Transport and Telecommunication, 2025, volume 26, no. 1, 13-22
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2025-0002

OPTIMIZATION OF RESTRICTED CONTAINER RELOCATION USING THE MONTE CARLO TREE SEARCH METHOD

***Abdelali Chaabane*[1]*, Khadidja Yachba*[2]*, Ladjel Bellatreche*[3]**

[1]*Relizane University*
Relizane, Algeria
abdelalichaabane74@gmail.com
[2]*Oran Graduate School of Economics*
Oran, Algeria
yachba.khedidja@ese-oran.dz
[3]*National Higher School of Mechanics and Aerotechnology*
Poitiers, France
ladjel.ballatreche@ensma.fr

This article explores how to improve operational performance in maritime ports by managing the flow of goods effectively. This study proposes an innovative approach based on Reinforcement Learning (RL), specifically the Monte Carlo Tree Search (MCTS) method, to address the restricted container relocation problem (RCRP). This method aims to determine an optimal sequence for container retrieval based on their respective priorities, in order to minimize the number of necessary relocations. By employing precise actions and a defined reward function, MCTS is guided towards the best possible solution. The efficiency and relevance of this method are demonstrated through various solved scenarios and compared to a literature-based approach using genetic algorithms. The results show that the MCTS approach is effective in addressing the complex challenges of goods flow management in maritime ports.

Keywords: Container relocation, optimization of maritime operations, container ship, reinforcement learning, Monte Carlo Tree Search

1. Introduction

Containerized cargo transport is a vital pillar of global trade, container terminals play a central role in this process by facilitating the efficient transfer of containers between different modes of transportation. In this context, optimal management of container relocation within terminals holds crucial importance in enhancing operational performance and reducing costs.

This article focuses on solving the RCRP, a complex challenge encountered in container terminal operations. The RCRP aims to minimise the number of relocations required during the container retrieval process, which can have a significant impact on the overall terminal performance.

The RCRP represents a major challenge in combinatorial optimization (Jovanovic *et al*., 2019) in the management of warehouses and yards. Due to space constraints, containers are often stacked or arranged side by side, resulting in various configurations in terms of width, height, and length. Careful planning is required for efficient container retrieval, aiming to minimize both the required relocations and associated costs.

Over the years, a multitude of approaches have been developed to address the Container Relocation Problem (CRP), each with its own advantages and disadvantages. Many studies have thus contributed to this issue by proposing various methods. More recently, the use of genetic approaches has led to several innovative proposals. One research introduced the HBRKGA (Da Silva Firmino & Cesário Times, 2024), a hybrid approach combining Biased Random-Key Genetic algorithm with a metaheuristic, aiming to minimize crane operating time while ensuring the retrieval of all containers. Similarly, another study (Đurasević & Đumić, 2024) adapted Genetic Programming (GP) to automatically design Relocation Rules (RR), highlighting their effectiveness compared to manually designed RRs (Maglić *et al*., 2019). Furthermore, the application of GP to derive property functions integrated into relocation rules has been examined to reduce energy consumption (Đurasević *et al.*, 2023). Research has also explored the use of GP to minimize relocations in real container bays at ports (Gulić *et al.*, 2022). Additionally, a study presented a rule-based SPFH heuristic aiming to minimize both the number of necessary relocations and the complexity of the resolution process (Zhou & Zhang, 2024).

Other research has also explored the integration of machine learning techniques, such as the use of the Trust Region Policy Optimization method (Wei *et al.*, 2021), an RL heuristic (Jiang *et al.*, 2021), and an MLUB upper bound method (Zhang *et al.,* 2020), to solve the CRP. In (Ye *et al.,* 2023), supervised learning models were used to analyze factors influencing container movements. The results demonstrate that using Random Forest (RF) achieves a classification accuracy rate of 94% for the restricted Block Relocation Problem (BRP) model. Furthermore, in (Liu *et al*., 2023), a deep reinforcement learning-based method is introduced for the first time for the CRP, a major challenge in port operations. This approach integrates a dynamic attention model to adapt to changes in the container yard configuration.

Moreover, various heuristic and exact algorithmic approaches (Tanaka & Voß, 2019) have been proposed to solve the CRP. Among them, a new heuristic approach to solving the Stochastic Container Relocation Problem (SCRP) is presented in (Bacci *et al.,* 2023), aiming to minimize the number of moves required to retrieve containers in uncertain retrieval orders. Another study focuses on the SCRP, introducing a flexible service policy to reduce relocations and waiting times (Feng et al., 2020) using stochastic dynamic programming and tree-based search algorithms. One study also examined the SCRP. Researchers proposed two main tools to solve these problems: an optimal algorithm called Pruning-Best-First-Search (PBFS) and a randomized approximate algorithm called PBFS-Approximate. These algorithms are designed to minimize the number of relocations required while considering the uncertainty associated with container retrieval order (Galle *et al.,* 2018). Another hybrid approach: the branch-and-bound algorithm combines a local search heuristic to solve the CRP *(Zweers et al*., 2020) by dividing it into two phases: preprocessing and relocation. A recent revision of the problem formulation aims to improve the efficiency of a mixed-integer programming model previously introduced by Galle, Barnhart, and Jaillet (Galle *et al.*, 2018). This revision corrects the shortcomings associated with the LIFO policy, thus enabling better problem resolution (Jin, 2020).

Other approaches focus on the BRP. Some studies combine it with Appointment Planning (Azab & Morita, 2022b, 2022a) to reduce both the number of required relocations and truck waiting times within the terminal, using a mixed-integer linear model. Additionally, methods such as the Bounded Beam Search algorithm (Bacci *et al*., 2019) and a mixed-integer linear model (Kimms & Wilschewski, 2023) are proposed to specifically address the BRP in its restricted version.

In older works, research also suggested two approaches B&B and heuristic rules based on probability theory (Kim & Hong, 2006) to select storage locations for relocated blocks and identify blocks to retrieve. Additionally, one study addressed the CRP in its restricted and unrestricted versions using iterative A* deepening algorithms, with fixed rules to produce an operational plan aiming to retrieve all containers in a given pickup sequence (Zhu *et al.*, 2012).

This work is a part of the research conducted in the fields of decision support systems, transportation, maritime transportation (Abdelali *et al*., 2023; Yachba *et al*., 2024; Yachba *et al.*, 2022), logistics (Yachba *et al*., 2021), optimization (Belayachi *et al.,* 2017; Amrani *et al*., 2018; Yachba *et al.,* 2018), and multicriteria methods (Yachba *et al*., 2015; Yachba *et al*., 2018; Tahiri *et al*., 2022; Tahiri *et al*., 2020).

Our objective is to contribute to the improvement of container terminal operations by adopting the MCTS method to solve the static restricted CRP. Unlike existing articles, which use the MCTS method to address variants of the CRP, such as the one dealing with the pre-marshalling problem (Wang *et al*. 2024), involving rearranging poorly stacked containers in the same bay by combining MCTS with eight composite rearrangement rules in the selection phases, and article (Zhang *et al.,* 2023), which tackles the online relocation problem using heuristic rules and probability approximation, our approach specifically focuses on adapting MCTS by proposing a formulation and reward function that guide the method towards the best CRP solution for the static and restricted context.

2. Problem description

In this article, we address the challenge of restricted relocation within port container storage areas, a crucial issue for optimizing port operations efficiency. The objective is to find an optimal retrieval sequence for containers within a given bay, based on their respective priorities, in order to minimize the number of necessary relocations. In this context, containers are arranged in blocks, with each block subdivided into several bays, themselves composed of multiple stacks of containers. Each stack contains multiple levels, forming a complex storage structure, as illustrated in Figure 1. The characteristics of the bays vary, with a capacity of up to 20 bays, and diverse configurations ranging from 2 to 10 stacks and from 3 to 7 levels per bay. The assumptions and underlying constraints provide a clear formulation of our model:

- The bay is characterized by two dimensions: the number of stacks and the number of levels.
- The problem is limited to one bay at a time.
- The initial configuration of the container bay is pre-known.
- All containers are the same size.
- Each container has a unique pre-defined priority.
- Information regarding container locations, their retrieval priorities, and operational constraints are available in advance.

Specific problem constraints include:

- Only relocations of containers blocking access to the highest priority container are allowed.
- The problem is static: no arrival of new containers is permitted during the retrieval process.
- Containers are relocated until the target stack is full.
- Only the topmost container can be handled at a time.
- Relocation operations occur between two stacks as long as it's possible, provided the target stack is not full.
- The maximum stacking height is determined by the maximum retrieval height of the latest Rubber Tyred Gantry (RTG) cranes.
- The bay's width corresponds to the reach of the latest RTG cranes.

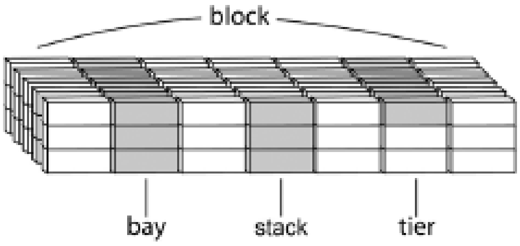

Figure 1. Block in a container yard (Ozcan & Eliiyi, 2017)

3. Our approach

3.1. Monte Carlo Tree Search (MCTS)

The MCTS method is a tree search algorithm first introduced in 2006 in the domain of games, notably with its application to the game of Go, where it has become a classic method of artificial intelligence. MCTS belongs to the family of RL methods, and similar to Monte Carlo methods, the agent adjusts its representations based on the results of random simulations. In MCTS, the agent accumulates reinforcements in a search tree, allowing it to gradually refine its knowledge of the problem (Fabbri, 2015).

MCTS is an algorithm that builds a tree in memory. Each node of the tree retains statistics indicating the frequency with which a move is played from a given state $N(s;a)$, the number of times each move is played from that state $N(s)$, and the average reward $Q(s;a)$ obtained after applying the move a in the state s. The tree is constructed iteratively by simulating actions in the game, choosing moves based on the statistics stored in the nodes (Perez *et al*., 2015).

Each iteration of MCTS can be divided into several stages, as introduced by (Chaslot *et al*., 2008): Tree Selection, Expansion, Monte Carlo Simulation, and Backpropagation (all summarized in Figure 2).

At the beginning of the algorithm, the tree is only formed by the root node, which contains the current state of the game. During the selection stage, the tree is traversed from the root until a maximum depth or the end of the game is reached (Perez *et al*., 2015).

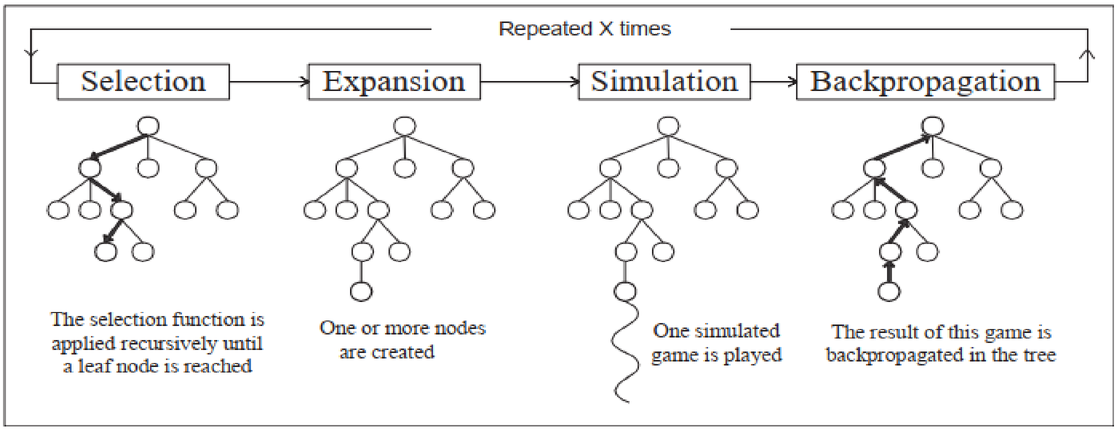

Figure 2. Steps of the MCTS algorithm (Chaslot *et al*., 2008)

In each of these action decisions, MCTS establishes a balance between exploitation and exploration. In other words, it chooses between an action that leads to states with the best result found so far and a move to explore less explored game states, respectively. To achieve this, Kocsis & Szepesvári (2006) applied the Upper Confidence Bound (UCB1, see Equation 1) as a tree policy.

$$a^* = \underset{a \in A(s)}{arg\,max} \left\{ Q(s,a) + C\sqrt{\frac{ln\,N(s)}{N(s,a)}} \right\}. \tag{1}$$

The balance between exploration and exploitation can be adjusted by changing C. Higher values of C give additional weight to the second term of Equation 1 of UCB1, preferring actions that have been less explored, at the expense of actions with the highest average reward $Q(s;a)$. A commonly used value for single-player games is √2, as it balances both facets of the search when values are normalized between 0 and 1. The value of C depends on the application and may vary from one game to another. It is worth noting that MCTS, when combined with UCB1, achieves asymptotically logarithmic regret on each node of the tree (Coquelin *et al.*, 2007).

If, during the tree selection phase, a node has fewer children than the number of available actions from a given position, a new node is added as a child of the current node (expansion phase), and the simulation step begins. At this stage, MCTS executes a Monte Carlo simulation (or rollout; default policy) from the expanded node. This simulation is conducted by choosing random actions (either uniformly random or biased) until the end of the game or until a predefined depth is reached, allowing for the evaluation of the game state (Perez *et al.*, 2015).

Finally, during the backpropagation step, the statistics $N(s)$, $N(s;a)$, and $Q(s;a)$ are updated for each visited node using the following formulas (Sutton & Barto, 2018):

$$R = r + \gamma R, \tag{2}$$

$$Q(s,a) = Q(s,a) + \frac{1}{N(s,a)}[R - Q(s,a)], \tag{3}$$

where R represents the updated cumulative discounted reward received from the simulation from s until the end of the simulation, r is the reward received at a given state, and γ is the discount factor.

3.2. Application of MCTS for solving the RCRP

We approach MCTS by modeling it as a Markov Decision Process (MDP). The latter is characterized by a set of 5-tuple (S, A, R, T, π), where:

- S is a finite set of states.
- A is a finite set of actions available from state S.
- R is the immediate reward (or expected immediate reward).
- T is the transition function from state s_t to state s_{t+1}.
- π is the strategy or policy.

The problem of MDP is to find a policy π that specifies the actions the decision-maker will choose when in state s_t.

The following notations will be used to describe our approach:

- B: Bay
- P: Stacks
- P_i: Stack i
- C: Containers
- s_t: state of the bay in time t
- $a_i(c, P_j)$: Action applied to container c
- N $(a_i(c, P_j)\ |s_t\ )$: Number of containers in the bay in state t after applying action a to the container c

MDP for RCRP

The MDP for RCRP is formulated according to the principles of MDP. Figure 3 illustrates an example of a container relocation MDP:

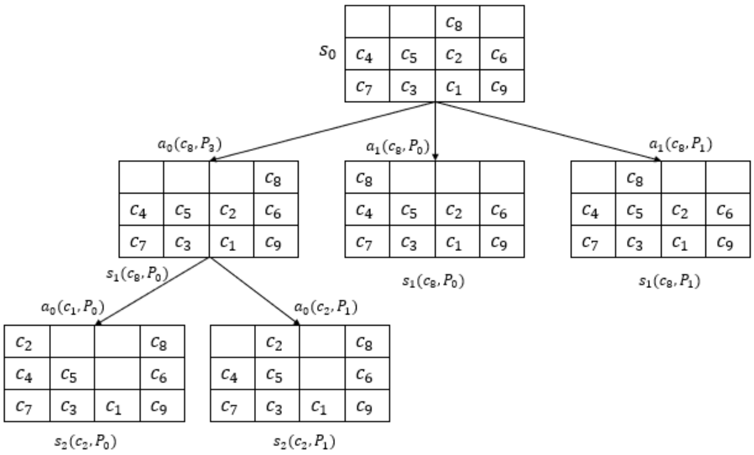

Figure 3. MDP for RCRP

Relocation State

In the context of RCRP, the variable t represents the relocation sequence, where $t = 0$ corresponds to the initial state of the bay. When $t = 1$, it is the next state after the first container has been retrieved or relocated, and so on. In Figure 3, s_0 represents the initial state of the bay, then s_1 represents the next state when the first container c_8 is relocated, then s_2. When in s_0, multiple actions are available, which involve relocating a container blocking the target container to an available slot. $s_1(c_8, p_3)$ means relocating container c_8 to stack 3 in state s_1. $s_1(c_8, p_0)$ means relocating container c_8 to stack 0 in state s_1. In s_1, two actions are possible after relocating container c_8: $s_2(c_2, p_0)$ or $s_2(c_2, p_1)$ meaning relocating c_2 to stack 0 or 1. Then state s_3 where container c_1 is retrieved.

Action applied for RCRP

In the context of restricted CRP, it involves finding the optimal location to move an obstructive container to retrieve a target container. The available actions vary depending on the state of the bay, selecting stacks that contain free slots except for the one containing the target container (Actions [stack 1, stack 2,..., stack N]).

Reward calculation for RCRP:

In the framework of RL, the reward reflects the dynamics of the environment. In the context of RCRP, the reward is primarily based on objectives and constraints. In our work, we relied on the number of relocations to evaluate the chosen action at each step:

$$NR_t = \begin{cases} +1 & if \quad N\left(a_i(c, P_j) \mid s_{t-1}\right) > N\left(a_i(c, P_k) \mid s_t\right) \\ -1 & Otherwise \end{cases} \tag{4}$$

- Each relocation is penalized as a negative action (-1).
- Each access to the target container is rewarded as a positive action (+1).

The total number of relocations is calculated by summing the individual relocation numbers between the initial state t and the final state f where all containers are retrieved, as represented by the following formula:

$$TNR = \sum_t^f NR_t. \tag{5}$$

The cumulative reward is calculated using Equation 2 with the discount factor γ, which is decreased at each step. This means that the value of the reward decreases at each step, indicating a less optimal solution. For example, $R = r + (\gamma^5)\, TNR$ if the final state is reached after 5 iterations, and $R = r + (\gamma^7) TNR$ if the final state is reached after 7 iterations.

4. Results and discussion

In this section, we present the results obtained with our approach, highlighting its effectiveness through a comparison with a previous method (Gulić et al. 2022) that utilizes a genetic algorithm. This method has demonstrated its efficiency compared to several other approaches, such as the methods proposed in (Caserta *et al.*, 2012; Jovanovic & Voß, 2014; Maglić *et al.*, 2019; Zhu *et al*., 2012).

This comparison revolves around two main criteria: fitness value and the execution time required to achieve an optimal result. The experimental program was developed using Python with the Anaconda environment and was executed on a personal computer equipped with an Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz 2.40 GHz processor and 8 GB of RAM. We conducted an evaluation of the MCTS method by varying several parameters, including the value of C for exploration/exploitation in Equation 1, γ for the discount factor in Equation 2, as well as the number of iterations, aiming to obtain high-quality solutions in minimal time. The best-found parameters are: $C = \sqrt{2}$, $\gamma = 0.8$, and $t = 200$ iterations.

The evaluation was performed on a set of randomly generated RCRP instances, covering different bay configurations ranging from 3 to 6 stacks and from 3 to 6 levels, in accordance with the specifications of the latest RTG cranes regarding the actual dimensions of the bays. For each configuration, we used 20 distinct instances to ensure adequate representation of the variability of encountered problems.

The number of containers in each bay is determined using the following formula, as presented in (Gulić *et al.,* 2022), to allow fair comparison:

$$N = W * H - (H - 1). \tag{6}$$

Here, N denotes the total number of stacked containers, while W and H respectively represent the total number of stacks and levels in a bay.

The objective function used for comparison is based on calculating the total number of relocations required. This includes counting the retrieval and relocation operations needed to retrieve all containers. Unlike the reward function we used in Equation 4, which assigns a value of -1 to relocation operations.

Table 1 presents a comparison between the performance of our method and the comparison approach (GA) in terms of fitness value and execution time for different bay configurations. Our results show that in most cases, our method has outperformed the comparison approach in terms of fitness value, with slight differences observed in some instances of the 6×6 configuration. Additionally, our method succeeded in obtaining optimal solutions in relatively short timeframes for small and medium-sized instances, although a slight increase in execution time was observed for instances highlighted in bold in the table. This increase in execution time in our MCTS method is mainly due to the number of available actions in each state, requiring deeper exploration to find an optimal solution, especially in the 6×6 cases where the number of possible actions in the initial state is higher. The results also confirm that our method offers optimal performance for small and medium-sized bays, demonstrating its effectiveness in solving container relocation problems.

Table 1. The results of the RCRP tests on the actual bay sizes obtained by both the suggested methods and the one proposed by GA (Gulić *et al*., 2022)

Nbr. of tiers × Nbr. of stacks		**The average total number of relocations**		**Average computation time (s)**	
	Nbr. of Containers	**GA**	**MCTS**	**GA**	**MCTS**
3 × 3	7	10.8	10.4	3.9905	0.28185743
4 × 3	9	13.8	13.6	5.261	1.02481785452
5 × 3	11	19.8	18.4	5.6295	1.371012779
6 × 3	13	20.6	18.8	5.978	1.43042665257
3 × 4	10	15.2	14.6	5.18525	0.502432
4 × 4	13	22.8333	22.1667	6.635	3.6602
5 × 4	16	31.6667	30.5	6.683	5.702949
6 × 4	19	38.1667	35.1667	7.511	5.9640222
3 × 5	13	18.4	18.2	6.16775	3.09212
4 × 5	17	28.4286	26.7143	7.163	5.3895063
5 × 5	21	42.3	40.14	**8.762**	**8.84455**
6 × 5	25	45.8333	44.5	**9.32**	**12.28037**
3 × 6	16	21.8	21.2	6.6205	3.956002
4 × 6	21	35.45	32.5	7.7516	5.807023
5 × 6	26	41.4	40.2	**12.534**	**15.6**
6 × 6	31	**64.83**	**65.32**	**44.837**	**52.33**

Furthermore, we conducted a comparison between the results obtained by our MCTS method and those of the GA method, using the 20 instances from the large-scale test set for a bay of dimensions 4×6. Figure 4 illustrates this comparison in terms of the objective function. It is notable that, for the majority of instances in the test set, the MCTS method manages to find a solution with fewer additional relocations than the GA method. Moreover, a clear difference in terms of execution time is observed in Figure 5, confirming the effectiveness and speed of our proposed approach compared to the GA method.

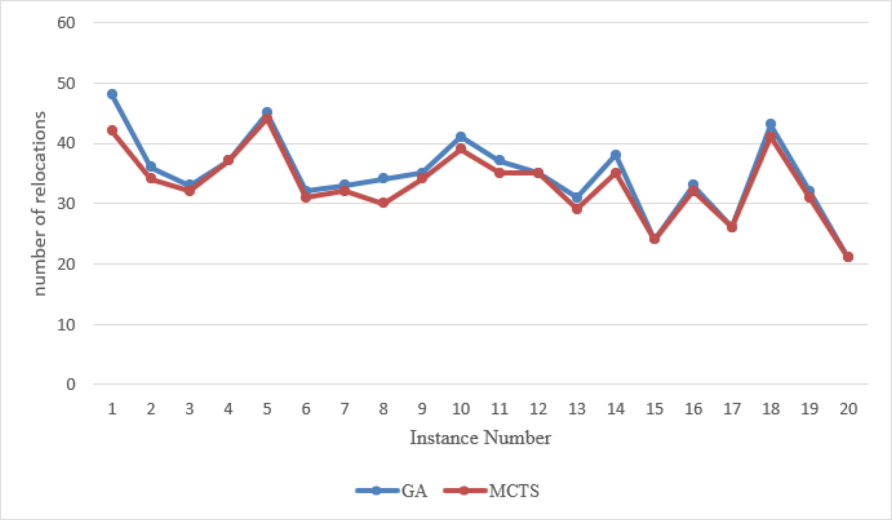

Figure 4. A comparative evaluation between the results of our approach and those of GA for the 20 instances of the test set, with a bay of dimensions 4×6, in terms of the value of the objective function

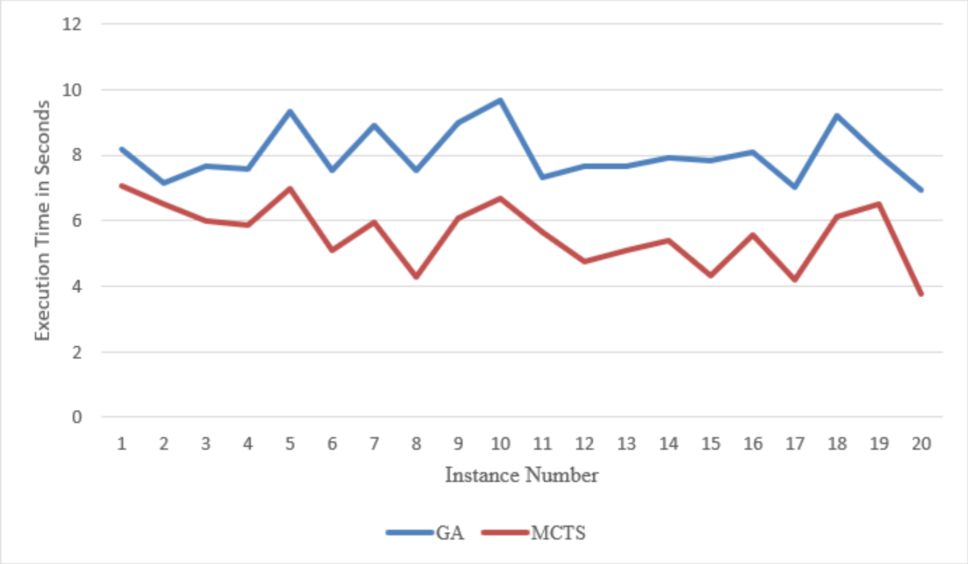

Figure 5. A comparison of the execution times between our approach and GA for the 20 instances of the test set, featuring a bay of dimensions 4×6

5. Conclusion

In the complex domain of transportation logistics, our work proposed an innovative approach for container relocation in maritime ports, using the Monte Carlo Tree Search (MCTS) method in a restricted context. Our approach aims to minimize the number of relocations by finding an optimal sequence for container retrieval. By adjusting parameters and constructing a well-suited reward function, we succeeded in extracting the best actions from the search tree, enabling MCTS to find perfect solutions within seconds. By comparing our method with a genetic algorithm-based approach in the literature, we observed promising results. Our experiments demonstrated that our MCTS approach generally provided higher-quality solutions in a shorter time frame. This increased efficiency can be attributed to MCTS's ability to rapidly explore a wide search space while making optimal decisions. Thus, our approach proves to be an effective and efficient solution for solving the challenge of container relocation in maritime ports, even in a restricted context.

Continuing our research on logistical challenges, our next project promises to be equally stimulating. We plan to apply our method to other contexts, such as the unrestricted relocation problem. Additionally, we intend to explore the use of advanced machine learning techniques to address other crucial aspects of transportation management. Finally, in-depth comparative studies with other optimization methods for container relocation could provide valuable insights into the specific advantages of the MCTS approach in this field.

References

1. Abdelali, C., Khadidja, Y., & Bellatreche, L. (2023) Optimization of dangerous goods stowage : A hybrid approach between faster R-CNN and coronavirus metaheuristic. In: *2023 IEEE International Conference on Big Data (BigData),* 2561-2569. https://doi.org/10.1109/BigData59044.2023.10386793.
2. Amrani, F., Yachba, K., Belayachi, N., & Hamdadou, D. (2018) A decision-making system for container storage management in a seaport using the ant-colony optimisation algorithm. *International Journal of Management and Decision Making,* 17(3), 348-367.
3. Azab, A., & Morita, H. (2022a) A zero-blockage based scheduling for import containers pickup operations at container terminal yards. In: *Proceedings of the 11th International Conference on Operations Research and Enterprise Systems*, 286-293. https://doi.org/10.5220/0010885000003117.
4. Azab, A., & Morita, H. (2022b) The block relocation problem with appointment scheduling. *European Journal of Operational Research*, 297(2), 680-694. https://doi.org/10.1016/j.ejor.2021.06.007.
5. Bacci, T., Mattia, S., & Ventura, P. (2019) The bounded beam search algorithm for the block relocation problem. *Computers & Operations Research*, 103, 252-264. https://doi.org/10.1016/j.cor.2018.11.008.
6. Bacci, T., Mattia, S., & Ventura, P. (2023) The realization-independent reallocation heuristic for the stochastic container relocation problem. *Soft Computing,* 27(7), 4223-4233. https://doi.org/10.1007/s00500-022-07070-3.
7. Belayachi, N., Gelareh, S., Yachba, K., & Bouamrane, K. (2017) The logistic of empty containers′ return in the liner-shipping network. *Transport and Telecommunication Journal,* 18(3), 207-219.
8. Bendaoud, Z., & Yachba, K. (2017) Towards a decision support system for optimization of container placement in a container terminal. *International Journal of Strategic Information Technology and Applications* (IJSITA), 8(3), 59-72.
9. Caserta, M., Schwarze, S., & Voß, S. (2012). A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research,* 219(1), 96-104. https://doi.org/10.1016/j.ejor.2011.12.039
10. Chaslot, G., Winands, M.H.M., Jaap Van Den Herik, H., Uiterwijk, J. W. H. M., Bouzy, B. (2008) Progressive Strategies for Monte-Carlo Tree Search. *New Mathematics and Natural Computation*, 04(03), 343-57. doi: 10.1142/S1793005708001094.
11. Coquelin, P.-A., Munos, R. (2007) Bandit algorithms for tree search. https://doi.org/10.48550/ARXIV.CS/0703062.
12. Da Silva Firmino, A., & Cesário Times, V. (2024) A hybrid biased random-key genetic algorithm for the container relocation problem. In M. Khosravy, N. Gupta, & O. Witkowski (Éds.), *Frontiers in Genetics Algorithm Theory and Applications,* 55-80. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-8107-6_4.
13. Đurasević, M., & Đumić, M. (2024) Designing relocation rules with genetic programming for the container relocation problem with multiple bays and container groups. *Applied Soft Computing*, 150, 111104. https://doi.org/10.1016/j.asoc.2023.111104
14. Đurasević, M., Đumić, M., Čorić, R., & Gil-Gala, F. J. (2023). *Automated design of relocation rules for minimising energy consumption in the container relocation problem* (arXiv:2307.01513). arXiv. http://arxiv.org/abs/2307.01513
15. Fabbri, A. (2015) Dynamique d'apprentissage pour Monte Carlo Tree Search : Applications aux jeux de Go et du Clobber solitaire impartial. *Université Claude Bernard - Lyon 1*, 214.
16. Feng, Y., Song, D.-P., Li, D., & Zeng, Q. (2020) The stochastic container relocation problem with flexible service policies. *Transportation Research Part B: Methodological*, 141, 116-163. https://doi.org/10.1016/j.trb.2020.09.006.

17. Galle, V., Barnhart, C., & Jaillet, P. (2018) A new binary formulation of the restricted Container Relocation Problem based on a binary encoding of configurations. *European Journal of Operational Research*, 267(2), 467-477. https://doi.org/10.1016/j.ejor.2017.11.053.
18. Galle, V., Manshadi, V. H., Boroujeni, S. B., Barnhart, C., & Jaillet, P. (2018) The Stochastic Container Relocation Problem. *Transportation Science*, 52(5), 1035-1058. https://doi.org/10.1287/trsc.2018.0828.
19. Gulić, M., Maglić, L., Krljan, T., Maglić, L. (2022) Solving the container relocation problem by using a metaheuristic genetic algorithm. *Applied Sciences,* 12(15), 7397. doi:10.3390/app12157397.
20. Jiang, T., Zeng, B., Wang, Y., & Yan, W. (2021) A new heuristic reinforcement learning for container relocation problem. *Journal of Physics: Conference Series,* 1873(1), 012050. https://doi.org/10.1088/1742-6596/1873/1/012050.
21. Jin, B. (2020) On the integer programming formulation for the relaxed restricted container relocation problem. *European Journal of Operational Research*, 281(2), 475-482. https://doi.org/10.1016/j.ejor.2019.08.041.
22. Jovanovic, R., Tanaka, S., Nishi, T., & Voß, S. (2019) A GRASP approach for solving the Blocks Relocation Problem with Stowage Plan. *Flexible Services and Manufacturing Journal*, 31(3), 702-729. https://doi.org/10.1007/s10696-018-9320-3.
23. Jovanovic, R., & Voß, S. (2014). A chain heuristic for the Blocks Relocation Problem. *Computers & Industrial Engineering*, 75, 79-86. https://doi.org/10.1016/j.cie.2014.06.010.
24. Khadidja, Y., Amina, F. I., & Eddine, B. S. (2024). Enhancing Transportation Efficiency with Optimal Container Placement Using the Bat Algorithm. *Transport and Telecommunication Journal,* 25(2), 200-208.
25. Kim, K. H., & Hong, G.-P. (2006). A heuristic rule for relocating blocks. *Computers & Operations Research,* 33(4), 940-954. https://doi.org/10.1016/j.cor.2004.08.005.
26. Kimms, A., & Wilschewski, F. (2023) A new modeling approach for the unrestricted block relocation problem. *OR Spectrum*, 45(4), 1071-1111. https://doi.org/10.1007/s00291-023-00728-w.
27. Kocsis, L., & Szepesvári, C. (2006) Bandit Based Monte-Carlo Planning. In J. Fürnkranz, T. Scheffer, & M. Spiliopoulou (Éds.), *Machine Learning: ECML 2006,* 4212, 282-293. Springer Berlin Heidelberg. https://doi.org/10.1007/11871842_29.
28. Liu, F., Ye, T., & Zhang, Z. (2023) Dynamic Attention Model – A Deep Reinforcement Learning Approach for Container Relocation Problem. In H. Fujita, Y. Wang, Y. Xiao, & A. Moonis (Éds.), *Advances and Trends in Artificial Intelligence. Theory and Applications,* 13926, 273-285. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36822-6_24.
29. Maglić, L., Gulić, M., & Maglić, L. (2019) Optimization of Container Relocation Operations in Port Container Terminals. *Transport*, 35(1), 37-47. https://doi.org/10.3846/transport.2019.11628.
30. Ozcan, S., & Eliiyi, D. T. (2017) A reward-based algorithm for the stacking of outbound containers. *Transportation Research Procedia*, 22, 213-221. https://doi.org/10.1016/j.trpro.2017.03.028.
31. Perez, D., Mostaghim, S., Samothrakis, S., & Lucas, S. M. (2015) Multiobjective Monte Carlo Tree Search for Real-Time Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4), 347-360. https://doi.org/10.1109/TCIAIG.2014.2345842
32. Sutton, R. S., & Barto, A. G. (2018) Reinforcement learning: An introduction (Second edition). The MIT Press.
33. Tahiri, H., Khadidja, Y., & Bouamrane, K. (2022) Towards a multi-criteria decision approach to solving the container storage problem: container ship loading component. *International Journal of Organizational and Collective Intelligence (IJOCI),* 12(1), 1-21.
34. Tahiri, H., Yachba, K., & Bouamrane, K. (2020) A container placement approach in a container-ship based on the electre 2 method. In: *4th International Symposium on Informatics and its Applications (ISIA*), IEEE, 1-5.
35. Tanaka, S., & Voß, S. (2019). An exact algorithm for the block relocation problem with a stowage plan. *European Journal of Operational Research*, 279(3), 767-781. https://doi.org/10.1016/j.ejor.2019.06.014.
36. Wang, Z., Chenhao, Zh., Ada, Ch., Jingkun, G. (2024) A policy-based Monte Carlo Tree Search method for container pre-marshalling. *International Journal of Production Research,* 62(13), 4776-92. doi:10.1080/00207543.2023.2279130.
37. Wei, L., Wei, F., Schmitz, S., Kunal, K., Noche, B. (2021) Optimization of container relocation problem via reinforcement learning. *Logistics Journal: Proceedings – ISSN 2192-9084,* 17. doi: 10.2195/lj_Proc_wei_en_202112_02.

38. Yachba, K., Bendaoud, Z., & Bouamrane, K. (2018) A technique for resolution of the assignment problem containers in a container terminal. In: *Handbook of Research on Biomimicry in Information Retrieval and Knowledge Management*, IGI Global, 120-132.
39. Yachba, K., Bendaoud, Z., & Bouamrane, K. (2018) Toward a decision support system for regulation in an urban transport network. *International Journal of Strategic Information Technology and Applications (IJSITA),* 9(2), 1-17.
40. Yachba, K., Bouamrane, K., & Gelareh, S. (2015) Containers storage optimization in a container terminal using a multimethod multi-level approach. In: *The International Conference on Computers & Industrial Engineering (CIE45),* 28-30.
41. Yachba, K., Chaabane, A., & Benadda, M.A. (2021) Un système d'aide à la décision pour l'optimisation de processus de distribution des produits finaux. In : *Séminaire international sur les mathématiques et l'informatique*, Février, Oran.
42. Yachba, K., Gelareh, S., & Bouamrane, K. (2016) Storage management of hazardous containers using the genetic algorithm. *Transport and Telecommunication Journal*, 17(4), 371-383.
43. Ye, R., Ye, R., & Zheng, S. (2023) Machine Learning Guides the Solution of Blocks Relocation Problem in Container Terminals. Transportation Research Record: Journal of the Transportation Research Board, 2677(3), 721-737. https://doi.org/10.1177/03611981221117157.
44. Zhang, C., Guan, H., Yuan, Y., Chen, W., & Wu, T. (2020) Machine learning-driven algorithms for the container relocation problem. *Transportation Research Part B: Methodological*, 139, 102-131. https://doi.org/10.1016/j.trb.2020.05.017.
45. Zhang, Zh., Tan, K.Ch., Qin, W., Li, Y., Ek Peng Chew, E.P., Xu, K. (2023) *Online container relocation problem with retrieval probability*. doi:10.2139/ssrn.4604944.
46. Zhou, S., & Zhang, Q. (2024) Real-time batch optimization for the stochastic container relocation problem. *Applied Sciences,* 14(6), 2624. https://doi.org/10.3390/app14062624.
47. Zhu, W., Qin, H., Lim, A., & Zhang, H. (2012) Iterative deepening A* algorithms for the container relocation problem. *IEEE Transactions on Automation Science and Engineering*, 9(4), 710-722. https://doi.org/10.1109/TASE.2012.2198642.
48. Zweers, B. G., Bhulai, S., & Van Der Mei, R. D. (2020) Optimizing pre-processing and relocation moves in the stochastic container relocation problem. *European Journal of Operational Research,* 283(3), 954-971. https://doi.org/10.1016/j.ejor.2019.11.067.