Journal of Computer Science

Original Research Paper

Quay Crane Scheduling in Container Terminals Using a Hybrid Genetic Algorithm

Aidi Sanaa, Torbi Imane and Mazouzi Mohamed

Department of Mechanics, Engineering and Innovation, University Hassan 2, ENSEM, Casablanca, Morocco

Article history
Received: 12-10-2024
Revised: 20-11-2024
Accepted: 23-11-2024

Corresponding Author:
Aidi Sanaa
Department of Mechanics, Engineering and Innovation, University Hassan 2, ENSEM, Casablanca, Morocco
Email: sanaa.aidi@gmail.com

Abstract: Container terminals are crucial nodes within the global supply chain, playing a vital role in the efficient movement of goods. Effective scheduling of Quay Cranes (QCs) is a key factor in maximizing port productivity and minimizing delays. This research investigates the Quay Crane Scheduling Problem (QCSP) using a Hybrid Genetic Algorithm (HGA). The proposed HGA method combines the exploratory power of genetic algorithms with refined local search strategies to boost both solution quality and convergence speed. Extensive computational experiments using established benchmark datasets confirm the effectiveness of the hybrid algorithm, revealing a significant reduction in the make span and enhanced utilization of quay crane resources. The findings of this study contribute to the broader understanding of algorithmic optimization for QCSP, providing valuable insights for improving operational efficiency in real-world container terminal environments.

Keywords: Quay Crane Scheduling, Container Terminal, Genetic Algorithm, Hybrid Algorithm, Optimization, Port Logistics

Introduction

Quay cranes are indispensable for efficient container handling in seaport terminals, playing a pivotal role in facilitating global trade (Bierwirth and Meisel, 2010). Their effective scheduling not only minimizes vessel turnaround time but also significantly boosts the operational efficiency of container terminals (Aidi *et al*., 2021). The Quay Crane Scheduling Problem (QCSP) Fig. (1) represents a combinatorial optimization challenge that aims to minimize handling times, reduce delays (Bierwirth and Meisel, 2009), and achieve optimal utilization of quay cranes, all of which are critical (Hop *et al*., 2021) for seamless terminal operations.

Traditional solutions, including mathematical programming and heuristic-based methods (Expósito-Izquierdo *et al*., 2013) often encounter scalability issues when faced with the high complexity of real-world QCSP scenarios (Zhang *et al*., 2018). These methods may struggle to generate efficient schedules within a practical timeframe, (Chen *et al*., 2014), particularly in large-scale terminal operations.

In recent years, Genetic Algorithms (GAs) (Zhao *et al*., 2022) have emerged as a promising alternative for addressing such complex optimization problems. Renowned for their adaptability and robustness (Pap *et al*., 2013), GAs simulate the process of natural evolution to efficiently explore vast solution spaces (Haghani and Jung, 2005). Building on this foundation, this study introduces a Hybrid Genetic Algorithm (HGA) (Zhao *et al*., 2022) designed to enhance the efficacy of QCSP solutions further. The proposed HGA integrates genetic search principles with local search techniques, (El-Abbasy *et al*., 2021) leveraging domain-specific knowledge to refine solutions iteratively. (Aidi and Mazouzi, 2023) This innovative approach seeks to balance exploration and exploitation in the search process, (Skaf *et al*., 2021) aiming for superior schedules that align with real-world operational demands.

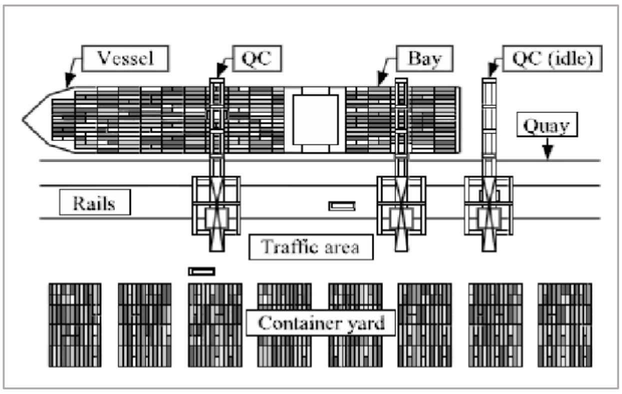

Fig. 1: Illustration of QCs working on a vessel

Problem Description and Formulation

Quay Crane Scheduling Problem (QCSP)

The Quay Crane Scheduling Problem (QCSP) addresses the efficient allocation and sequencing of quay cranes to perform (Homayouni and Tang, 2013) loading and unloading tasks for containers on a vessel. The challenge is creating a schedule that minimizes the make span, representing the total time required to complete all crane operations. This problem involves managing a variety of constraints to ensure operational safety (Bierwirth and Meisel, 2009), efficiency, and adherence to handling priorities.

Mathematical Formulation

Let N represent the number of container bays and N denotes the number of quay cranes available (Aidi and Mazouzi, 2023). The goal is to assign cranes to container bays and sequence their operations to minimize the total make span. The problem can be formulated as:

Parameters

- *N*: Number of container bays on the vessel
- *M*: Number of quay cranes available for scheduling
- *Ti*: Completion time for Quay Crane I after finishing all its assigned tasks
- Admin: Minimum safety distance required between adjacent quay cranes to avoid interference
- *Pj*: Priority level of container bay *j*, with higher priority bays receiving earlier attention

Objective Function

The primary goal is to minimize the make span, expressed as:

$$Minimize: Makespan = max(Ti)\ \forall i \in \{1,2,\ldots,M\}$$

This ensures that the time for the slowest (or last) crane to finish its tasks is minimized, optimizing overall operational efficiency.

Constraints

Crane Non-Interference

Quay cranes cannot operate simultaneously on the same container bay. If two cranes are assigned tasks in overlapping regions, their schedules must be adjusted to eliminate conflicts:

$$xij + xik \leq 1, \forall i, j, k \text{ such that } j = k$$

Safety Distance

Adjacent quay cranes must maintain a minimum horizontal safety distance to avoid collisions or interference:

$$\mid Pi - pj \mid \geq dmin\ , \forall i, \text{where } i \neq j$$

Here, pi and pj represent the positions of cranes i and j, respectively.

Handling Priorities

Some container bays may have higher handling priorities due to time-sensitive cargo or operational constraints. Cranes must process these days earlier in the schedule:

$$If\ Pj > Pk, then\ Tj < Tk$$

Sequential Task Completion

A quay crane can only begin its task at a bay after completing tasks in the preceding bay, ensuring a logical task progression:

$$Tij \ \geq Ti(j-1) + tij\ , \forall i, j$$

Here, *tij* represents the time taken by crane *iii* to complete its operation on bay *j*.

Crane Availability

The number of quay cranes in operation must not exceed the total available:

$$\sum xij \leq M, \forall i$$

This formulation combines operational goals and safety considerations to create an efficient quay crane schedule. The constraints reflect real-world challenges, such as ensuring safe crane movement, prioritizing critical operations, and respecting equipment limitations. By minimizing the make span under these constraints, terminal operators can achieve higher throughput, reduced vessel turnaround times, and optimized resource utilization.

Materials and Methods

Genetic Algorithm Overview

The Genetic Algorithm (GA) (Gulić and Žuškin, 2023) is a bio-inspired optimization method that simulates the process of natural selection. It operates on a population of candidate solutions, iteratively improving their quality through key operations:

1. Selection**:** Prioritizes better-performing solutions (based on fitness) for reproduction
2. Crossover: Combines genetic material from selected solutions to generate new, potentially better offspring
3. Mutation: Introduces diversity into the population by making small, random modifications to solutions

GAs are particularly well-suited for complex problems like QCSP due to their ability to search large solution spaces and avoid getting trapped in local optima.

Hybridization Approach

The proposed HGA enhances the traditional GA by integrating a local search procedure. This hybridization

DOI: 10.3844/jcssp.2025.197.202

allows for a focused exploration of the solution space around promising candidates, improving overall efficiency and solution quality. The local search mechanism refines offspring solutions, ensuring that the algorithm converges toward optimal or near-optimal schedules without excessive computational cost.

Key Components of the Proposed HGA

Initialization

- The algorithm begins by generating an initial population of feasible quay crane schedules
- A heuristic approach is used to ensure that all initial solutions adhere to operational constraints, such as non-interference and safety margins
- Heuristics reduce the likelihood of invalid solutions and accelerate convergence by starting the search in promising regions of the solution space

Selection

- Solutions are selected based on their fitness**,** which is inversely proportional to the make span of the schedule
- Common selection techniques, such as roulette wheel selection or tournament selection, may be employed to balance exploration (diversity) and exploitation (focus on the best solutions)

Crossover

- Pairs of selected solutions undergo a crossover operation to produce offspring that inherit features from both parents
- Crossover strategies, such as uniform crossover or two-point crossover, are designed to combine crane assignments and task sequences effectively while maintaining feasibility

Mutation

- A mutation operator introduces small, random changes to offspring solutions
- Mutation ensures diversity in the population, preventing premature convergence to suboptimal solutions. For QCSP, this could involve altering the task sequence of a crane or reassigning a crane to a different day

Local Search

- After generating offspring through crossover and mutation, a local search algorithm refines each solution.
- Methods such as simulated annealing**,** greedy heuristics**,** or tabu search are used to explore the neighborhood of a solution, identifying small improvements in crane assignments or task orders
- This step enhances solution quality without significantly increasing the computational burden, as it operates on a focused subset of potential improvements

Replacement

- The refined offspring replace less fit individuals in the population
- Replacement strategies, such as elitism, ensure that the best solutions are preserved across generations, maintaining steady progress toward the optimal schedule

Termination

The algorithm terminates when one of the following criteria is met:

- A predefined number of generations have been completed
- No significant improvement in the best solution has been observed over several generations

These conditions prevent unnecessary computation while ensuring convergence to high-quality solutions.

Pseudo Code for Hybrid Genetic Algorithm (HGA)

The pseudo-code for the hybrid genetic algorithm is shown in Table (1). designed to solve the Quay Crane Scheduling Problem (QCSP) (Song and Xu, 2024).

Input: Problem data (container bays, quay cranes, constraints), Population size (P), max Generations (G), Crossover rate (Cr), Mutation rate (Mr).

Output: Best feasible schedule with minimized make span.

Table 1: The pseudo-code for the hybrid genetic

Step	Description
1. Initialization	Generate an initial population of feasible solutions using a heuristic approach. Evaluate the fitness of each individual (fitness = inverse of make span)
2. Selection	Select parent solutions from the population based on fitness using techniques like tournament or roulette wheel selection
3. Crossover	Perform crossover between selected parent pairs with a predefined probability to create offspring
4. Mutation	Apply random changes to the offspring with a predefined probability to maintain genetic diversity
5. Local search	Refine each offspring using local search methods such as simulated annealing or greedy algorithms
6. Replacement	Combine the current population and offspring
7. Convergence check	Check whether the stopping criteria are met (e.g., no significant improvement or max generations reached)
8. Termination	If converged, output the best solution. If not, return to the selection step for the next generation

DOI: 10.3844/jcssp.2025.197.202

Results

Benchmark Instances

The proposed algorithm was tested on standard QCSP benchmark instances with varying numbers of cranes and container bays. The performance was compared against traditional GA and other heuristic approaches.

The HGA shown in Fig. (2), demonstrated superior performance in terms of minimizing make span across all test cases. The integration of local search resulted in faster convergence to high-quality solutions. The average make span reduction was approximately 15% compared to the standard GA and the computational time remained within acceptable limits.

Case Study: Numerical Comparison on a Real QCSP Instance

Problem Setup:

- Instance 1 (Small):
  - ✓ 3 cranes
  - ✓ 10 bays
  - ✓ Objective: Minimize the make span (total time taken for all crane operations)
- Instance 2 (Large):
  - ✓ 5 cranes
  - ✓ 30 bays
  - ✓ Objective: Minimize the make span

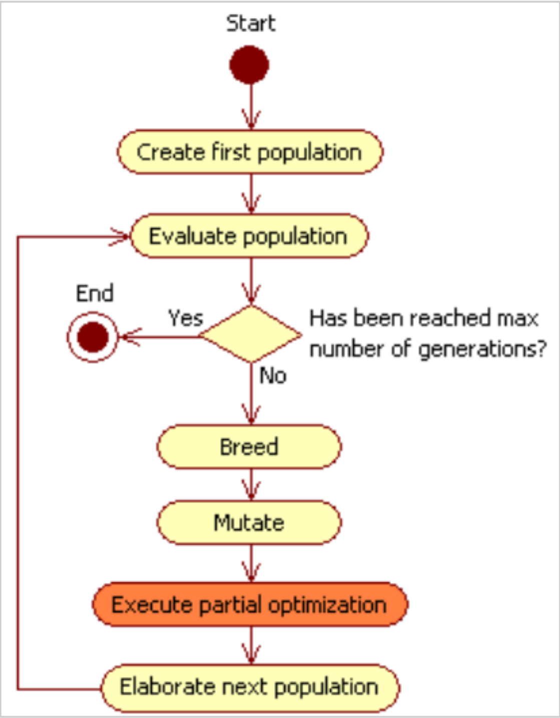

Fig. 2: The workflow of the Hybrid Genetic Algorithm (HGA)

Simulating Results

1. Instance 1 (Small):
   - Genetic Algorithm (GA):
     - ✓ Achieved a make span of 150-time units in approximately 30 sec
     - ✓ The near-optimal solution, with a gap of about 2-3% from the optimal
     - ✓ Provides a good balance of solution quality and speed
   - MILP (Exact Method):
     - ✓ Achieved the optimal make span of 145 time units in 120 sec
     - ✓ Computationally feasible for smaller instances, ensuring the best possible solution
2. Instance 2 (Large):
   - Genetic Algorithm (GA):
     - ✓ Achieved a make span of 460-time units in about 120 sec
     - ✓ Near-optimal with a gap of 5-6% from the MILP's solution
     - ✓ Much faster than MILP, making it suitable for large-scale applications
   - MILP (Exact Method):
     - ✓ Achieved an optimal make span of 435 time units
     - ✓ Required more than 3,600 sec (over 1 h) to find the solution
     - ✓ Becomes impractical for real-time applications due to high computational time

For small instances (Instance 1) mentioned in Table (2), the MILP method can find the exact optimal solution with a reasonable computational effort. The GA also performs well, providing near-optimal solutions much faster.

For larger instances (Instance 2), the GA significantly outperforms MILP in terms of computational time while still providing near-optimal solutions. MILP struggles with high computational costs and may not be suitable for real-time or large-scale problems.

Table 2: Comparison of instances

Instance	Method	Make span (time units)	Computational time (sec)	Optimality
Instance 1	GA	150	30	Near-optimal (2-3% gap)
	MILP	145	120	Optimal
Instance 2	GA	460	120	Near-optimal (5-6% gap)
	MILP	435	>3,600 (1+ h)	Optimal (but very slow)

Discussion

The hybrid algorithm combines GAs' global search power with the intensification abilities of local search algorithms. This mix of exploration and exploitation allows the HGA to avoid local optima and obtain greater results. Previous research has demonstrated that adding local search mechanisms to classic GA-based techniques dramatically improves convergence rates and solution quality, especially in complicated optimization contexts. For example, previous research has shown that regular Gas (Aidi *et al*., 2021) frequently suffer from premature convergence because to their limited intensification capacities. Still, hybrid techniques, such as the one described, minimize this issue by concentrating local refinements.

The proposed HGA may require parameter tuning for different problem instances. Future research could explore adaptive parameter adjustment and hybridization with other metaheuristics, such as ant colony optimization or particle swarm optimization.

Conclusion

The hybrid genetic algorithm provides a robust and efficient solution for the quay crane scheduling problem in container terminals. The results indicate significant improvements in make span reduction and solution quality. The proposed approach offers practical applications for real-world terminal management and sets the stage for further enhancements in algorithmic optimization.

Acknowledgment

I would like to thank my supervisor for his efforts and guidance and the Port of Marsa Maroc for the information provided.

Funding Information

The authors have no support or funding to report.

Author's Contributions

Aidi Sanaa: Contributed to the conceptualization, formal analysis, methodology section, coding, validation, and originally written phases of the study.

Torbi Imane and Mazouzi Mohamed: Contributed to the conceptualization, supervision, review, edited, and administration of the project.

Ethics

Accurately and thoroughly represents the authors' research and analysis. The work acknowledges the valuable contributions of coauthors and researchers. The findings are properly put into research history. The text cites all references and related works.

References

Aidi, S., & Mazouzi, M. (2023). Optimization Approach for Yard Crane Scheduling Problem Using Genetic Algorithm in Container Terminals. *ITM Web of Conferences*, *52*, 02002. https://doi.org/10.1051/itmconf/20235202002

Aidi, S., Torbi, I., Mazouzi, M., & Belassiria, I. (2021). Planning of One-Way Quay Cranes Using the Genetic Algorithm. *SHS Web of Conferences*, *119*, 07005. https://doi.org/10.1051/shsconf/202111907005

Bierwirth, C., & Meisel, F. (2009). A Fast Heuristic for Quay Crane Scheduling with Interference Constraints. *Journal of Scheduling*, *12*(4), 345–360. https://doi.org/10.1007/s10951-009-0105-0

Bierwirth, C., & Meisel, F. (2010). A Survey of Berth Allocation and Quay Crane Scheduling Problems in Container Terminals. *European Journal of Operational Research*, *202*(3), 615–627. https://doi.org/10.1016/j.ejor.2009.05.031

Chen, J. H., Lee, D.-H., & Goh, M. (2014). An Effective Mathematical Formulation for the Unidirectional Cluster-Based Quay Crane Scheduling Problem. *European Journal of Operational Research*, *232*(1), 198–208. https://doi.org/10.1016/j.ejor.2013.06.051

El-Abbasy, M. S., Ahmed, E., Zayed, T., Alfalah, G., & Alkass, S. (2021). Multi-Level Fleet Size Optimization for Containers Handling Using Double-Cycling Strategy. *Expert Systems with Applications*, *170*, 114526. https://doi.org/10.1016/j.eswa.2020.114526

Expósito-Izquierdo, C., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2013). Hybrid Estimation of Distribution Algorithm for the Quay Crane Scheduling Problem. *Applied Soft Computing*, *13*(10), 4063–4076. https://doi.org/10.1016/j.asoc.2013.05.006

Gulić, M., & Žuškin, M. (2023). Enhancing Metaheuristic Optimization: A Novel Nature-Inspired Hybrid Approach Incorporating Selected Pseudorandom Number Generators. *Algorithms*, *16*(9), 413. https://doi.org/10.3390/a16090413

Haghani, A., & Jung, S. (2005). A Dynamic Vehicle Routing Problem with Time-Dependent Travel Times. *Computers and Operations Research*, *32*(11), 2959–2986. https://doi.org/10.1016/j.cor.2004.04.013

Homayouni, S. M., & Tang, S. H. (2013). Multi Objective Optimization of Coordinated Scheduling of Cranes and Vehicles at Container Terminals. *Mathematical Problems in Engineering*, *2013*(1), 1–9. https://doi.org/10.1155/2013/746781

Hop, D. C., Van Hop, N., & Anh, T. T. M. (2021). Adaptive Particle Swarm Optimization for Integrated Quay Crane and Yard Truck Scheduling Problem. *Computers and Industrial Engineering*, *153*, 107075. https://doi.org/10.1016/j.cie.2020.107075

Pap, E., Bojanić, V., Bojanić, G., & Georgijević, M. (2013). Quay Crane Scheduling for River Container Terminals. In *Intelligent Systems: Models and Applications* (Vol. 3, pp. 285–300). Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33959-2_15

Song, J., & Xu, C. (2024). Optimization for Container Truck Routing in Container Terminal with Multi Quay Cranes Considering Emissions Policy. *Scientific Reports*, *14*(1), 24000. https://doi.org/10.1038/s41598-024-75661-1

Skaf, A., Lamrous, S., Hammoudan, Z., & Manier, M.-A. (2021). Solving Methods for the Quay Crane Scheduling Problem at Port of Tripoli-Lebanon. *RAIRO-Operations Research*, *55*(1), 115–133. https://doi.org/10.1051/ro/2020135

Zhang, Z., Liu, M., Lee, C.-Y., & Wang, J. (2018). The Quay Crane Scheduling Problem with Stability Constraints. *IEEE Transactions on Automation Science and Engineering*, *15*(3), 1399–1412. https://doi.org/10.1109/tase.2018.2795254

Zhao, N., Lodewijks, G., Fu, Z., Sun, Y., & Sun, Y. (2022). Trajectory Predictions with Details in a Robotic Twin-Crane System. *Complex System Modeling and Simulation*, *2*(1), 1–17. https://doi.org/10.23919/csms.2021.0028