In this study, the Taguchi experimental design technique is employed with factors determined based on the parameters used in ACO and GA methods and aligned with the relevant literature. No interaction between the factors is assumed, and factor levels are chosen to achieve results that closely approximate the optimal solution, in accordance with existing studies. Experiments are conducted for scenarios of varying sizes using the specified number of trials for ACO and GA. Variance analysis has been applied to the obtained experimental results.

For ACO, parameter tuning is limited to Scenarios 1 and 2, where the algorithm is applied. The analysis results, including parameters such as pheromone increase rate (Delta), pheromone evaporation rate (Rough), and the number of ants (α), are presented in Table 7. Notably, Delta and Rough values increased with the problem size, while the number of ants remained constant at 50 across both scenarios.

Table 7. ACO parameters for scenarios 1 and 2.

	Scenario-1	Scenario-2
Δ	1	2
ρ	0.2	0.4
α (Number of ants)	50	50

For GA, parameter analysis is conducted for all scenarios, and the optimized values are summarized in Table 8. Key observations include a proportional increase in population size based on the problem scale (e.g., 50 for Scenario-1, 70 for Scenarios 2 and 3, and 90 for Scenario-4). Similarly, the elitism ratio increased with chromosome size, while other parameter adjustments were problem-specific, such as mutation types and crossover strategies.

Table 8. GA parameters for scenarios 1, 2/3, and 4.

	Scenario-1	Scenario-2/3	Scenario-4
Computation Method	Free Population	Free Population	Free Population
The Number of Replication	5	5	5
Population Size	50	70	90
Elitism Rate	1	5	5
Crossover Type	Uniform	Uniform	Uniform
Crossover Rate (%)	60	90	60
Chromosome Selection Method	Rank-Based Weighting	Roulette Wheel	Tournament
Mutation Type	Replacement	Replacement	Replacement
Mutation Rate (%)	60	30	30
Mutation Decay Rate (%)	10	0	10
Number of Replacements	15	5	15
C	3	3	3

All experiments are performed on a standard computer with 16 GB RAM and an Intel i7 processor to ensure consistent computational conditions across all scenarios. These parameter settings provided the foundation for evaluating the performance of the proposed methods in the subsequent numerical analyses.

4.2. Discussion of the Results

The results obtained from the numerical experiments are discussed scenario-wise to evaluate the performance differences among the proposed methods. A comprehensive comparison of fitness values, computation times, and deviations from the optimal solution is presented in Table 9. All analyses have been conducted with 5 replications to ensure robustness.

Table 9. Comparative results of the methods across different scenarios.

		Method 1	Method 2	Method 3
Scenario-1	Fitness Value (min)	733	597.80	564.80
	Deviation from Best Value (%)	29.78	5.84	0
	Computation Time (sec)	-	152.41	22.09
	Deviation from Best Value (%)	-	589.95	0
Scenario-2	Fitness Value (min)	8570.85	10,020.60	8201.60
	Deviation from Best Value (%)	4.50	22.18	0
	Computation Time (sec)	-	10,020.60	121.54
	Deviation from Best Value (%)	-	8144.69	0
Scenario-3	Fitness Value (min)	10,024.80	-	9365.20
	Deviation from Best Value (%)	7.04	-	0
	Computation Time (sec)	-	-	122.66
	Deviation from Best Value (%)	-	-	-
Scenario-4	Fitness Value (min)	29,611.53	-	28,281.80
	Deviation from Best Value (%)	4.70	-	0
	Computation Time (sec)	-	-	260.69
	Deviation from Best Value (%)	-	-	-

The numerical results obtained for Scenario-1, as shown in Table 4, indicate that Method 3 achieves the best fitness value of 564.8 min. Method 2 follows with a 5.84% deviation, while Method 1 lags with a 29.78% deviation from the best solution. Notably, Method 3 also completes the computation in 22.09 s, indicating a fivefold speed improvement over Method 2, which requires more than 100 s. In Scenario-2, Method 3 continues to exhibit superior performance with a fitness value of 8201.6 min, representing a 22.18% improvement over Method 2 and a 4.50% improvement over Method 1. Furthermore, Method 3 achieves this result in 121.54 s, while Method 2 requires over 10,000 s to complete the same analysis. The computational time for Method-1 is not provided as it involves manual calculations. Additionally, Method-2 does not yield solutions for Scenario-3 and Scenario-4, hence the corresponding values are absent. The results for Scenario-3 further highlight the dominance of Method 3, which provides the best fitness value at 9365.2 min with a 7.04% improvement over Method 1. Additionally, Method 3 efficiently completes the analysis in 122.66 s, maintaining its computational advantage. Finally, for Scenario-4, Method 3 again achieves the best fitness value at 28,281.8 min, a 4.70% improvement over Method 1. The computational time for Method 3 remains competitive at 260.69 s and highlights its scalability and suitability for larger problem sizes.

Overall, these findings reinforce the dominance of Method 3 in both solution quality and computational efficiency. This performance demonstrates its potential as the most effective approach for QCSP. Although Method 1 is occasionally competitive, especially for moderate-scale problems, it lags behind for larger scenarios. Additionally, the significant computational challenges faced by Method 2 in larger scenarios further solidify the recommendation of Method 3 for practical DSS development. The findings also indicate that Method-2 struggles with larger problem sizes, rendering it unsuitable for Scenarios 3 and 4. Consequently, the superior performance of GA, used in Method-3, underscores its suitability for handling complex QCSP effectively and efficiently.

5. Proposed DSS Structure

The proposed Decision Support System (DSS) for the QCSP consolidates comprehensive analyses, including business requirements, data modeling, and software architecture design. Named KUTAY (initial version), it optimizes terminal operations by effectively tackling challenges in planning, scheduling, and resource allocation. KUTAY's structure and functionality are presented through its core components: business analysis, system architecture, and algorithmic framework.

5.1. Business Analysis and System Requirements

The foundation of KUTAY lies in a meticulous business analysis process aimed at identifying the specific needs of users managing terminal operations. The business analysis phase involved determining the required system functionalities and identifying critical areas to address. Figure 5 outlines the core sections identified as essential for the DSS.

In particular, one of the primary user requirements is the ability to input and manage vessel arrival and departure plans. These plans, defined at the bay level, include information on containers to be unloaded, reshuffled, or loaded. The detailed bay plan (commonly referred to as the stowage plan) constitutes the cornerstone of terminal planning operations. An illustration of such a plan, depicting the loading and unloading workflow for container bays, is presented in Figure 3 (Section 2).

Along with vessel plans, specifying terminal resources is equally critical. Notably, the DSS accommodates the configuration of the QCs and their operational velocities, as well as block-to-quay distances and other terminal-specific parameters. In combination, these adjustable settings enable optimal resource allocation and foster a scheduling approach that adapts to diverse scenarios. Additionally, the DSS's planning module supports the QC scheduling and the container assignments at the bay level. Through this module, users can define and modify system parameters in a flexible manner, thus tailoring the system to a wide range of operational conditions. Key performance indicators, such as resource usage, operational timelines, and Gantt charts, are integrated to furnish actionable insights for decision-makers.

Figure 5. The key screens of the DSS: (**a**) home screen; (**b**) terminal tab; (**c**) resources sub-tab; (**d**) sequencing sub-tab; (**e**) scheduling sub-tab; (**f**) reports tab.

5.2. Data Models and Software Architecture

The DSS is underpinned by comprehensive data models that capture the intricate nature of terminal operations which include entities such as vessels, QCs, and containers (see Figure 6).

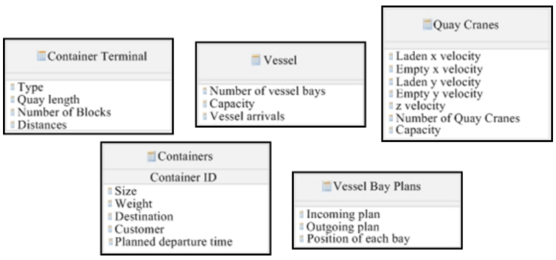


Figure 6. The data model for the terminal operations.

The system is built using object-oriented programming in C#, which promotes platform independence and straightforward error handling, to facilitate real-time decision-making. Overall software architecture employs Visual Studio as its primary IDE, supported by CodeMap and Entity Framework for efficient class-level and database management. Figure 7 demonstrates how the class diagram of the GA fits into and interacts with the other DSS components. The asterisks in the diagram represent multiplicity in the unified modeling language, indicating that a class can be associated with multiple instances of another class, allowing for flexible and scalable object relationships.

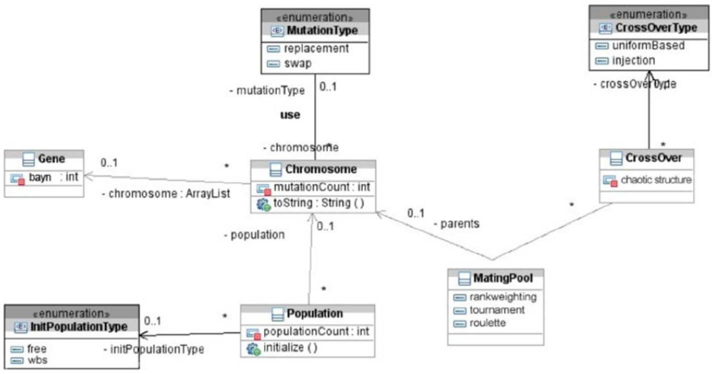


Figure 7. The class diagram of the GA.

5.3. Functional Modules

The DSS comprises several functional modules, each addressing specific aspects of terminal operations:

1. Terminal Parameter Definition: Enables the input and configuration of terminal attributes, such as crane velocities, block positions, and transport resources. The parameters are managed under the Terminal tab (Figure 5b).
2. Resource Management: Facilitates the definition and allocation of QCs and other resources. The Resources sub-tab allows users to specify crane velocities and operational constraints (Figure 5c).
3. Scheduling and Sequencing: Integrates advanced optimization algorithms for QC scheduling and container sequencing. Users can input bay-level arrival and departure plan under the Sequencing sub-tab and generate schedules using the Scheduling sub-tab (Figure 5d,e).
4. Reporting and Visualization: Provides detailed reports, including Gantt charts and resource utilization metrics, to assess operational performance. These outputs are accessible under the Reports tab (Figure 5f).

5.4. Algorithmic Framework

To ensure realistic outcomes, GA incorporates crane travel times and safety gaps as constraints, thus reflecting real-time operational dynamics. Figure 8 details the integration of sequencing, scheduling, and reporting modules, while Figure 9 provides a visual depiction via Gantt charts and fitness evolution plots.

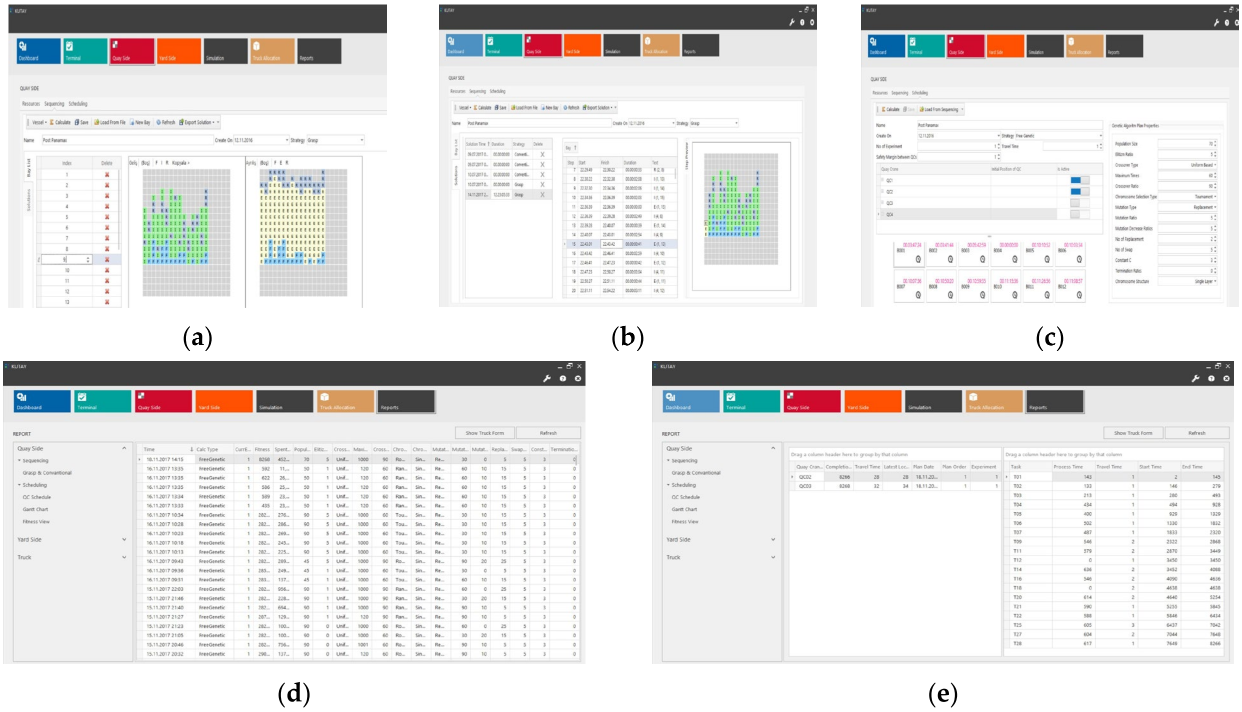


Figure 8. The integration of sequencing, scheduling, and reporting modules: (**a**) sequencing sub-tab including bay list and stowage plans; (**b**) sequencing sub-tab including handling sequence of a bay; (**c**) scheduling sub-tab including QC schedules of each bay; (**d**) schedule report; (**e**) QC based schedule report.

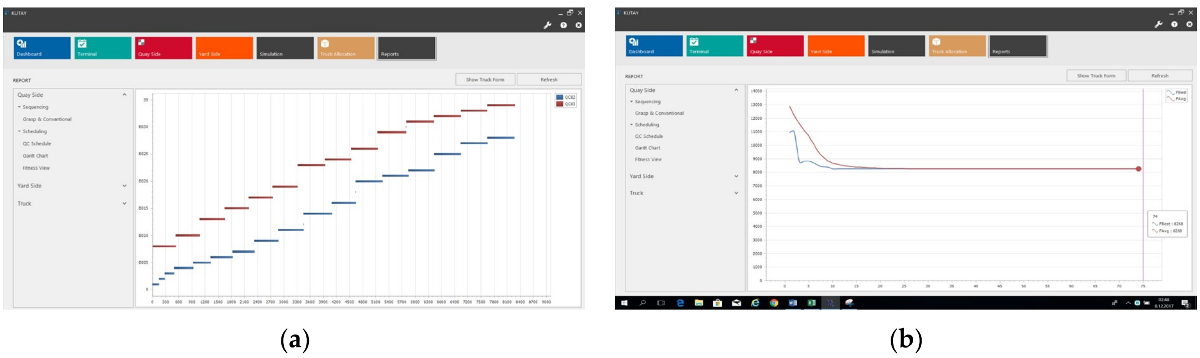


Figure 9. Gantt chart and fitness view for QC scheduling operations: (**a**) Gantt chart; (**b**) fitness view.

5.5. Comprehensive Integration

Overall, KUTAY's architecture seamlessly integrates each functional module, enabling both independent and collective management of terminal operations. The DSS equips users with pertinent analytics and optimization features spanning real-time container sequencing to QC scheduling. As indicated by Figures 5 and 8, both a user-friendly interface and a robust backend algorithmic core converge to offer a holistic management solution for container terminals. This fully integrated approach orchestrates real-time container sequencing, QC scheduling, and actionable insights, thus streamlining resource deployment and enhancing operational productivity.

6. Conclusions

This study investigates the Quay Crane Scheduling Problem (QCSP) within container terminals and offers novel strategies to boost operational performance. Specifically, we address a single-vessel QCSP with multiple quay cranes under non-crossing constraints by incorporating double-cycling to unify loading and unloading cycles. Within the rising demand for containerized transport and its crucial role in global supply networks, the QCSP presents multiple challenges. The study contributes to strengthening the terminal operations through efficient resource allocation, effective job scheduling, and realistic handling of real-world constraints by proposing a series of methods and developing a robust Decision Support System (DSS).

Comparative experiments with diverse scenarios confirm that the hybrid GRASP and GA approach (Method 3) outperforms both the classical scheduling (Method 1) and the ACO-based (Method 2) methods. In particular, Method 3 consistently demonstrates higher-quality solutions and shorter computation times, especially when dealing with larger problem instances. Moreover, a detailed parameter analysis for the meta-heuristic components underlines the resilience and efficiency of the proposed algorithms. The DSS further supports the feasibility of these solutions by presenting a unified platform that facilitates resource distribution, scheduling jobs, and operational planning in a seamless manner.

Several important contributions emerge from this study. First, the hybrid meta-heuristic framework that merges GRASP and GA addresses the QCSP's intricacies by alleviating the drawbacks of existing approaches and showing considerable gains in solution accuracy and efficiency. Second, the KUTAY offers a user-centric interface, integrating advanced optimization procedures with pragmatic constraints, such as crane separation, safety margins, and precedence relationships, into a single environment. This unification enables timely and actionable insights for terminal operators. Lastly, the scenario-based exploration, particularly aimed at mid-sized container terminals in Turkey, emphasizes localized operational settings and provides context-specific strategies that align with real-world requirements.

Despite its valuable insights, the study has certain limitations. To begin with, a predefined set of scenarios is used, and this overlooks possible real-life complexities like sudden weather changes or unforeseen equipment failures. Additionally, relying on fixed resource parameters (standard crane velocities) helps maintain consistency. However, it may not fully represent the various conditions in different terminal environments. Finally, although the computational setup suffices for the research scope, more extensive or time-sensitive applications in larger ports could pose scalability challenges to the proposed framework.

Furthermore, direct comparisons with standard QCSP heuristics remain challenging due to the specialized constraints in our approach—such as real-time crane travel times, double-cycling, and mid-sized terminal configurations. These features diverge from classical assumptions, making a one-to-one numerical comparison with existing references less meaningful. Future work may involve adapting the problem to more traditional QCSP benchmarks where feasible, while preserving key real-world complexities necessary for practical relevance.

Looking ahead, these findings open up multiple avenues for future work. One promising direction is the integration of dynamic factors, where real-time data (such as unexpected vessel delays or crane malfunctions) could enable adaptive and robust scheduling solutions. Further exploration of next-generation meta-heuristics, including reinforcement learning or hybrid swarm intelligence, may enhance both scalability and performance in QC planning.

Author Contributions: Conceptualization, M.E.T.; Validation, M.E.T.; Formal analysis, M.E.T.; Investigation, M.E.T.; Resources, M.E.T.; Data curation, M.E.T.; Writing—original draft preparation, M.E.T.; Visualization, M.E.T.; Methodology, M.E.T. and F.E.; Software, M.E.T. and F.E.; Writing—review and editing, M.E.T. and F.E.; Supervision, F.E.; Project administration, F.E.; Funding acquisition, F.E. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through project number MoE-IF-UJ-R2-22-20560-1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
GRASP	Greedy Randomized Adaptive Search Procedure
GA	Genetic Algorithm
DSS	Decision Support System
UNCTAD	United Nations Conference on Trade and Development
QCs	Quay Cranes
QCSP	Quay Crane Scheduling Problems
AGVs	Automated Guided Vehicles
PSO	Particle Swarm Optimization
MGPSO	Multiple-Group PSO
DT	Digital Twin
IoT	Internet of Things

Appendix A

Below is the 34-bay vessel layout plan used for Scenario-3. Bays 4, 15, 18, and 28 are omitted from the figure as there are no handling operations scheduled for these locations.

Table A1. Stowage plan for Sceanario-3.

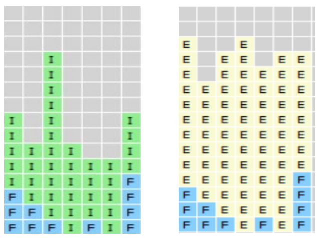
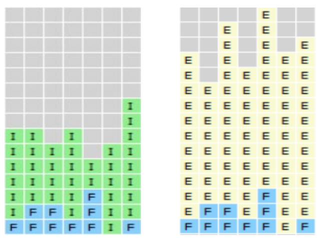
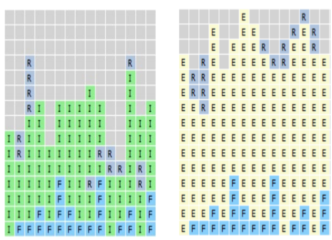
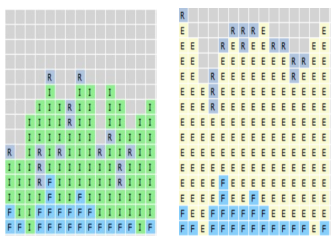
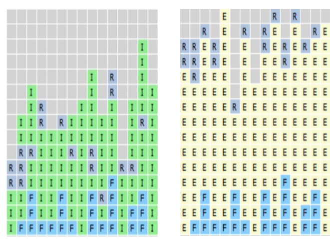
<table>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 1
I: 39; E: 72; F: 11; R: 0</td><td colspan="2">Bay 2
I: 37; E: 77; F: 10; R: 0</td><td colspan="2">Bay 3
I: 58; E: 104; F: 17; R: 3</td></tr>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 5
I: 90; E: 158; F: 23; R: 14</td><td colspan="2">Bay 6
I: 90; E: 158; F: 23; R: 14</td><td colspan="2">Bay 7
I: 90; E: 158; F: 23; R: 18</td></tr>
</table>

Table A1. *Cont.*

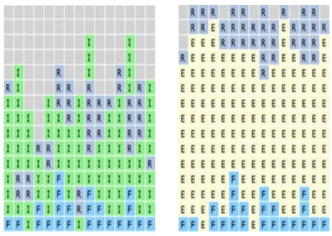
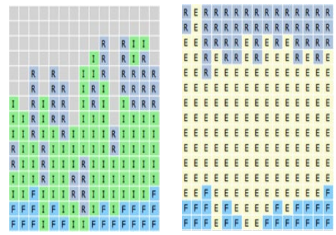
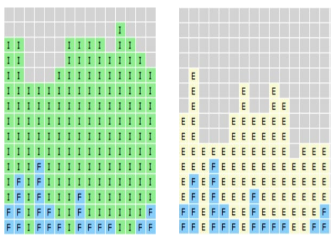
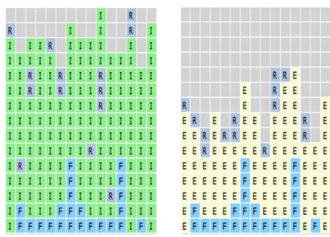
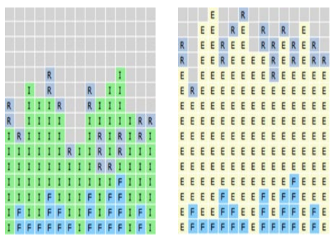
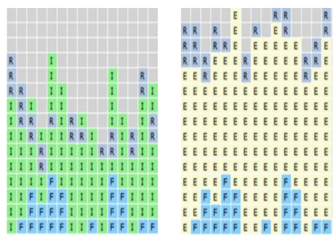
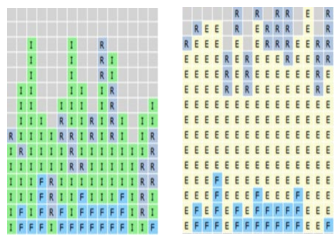
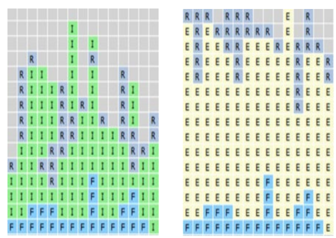
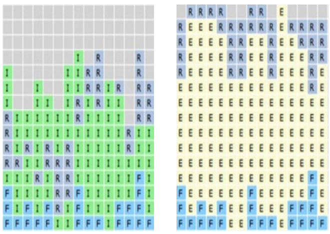
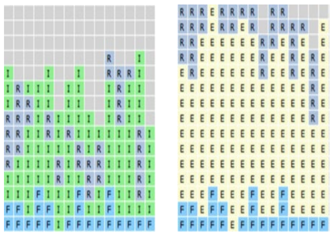
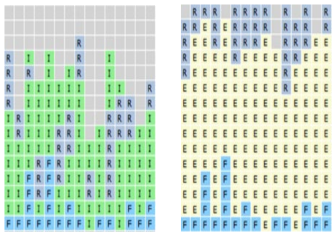
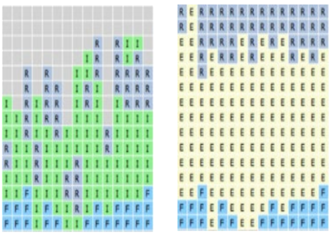
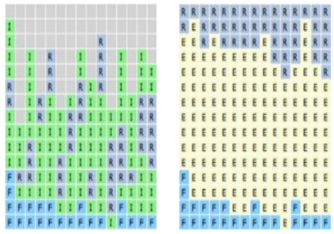
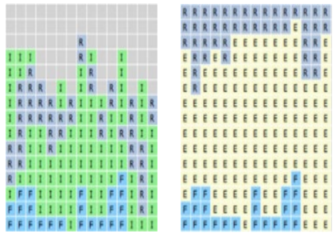
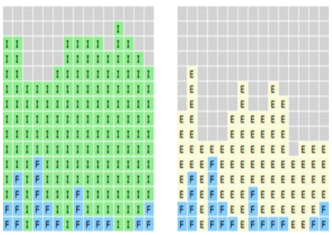
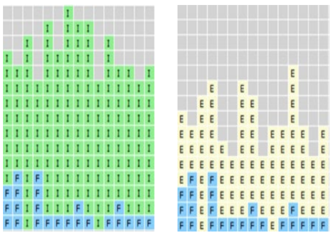
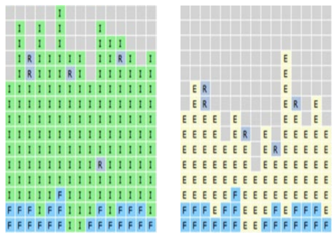
<table>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 8
I: 90; E: 158; F: 23; R: 23</td><td colspan="2">Bay 9
I: 90; E: 158; F: 23; R: 28</td><td colspan="2">Bay 10
I: 90; E: 158; F: 23; R: 30</td></tr>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 11
I: 90; E: 158; F: 23; R: 36</td><td colspan="2">Bay 12
I: 90; E: 158; F: 23; R: 41</td><td colspan="2">Bay 13
I: 90; E: 157; F: 23; R: 45</td></tr>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 14
I: 158; E: 90; F: 23; R: 0</td><td colspan="2">Bay 16
I: 158; E: 90; F: 23; R: 14</td><td colspan="2">Bay 17
I: 158; E: 90; F: 23; R: 18</td></tr>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 19
I: 90; E: 158; F: 23; R: 18</td><td colspan="2">Bay 20
I: 90; E: 158; F: 23; R: 23</td><td colspan="2">Bay 21
I: 90; E: 158; F: 23; R: 28</td></tr>
<tr><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th><th>Arrival</th><th>Departure</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">Bay 22
I: 90; E: 158; F: 23; R: 28</td><td colspan="2">Bay 23
I: 90; E: 158; F: 23; R: 32</td><td colspan="2">Bay 24
I: 90; E: 158; F: 23; R: 32</td></tr>
</table>

Table A1. *Cont.*

Arrival	Departure	Arrival	Departure	Arrival	Departure
					
Bay 25 I: 90; E: 158; F: 23; R: 36		Bay 26 I: 90; E: 158; F: 23; R: 36		Bay 27 I: 90; E: 158; F: 23; R: 36	
Arrival	Departure	Arrival	Departure	Arrival	Departure
					
Bay 29 I: 90; E: 157; F: 23; R: 45		Bay 30 I: 90; E: 157; F: 23; R: 45		Bay 31 I: 90; E: 157; F: 23; R: 45	
Arrival	Departure	Arrival	Departure	Arrival	Departure
					
Bay 32 I: 158; E: 90; F: 23; R: 0		Bay 33 I: 158; E: 90; F: 23; R: 0		Bay 34 I: 158; E: 90; F: 23; R: 5	

References

1. Review Of Maritime Transport 2022. Available online: https://unctad.org/system/files/official-document/rmt2022_en.pdf (accessed on 1 December 2024).
2. Kim, K.H.; Park, Y.M. A crane scheduling method for port container terminals. *Eur. J. Oper. Res.* **2004**, *156*, 752–768. [CrossRef]
3. Lee, D.H.; Wang, H.Q.; Miao, L.X. Quay crane scheduling with non-interference constraints in port container terminals. *Transp. Res. E Logist. Transp. Rev.* **2008**, *44*, 124–135. [CrossRef]
4. Bierwirth, C.; Meisel, F. A fast heuristic for quay crane scheduling with interference constraints. *J. Sched.* **2009**, *12*, 345–360. [CrossRef]
5. Legato, P.; Trunfio, R.; Meisel, F. Modeling and solving rich quay crane scheduling problems. *Comput. Oper. Res.* **2012**, *39*, 2063–2078. [CrossRef]
6. Meisel, F. The quay crane scheduling problem with time windows. *Naval Res. Logist.* **2011**, *58*, 619–636. [CrossRef]
7. Chung, S.H.; Choy, K.L. A modified genetic algorithm for quay crane scheduling operations. *Expert Syst. App.* **2012**, *39*, 4213–4221. [CrossRef]
8. Park, Y.M.; Kim, K. A scheduling method for berth and quay cranes. In *Container Terminals and Automated Transport Systems*; Günther, H.O., Kim, K.H., Eds.; Springer: Berlin, Germany, 2005; pp. 159–181.
9. Moccia, L.; Cordeau, J.F.; Gaudioso, M.; Laporte, G. A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Res. Logist.* **2006**, *53*, 45–59. [CrossRef]
10. Liang, C.J.; Huang, Y.F.; Yang, Y. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Comput. Ind. Eng.* **2009**, *56*, 1021–1028. [CrossRef]

11. Zeng, Q.C.; Yang, Z.Z.; Hu, X.P. Disruption recovery model for berth and quay crane scheduling in container terminals. *Eng. Opt.* **2011**, *43*, 967–983. [CrossRef]
12. Iris, Ç.; Pacino, D.; Ropke, S.; Larsen, A. Integrated Berth Allocation and Quay Crane Assignment Problem: Set partitioning models and computational results. *Transp. Res. E Logist. Transp. Rev.* **2015**, *81*, 75–97. [CrossRef]
13. Iris, Ç.; Pacino, D.; Ropke, S. Improved formulations and an Adaptive Large Neighborhood Search heuristic for the integrated berth allocation and quay crane assignment problem. *Transp. Res. E Logist. Transp. Rev.* **2017**, *105*, 123–147. [CrossRef]
14. Carlo, H.J.; Vis, I.F.A.; Roodbergen, K.J. Storage yard operations in container terminals: Literature overview, trends, and research directions. *Eur. J. Oper. Res.* **2014**, *235*, 412–430. [CrossRef]
15. Taner, M.E.; Kulak, O.; Koyuncuoğlu, M.U. Layout analysis affecting strategic decisions in artificial container terminals. *Comput. Ind. Eng.* **2014**, *75*, 1–12. [CrossRef]
16. Wiese, J.; Suhl, L.; Kliewer, N. Mathematical models and solution methods for optimal container terminal yard layouts. *OR Spectr.* **2010**, *32*, 427–452. [CrossRef]
17. Lee, B.K.; Kim, K.H. Optimizing the block size in container yards. *Transp. Res. E Logist. Transp. Rev.* **2010**, *46*, 120–135. [CrossRef]
18. Ng, W.C.; Mak, K.L.; Li, M.K. Yard planning for vessel services with a cyclical calling pattern. *Eng. Opt.* **2010**, *42*, 1039–1054. [CrossRef]
19. Park, T.; Choe, R.; Kim, Y.H.; Ryu, K.R. Dynamic adjustment of container stacking policy in an automated container terminal. *Int. J. Prod. Econ.* **2011**, *133*, 385–392. [CrossRef]
20. Fu, Z.; Li, Y.; Lim, A.; Rodrigues, B. Port space allocation with a time dimension. *J. Op. Res. Soc.* **2007**, *58*, 797–807. [CrossRef]
21. Guldogan, E.U. Simulation-based analysis for hierarchical storage assignment policies in a container terminal. *Simulation* **2010**, *87*, 523–537. [CrossRef]
22. Petering, M.E.H.; Wu, Y.; Li, W.; Goh, M.; de Souza, R. Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals. *OR Spectr.* **2009**, *31*, 801–835. [CrossRef]
23. Stahlbock, R.; Voss, S. Efficiency considerations for sequencing and scheduling of double-rail-mounted gantry cranes at maritime container terminals. *Int. J. Shipp. Transp. Logist.* **2010**, *2*, 95–123. [CrossRef]
24. Kim, K.H.; Hong, G.P. A heuristic rule for relocating blocks. *Comput. Op. Res.* **2006**, *33*, 940–954. [CrossRef]
25. Yang, J.; Kim, K. A grouped storage method for minimizing relocations in block stacking systems. *J. Intell. Manuf.* **2006**, *17*, 453–463. [CrossRef]
26. Hirashima, Y.; Takeda, K.; Harada, S.; Deng, M.; Inoue, A. A Q-learning for group-based plan of container transfer scheduling. *JSME Int. J. Ser. C Mech. Syst. Mach. Elem. Manuf.* **2006**, *49*, 473–479. [CrossRef]
27. Choe, R.; Park, T.; Oh, M.S.; Kang, J.; Ryu, K. Generating a rehandling-free intra-block remarshaling plan for an automated container yard. *J. Intell. Manuf.* **2011**, *22*, 201–217. [CrossRef]
28. Ng, W.C.; Mak, K.L.; Zhang, Y.X. Scheduling trucks in container terminals using a genetic algorithm. *Eng. Opt.* **2007**, *39*, 33–47. [CrossRef]
29. Cao, J.X.; Lee, D.H.; Chen, J.H.; Shi, Q. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transp. Res. E Logist. Transp. Rev.* **2010**, *46*, 344–353. [CrossRef]
30. Petering, M.E.H. Development and simulation analysis of real-time, dual-load yard truck control systems for seaport container transshipment terminals. *OR Spectr.* **2010**, *32*, 633–661. [CrossRef]
31. Kulak, O.; Polat, O.; Gujjula, R.; Günther, H.O. Strategies for improving a long-established terminal's performance: A simulation study of a Turkish container terminal. *Flex. Serv. Manuf. J.* **2013**, *25*, 503–527. [CrossRef]
32. Chen, L.; Bostel, N.; Dejax, P.; Cai, J.; Xi, L. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *Eur. J. Oper. Res.* **2007**, *181*, 40–58. [CrossRef]
33. Zeng, Q.; Yang, Z. Integrating simulation and optimization to schedule loading operations in container terminals. *Comput. Op. Res.* **2009**, *36*, 1935–1944. [CrossRef]
34. Homayouni, S.M.; Tang, S.H. Multi objective optimization of coordinated scheduling of cranes and vehicles at container terminals. *Math. Probl. Eng.* **2013**, *2013*, 1–9. [CrossRef]
35. Lu, Y.; Le, M. The integrated optimization of container terminal scheduling with uncertain factors. *Comput. Ind. Eng.* **2014**, *75*, 209–216. [CrossRef]
36. Iris, Ç.; Christensen, J.; Pacino, D.; Ropke, S. Flexible ship loading problem with transfer vehicle assignment and scheduling. *Transp. Res. B Methodol.* **2018**, *111*, 113–134. [CrossRef]
37. Safaeian, M.; Etebari, F.; Vahdani, B. An integrated quay crane assignment and scheduling problem with several contractors in container terminals. *Sci. Iran.* **2021**, *28*, 1030–1048. [CrossRef]
38. Nourmohammadzadeh, A.; Voss, S. A robust multiobjective model for the integrated berth and quay crane scheduling problem at seaside container terminals. *Ann. Math. Artif. Intell.* **2022**, *90*, 831–853. [CrossRef]
39. Behjat, S.; Nahavandi, N. A mathematical model and grouping imperialist competitive algorithm for integrated quay crane and yard truck scheduling problem with non-crossing constraint. *Int. J. Eng.* **2019**, *32*, 1464–1479.

40. Hsu, H.P.; Wang, C.N.; Fu, H.P.; Dang, T.T. Joint scheduling of yard crane, yard truck, and quay crane for container terminal considering vessel stowage plan: An integrated simulation-based optimization approach. *Mathematics* **2021**, *9*, 2236. [CrossRef]
41. Hsu, H.P.; Wang, C.N. Resources planning for container terminal in a maritime supply chain using multiple particle swarms optimization (MPSO). *Mathematics* **2020**, *8*, 764. [CrossRef]
42. Xin, J.; Meng, C.; D'Ariano, A.; Wang, D.; Negenborn, R.R. Mixed-integer nonlinear programming for energy-efficient container handling: Formulation and customized genetic algorithm. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 10542–10555. [CrossRef]
43. Jiang, M.X.; Ma, F.Z.; Zhang, Y.Q.; Lv, S.Y.; Pei, Z.; Wu, G.H. Collaborative scheduling optimization of container port berths and cranes under low-carbon environment. *Sustainability* **2024**, *16*, 2985. [CrossRef]
44. Hsu, H.P.; Wang, C.N.; Nguyen, T.T.T.; Dang, T.T.; Pan, Y.J. Hybridizing WOA with PSO for coordinating material handling equipment in an automated container terminal considering energy consumption. *Adv. Eng. Informatics* **2024**, *60*, 102410. [CrossRef]
45. Yang, X.M.; Jiang, X.J. Yard crane scheduling in the ground trolley-based automated container terminal. *Asia-Pacific J. Op. Res.* **2020**, *37*, 2050007. [CrossRef]
46. Wang, R.; Xu, X.; Wang, Z.; Ji, F.; Mu, N. A pre-selection-based ant colony system for integrated resources scheduling problem at marine container terminal. *Comput. Mater. Contin.* **2024**, *80*, 2363–2385. [CrossRef]
47. Rouky, N.; Abourraja, M.N.; Boukachour, J.; Boudebous, D.; Alaoui, A.E.H.; Khoukh, F.E. Simulation optimization based ant colony algorithm for the uncertain quay crane scheduling problem. *Int. J. Ind. Eng. Comput.* **2019**, *10*, 111–132. [CrossRef]
48. Li, J.; Xu, B.; Yang, Y.; Wu, H. Quantum ant colony optimization algorithm for AGVs path planning based on block coordinates of pheromones. *Nat. Comput.* **2020**, *19*, 673–682. [CrossRef]
49. Castilla-Rodríguez, I.; Expósito-Izquierdo, C.; Melián-Batista, B.; Aguilar, R.M.; Moreno-Vega, J.M. Simulation-optimization for the management of the transshipment operations at maritime container terminals. *Expert Syst. Appl.* **2020**, *139*, 112852. [CrossRef]
50. Klar, R.; Fredriksson, A.; Angelakis, V. Digital twins for ports: Derived from smart city and supply chain twinning experience, digital twins for ports: Derived from smart city and supply chain twinning experience. *IEEE Access* **2023**, *11*, 71777–71799. [CrossRef]
51. Ding, Y.; Zhang, Z.; Chen, K.; Ding, H.; Voss, S.; Heilig, L.; Chen, Y.; Chen, X. Real-time monitoring and optimal resource allocation for automated container terminals: A digital twin application at the Yangshan Port. *J. Adv. Transp.* **2023**, *2023*, 6909801. [CrossRef]
52. Wang, K.; Xu, H.; Wang, H.; Qiu, R.; Hu, Q.; Liu, X. Digital twin-driven safety management and decision support approach for port operations and logistics. *Front. Mar. Sci.* **2024**, *11*, 1455522. [CrossRef]
53. Abril, D.; Paternina-Arboleda, C.D.; Velasquez-Bermudez, J. An integrated event-driven real-time tactical–operational optimization framework for smart port operations planning. *Logistics* **2024**, *8*, 65. [CrossRef]
54. Bilican, M.S.; Iris, Ç.; Karatas, M. A collaborative decision support framework for sustainable cargo composition in container shipping services. *Ann. Oper. Res.* **2024**, *342*, 79–111. [CrossRef]
55. Meisel, F.; Wichmann, M. Container sequencing for quay cranes with internal reshuffles. *OR Spectr.* **2010**, *32*, 569–591. [CrossRef]
56. Liu, M.; Chu, F.; Zhang, Z.; Chu, C. A polynomial-time heuristic for the quay crane double-cycling problem with internal-reshuffling operations. *Transp. Res. E Logist. Transp. Rev.* **2015**, *81*, 52–74. [CrossRef]
57. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [CrossRef]
58. Feo, T.A.; Resende, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. *Op. Res. Lett.* **1989**, *8*, 67–71. [CrossRef]