CIB World Building Congress 2019
Hong Kong SAR, China
17 – 21 June 2019

Constraint-free Crane Path Re-Planning

Xiao LI,
Department of Building and Real Estate, The Hong Kong Polytechnic University
(email: shell.x.li@connect.polyu.hk)
Hung-lin CHI,
Department of Building and Real Estate, The Hong Kong Polytechnic University
(email: hung-lin.chi@polyu.edu.hk)
Geoffrey Qiping Shen,
Department of Building and Real Estate, The Hong Kong Polytechnic University
(email: Geoffrey.shen@polyu.edu.hk)

Abstract

Lack of constraint-free crane path planning is one of the critical concerns in the dynamic on-site assembly process of prefabrication housing production (PHP). For decades, researchers and practitioners have endeavored to improve both the efficiency and safety of crane path planning from either static environment or re-planning the path when colliding with constraints or periodically updating the path in the dynamic environment. However, there is a lack of approach to the in-depth exploration of the nature of dynamic constraints and assist the crane operators in making adaptive path re-planning decisions by treating these constraints in different categories. To address this issue, this study develops a simulation-based service which embraces the core characteristics of adaptivity, sociability, and autonomy to achieve autonomous initial path planning, networked constraints classification, and adaptive decisions on path re-planning. This service is verified in the BIM environment, and it is found that it can generate the constraint-free path when it is necessary.

Keywords: Crane path planning, prefabrication housing production, constraints management, BIM

Introduction

The situation of unbalanced public residential housing (PRH) supply and demand becomes more and more stringent in Hong Kong. According to the Housing Authority of Hong Kong (2018), there were more than 153,300 general applicants for PRH and the average waiting time for them was 5.1 years (at its highest in last two decades). In order to expedite the supply of PRH, the PRH in Hong Kong has benefited and will continue to benefit significantly from prefabrication housing production (PHP), which is an innovative solution that the prefabricated material, component, module, and unit are manufactured efficiently at different locations and then converge at the site for installation. The popularity of PHP, also known as prefabricated construction, is productively boosting in the construction industry of Hong Kong as it meets market demand for improving industry-wide performance in the aspects including fast-track process, alleviating the on-site labor shortage, more sustainable and safer working environment (Li et al., 2017). However, the supply of PRH is still plagued by the pathological schedule delay of PHP. For example, the government planned to construct 13300 flat units of public housing in the financial year of 2016-2017. However, the actual amount of public housing production is 11276 units, and 15.22% delay occurred (Housing Authority, 2018). The uncertainties and constraints in the fragmented PHP process have been proved to be the primary drivers (Li et al., 2018). Uncertainty refers to something that may occur, whereas constraint (e.g., limited space and buffers) is something that will happen. The constraints are the obvious bottlenecks and thus are more predictable than the uncertainties to be removed in the task executions (e.g., four-day assembly cycle process). As such, the reliable schedules with constraint-free are vital for achieving an industrialized construction environment particularly in the on-site assembly process, which is the driver center for delivering the final products (Li et al., 2017).

The reliability of PHP schedules can be enhanced via proactive constraints management, which is the process of modeling, optimization, and monitoring of bottlenecks (e.g., lack of collision-free path planning, optimal buffer layout, and optimal installation sequence) to ensure that work package-level tasks assigned to workers can be successfully executed. Managing constraints in PHP processes is to prepare more (e.g., on detailed and dynamic planning with lean solutions) and act fast (e.g., on decision-making and collaborative working) using available information and knowledge. As such, the principal objective of constraints management is to continually improve the reliability of workflow by guaranteeing that precise information is always available at the right time in the right format to the right person.

As the tower crane leads the progress of site activities and makes it the hub of such PHP projects, and overall performances including productivity and safety are connected to smooth crane operations (Al Hattab et al., 2018). The constraint “lack of crane collision-free path planning” can be one of the most influential constraints to the duration of the on-site assembly process. Path planning has frequently been required in various fields (e.g., air, land, underwater) to provide the safe route from the start to the end point with optimized costs (e.g., time, motion, distance, and energy) (Cai et al., 2018). Additionally, it is recurrently inevitable to replan a path under the dynamic environment. Previous studies have focused on developing various simulations and algorithms to facilitate the path planning process in the field of robotics. These include sampling-based algorithms (e.g., rapidly-exploring random trees (RRT), probabilistic roadmaps (PRM)) (Zucker et al., 2007), node-based optimal algorithms (e.g., Dijkstra, A*, D*) (Koenig and Likhachev, 2005), bioinspired algorithms (e.g., genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO)) (Zhang et al., 2016), and mathematic model-based algorithms (e.g., mixed-integer linear programming) (Yilmaz et al., 2008). These techniques have also been widely applied to the crane path planning in the construction field. For example, Sivakumar et al. (2003) adopted A* to automate the crane path-planning task and found that A* search can provide near optimal paths. However, it was time-consuming. Ali et al. (2005) introduced the GA into the crane path planning to lessen the search time and enhance the quality of solutions. Chang et al. (2012) developed a method for near real-time path planning by using PRM. As the construction sites are complex and dynamic, the variable of time should be included in the path planning, and near all inputs may turn into varying instead of constant. This multiplies the complicatedness of the problem and leads to the demand for more smart algorithms to improve the path re-planning. Regarding the simulations and algorithms to re-plan the path in a dynamic environment, most solutions combine the previous algorithms or improve them based on the real situation. For example, Zhang and Hammad (2012) improved the RRT by using real-time information deriving from sensory feedback to achieve the dynamic path re-planning. Chi et al. (2014) combined the PRM and A* as a balance mechanism between efficiency and solution quality to achieve path re-planning in a dynamic virtual environment. Cai et al. (2018) proposed a multiobjective master-slave parallel genetic algorithm to assist the path planning in narrow and dynamic high-dimensional spaces.

Although considerable studies on dynamic crane path re-planning have been initiated, most of them concentrate on developing an algorithm to generate a route when constraints occur in the path. And these paths were re-generated either periodically or reaction to collisions between constraints and the current path. To the authors' knowledge, there is no study that proposes a constraints optimization service for crane operators with decision-making mechanism with respect to re-generate crane path in a changing environment. Moreover, previous methods may not evidently demonstrate the nature of a dynamic construction environment, as constraints vary over time, and thus the importance of changes may be negligible or significant at a given flash, and the magnitude of changes will alter based on the situation of the current path (Han and Seo, 2018). Namely, if the crane path is re-planned when collided with constraints or over a specific time interval, it may let the best time for replanning slip or generate unavailing runs of path replanning. Thus, it is an urgent need for an efficient approach to deciding whether a path should be replanned or remain unchanged when the constraints dynamically occur in the construction environment.

Thus, this study aims to develop an automatic path re-planning service with decision mechanism to assist the crane operator in computing path values with reference to specific constraints for identifying the impact of environmental changes and then decide the necessity of path re-planning. The specific objectives of this study are presented below: (1) to classify a set of constraints that can disturb the crane path and instantiate some of them into the physical construction site, instead of focus on all dynamic constraints in construction site; (2) to enable decision making for smart path re-planning by using cost values (distance) from a path to a specific constraint; (3) to simulate the decision making results (e.g., shorter path exists, or no need re-planning in current path in the building information modelling (BIM) environment.

2. Proposed Solution

2.1 Assumptions

Inspired by the methods in Han and Hasan (2018), and Chi et al. (2014), an adaptive decision-making approach can be developed for crane path re-planning in a dynamic construction environment through using path cost values that diffused from a specific group of constraints. Accordingly, several assumptions should be proposed in the following: (1) The roadmap graph in Configuration space (C-space) is displayed on a three-dimensional grid that composed by the equidistant triangles. The path planning method proposed in this study is built on the Probabilistic Roadmap (PRM), which equidistant geometrical points are sampled and connected (including the start and goal point) in the C-space. The process of PRM can be shown in Figure 1. The graph structure G in C-space can be formulated as:

$$\mathrm{G} = (v, e) \qquad (1)$$

Where v is the vertex that represents each geometrical point, and e is the side that connects the vertexes. Because the roadmap graph is assumed to be a three-dimensional equidistant triangular grid, the connections between vertexes are the sides of triangles. In order to identify loads of a tower crane in the sampled grid, the configuration space (C-Space) transformation is adopted (Chi et al., 2014). As shown in Figure 2, the location of the loads can be transformed from Cartesian space (X, Y, Z) to the 3-DOF tower crane's configuration (θ, γ, l).

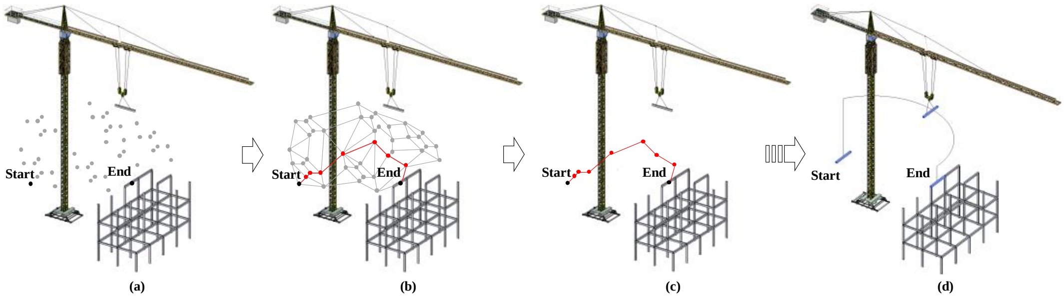

Figure 1. The process of PRM in Cartesian space: (a) sampling; (b) graphing; (c) solution finding; and (d) refining phases (Chi et al., 2014)

Where θ denotes the rotation angle of the crane turntable, γ stands for the rotation radius of the crane jib along with the distance between the current trolley and the mast, and l means its current hoisting distance between jib and hook. All the motions of the crane can be transformed into a point in the C-space. However, crane operators usually can only maneuver 2-DOFs of a tower crane together (Chi et al., 2014). For example, although rotating the jib while hoisting the loads can reduce the operation time, they are limited by the perception capacity of operators. This situation is not considered by the PRM, which may allow the generated path by PRM to be infeasible in practice. To deal with this issue, the triangular grid sampling method is proposed in C-space. As shown in Figure 3 (using 2D graph can easily illustrate its theorem), the triangular grid is established, which points are linked horizontally for a single DOF configuration and vertex can be connected diagonally for a 2-DOF configuration. The sides of the triangles can only be connected between neighboring points. Compared with the square grid graph, the path generated on this graph can ensure both the capacity of human manipulation (e.g., control sticks) and the assumption (3) that a movement conforms to a time interval with the same distance even though on the diagonal.

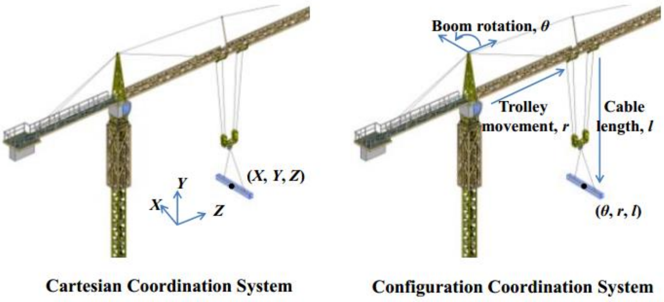

Figure 2. The transformation from the Cartesian space to configuration space (Chi et al. 2014)

(2) The dynamic constraints occur in a known construction environment. It means that the points of start and end are known and pre-determined, and dynamic constraints can move in any direction at any speed. Each constraint in Cartesian space transforms to the various polygons overlapped on the grid vertexes in the C-space (See Fig.3), which is C-obstacle. The C-obstacle denotes a cluster of motion (θ, γ, l) of tower crane must be avoided.

(3) A load of crane moves one side of the triangle for each time interval. The notation for this study has been listed in Table 1. The runtime is represented as T, which also denotes the movement times due to the assumption that one move is generated during each time interval. A path P_T can be defined according to T, and $v^{t,T}$ can be denoted as a set of grid vertexes near the path P_T that from the current vertex to the goal vertex. Variable t represents grid vertex order in P_T as $t = \{1, 2, \ldots, N_T\}$ where N_T is the length of P_T, and it also signifies the count of grid vertexes from the current vertex to the goal.

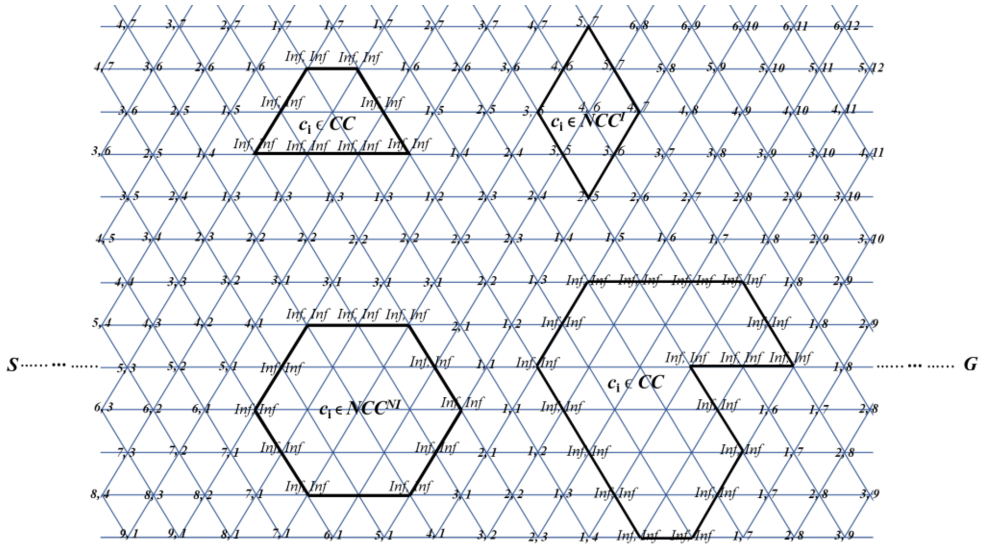

Figure 3. Cost diffusion in the triangular grid map

Table 1. Notation

Code	Definition
V	List of vertexes sampled in the roadmap graph
v	Grid vertex, $v \in V$
T	Runtime/ movements times on the path
P_T	A path at T represented by the vertexes from the current to the goal
$v^{t,T}$	t th vertex in P_T
C	List of constraints
c_i	i th constraint $i \in \lvert C \rvert$
V_f	List of feasible vertexes in the roadmap graph
v_f	Feasible grid vertex, $v_f \in V_f$
v_{start}	The start vertex of the loads, $V_{start} \in V_f$
v_{goal}	The goal vertex of the loads, $V_{goal} \in V_f$
s	The side length of each triangle

2.3 Problem Formulation

After the establishment of the above assumptions, the constraint-free path re-planning in this study can become an optimization problem which contains two stages: initial path planning and path re-planning decision-making.

2.3.1 Formulation of initial path planning

The initial path planning is to detect the optimal side combinations from the start vertex to the goal vertex based on the condition of initial constraints. The distance of the triangle side (connection between two vertexes) indicates the unit of cost value, and the optimal path is to search for the route with minimum cost values. Compared with Dijkstra's algorithm that is time-consuming to assess all sides of the grid to find an optimal solution, A* search algorithm with a partial heuristic function can help balance efficiency and performance for evaluating the quality of current solutions and removing impossible paths during search processes (Chi et al., 2014). As shown in Equation (2):

$$f(T) = g(T) + h(t) \qquad (2)$$

Where g(T) denotes the function computing the precise cost values from the starting vertex to the current vertex and $h(t)$ stands for a heuristic estimate function to estimate the predicted cost values from the current vertex to the goal. In addition, detection of initial constraints is an essential part of initial path planning, and it can guarantee each part of the path does not collide with neighboring constraints in the C-space. The paths obstructed by constraints may provide inoperable guidance for operators. The collisions can be identified by the ray tracing method, which check whether the two vertexes of a side can "see" each other or not (Chi et al., 2014).

2.3.2 Formulation of path re-planning

The path re-planning decision-making process starts with the categorization of constraints according to their priorities and positions, rather than treating all constraints uniformly. The constraints classification in crane operations has been summarized in Table 2. The movements of constraints can lead to changes in the distance between the planned path and specific constraints. These changes are recorded by the cost values of the path, which is computed by the diffusion of specific constraints. Decisions of path re-planning are then made according to the dynamic differences in cost values of the path.

Table 2. A list of dynamic constraints related to path planning task

Concern	**Description**	**Priority**	**Constraint**
Collision	On-site moving obstacles directly disturbing loads path with a random frequency (e.g., other moving cranes) can be assumed as the critical dynamic constraints.	Critical	External
Worker/vehicles	Site crews/vehicles pass through the crane's working area can be assumed as back and forth movements of constraints with the random speed and frequency	Non-critical & non-ignorable	External
Environment	Normal wind/rain during lifting can be assumed as one-way movements of constraints that pass through the load's path with the normal speed and frequency	Non-critical & ignorable	External

(1) Formulation of constraints and cost value

All constraints can be classified based on the priority listed in Table 2 to form the categorization in Table 3. To define each sub-class of constraints accurately, for $c_i \in C$, let shortest path at T be $P_T^{-c_i}$ and P_T^{-C} for c_i does not exist and there is no constraint, respectively. Additionally, allow a function $Z(\bullet)$ return one if there are more than one intersection between a constraint and a path, otherwise zero.

Table 3. Constraint Classification and Definition

No.	Constraints	Definition
1	Critical Constraints (CC)	Def1: $CC = \{c_i \mid Z(c_i,\ P_T^{-C}) = 1 \text{ and } Z(c_i, P_T^{-c_i}) = 1, \forall\ c_i \in C\}$
2	Non-Critical Constraints (NCC)	Def2: $NCC = C$ - CC
2.1	NCC-Not-ignorable (NCC^{NI})	Def3: $NCC^{NI} = \{c_i \mid Z(c_i, P_T^{-c_i}) = 1, \forall\ c_i \in NCC\}$
2.2	NCC-Ignorable (NCC^{I})	Def4: $NCC^{I} = \{c_i \mid Z(c_i, P_T^{-c_i}) = 0, \forall\ c_i \in NCC\}$

After the definition of different constraints, the numerical influence of CC and NCC^{NI} on a path can be computed by considering them as objects that diffuse influence. And all feasible grid vertexes can obtain the influence values from CC and NCC^{NI}. The larger influence values (smaller cost values) are attached to vertexes near to CC or NCC^{NI} , while smaller influence values (larger cost values) are attached to vertexes distant from CC or NCC^{NI}. This influence diffusion process assumes that NCC^{I} does not affect the path. Also, the grid vertexes on constraints can be treated as infeasible vertexes which the influence values show with “Inf.”

To define the detail process of cost value diffusion (the below context will use cost value instead of influence value for consistency) from CC and NCC^{NI}, let the cost values from CC and NCC^{NI} attached to vertex v be represented as $U_{CC}(v)$ and $U_{NCC}^{NI}(v)$, respectively, and let V_{CC} and V_{NCC}^{NI} be lists of vertexes that already acquire cost values diffused from CC and NCC^{NI}, respectively. As the process of cost value diffusion is the same for CC and NCC^{NI}, Eqs. (3)-(6) take CC as the example to formulate its cost value diffusion process. For all other vertexes not yet attached cost values from CC or NCC^{NI}, a set of vertexes adjacent to v_f are $A(v)$, which can be defined in Eq.(3)

$$A(v) = \left\{v' \middle| \left\|v' - v_f\right\| \leq s, \forall v' \in \left\{V_f \backslash V_{CC}\right\}\right\} \quad (3)$$

Where s is the side length of each triangle. Diffusion is implemented by attaching a cost value increased by one from the current value on adjacent vertexes. Then, V_{CC} and V_{NCC}^{NI} are updated. As $A(v_f)$ obtains cost values, and $A(v_f)$ turns into v_f for the next diffusion. Eqs. (4)-(6) are repeated until all v_f acquire cost values, which means diffusion completed and then V_{CC} or V_{NCC}^{NI} becomes the same as V_f .

$$U_{CC}\left(A\left(v_f\right)\right) = U_{CC}\left(v_f\right) + 1 \quad (4)$$
$$V_{CC} = V_{CC} \cup A\left(v_f\right) \quad (5)$$
$$v_f \leftarrow A\left(v_f\right) \quad (6)$$

(2) Formulation of dynamic scenarios

Figure 5 demonstrates the cost value diffusion process on the roadmap with two, one, and one constraint in CC, NCC^{NI}, and NCC^{I}, respectively. The cost values determined by CC and NCC^{NI} can be represented with the form $(U_{CC}(v), U_{NCC}^{NI}(v))$. There is no diffusion around NCC^{I} because it is not an object with the capacity of diffusing influence. The decision to re-plan a path depends on the continuation of alterations in the cost values of the path, and the cost values are calculated by the diffusion process from CC and NCC^{NI}. Namely, when constraints move, the changes in the path cost values caused by CC and NCC^{NI} are observed through moving constraints, and a decision on re-planning the path is made according to this observation. To clarify the changes in cost values in different situations, the six scenarios are proposed in Table 4.

Table 4. Dynamic constraints scenarios

Constraints	Collision with P_T	No collision with P_T
Movements of CC	scenario 1	scenario 2
Movements of NCC^{NI}	scenario 3	scenario 4
Movements of NCC^{I}	scenario 5	scenario 6

The cost values attached to P_T from CC and NCC^{NI} can be denoted as $U_{CC}(P_T)$ and $U_{NCC}^{NI}(P_T)$, respectively. In these six scenarios, s is believed to be adequately small to assess the need for path replanning from the changes in cost values caused by the dynamic constraints. Actually, If a collision between constraints (CC or NCC^{NI})and P_T, it is apparent that the current solution (P_T) is an infeasible path. Therefore, $U_{CC}(P_T)$ and $U_{NCC}^{NI}(P_T)$ display “Inf” in overlaying vertexes. The decision is made to re-plan due to a change in cost values of the path. Conversely, if there is no-collision, changes in cost values of path rely on the situations that group of constraints moved. Thus a decision can be made to replan a path when the current path can be improved in cost values.

3. Simulation Verification

3.1 Experiment Design

To test the performance of the proposed path re-planning approach, a simulation-based constraints optimization service was developed and demonstrated in both C-space and BIM environment. This service is developed on the cross-platform game engine named Unity 3D, which offers the scripting application programming interface (API) in C# with inbuilt physics library to simulate the crane operation tasks, on-site assembly environment, and dynamic constraints. The implemented algorithms include modified PRM method (sample equidistant triangles), A*, and *SWP_PathPlanner*.

3.2 Simulation Results

Figure.5 demonstrates the original environment with the initial path planning for $T=0$ in both the C-space and BIM environment. (1) In the set of figures for the original environment, the green dashed line of BIM environment is the shortest path without constraint (P_T^{-C}) and also represents the shortest path after the cost value diffusion process ($U_{CC}\,(P_T)$ and $U_{NCC}{}^{NI}\,(P_T)$). Constraints in CC and NCC^{NI} are displayed by red and yellow solid lines in C-space. (2) In the set of figures for the dynamic environment, static constraints are colored with blue in the C-space, respectively. And they are also attached with the matching entity name and ID in the BIM environment. Dynamic $U_{CC}\,(P_T)$ and dynamic $U_{NCC}{}^{NI}\,(P_T)$ are updated on the green dashed line. (3) In the set of figures for the environment with the decision, the comparisions of path values are conducted and the path is re-planned if any difference occurred. The new path is updated on the green dashed line.

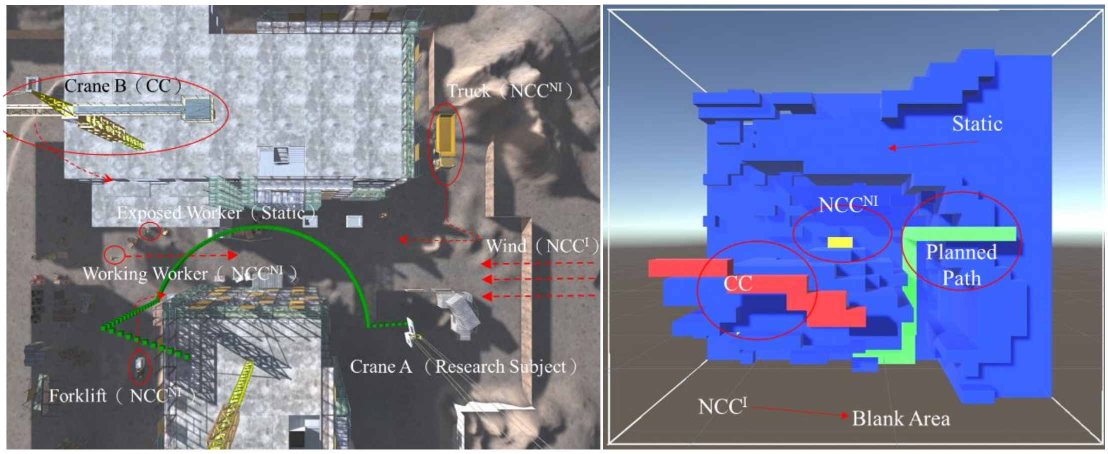

Fig.5 The initial path planning (T=0) in the simulation environment

In general, the results represent that the crane lifting task completed from the start to the goal with 191 movements (T). The total number of the dynamic constraints for all iterations was 5. Path re-planning conducted 3 times in the 191 movements. signifying that path re-planning was not essential for each T, even more than 25% dynamic constraints existed at each T. The detailed results corresponding to the six scenarios are discussed in the following.

(1) In T=61 (See Figure 6), scenario 1 can be validated by the evidence that moving crane B in CC collided with the path resulting in the path values changed and path re-planning. Contrarily, scenario 2 is simulated in T=60 (See Figure 7). It shows that moving crane B in CC became more distant from the path, and there was no collision, but it also led to path values changed and re-planning.

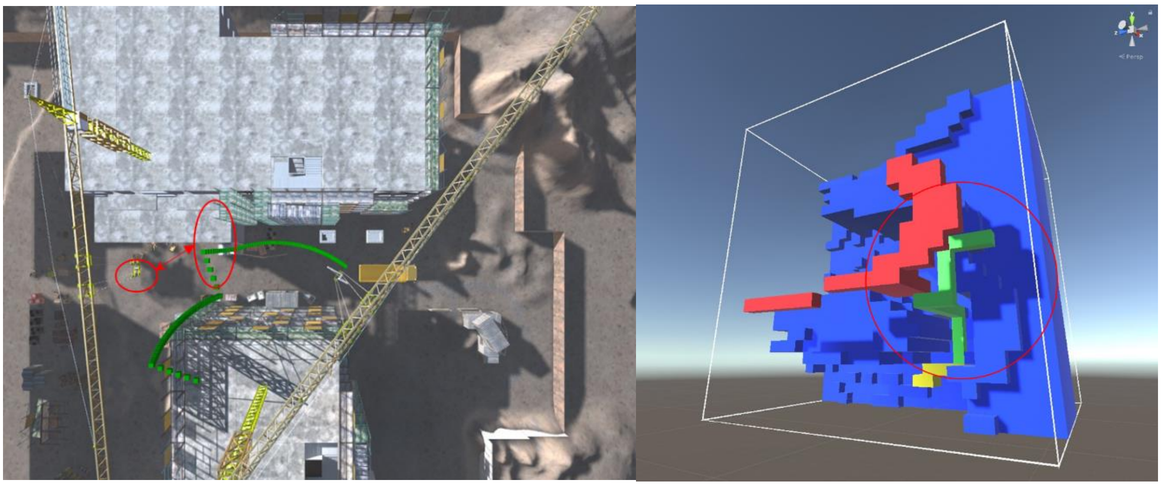

Fig.6 Scenario 1

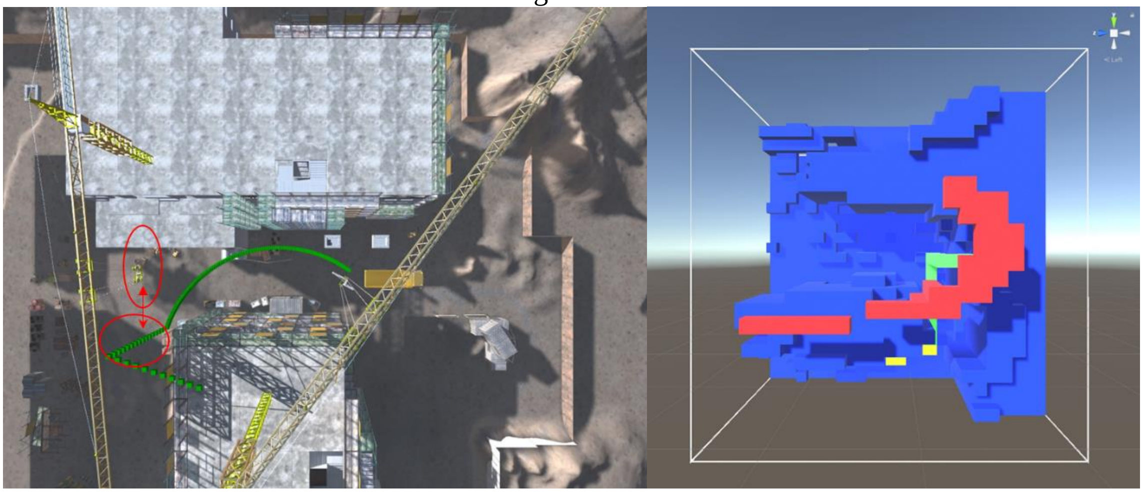

Fig.7 Scenario 2

(2) In T=22 (See Figure 8), scenario 3 occurred that moving vehicle in NCC^{NI} collided with the path and the result is the same as the scenario 1. However, the scenario 4 happened in T=20 (path value changed and path re-planned) (See Figure 9) and T=0 (path value changed and keep the original path) (See Figure 5) was totally different because it depends on whether the moving vehicle surrounded or were distant from the path.

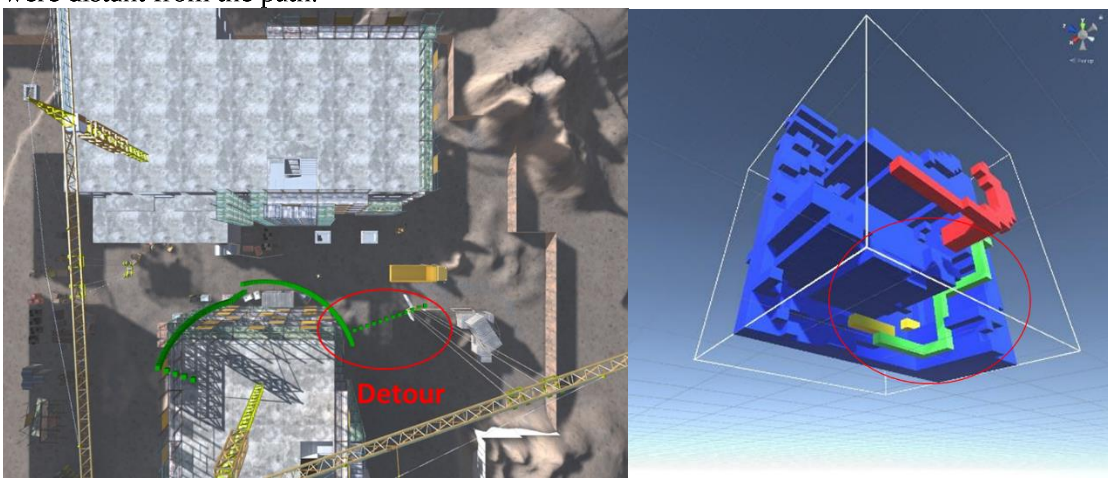

Fig.8 Scenario3

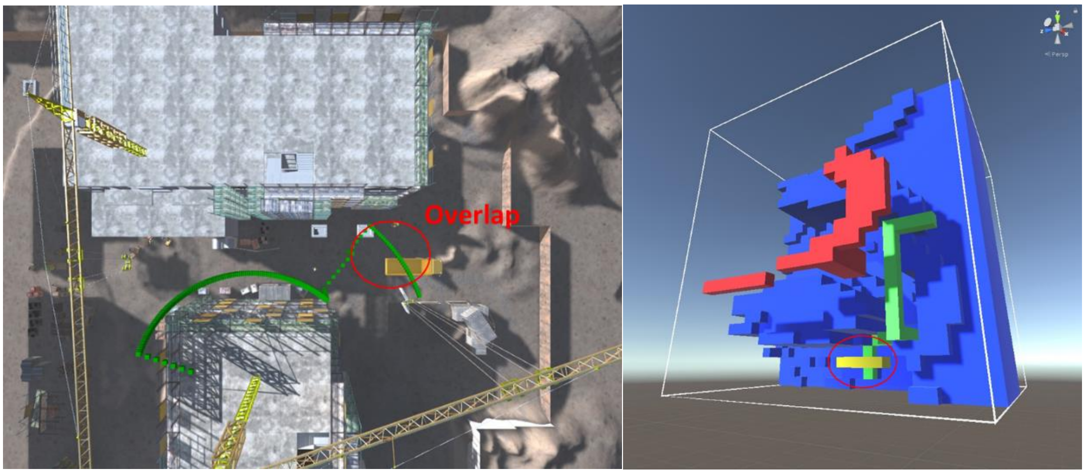

Fig.9 Scenario 4

(3) In T=0 (See Figure 5), scenario 5 and 6 are assessed to show that there was no path re-planning regardless of the normal wind and rain in NCC^I collided or did not collide with the path.

4. Conclusion

This study provides an in-depth exploration of smart decisions with adaptivity in path re-planning under a dynamic crane operation environment. Deviating from traditional methods that are re-planning a lift path when collided with constraints or over a specific time interval, this study argues for the adaptivity of a decision-making mechanism to update a path when necessary. Targeting a real PHP project in Hong Kong, this path re-planning service is validated in a BIM environment. The results of this simulation indicate the feasibility of applying this service into practice.

References

Al Hattab, M., Zankoul, E., Barakat, M., & Hamzeh, F. (2018). Crane overlap and operational flexibility: balancing utilization, duration, and safety. *Construction Innovation, 18*(1), 43-63.

Ali, M. A. D., Babu, N. R., & Varghese, K. (2005). Collision-free path planning of cooperative crane manipulators using a genetic algorithm. *Journal of Computing in Civil Engineering, 19*(2), 182-193.

Cai, P., Chandrasekaran, I., Zheng, J., & Cai, Y. (2018). Automatic Path Planning for Dual-Crane Lifting in Complex Environments Using a Prioritized Multiobjective PGA. *IEEE Transactions on Industrial Informatics*, *14*(3), 829-845.

Chang, Y. C., Hung, W. H., & Kang, S. C. (2012). A fast path planning method for single and dual crane erections. *Automation in Construction, 22*, 468-480.

Chi, H. L., Kang, S. C., Hsieh, S. H., & Wang, X. (2014, January). Optimization and Evaluation of Automatic Rigging Path Guidance for Tele-Operated Construction Crane. *In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 31, p. 1). Vilnius Gediminas Technical University, Department of Construction Economics & Property.

Han, J., & Seo, Y. (2018). Path regeneration decisions in a dynamic environment. *Information Sciences, 450*, 39-52.

Housing Authority (2018), Number of Applications and Average Waiting Time for Public Rental Housing, https://www.housingauthority.gov.hk/en/about-us/publications-and-statistics/prh-applications-average-waiting-time/

Koenig, S., & Likhachev, M. (2005). Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics, 21*(3), 354-363.

Li, X., Shen, G. Q., Wu, P., Fan, H., Wu, H., & Teng, Y. (2017). RBL-PHP: Simulation of Lean Construction and Information Technologies for Prefabrication Housing Production. *Journal of Management in Engineering, 34*(2), 04017053.

Li, C. Z., Xue, F., Li, X., Hong, J., & Shen, G. Q. (2018). An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction. *Automation in Construction, 89*, 146-161.

Sivakumar, P. L., Varghese, K., & Babu, N. R. (2003). Automated path planning of cooperative crane lifts using heuristic search. *Journal of Computing in Civil Engineering, 17*(3), 197-207.

Yilmaz, N. K., Evangelinos, C., Lermusiaux, P. F., & Patrikalakis, N. M. (2008). Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE Journal of Oceanic Engineering, 33*(4), 522-537.

Zhang, C., & Hammad, A. (2012). Improving lifting motion planning and re-planning of cranes with consideration for safety and efficiency. *Advanced Engineering Informatics*, *26*(2), 396-410.

Zhang, X., Zhao, Y., Deng, N., & Guo, K. (2016). Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm. *International Journal of Advanced Robotic Systems, 13*(3), 91.

Zucker, M., Kuffner, J., & Branicky, M. (2007, April). Multipartite RRTs for rapid replanning in dynamic environments. *In Robotics and Automation, 2007 IEEE International Conference on* (pp. 1603-1609). IEEE.