Joint CSCE Construction Specialty & CRC Conference 2025
Conférence conjointe spécialisée en construction de la SCGC et CRC-2025

Montreal, Quebec
July 28-31, 2025 / *28-31 juillet 2025*

PATH PLANNING PENALTY-BASED OPTIMIZATION FOR MOBILE CRANES

Boshen Lin[1], A. Bouferguene[1], M. Al-Hussein[1] and J. Kosa[2]

[1] Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Canada
[2] NCSG Crane & Heavy Haul Corporation, Acheson, AB, Canada

ABSTRACT: Modern construction projects are increasingly complex to a point where scheduling analyses can no longer be performed manually using pencil and paper as they once were. As a result, not only do we need to rely on computer technology to model the large number of parameters (degrees of freedom) but more importantly new algorithms may need to be elaborated in order to handle the spatial complexity of construction sites. In this respect, when cranes are used to move loads from their pick locations to their final destinations, it is paramount to have a minute understanding of the trajectories of each payload.
Modern heavy construction industry relies heavily on cranes since projects are often in terms of modules that are prefabricated off-site and front-loaded with as many functionalities as possible to minimize on-site activities. Front-loading modules increases their weights thus making high-capacity mobile cranes indispensable for on-site assembly. To add to the complexity of the lifting problem, it is important to mention that nowadays construction sites can be congested right from the start, which makes path analysis and planning even more critical to projects' success. Building on the success of A* algorithm, which has been used mostly to represent 2D crane trajectories, a variant of this algorithm is proposed to represent 3D trajectories. However, users are allowed to introduce penalties to take into account elements that could be related to efficiency or safety. The results of this contribution illustrate the ability of the proposed algorithm to balance the length of the trajectory, which happens to be the sole objective function of the traditional A*, with other user-defined criteria that can lead to smoother or safer trajectories.

1. INTRODUCTION

In recent years, heavy construction practitioners have turned to modularization as a means for increasing the efficiency and flexibility of project delivery. In this context, regardless of their complexities, projects are thought of as a set of modules that can be prefabricated in controlled environments before being shipped to construction sites for assembly. With a view to optimize project performance metrics, practitioners adopting the modular construction paradigm strive to minimize the amount of work carried out on-site by front-loading the modules with as many functionalities as practically possible. Consequently in contrast to the modules of the first generation that were light because they were practically empty shells, the most recent generations of modules can be extremely heavy. In fact, in the context of modern heavy construction industry (which is relevant to this contribution), it is not unusual to come across modules weighing over 500 metric tons. As a result, high capacity mobile cranes became one of the most important equipment allowing modules to be assembled effectively while adhering to the industry safety standards (Shapira et al. 2007, Peurifoy et al. 2011). It is important to note that since crane capacity is closely linked to mechanical and structural complexity, the mobilization of high capacity cranes come at a substantial cost (Taghaddos et al. 2019). The impact of this financial aspect on project profitability leads project managers and lift engineers to spend countless hours developing minute lift plans that will maximize crane utilization especially when adverse conditions, such as wind, bad weather, or delays in module delivery, that have the potential to disrupt project timelines need to be considered (Zhang and Hammad 2012, ElNimr et al. 2016, Hu et al. 2021). In addition to the assembly phase, where crane operation planning is paramount for an effective execution of each lift, planning during the bidding stage is equally important. In fact, at this stage, lift engineers must present their ideas with clarity in order to convince owners that they fully understand the challenges associated with each potential lift. Furthermore, information volatility during the bidding phase adds to the complexity of the decision making process since engineers are required to react rapidly to changes in project scope and timelines. In fact, as discussions with owners move forward, lift engineers can at times find themselves developing lift plans numerous times as a response to the aforementioned changes. This iterative refinement of projects at the bidding stage is costly and new technologies can provide invaluable help since it allows lift engineers to provide detailed and accurate answers to owners. Central to lift studies is crane path planning which is essential to avoid guesswork on-site especially when congestion is high and activities are to be conducted under strict time constraints (Wei-Han et al. 2015, Hung et al. 2016). Although, the last word on the lift is left to the crane operator, providing information beforehand in the form of crane operation feasibility and visualization can prove very useful for owners as well as lift engineers and crane operators. As a result, in this contribution, a variant of the A* algorithm that is specifically adapted to 3D motion is described. Although in previous work, the A* algorithm has been used for crane operations planning, its use was limited to generating payload paths as if they were moving in a two-dimensional plane (Olearczyk et al. 2014). In this context, three-dimensional obstacles are represented by their footprint. This limitation not only led to sub-optimal paths (typical in highly congested sites) but more importantly to the difficulty of gaining better insight into moving payloads over existing obstacles. Furthermore, since the standard A* algorithm is designed to give the shortest path, this new variation allows the user to enter penalties in order to generate a path that balances the length of the path with other constraints that could be important for safety.

2. LITERATURE REVIEW

In the heavy construction industry, mobile cranes are essential not only for construction (e.g., modular assembly) but also for post-construction activities, including maintenance in all of its forms (e.g., preventive, corrective, reactive, etc.). These cranes are usually the preferred choice of practitioners mainly for their wide capacity ranging from 10 to 1200 metric tons, their diverse geometric configurations and their relatively short assembly/disassembly time. These qualities make them the preferred choice for large-scale modular projects spanning several months as well as for maintenance projects with strict time constraints, such as replacing components in refineries during shutdown time. However, because their base is located on the ground, they can present serious challenges when it comes to collisions with obstacles. The following literature review explores briefly important advances made in the area of crane operations path planning. Generally, planning for optimized crane operations considers three interrelated aspects. (1) The selection of a crane location allows the maximization of the number of lifts without relocation. (2) The selection of an appropriate crane configuration (e.g., type of superstructure, carrier's geometry, boom, jib, etc.) that ensures adequate capacity and least amount of interferences with any surrounding obstacles; and (3) the option of using a pick-and-walk strategy, unique to crawler cranes, as a means to avoid relocation. From the perspective of path planning for crane operations, many algorithms have been proposed over the years. In what follows, a few of these methods will be examined briefly.

- Linear programming (LP) is one the key methods used for optimization in operations research. In the context of this approach, researchers established an objective function that minimizes either carbon emissions and/or the total time required to transport the payloads from their pick locations to their final resting positions. Although LP is not specifically used to find feasible paths, it allows practitioners to select from the set of feasible (collision-free) paths, the candidate(s) that can optimize other objective functions (Jeng et al. 2010, Matjaz and Uros 2023). Decision variables include crane and payload locations, the configuration of the crane (e.g., length of the boom, crane capacity, etc.), a binary variable associated with the option of walk-and-pick scenario, etc. In addition to mechanical characteristics of the crane, researchers have also considered dynamic factors such as wind speed and ground condition.

- Evolutionary and metaheuristic algorithms were introduced for path planning especially in the context of complex layouts. Optimizing the length of the trajectories while avoiding obstacles can be a difficult endeavor. Although devoted to tower cranes, Zhu et al noted the complexity of planning crane trajectories for modular integrated construction. As a result, they have proposed a metaheuristic procedure that combines PSO (Particle Swarm Optimization) with SA (Simulated Annealing). A comparison of six lifting scenarios let the authors to conclude that their PSO-SA algorithm outperformed other search procedures, e.g., GA-SA, SA, GA and PSO in finding a feasible collision-free path. More recently, Park (Park 2020) and Park and Han (Park and Han 2019) conducted a comparative analysis three major algorithms A*, RRT and GA (Genetic Algorithm) to optimize the trajectories of a mobile crane lifting trajectory for a modular project in the heavy construction industry. The authors noted that while A* provides a near optimal path, GA led to a reasonable path (lengthwise) while allowing for multiple constraints to be taken into account. The RRT had the worst performance of all.
- Reinforcement learning emerged in the past few years as a groundbreaking approach that has been widely adopted for the development of the intelligence needed in autonomous vehicles. It was only a matter of time before RL found its way to applications in the construction industry (Zhu et al. 2020). In this context, a growing number of case studies focused on construction equipment including cranes with a long-term objective of making them autonomous. Clearly, an inevitable yet important byproduct when using RL for crane operations is the determination of collision-free paths that provide a reasonable level of optimality as to their lengths. While this approach is very attractive since it can be used in combination with virtual reality to illustrate the optimization process, it is however computationally intensive. This may limit its use when at the phases where project information is volatile and subject to frequent changes.

Despite being a classic problem, crane operations in all of its facets is still an important problem for the construction industry. Significant progress has been made in terms of algorithms and visualization as reflected in the vast body of scientific literature. However, since practitioners generally acknowledge (anecdotally) that no two-construction projects are alike, research built upon case studies is likely to adapt or combine one or more existing algorithms to develop specialized procedures tailored to specific case under investigation.

3. METHODOLOGY

The proposed algorithm enhances the traditional A* pathfinding approach to address the unique challenges of crane operations in modular construction environments. It prioritizes safety and computational efficiency by leveraging a spherical coordinate system and integrating risk-based cost penalties. This section details the algorithm’s design, workflow, and validation process, structured as follows:

3.1 Problem Structure

Crane path planning involves finding collision-free sequences of movements that adhere to the kinematic constraints. A mobile crane’s configuration is defined by three active degrees of freedom (DOF) that can be measured by the angles θ_{yaw}, θ_{pitc} and the length l_{ext} (see Figure 1).

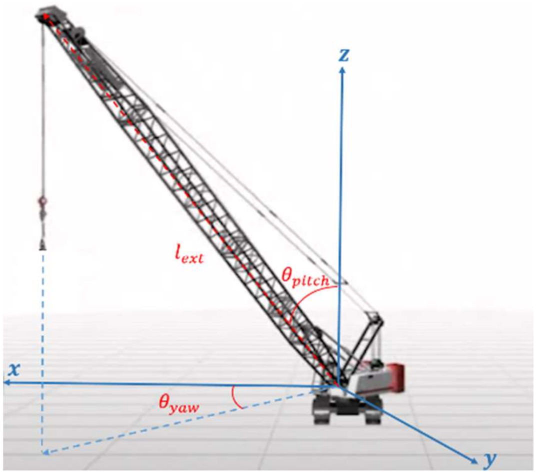

Figure 1: Yaw, Pitch and length (boom) extension coordinates

The yaw measures the rotation of the carrier around the vertical axis(z); the pitch measures the vertical tilt of the boom with respect to z and l_{ext} is the radial extension of the boom measured from its base. Using these variables, the configuration space representing the locations where the payload does not interfere with its surroundings is represented by an ensemble of nodes that could be viewed as a volumetric (irregular) tessellation of a sphere. These nodes are written as triplets $M_i = (\Theta_{\text{yaw}}, \Theta_{\text{pitch}}, l_{ext})$ where M_i defines the i-th node in the path. The objective is, therefore, to find a sequence of nodes $\mathrm{P} = M_0, M_1, \dots, M_n$ that constitute a path from the pick point M_0 to the set point M_n, which not only avoids collisions but also satisfies other user-defined constraints such as safety clearances or other site-specific limitations.

3.2 Algorithm Overview

In essence, crane path planning consists in finding a collision-free trajectory for a payload from its pick point to its final resting position. While for a given lift, multiple collision-free trajectories may exist, additional constraints, such as minimizing variations in the values of the parameters $\Theta_{yaw}, \Theta_{pitch}, l_{ext}$ or keeping the load close to the ground, can be considered as a means to improve efficiency (i.e., ensuring a smooth trajectory) and/or safety. Although the A* is designed to find the shortest path, its complexity can be high; this is particularly true in 3D space where the branching factor (i.e. the number of neighbors for each node) ranges from 6 to 26 voxels assuming a regular. Clearly, under these circumstances, the computational performance of the A* is expected to degrade significantly. Furthermore, given mobile kinematics which is controlled by the yaw, pitch and boom extension variables, the traditional A* needs to be modified to account for these constraints. As a result, the proposed solution adopts a sphere-based discretization of the search space, where nodes represent positions along radial directions at varying distances from the crane's pivot point. The algorithm evaluates paths using a multi-objective cost function that balances path length, obstacle proximity, elevation changes, and angular safety margins.

3.3 Sphere-Based Node Generation

Nodes are defined using spherical coordinates relative to the crane's center as depicted in Figure 1. The values of yaw and pitch angles are generated by intersecting radial rays with concentric spheres. The first step is to find all the critical points which consist of the start position, the goal position, and all the obstacle vertices expanded by a safety margin δ ($V_{expanded} = V_{obstacle} + \delta$). For simplicity, this process is illustrated on a 2D representation (see Figure 2) in which rays correspond to forward and backward translations (controlled by the pitch and boom extension) whereas circles describe swinging motion (controlled by the yaw angle).

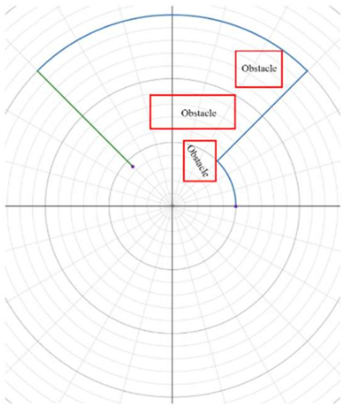

Figure 2: Example of space partitioning in 2D

In figure 2, the A* searches a free-collision path by following rays and arcs from the pick point to the final position. Clearly, when multiple obstacles are close to each other, the variation of the A* that would either find a path that has multiple rotation-translation changes, i.e. having many cusp points, or a path that is too long since the payload will be moved outward. In 3D space, concentric circles become concentric spheres. However, the concept of having nodes at the intersection of the rays and the circle (or spheres in 3D) remains similar. The ray should start from the crane's position and pass through the critical point (i.e., a point at which the ray or the spherical layer –or shell- intersect an obstacle). The sphere layer should be incident with the key point and center at the crane's position. Finally, storing the intersections in an appropriate node structure for the modified A* algorithm. The structure should contain at least the following properties: (spherical) layer ID, ray ID, g-cost, heuristic value, f-cost, parent, and the 3D location of the intersection. The inclusion of the layer ID and ray ID makes it efficient to search for neighbors since nodes on the same ray or the surface of the same sphere layer are considered neighbors.

```
Node
layer_id    // Sphere radius tier
ray_id      // Directional ray identifier
g           // Cost from start
h           // Heuristic to goal
f           // Total cost (g + h)
parent      // Previous node
world_position // 3D coordinates
```

Figure 3: Data structure defining the nodes used for 3D path searching

As described, the path searching algorithm proceeds by creating concentric spherical layers (or shells) where each node represents a potential boom configuration defined by its current length and the yaw and pitch angles. It is important to note that, for the sake of efficiency, the nodes are pre-computed based on the locations of the obstacles and considering the user-defined clearances that are critical to avoid collisions and more importantly to improve lifting safety. Figure 4, provides the pseudo-code describing salient implementation details.

Algorithm 1 Sphere-Radial Node Generation

Define: Origin (crane base position)

Collect critical points:
- Start/end positions
- Obstacle vertices expanded by clearance margin

for each critical point **do**

4: Calculate radial distance from origin
5: Create virtual sphere at calculated radius
6: Cast radial rays to sphere

7: **Generate** cNode objects at ray-sphere intersections:
- layer_id = radial distance tier
- lay_id = angular direction identifier
- world_position = 3D coordinates

8: **Prune nodes colliding with obstacles**

The strategy used in the above algorithm reduces significantly the number of nodes (hence the search space) compared to a full exhaustive search where all nodes are considered. However, although a number of alternatives are not considered, for the sake of efficiency, the proposed algorithm still maintains sufficient resolution to be practically useful for crane operations on construction sites. Fewer nodes enable faster computation. Movement along spheres mimics real-world crane operations consisting of boom extension/retraction, and rotation.

3.4 Modified A* Path finding and path construction

The traditional A* algorithm is enhanced with a multi-objective cost function balancing path length and safety criteria which can be expressed by steering the search towards a smooth trajectory or keeping the load preferably close to the ground while satisfying the clearance conditions. Accordingly, the multi-objective cost function combines four components: Base Cost, Obstacle Risk, Height Risk, and, Angular Risk.

- The base cost, referred to as the g-value in the data structure description in Figure 3, is a cost associated with the distance travelled. It is accumulated by the movement cost, which could be a fixed penalty value or the distance between the node and the crane's position.

$$[1]\ g = \sum_{i=0}^{n-1} d(M_i, M_{i+1}) \quad \text{where the path } P = \{M_0, M_1, \cdots, M_n\}$$

- The obstacle risk R_{obs} and the height risk $R_h$are the penalties for proximity to obstacles within a safety radius and for ascending (i.e., increasing the height of the payload) respectively. These two risks are key for retrieving a safe path instead of the shortest path. The algorithm will tend to avoid obstacles by walking around them or going over them because of the obstacle risk. However, going over obstacles can sometimes be the least preferred alternative. When a heavy module is suspended over a structure, unexpected movements due to wind gusts or load swinging increase the risk of accidents that can potentially damage both the module and the obstacle. Therefore, the height risk is introduced to restrict but not ban the act of going over obstacles. For coding purposes, the obstacle risk is set to infinity to prevent having nodes inside obstacles or below the limit of the crane undercarriage.

$$[2]\ R_{obs} = \begin{cases} \infty, & \text{if } N.pos.z \leq 0 \cup \exists O_i\text{: } N.pos \in \text{Space}(O_i) \\ \sum_{k=1}^{m} max(0, r_s - \|N - O_k\|) & \text{otherwise} \end{cases}$$

In which R_{obs} is the obstacle risk, calculated for node N and obstacle O_k subjected to the radius safety constraint, r_s. As for the height risk, it is given by, $R_h = \max(0, N_{next}.pos.z - N_{cur}.pos.z)$.

- The angular risk is designed to filter out the nodes that cannot be reached. Due to the crane kinematics since the boom, can only tilt up or down within a specific angular range. For coding purposes, the angular risk is set to infinity if the angle is not within the boom's luffing limits. In this way, the algorithm will never choose a node that is not reachable for the crane.

$$
[3] \quad \begin{aligned} R_{ang} &= \begin{cases} \infty & if\ \theta_{pitc} \notin [\theta_{min}, \theta_{max}] \\ 0 & \text{otherwise} \end{cases} \\ comp_G &= R_{obs}(N_i, V) + R_h(N_i, N_{i-1}) + R_{ang}(N_i, \theta_{min}, \theta_{max}) \\ f(N_i) &= g(N_i) + h(N_i) + comp_G(N_i, N_{i-1}, V, \theta_{min}, \theta_{max}) \end{aligned}
$$

To finalize the construction of the path, the nodes that are determined by the modified A* are post-processed in such a way as to satisfy the constraints associated with the geometric properties of crane kinematics. In this respect, three rules are applied:

- Consecutive nodes on the same spherical layer are connected by circular arcs
- Axial transitions between spherical layers use straight segments
- The final path combines arcs and lines through which symbolically can be expressed as,

$$
[4] \quad Path = \bigcup \left[Arc\left(Center, M_i, M_j\right) \otimes Line(M_k, M_l) \right]
$$

4. RESULTS AND DISCUSSION

4.1 Case study

To evaluate the effectiveness of the algorithm developed in this work, a case study was simulated in 3ds-Max. A crane with a telescopic boom was chosen for this application in order to allow its length l_{ex} to be adjusted since this can have an impact on the number of alternative trajectories. It is important to mention that while a real case study may be preferred by practitioners, simulation offers significant advantages during the development phase since it provides greater control over various parameters, including the congestion levels. A comparative analysis of the performance of the proposed sphere-based A* algorithm against traditional grid-based A* is conducted. The impact of the risk-aware cost function is analyzed.

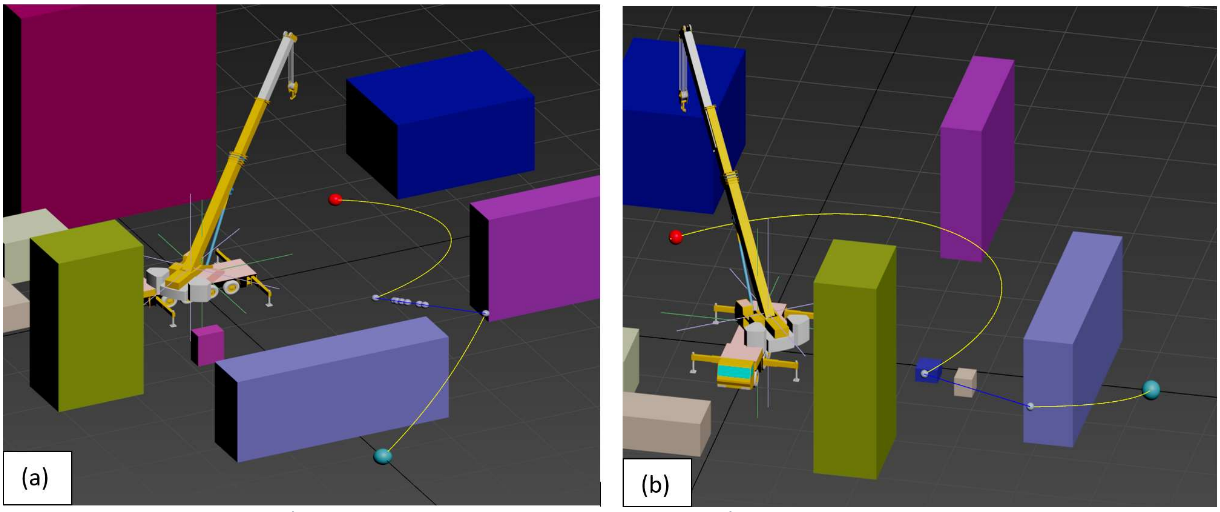

Figure 4: Bird's-eye view of the simulated case study along with the lift path. (a) Without penalties; (b) With penalties

A simulated case study of a modular construction site with 10 obstacles was used for validation. Key metrics include the number of nodes, computation time, path length, and safety compliance. Figure (4) provides a bird's view of the simulated case study along with the paths generated by the proposed algorithm in the case where no risk penalties are considered (a) and when they are taken into account (b). In this visualization, the arc components of the path are colored in yellow, while the ray components are colored in blue. This visual comparison of paths generated with and without the risk-aware cost function illustrates the algorithm's safety-first philosophy. The risk-aware path adopts a longer arc trajectory, circumnavigating obstacles with a conservative clearance margin. This increases the length of the path compared to its penalty-free counterpart, which is acceptable as the trade-off. While the penalty-free path is shorter, it takes more vertical movements, which may be viewed as risky. In contrast, the risk-aware path maintains a consistent low altitude.

4.1 Sphere-Based vs. Grid-Based A*

Table 1: Algorithmic complexity evaluation of the Sphere based A and Grid-based A**

Metric	Sphere-based A*	Grid-based A*: Resolution 1 node per cubic meter	Improvement
Number of nodes in the 3D map	6724	262997	~97% fewer nodes
Computation time (in terms of loops) –see Figure 5-	138	> 10000	Up to 98% fewer loops

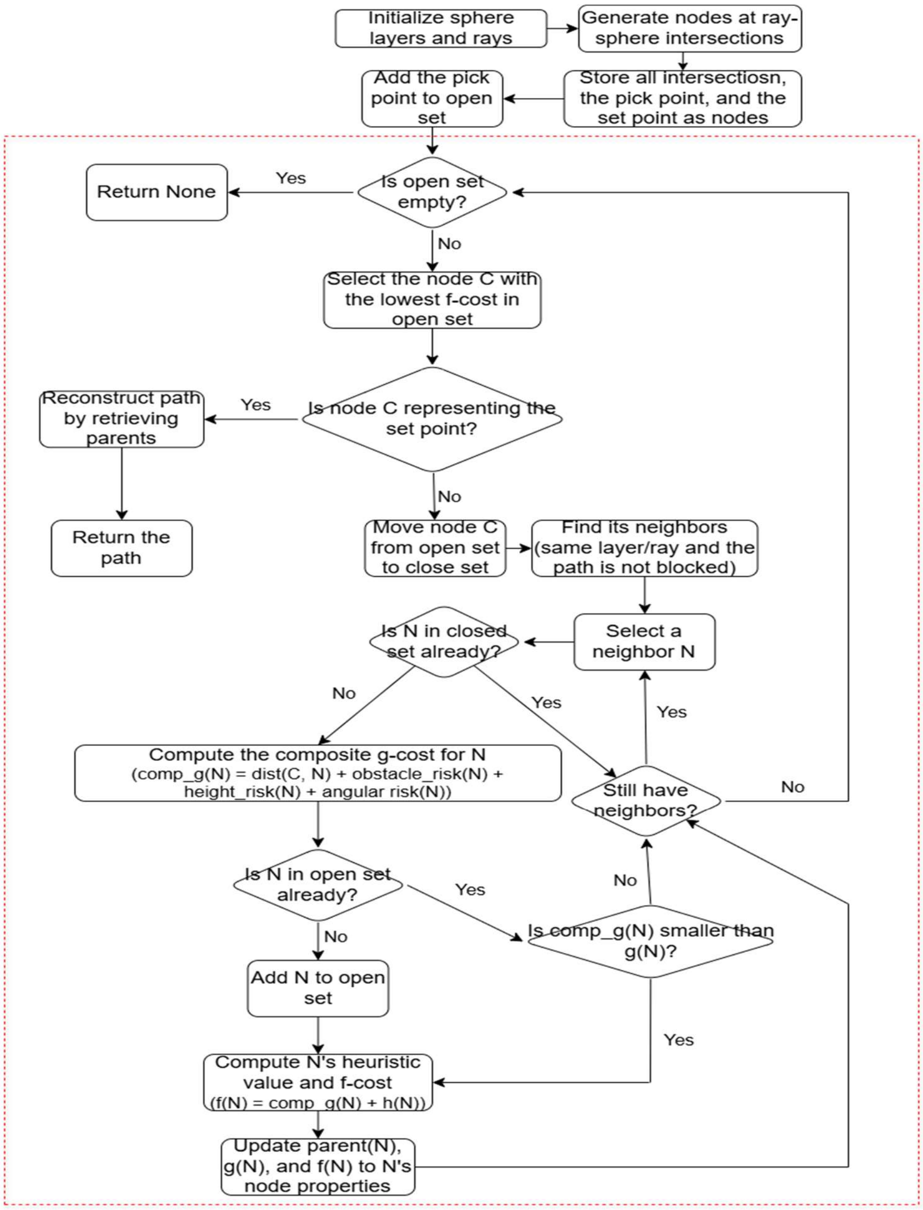

Figure 5: Flow diagram showing the main loop used as an efficiency metric

According to the table above, the proposed sphere-based A* algorithm demonstrates significant efficiency improvements over traditional grid-based A*. The sphere-based approach generated 6724 nodes, a 97.4% reduction compared to the grid-based method's 262997 nodes. This reduction stems from the spherical coordinate system's inherent alignment with crane kinematics, which avoids exhaustive 3D grid discretization. Instead of uniformly sampling the entire workspace, nodes are strategically placed along radial directions and angular sweeps, mirroring realistic boom movements. In addition, computation time, measured in loop iterations, further highlights the efficiency. The sphere-based A* required 138 loops to find a collision-free path, while the grid-based method exceeded 10000 loops, which means a 98.6% reduction at least. This efficiency is critical for real-time path planning that may be required in case of unexpected changes in the lifting schedule or at the bidding phase where project information is volatile.

4.2 Impact of Risk Penalty

Table 2: Performance in the optimized path planning: with and without risk penalties

Metric	With risk penalty	Without risk penalty
Number of nodes in the path	4	24
Path length	113.2	97.2
Cumulative height	18.7	169.2
Avg. height per node	4.7	7.0
Total distance from obstacles	14098.6	79228.6
Avg. distance from obstacles per node	3524.6	3301.2
Computation time (in terms of number of loops) –see Figure 5-	1671	138

The risk-aware cost function introduces critical safety trade-offs. With the penalty enabled, the path consisted of only 4 nodes, compared to 24 nodes without penalties. The penalized path length (113.171 m) was longer than the penalty-free path (97.226 m), reflecting deliberate detours to maintain obstacle clearance.

Notably, the accumulative height of the penalized path (18.674 m) was 89% lower than the unpenalized path (169.174 m), demonstrating effective avoidance of risky vertical movements. Safety metrics further validate the penalty's role. The average distance from obstacles to a node for the penalized path (3524.64 m) was 6.7% higher than the unpenalized path (3301.19 m), ensuring safer margins.

The penalty increased computation loops (1671 *vs* 138), as the algorithm evaluates safety criteria iteratively. This trade-off is acceptable in safety-critical environments, where collision avoidance outweighs minor delays. The result confirms that the risk penalty successfully prioritizes safety over raw efficiency, aligning with crane operation requirements.

5. CONCLUSIONS

In the heavy construction industry, projects typically fall in one of two main classes: (1) large-scale modular projects extending over several months and (2) maintenance projects that need to be completed within tight time windows. For instance, in some cases replacing old equipment needs to be completed over a single weekend to avoid costly interruption of production. However, be it for modular assembly or maintenance, planning crane operations is a critical activity that needs to be minutely conducted not only for efficiency but also importantly for safety. Although the path planning part of crane operations has been addressed in several research endeavors, there is room for exploring other alternatives that can enhance the existing toolkit used by practitioners and researchers. In this contribution, a variation of the A* path searching algorithm is proposed. The key feature of this algorithm is its ability to generate efficiently a trajectory of a crane payload in a 3D space. While a naïve A* algorithm operates in a 3D space by exploring unoccupied voxels (i.e., obstacle-free) seeking to minimize the length, the proposed variation allows the user to introduce additional constraints as a means to take safety factors into account. Furthermore, to keep the complexity of the algorithm within reasonable bounds, a heuristic requiring the trajectory to move on the edges of the obstacle is added. As a result, the flexibility in adding user-constraints and the heuristic thinking led to a time-efficient path generation procedure, which not only yields paths satisfying user conditions but also are reasonably optimal. Future work intends to explore optimality of the trajectories under external constraints.

REFERENCES

Peurifoy, R.L., Schexnayder, C.J, Shapira, A. and Schmitt, R.L. 2011. Construction Planning, Equipment and Methods. McGraw Hill, New York, USA.

Taghaddos, H., Eslami, A., Hermann, U., AbouRizk, S. and Mohamed, Y. 2019. Auction-based Simulation for Industrial Crane Operations. Automation in Construction, 104: 107-119.

Shapira, A., Lucko, G., Schexnayder, C.J., 2007, Cranes for Building Construction Projects. Journal of Construction Engineering and Management, ASCE, 133: 690-700.

Hu, S.B, Fang, Y.H. and Bai, Y. 2021, Automation and Optimization in Crane Lift Planning: A Critical Review. Advanced Engineering Informatics, 49:101346

ElNimr, A., Fagiar, M., Mohamed, Y., 2016, Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changing site layout. Automation in construction, 68:235-248.

Hung, WH, Liu, CW, Liang, CJ and Kang, SC, 2016. Strategies to accelerate the computation of erection paths for construction cranes. Automation in Construction, 62:1-13

Zhang, C. and Hammad, A. 2012. Improving Lifting Motion Planning and Re-planning of Cranes with Consideration for Safety and Efficiency. Advanced Engineering Informatics, 26:396-410

Park, S. 2020, Comparison of Lift Path Planning Algorithms for Mobile Crane Operations in Heavy Industrial Projects. MSc Thesis, Concordia University, Montreal, QC, Canada

Jeng, SL, Yang, CF, Chieng, WH, 2010. Outrigger Force Measure for Mobile Crane Safety Based on Linear Programming Optimization. Mechanics Based Design of Structures and Machines, 38(2):145-170

Matjaz, H. and Uros, K., 2023. Optimal Positioning of Mobile Cranes on Construction Sites Using Nonlinear Programming with Discontinuous Derivatives. Sustainability, 15:16889.

Park, S. and Han S. 2019. Comparison of Lift Path Finding Algorithms for Mobile Crane Operation in Heavy Industrial Projects. Proceedings, Annual Conference – Canadian Society for Civil Engineering, Laval (Greater Montreal), QC, Canada

Olearczyk, J., Bouferguene, A., Al-Hussein, M. and Ulrich, H. 2014. Automating Motion Trajectory of Crane-Lifted Loads. Automation in Construction, 45:178-186

Zhu, W., Guo, X., Fang, Y. and Zhang, X., 2020, A Path-Integral-Based Reinforcement Learning Algorithm for Path Following of an Autoassembly Mobile Robot. IEEE Transactions on Neural Networks and Learning Systems, 31(11):4487-4499