Integrated Task and Motion Planning for Keel Block Optimisation at Captain Cook Graving Dock

William Jones[1], **Hanna Kurniawati**[2], **Patrik Haslum**[2], **Ian Manchester**[1]
[1]Australian Robotic Inspection and Asset Management (ARAIM) Research Hub,
The University of Sydney, Australia
[2]Australian Robotic Inspection and Asset Management (ARAIM) Research Hub,
Australian National University, Australia

william.jones@sydney.edu.au, hanna.kurniawati@anu.edu.au,
patrik.haslum@anu.edu.au, ian.manchester@sydney.edu.au

Abstract

Efficient repositioning of keel blocks in dry docks is critical for vessel maintenance operations but is challenging due to tight spatial constraints, complex sequencing dependencies, and reliance on expert personnel. This paper presents a preliminary study that frames the problem of keel block arrangement in the Captain Cook Graving Dock, Sydney, as an integrated Task and Motion Planning problem. Our approach combines binary linear assignment for optimal block assignments, quadtree-based spatial decomposition for rapid collision-aware motion planning, and heuristic search algorithms for efficient move sequencing. Comparative experiments using realistic dock scenarios demonstrate the value of the method, highlighting trade-offs between computational speed and solution optimality. Close collaboration with domain experts ensured practical relevance and alignment with operational constraints. Ongoing research will explore alternative search algorithms that achieve near-optimal results with reduced computational effort, adaptive replanning strategies, and real-world deployment to fully assess operational impacts. The proposed framework provides a robust foundation for broader automation initiatives in heavy industrial logistics and asset management.

Figure 1: Keel blocks supporting a ship in the Captain Cook Graving Dock at Garden Island. Image source: (9News 2023)

Introduction

The Captain Cook Graving Dock (CCGD) at Garden Island, Sydney, is the largest ship repair and maintenance dock in the Southern Hemisphere and a critical asset for Australia's defence capabilities. The dock supports both the Royal Australian Navy's operational readiness and commercial maritime activities, playing a crucial role in national security and economic stability. High demand for docking facilities places significant operational pressure on the facility, with limited capacity to accommodate vessel maintenance and repair schedules each year.

A primary operational challenge at CCGD is the repositioning of wooden-topped concrete keel blocks, which support ships during maintenance periods out of the water (see Figure 1). The approx. 25 tonne blocks are manoeuvred by a very large forklift truck. The several hundred blocks that may be required to support some ships must be positioned very accurately, plus or minus 10mm, to ensure the ship is supported at the correct loading points without causing damage. Currently, the placement and arrangement of these keel blocks between dockings is labour-intensive, time-consuming, and heavily reliant on the expertise of skilled dockmasters and their teams. This dependency creates vulnerabilities to delays arising from human error, scheduling conflicts, and the availability of experienced personnel. Minimising the time required for this setup process is thus essential for maximising the dock's operational efficiency and capacity.

The objective of this research is to address the above challenges by developing and evaluating a preliminary computational framework that optimises the arrangement and movement of keel blocks during ship docking operations. Specifically, this work frames the keel block reconfiguration as an integrated *Task and Motion Planning* (TAMP) problem, a perspective widely used in robotics for planning and executing complex, constrained actions, and develops a set of tools to generate optimised reconfiguration plans. This promises significant reductions in downtime between dockings, improved operational reliability, and reduced dependency on expert labour. Initially, these plans will guide forklift drivers, however, the ultimate goal is they guide an autonomous forklift. The significance of this research lies not only in its potential to enhance efficiency at CCGD but also in advancing the broader adoption of robotic technologies and digitisation within the maritime and defence sectors.

This research was conducted in close collaboration with the dockmaster and operations team at the CCGD. Their extensive domain knowledge was critical in shaping the problem formulation, identifying operational constraints, and validating the practical relevance of the proposed planning approach. Regular engagement with practitioners ensured the developed methods address the real needs and challenges of dock operations, increasing the likelihood of successful adoption and long-term impact.

This research makes four primary contributions:

1. Frames the keel block reconfiguration task as an integrated TAMP problem and formulates it as a sequential optimisation problem under realistic operational constraints.
2. Develops an integrated TAMP approach combining binary linear assignment for optimal block assignments, quadtree-based motion planning for efficient collision checking, and heuristic-guided search for effective sequencing.
3. Demonstrates the practical effectiveness of the proposed methodology through a comparative study of Depth-First Search (DFS) and A* algorithms in realistic dock scenarios, highlighting critical performance trade-offs.
4. Validates the practicality of the formulation via sustained engagement with dock operations staff, encoding key operational constraints (stacking order, caisson-crossing policies, packing reuse) into the planner.

Related Work

Application Domain

The complexity of rearranging large, heavy assets in confined and dynamic industrial spaces has driven research into advanced planning and scheduling approaches. In the maritime sector, much of the literature has focused on dock scheduling, resource allocation, berth planning and maintenance scheduling using operations research techniques such as heuristic optimisation, mixed-integer programming, and discrete event simulation (Guan, Wang, and Meng 2025; Hilliard, Lin, and Sparks 2020; Boudreault et al. 2022; Legato and Mazza 2001). These studies address when vessels dock, how to allocate quay or workforce resources and plan maintenance activities, but do not address the detailed spatial arrangement and physical movement of large support infrastructure such as keel blocks.

Chen and Huat examined keel block layout problems specifically. Their approaches often relied on rule-based heuristics and a genetic algorithm, achieving reductions in setup times and supporting manual planning. However, they focus only on the static arrangement of blocks, neglecting the underlying motion planning challenge of how to move each block into place efficiently and safely, while avoiding collisions and accounting for the evolving workspace.

TAMP

The robotics and AI communities have developed a rich body of work in TAMP, which unifies symbolic task planning (e.g., what sequence of tasks is needed) with geometric motion planning (e.g., how to move to accomplish each task) (Cambon, Gravot, and Alami 2004; Garrett et al. 2021). TAMP frameworks have enabled robots to operate autonomously in warehouses, factories, and service domains by integrating discrete sequencing with real-time feasibility checks. Recent surveys highlight both the theoretical underpinnings and practical advances in TAMP, emphasising the need for search algorithms that can effectively navigate large combinatorial state spaces while performing efficient geometric reasoning (Garrett et al. 2021).

The concept of TAMP is also closely related to the classical problem of motion planning among movable obstacles (Latombe 2012, Ch. 11), in which the environment contains objects that must themselves be manipulated or temporarily relocated to enable a robot's primary task. This paradigm, originally introduced in the context of robot motion planning (Latombe 2012, Ch. 11) (first edition 1991) and further developed in later work (Stilman and Kuffner 2005), is directly relevant to keel block rearrangement, where intermediate placements of blocks can be viewed as temporarily creating or resolving obstacles for subsequent moves.

Beyond classical formulations, recent integrated TAMP approaches tightly couple discrete task choice with continuous motion feasibility. Researchers propose a unified sampling-based framework that interleaves task and motion sampling (Thomason and Knepper 2019), while others have developed Rapidly-exploring Random Trees (RRT) derived planners that propagate task feasibility along motion trees (Caccavale and Finzi 2022; Saccuti, Monica, and Aleotti 2023). Our setting differs in being effectively two-dimensional (2D) and dominated by large movable obstacles and tight clearances; hence, we favour a structured space decomposition (quadtree) with heuristic search over sampling-based planners, which are advantageous in higher-dimensional configuration spaces. Nevertheless, these methods are complementary.

Several key technical developments from robotics are especially relevant to the dock environment. First, hierarchical spatial decompositions such as quadtrees have proven effective for scalable collision checking and pathfinding in cluttered or dynamic environments (Finkel and Bentley 1974; Samet 1984; LaValle 2006). In our keel block rearrangement problem, the robot's state space is essentially 2D, making quadtree representations particularly advantageous. While sampling-based roadmaps such as RRT and Probabilistic Roadmaps (PRM) are powerful tools for high-dimensional configuration spaces (LaValle and Kuffner 2001; Kavraki et al. 1996), they offer less benefit in low-dimensional (2D) settings like the dock, where the principal challenge lies in efficiently navigating around a large number of dynamic and static obstacles. By adapting spatial resolution locally, quadtrees allow motion planners to focus computational effort where it is needed most, such as around obstacles, narrow passages, or frequently changing layouts, without incurring the memory or speed penalties of uniform grids. Combined with efficient search algorithms (e.g., A*, weighted A*), these methods enable practical real-time planning in spaces with many obstacles and constraints (Sun et al. 2016).

Despite these advances, integrated TAMP methods have not been previously applied to the keel block reconfiguration problem in dry docks, a setting with uniquely stringent spatial, operational, and safety-critical constraints. To our knowledge, this is the first work to frame keel block rearrangement as a TAMP problem, unifying block assignment, high-resolution collision-aware motion planning, and task sequencing. Furthermore, we demonstrate the practical application of quadtree-based methods to this domain, which has not previously been reported in the literature.

This work bridges these gaps by (1) formulating the keel block rearrangement as a sequential decision-making problem under multiple constraints; (2) using binary linear assignment for block assignment or linear programming for multi-cycle block assignment; (3) leveraging quadtree-based motion planning for efficient and scalable pathfinding; and (4) integrating both into a TAMP framework that systematically explores and evaluates feasible reconfiguration sequences. Our methodology combines and extends ideas from operations research and robotics, demonstrating their value for real-world maritime automation.

Operational Context and Problem Description

The reconfiguration of keel blocks within the CCGD is a complex, high-stakes activity that directly affects the dock's efficiency and throughput. Each ship that enters the dock must be supported by a bespoke configuration of wooden-topped concrete keel blocks. This configuration is designed specifically for that vessel at the time of its construction. While the blocks themselves are generic, reusable assets permanently held in the dock, their placement is tailored to each vessel based on structural support points defined at the time of the ship's construction. As such, blocks must be repositioned ahead of each docking to accommodate the incoming vessel's unique support requirements. These configurations specify not only the position of each block beneath the ship but also the type of block required at each location. The dock holds approximately five different types of blocks, each varying in size and shape to support different parts of the hull. Some ships require additional height, necessitating multi-layer configurations in which blocks are stacked. These arrangements introduce further complexity, as the stacking order must also be considered when planning block assignments and movements.

The dock floor is not flat. By design, it is highest at the centre and it slopes gently toward the edges to allow for drainage, resulting in a height difference of around 10mm every 10 metres. Although just a small slope, it is necessary to compensate for this difference if a level surface is required to dock a ship. To achieve this adjustment, wooden packing is placed on top of the blocks to achieve the exact vertical position required for proper vessel support. This wooden packing must raise each block to a height that meets the ship's hull geometry.

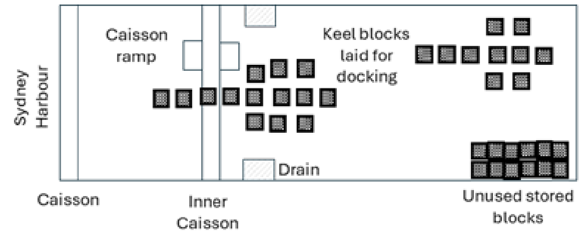

Figure 2: Simplified schematic of the dock environment. Example keel block positions, internal caisson and crossing ramp, and drains are shown. Spatial constraints require all block placements to avoid fixed obstructions.

Not only is packing material expensive, but time and labour can also be saved if packing does not need to be replaced between dockings. A small amount of packing is inevitably damaged during use and must be replaced, however, the ability to preserve usable packing during block relocation offers both a direct cost saving and a reduction in turnaround time. The goal of reusing as much packing as possible is an added constraint that significantly increases the complexity of block placement.

Space within the dock is extremely limited, which constrains the order in which blocks can be moved. Often, multiple ships are scheduled for docking simultaneously. Each ship's required configuration must be laid out in advance. Unused blocks that are not part of a current ship's configuration must be stored in designated areas within the dock in a way that does not interfere with the placement of blocks for other ships or obstruct work planned for once ships are docked. As a result, the problem is not only one of where to place blocks but also the sequencing of their movement from current to target positions.

A further complexity arises from obstructions present on the dock floor, the most notable of which is the internal caisson crossing point. Situated approximately one-third along the dock, this crossing is a designated location where a movable caisson (a watertight barrier) can be inserted to divide the dock. The crossing features a ramp that allows forklift access between dock sections, but its usability is highly dependent on keeping the area clear. Frequently, blocks are positioned in such a way that they partially or entirely obstruct the ramp, temporarily preventing the forklift from crossing. Consequently, the order in which blocks are placed becomes critical; blocks obstructing the caisson crossing ramp must be scheduled for placement only after all necessary movements across the ramp are complete. Additional fixed obstructions such as drains are distributed across the dock floor and further constrain the feasible paths and manoeuvring options available to forklift operators. Together, these physical constraints strongly influence both the spatial arrangement and the feasible sequencing of block movements, demanding careful consideration in any automated planning approach.

While each docking cycle presents unique requirements, significant efficiency gains can be realised by considering block placement over multiple cycles. By considering the anticipated sequence of incoming ships and their required block configurations, it is possible to optimise block movements over a horizon of multiple dockings. For instance, a block may be placed slightly off from its optimal position for the next docking, but closer to its future required location, reducing the cumulative movement effort across cycles. This introduces a trade-off between per-cycle optimality and long-term efficiency which the proposed method can address.

The actual movement of blocks is performed using a large, specialised forklift capable of handling the blocks that weigh approximately 25 tonnes each. Due to the mass and precision required, each block must be positioned within an accuracy of 10mm, the task is both time-consuming and labour-intensive. It demands highly skilled drivers with significant experience operating this type of equipment in tight conditions. New operators require extensive training, and shortages of experienced drivers present a clear operational bottleneck.

Given these factors, the process of keel block repositioning is a major contributor to delays between dockings and represents a critical vulnerability in dock operations. There is a pressing need for intelligent planning tools that can assist with reducing the time of this process. Optimising block movements will minimise time between docking and material costs. The methodology proposed in this paper directly addresses these challenges, offering a TAMP methodology capable of generating feasible and efficient plans for complex keel block reconfiguration tasks in constrained environments.

Problem Formulation

The keel block reconfiguration task is modelled as a sequential optimisation problem under spatial and operational constraints. Let $\mathcal{B} = \{b_1, \dots, b_N\}$ denote the set of keel blocks, each with a known type and initial position p_i^0 for block $b_i \in \mathcal{B}$. The objective is to move each block to a specified target position p_i^T, while minimising cumulative movement cost and satisfying all operational constraints. In the collision checking process, rather than explicitly modelling the forklift, we account for its physical footprint by inflating the obstacles corresponding to the carried block with an appropriate margin. Thus, collision checks are performed between the moving block (with margin) and other obstacles in a two-dimensional workspace, considering translation-only motions. In certain configurations, blocks may be stacked (e.g., placed on top of other blocks), so both ground-level and stacked positions are considered when defining the allowable moves.

The solution to the reconfiguration problem consists of a sequence of high-level *block-moving tasks*. Each task results in a distinct change in the positions of the keel blocks, and corresponds to a new node in the search tree. Specifically, we define two primary task types:

1. **Move**(b, p, p'): Relocating block $b \in \mathcal{B}$ from its current position $p \in \mathbb{R}^2 \times [1, 2]$ directly to its final target position $p' \in \mathbb{R}^2 \times [1, 2]$, where $[1, 2]$ indicates the discrete stack level (1 = ground level, 2 = stacked above another block). When planning a move to or from a stacked position, the feasibility of the move is contingent on the stacking order: the supporting (bottom) block must be in place before placing the upper block, and the upper block must be removed before the supporting block can be relocated.
2. **Move_Out_of_the_Way**(b, p, p_{temp}): Temporarily relocating block $b \in \mathcal{B}$ from its current position $p \in \mathbb{R}^2$ to a temporary position $p_{temp}\mathbb{R}^2 \times [1]$, dynamically chosen from a predefined set of feasible intermediate locations. Such tasks are necessary when a block's final destination is temporarily blocked or to clear access to another block.

Additionally, the feasibility of transitions between these tasks implicitly requires verifying that there is sufficient space for the forklift to perform an unloaded move, i.e., moving without carrying a block from its current position to the next required position. Although essential for ensuring motion feasibility, transit moves do not change the configuration of the blocks themselves. Instead, transit feasibility checks act as an intermediate geometric validation step between successive block-moving tasks. For these checks, the forklift is treated as an object with its own clearance requirements, and the path is validated to ensure that the forklift can move freely through the workspace, even when not carrying a block.

Each of these tasks corresponds to a high-level action that must be translated into a collision-free trajectory at the motion planning level. Tasks such as picking up or placing a block are implicitly considered feasible if their associated movement tasks are feasible, as successful completion of a movement task means the forklift is already at the required position for these actions.

The following constraints must be satisfied throughout the planning and execution:

- **Block Type and Packing Constraints:** Each block must be assigned to a target position compatible with its type. All concrete keel blocks are topped with wooden packing used to level the blocks. Ideally, blocks should be matched from current to target positions such that minimal repacking is required. Blocks can be relocated to temporary intermediate positions without changing packing; packing adjustments only occur once the block is placed in its final target position for supporting a ship. Minimising the need for repacking reduces material costs and labour time significantly.
- **Motion and Spatial Constraints:** Every move (including transit moves of the forklift without blocks) must be physically feasible given the current state of the dock. The forklift's path must avoid collisions with obstacles such as other blocks and drains.
- **Sequencing Constraints:** Some moves become feasible only after others, for example, when one block physically obstructs another, or when access to critical areas (e.g., the caisson crossing ramp) must be preserved until all required movements through that area are complete. In the case of stacked blocks, a block at level two (top) can only be placed if the level one (bottom) block is already present, and a block at level one cannot be moved if there is a block on top of it.

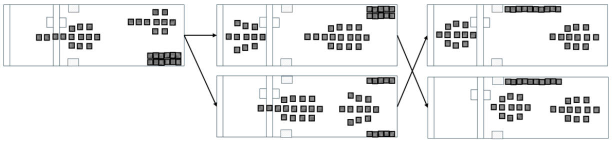

Figure 3: Example of a multi-cycle docking sequence. The current docking block configuration is shown, along with two alternative block configurations for each of two subsequent docking cycles. Arrows represent transition costs between configurations. For certain combinations of ships, multiple docking positions are possible, leading to numerous potential block arrangements. Strategic selection of ship positions, and thus block placements, can significantly reduce cumulative block movement compared to independently optimising each docking cycle

The optimisation objective is to minimise the total forklift travel distance, subject to the constraint that block matching between current and target positions preserves existing packing whenever possible. Specifically, blocks are initially matched by enforcing a hard constraint that prioritises packing preservation. Once this matching is determined, the algorithm minimises forklift travel distances to achieve the predetermined matching. While temporary intermediate block positions are sometimes necessary, the optimisation implicitly discourages their use, as additional temporary placements typically increase the overall travel distance of the forklift.

Formally, let x_{ij} be a binary variable indicating if block i (at current position p_i^0) is assigned to target position j (at p_j^T), and let d_{ij} denote the travel distance between p_i^0 and p_j^T. Let $\mathcal{P}$ denote the set of feasible assignments that preserve packing. The objective is:

$$\begin{array}{ll} minimise & \sum_{i,j} d_{ij}\, x_{ij} \\ subjectto & \sum_j x_{ij} = 1 \qquad \forall i \\ & \sum_i x_{ij} = 1 \qquad \forall j \\ & x_{ij} \in \{0,1\} \qquad \forall i,j \\ & x_{ij} = 0 \; if \; (i,j) \notin \mathcal{P} \end{array} \tag{1}$$

The problem can be extended to a multi-cycle planning horizon, where future block configurations for subsequent docking configurations are considered, allowing docking configurations that reduce total cumulative movement to be identified.

The following sections describe our integrated methodology for solving this problem, combining binary linear assignment for block assignment, quadtree-based motion planning for feasible paths, and a TAMP framework for generating and searching feasible move sequences.

Methodology

Our proposed methodology employs an Integrated TAMP framework that systematically combines task-level planning (block assignment using binary linear assignment) and motion-level planning (collision-free pathfinding using quadtrees and heuristic search) to efficiently sequence feasible keel block movements.

Execution Model and Search Pipeline

We first solve a binary linear assignment to match current blocks to targets under packing/type constraints. The search tree is then expanded over block configurations. Expanding a node considers either a direct MOVE to the assigned target or a MOVE_OUT_OF_THE_WAY to a temporary position. Each candidate is accepted only if two motion checks pass: (i) an *unloaded transit* from the forklift's current pose to the pickup, and (ii) a *loaded relocation* from pickup to the candidate drop. Pick/place are atomic actions parameterised by constants to give operational time estimates; they are not motion-planned beyond local clearance at the placement footprint. Here we consider the case of a single forklift operating in the dock only.

Constructing the Tasks via Binary Linear Assignment

To determine optimal assignments of keel blocks to target positions, we formulate and solve an binary linear assignment problem using the positions and constraints defined previously. Specifically, we match each block $b_i \in \mathcal{B}$ from its initial position p_i^0 to a final position p_i^T for each docking cycle, ensuring compatibility constraints (block types and packing) and spatial constraints (e.g., caisson crossings) are satisfied.

The optimisation is defined as:

$$\begin{array}{ll} minimise & \sum_k \sum_{i,j} \left[c_{ij}^k - \beta_{ij}^k\right] x_{ij}^k \\ subjectto & \sum_j x_{ij}^k = 1, \quad \forall i,k \\ & \sum_i x_{ij}^k = 1, \quad \forall j,k \\ & x_{ij}^k \in \{0,1\}, \quad \forall i,j,k \end{array} \tag{2}$$

Here, as previously defined, x_{ij}^k indicates the assignment of block i to target position j in cycle k, with costs c_{ij}^k incorporating distance and penalty factors and optional bonuses β_{ij}^k to encourage efficient stacking transitions.

This bonus term may be activated for ships or configurations where multi-layer stacking is required, and can be omitted for single-layer block arrangements. By including β_{ij}^k in the objective, the model is able to preferentially assign blocks to positions that allow for efficient transitions between stacked configurations, reducing unnecessary lifting or repositioning between cycles i.e., the top block in a previous configuration can become the bottom in a new configuration, avoiding the need for it to be placed in a temporary position while the bottom block is moved.

A key feature of the dock environment is the internal caisson crossing, which often acts as a bottleneck during block movements. The binary linear assignment model applies additional penalties to assignments that require blocks or the forklift to traverse the caisson crossing, reflecting the real operational cost associated with the forklift having to slow down for the crossing. This ensures that, whenever possible, the plan will favour assignments that minimise unnecessary traversals, thereby improving overall efficiency.

The binary linear assignment formulation can be naturally extended to plan across multiple docking cycles. Instead of matching current block positions to a single future configuration, we consider a sequence of target layouts across successive cycles. The objective is to minimise the cumulative distance that blocks must be moved over this sequence. This is encoded by defining a multi-layer bipartite graph, where each layer represents a docking cycle, and inter-layer edges model block transitions between configurations. Solving this extended linear program enables the planner to identify intermediary placements that may deviate from the immediate optimal but reduce total movement across all planned cycles. This strategy enhances planning foresight and reduces operational load across an entire maintenance schedule.

The assignment output from the binary linear assignment model provides a target layout for each block. However, translating these assignments into feasible sequences of physical moves requires accounting for spatial constraints and dynamic obstacles, as detailed in the following motion planning section.

We implement the assignment in Python using PuLP with an open-source backend. For the pure assignment and for the multi-cycle variant, all instances with many hundreds of blocks solve in well under one second on a commodity laptop. Runtimes are negligible relative to search and motion planning.

Quadtree-Based Motion Planning

Although blocks may be stacked in some configurations, all forklift motion planning is performed in the ground plane ($\mathbb{R}^2$). When a block is placed on top of another, only the ground-level block at the stacking destination is ignored as an obstruction for that particular move; otherwise, all blocks and obstacles are treated as ground-level obstructions for collision checking.

To efficiently plan paths within the highly constrained and cluttered dock environment, we adopt a quadtree decomposition for spatial representation. A quadtree is a hierarchical, tree-based data structure that recursively partitions two-dimensional space into four quadrants or regions, subdividing further only where higher spatial resolution is required (Finkel and Bentley 1974; Samet 1984). This enables a compact encoding of complex environments by focusing computational effort on regions where obstacles or free space boundaries exist, while coarser subdivisions are used in more homogeneous areas.

In our application, obstacles such as existing keel blocks, dock boundaries, caisson crossings, drains, and other permanent features are each represented as polygons. The dock floor is then recursively subdivided: if a quadrant is partially occupied by an obstacle, it is further subdivided until either a desired minimum resolution is reached or the space within a quadrant is homogeneously free or blocked. The result is a quadtree where each leaf node represents either free space or an obstacle.

The primary advantage of the quadtree approach in the dock context is its ability to represent large open areas with a few nodes and focus detail only where needed around obstacles, block clusters, or narrow passages such as the caisson crossing ramp. This is especially effective in ship docking operations, where the majority of the floor is empty or sparsely populated by blocks, but critical regions require fine-grained analysis to ensure collision-free movement.

Once the quadtree is constructed, nodes representing a region with no object in it are interpreted as navigable regions and adjacency relationships between nodes are used to construct a motion graph. Classical pathfinding algorithms such as A* are then used to compute feasible routes from the forklift's current location to pick-up and drop-off points. When blocks are moved, the quadtree representation is dynamically updated by marking previously occupied quadrants as free and newly occupied ones as obstacles, enabling rapid re-planning within a dynamic environment. Although quadtrees inherently represent static environments, incremental updates are computationally efficient due to their hierarchical structure. This ensures efficient, scalable, and collision-aware motion planning throughout the reconfiguration process.

Compared to uniform grids, quadtrees are memory-efficient and quick to update. Although visibility graphs may use less memory, quadtrees offer simpler, more efficient implementations, particularly beneficial in frequently changing low-dimensional environments like ours (Samet 1984; LaValle 2006). Sampling-based methods such as RRT and PRM are highly effective in high-dimensional spaces (e.g., robot manipulators) but less advantageous for our scenario due to their inability to directly leverage structured low-dimensional spatial information (Kavraki et al. 1996; LaValle and Kuffner 2001).

We approximate body clearance conservatively by inflating obstacles with the Minkowski sum of (i) the carried block polygon and (ii) a rectangle matching the forklift footprint plus a safety margin. This bounds collisions involving the forklift body without planning full nonholonomic, orientation-coupled paths. A full orientation–kinematics model (e.g., curvature and swept-volume along turns) is left for future work.

Having established a method for rapid collision checking and pathfinding, we integrate this method into the higher-level planning framework, described next, to guide and prune the search for a feasible reconfiguration sequence.

Figure 4 illustrates a typical application of the motion planning methodology in the dock environment. Obstacles and the dock boundary are represented as polygons, and the quadtree partitioning allows efficient identification of free and blocked regions. The forklift's planned path is determined by searching the navigable nodes of the quadtree graph using the A* algorithm.

Integrating Task and Motion Requirements in Planning

The rearrangement of keel blocks in the dock is fundamentally a sequential decision-making problem, where each block movement alters the environment for subsequent actions. We employ an Integrated TAMP framework to efficiently find feasible and near-optimal solutions, systemati-

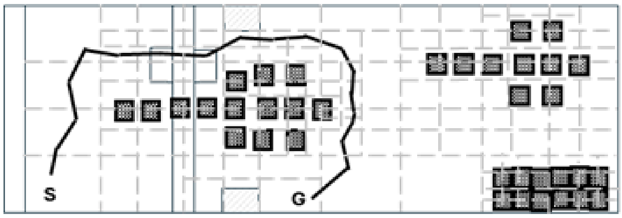

Figure 4: Illustration of quadtree-based motion planning concept in the dock environment. Obstacles such as keel blocks, caisson, and drains are shown as polygons. The dock floor is subdivided recursively into quadrants, creating finer partitions near obstacles and coarser partitions in open areas. Navigable regions form a motion graph used for efficient A* pathfinding. The forklift's planned path from the start (S) to the goal (G) is illustrated.

Algorithm 1: Integrated TAMP Planning for Keel Block Reconfiguration

Input: current positions $\{p_i^0\}$ and target positions $\{p_i^T\}$ for each keel block $b_i \in \mathcal{B}$; predefined temporary holding positions *Hold*
Assignment: solve the binary linear assignment so that each block b has a designated target p_b^T
Initialise search tree: create root node with configuration $\{p_i\} \leftarrow \{p_i^0\}$; place root in the set of *unexplored* nodes
while there are unexplored nodes in the search tree **do**
  Select one unexplored node (DFS, A*, or weighted A* selection policy)
  if $p_b = p_b^T$ for all $b \in \mathcal{B}$ in this node **then**
    Return: the sequence of actions from the root to this node (feasible solution)
  end if
  for each block $b \in \mathcal{B}$ with $p_b \neq p_b^T$ **do**
    Candidate destinations $S \leftarrow \{p_b^T\} \cup$ *Hold* {target and temporary options}
    for each $s \in S$ **do**
      Check unloaded transit feasibility: A* on the quadtree from current forklift pose to p_b
      Check loaded relocation feasibility: A* on the quadtree from p_b to s using inflated obstacles
      if both checks succeed **then**
        EXPAND: create a child node by applying $\textsc{Pick}(b)$, $\textsc{Move}(b \rightarrow s)$, $\textsc{Place}(b, s)$; update positions by setting $p_b \leftarrow s$ (and update forklift pose)
        (If using A*/wA*) attach a new cumulative cost equal to parent cost plus the cost of this relocation
        add the child node to the set of *unexplored* nodes
      else
        PRUNE: discard this candidate (b, s) for this node
      end if
    end for
  end for
  mark the selected node as *explored*
end while

cally exploring possible sequences guided by both task-level requirements and geometric feasibility.

The methodology constructs and searches an expanding tree of potential block configurations (Figure 5). The root node represents the initial positions of all blocks. Each new node is created by relocating a block, either directly to its assigned target or temporarily to an intermediate position, resulting in a new configuration. At each step, the planner considers blocks eligible for relocation; if a direct move to target is blocked, a feasible temporary position is dynamically selected from predefined holding areas.

A central feature of this approach is that each candidate move, both the unloaded forklift transit and the loaded block relocation, is checked for geometric feasibility using the motion planner (quadtree-based). Only if both motions are collision-free does the algorithm add the new configuration as a node in the search tree; otherwise, the branch is pruned immediately. This ensures that every explored sequence of block moves is physically executable, efficiently excluding infeasible options as the search progresses.

The search for an efficient rearrangement sequence is thus cast as incremental exploration of a feasibility-pruned search tree. Various search algorithms can be applied depending on the desired trade-off between computational effort and solution quality. For example, depth-first search (DFS) can be used to find any feasible plan, while heuristic-guided approaches such as A* or weighted A* can be employed to preferentially explore branches likely to lead to optimal or near-optimal solutions (e.g., minimal total travel distance or time). The heuristic may estimate the cumulative cost from the current state to the goal based on the remaining block movements and spatial layout.

A branch of the tree is considered a complete solution if it reaches a node where all blocks are in their required target positions. The TAMP approach guarantees that, given sufficient computational resources and accurate feasibility checks, a feasible rearrangement plan will be found if one exists.

In summary, our TAMP methodology organises the reconfiguration problem as an expanding search tree, systematically evaluating feasible sequences of block moves. By integrating geometric motion checks into each step, the framework guarantees executable solutions. Algorithm 1 summarises this integrated approach.

Results

To demonstrate the performance of the proposed TAMP framework, we applied both DFS and A* search algorithms to an identical keel block reconfiguration scenario representative of a typical docking. The test scenario consists of rearranging 93 keel blocks from their current positions to a specified target configuration, incorporating realistic dock constraints. This example used an initial configuration

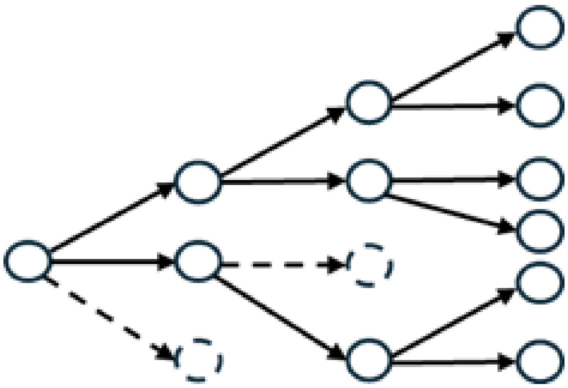

Figure 5: Schematic illustration of the expanding TAMP search tree. Each node represents a unique configuration of keel blocks. Solid lines denote feasible block moves (direct or temporary) i.e., both unloaded transit and loaded relocation pass quadtree feasibility. Dashed lines represent infeasible moves (i.e., failed one of the quadtree checks) pruned due to collisions or inaccessible paths. The search continues until the target configuration is reached. Realistic scenarios involve substantially greater complexity and branching.

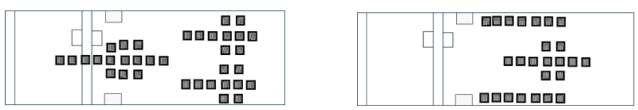

Figure 6: Simplified illustration of (**left**) initial configuration with keel blocks positioned for three docked ships and (**right**) target configuration for one ship, with remaining blocks relocated to storage along dock walls. The scenario involves repositioning 93 blocks within the dock while adhering to operational constraints

of 3 real docked ship block configurations transitioned to 1 docked ship with spare blocks moved to storage areas. Both search algorithms leveraged the same binary linear assignment, hence, the same initial and target positions were matched and packing preserved. Identical quadtree-based motion planning was used for assessing the feasibility of moves in sequence. The primary objective was to minimise total movement cost.

To compare the two search methods, we report the total number of individual forklift movements (where a single block relocation involves at least four moves: move unloaded, pick, move loaded, place), the aggregate distance travelled, and the computation time required to find a solution. Both approaches produced plans that were fully feasible and met all operational constraints.

Table 1 summarises the comparative results for DFS and A* in the selected scenario. The chosen metrics are directly linked to operational efficiency, impacting labour costs, maintenance turnaround times, and equipment utilisation. Thus, improvements in these metrics translate to tangible operational benefits.

As demonstrated in this example, both algorithms pro-

Table 1: Comparison of DFS and A* performance on keel block reconfiguration

Metric	**DFS**	**A***
Total number of forklift moves	651	360
Total travel distance (m)	8800	8144
Computation time ($\approx$ mins)	3	60

duced feasible reconfiguration plans, but significant differences in operational efficiency were observed. The A* algorithm, consistently in this example and others, generates solutions requiring fewer forklift moves and reduced total travel distance compared to DFS, reflecting its optimality. DFS typically generates longer, less efficient sequences, often including unnecessary temporary moves. Importantly, considering that the practical task of picking up and placing keel blocks is notably slower and more labour-intensive than simply driving unloaded, DFS's reliance on substantially more forklift moves, including frequent repositioning to temporary spots, results in significantly greater overall operational time. Thus, the reduced number of moves generated by A* not only represents an optimal theoretical solution but also delivers substantial practical time and cost savings despite longer computational planning times.

Although we optimise distance, the discrete plan can be post-processed to yield realistic time estimates by attaching empirically calibrated pick/place duration distributions and speed/acceleration profiles to each primitive move, enabling simulation-based (or aggregated) makespan estimates without modifying the planner.

Discussion

The methodology and sample results presented demonstrate the feasibility and practical value of applying integrated TAMP methods to the complex, real-world problem of keel block reconfiguration in a constrained dock environment. The proposed framework, combining binary linear assignment, quadtree-based motion planning, and feasibility-checked sequencing, produces executable plans that account for both spatial constraints and operational priorities.

A key strength of this approach is its flexibility and extensibility. By explicitly modelling domain constraints such as block stacking, caisson crossing penalties, and packing reuse, the methodology accommodates the wide variability of real docking scenarios. The integration of geometric and symbolic reasoning within the TAMP framework enables efficient pruning of infeasible sequences, resulting in practical computational performance even as the scale and complexity of the environment increase.

The project benefited greatly from close collaboration with the dockmaster and operations team at CCGD. Their expert insights were essential for defining realistic parameters, identifying constraints, and validating that generated plans were not only theoretically feasible but operationally acceptable. This partnership ensured the model captured domain-specific nuances and enhanced the relevance and credibility of the research outcomes.

Despite these promising results, several limitations remain. The current system operates in a simulated environment and has not yet been deployed for live dock operations. As such, quantifying the full-time and cost savings compared to existing manual approaches is not yet possible. A first prototype for use by docking planners and forklift operators has been developed. Ongoing work will involve piloting the tool in operational settings, collecting empirical performance data, and iteratively refining both the model and user interface based on operator feedback.

A limitation is that some real-world complexities are not yet fully modelled. Sometimes blocks are damaged and cannot be used, however, this may only be discovered by the driver during the setup process. Extending the methodology to explicitly address uncertainty, dynamic re-planning, and human-in-the-loop adjustments represents an important avenue for future research. We are working towards scalable techniques for such extensions.

Finally, the methodology's applicability extends beyond the specific case of keel block reconfiguration. The integrated approach to planning and execution is well-suited to other heavy industrial logistics and asset management challenges, such as yard management, port operations, and factory floor reconfiguration. The success of this project demonstrates the broader value of adapting advanced robotics and AI planning tools for practical, safety-critical industrial domains.

Conclusion

This work has demonstrated the feasibility and utility of an integrated TAMP approach for the reconfiguration of keel blocks in a large, operational dry dock. By combining binary linear assignment for block assignment, quadtree-based motion planning, and feasibility-checked sequencing, the proposed methodology systematically addresses the key operational challenges associated with block repositioning, including spatial constraints, stacking requirements, and sequencing dependencies such as caisson ramp accessibility and packing reuse.

Through comparative evaluation on a realistic scenario, we have shown that the TAMP framework is capable of generating executable, efficient reconfiguration plans using both DFS and A* search. The results highlight clear trade-offs between computational efficiency and plan optimality, with A* producing more concise movement sequences at the expense of longer computation times. These insights provide practical guidance for the selection of planning algorithms based on the operational context and available computational resources.

A central factor in the success of this work has been the close collaboration with the dockmaster and operations team, whose expertise ensured that the developed methods align with real-world practices and constraints. This engagement has been crucial for the adoption-readiness and long-term impact of the solution.

While the methodology is currently validated in simulation, the first prototype has been developed for use by docking planners and forklift operators. Ongoing and future work will focus on deployment in live operations, collecting empirical data, and further refining the system to handle uncertainties and human-in-the-loop requirements. Another important avenue for future research is the investigation and testing of alternative search algorithms for TAMP. There is significant potential in exploring optimisation algorithms that can better balance the trade-off between solution optimality and computational speed, such as greedy best-first search, iterative deepening A*, or other algorithms, potentially achieving near-optimal results with reduced computation time. Hybrid strategies that adaptively combine heuristic search and rapid feasibility checks, as well as reinforcement learning approaches to prioritise promising branches of the search tree, also warrant further investigation. The application of these to create efficient and robust keel block arrangement plans over multiple docking cycles has the potential to unlock significant value at the CCGD.

Beyond the immediate application, the proposed TAMP framework is broadly applicable to other heavy logistics, industrial asset management, and reconfiguration challenges in constrained environments. The results demonstrate the potential for advanced planning and robotics methods to deliver tangible benefits in safety-critical, labour-intensive domains, accelerating the adoption of digital and autonomous solutions within the maritime sector and beyond.

Acknowledgment

This work was supported by the ARC Research Hub in Intelligent Robotic Systems for Real-Time Asset Management (IH210100030).

References

9News. 2023. Behind the scenes at Garden Island's dry dock. https://www.9news.com.au/national/behind-the-scenes-at-garden-island-s-dry-dock/72af649e-980d-441b-be1f-263f92356e9c. Accessed: June 2025.

Boudreault, R.; Simard, V.; Lafond, D.; and Quimper, C.-G. 2022. A constraint programming approach to ship refit project scheduling. In *28th International Conference on Principles and Practice of Constraint Programming (CP 2022)*, 10–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Caccavale, R.; and Finzi, A. 2022. A rapidly-exploring random trees approach to combined task and motion planning. *Robotics and Autonomous Systems*, 157: 104238.

Cambon, S.; Gravot, F.; and Alami, R. 2004. A robot task planner that merges symbolic and geometric reasoning. In *ECAI*, volume 16, 895.

Chen, C.; and Huat, C. K. 2013. Optimal dock block arrangement considering subsequent ships solved by genetic algorithm. In *Proceedings of the 4th International Conference on Engineering, Project, and Production Management (EPPM 2013). October 23-25*, 770–783.

Finkel, R. A.; and Bentley, J. L. 1974. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4(1): 1–9.

Garrett, C. R.; Chitnis, R.; Holladay, R.; Kim, B.; Silver, T.; Kaelbling, L. P.; and Lozano-Pérez, T. 2021. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4: 265–293.

Guan, H.; Wang, M.; and Meng, Q. 2025. Syncrolift dry dock scheduling with a capacitated ship transfer system. *Computers & Operations Research*, 107138.

Hilliard, A.; Lin, K. Y.; and Sparks, R. D. 2020. Surface ship drydock schedule planner. *Naval Engineers Journal*, 132(1): 63–72.

Kavraki, L.; Svestka, P.; Latombe, J.-C.; and Overmars, M. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4): 566–580.

Latombe, J.-C. 2012. *Robot motion planning*, volume 124. Springer Science & Business Media.

LaValle, S. M. 2006. *Planning algorithms*. Cambridge University Press.

LaValle, S. M.; and Kuffner, J. J. 2001. Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. *Algorithmic and computational robotics*, 303–307.

Legato, P.; and Mazza, R. M. 2001. Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research*, 133(3): 537–547.

Saccuti, A.; Monica, R.; and Aleotti, J. 2023. PROTAMP-RRT: a probabilistic integrated task and motion planner based on RRT. *IEEE Robotics and Automation Letters*, 8(12): 8398–8405.

Samet, H. 1984. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2): 187–260.

Stilman, M.; and Kuffner, J. J. 2005. Navigation among movable obstacles: Real-time reasoning in complex environments. *International Journal of Humanoid Robotics*, 2(04): 479–503.

Sun, W.; Torres, L. G.; van den Berg, J.; and Alterovitz, R. 2016. *Safe Motion Planning for Imprecise Robotic Manipulators by Minimizing Probability of Collision*, 685–701. Cham: Springer International Publishing. ISBN 978-3-319-28872-7.

Thomason, W.; and Knepper, R. A. 2019. A unified sampling-based approach to integrated task and motion planning. In *The International Symposium of Robotics Research*, 773–788. Springer.