

# Automated Crane-lift Path Planning Using Modified Particle Swarm Optimization for High-rise Modular Integrated Construction

A.M. Zhu<sup>a</sup> and W. Pan<sup>a</sup>

<sup>a</sup> Department of Civil Engineering, The University of Hong Kong, Hong Kong SAR, China  
E-mail: [zhuaimin@connect.hku.hk](mailto:zhuaimin@connect.hku.hk), [wpan@hku.hk](mailto:wpan@hku.hk)

**Abstract –**

**Modular integrated construction (MiC) is a most advanced off-site technology. In a MiC project, it is critical but challenging to install prefabricated volumetric modules efficiently and safely, as they are much heavier and larger than conventional construction components and materials. However, the current crane-lift executions are heavily reliant on operators' subjective judgement, which turns out to be time-consuming and error-prone, most notably when jobsites are congested. Automatic crane-lift path planning has been regarded as an important research topic for addressing this problem. Nevertheless, most previous studies did not consider the MiC-specific features of a crane lift such as correlation between module weight and crane trolley movement. Therefore, this paper aims to propose a modified particle swarm optimization (PSO) algorithm to automatically conduct crane-lift path planning for high-rise MiC. A new fitness function and three auxiliary engines are designed for executing the proposed PSO path planner. This novel algorithm is validated using a real-life MiC project. The findings reveal that the optimized algorithm outperforms existing metaheuristics in terms of convergence characteristics and path smoothness. A collision-free crane-lift path can be worked out with a small population size and a few iterations. Practically, this paper should facilitate safe and efficient project delivery of high-rise MiC. Scientifically, the paper contributes to the theoretical development of smart and automated technologies and algorithms in construction.**

**Keywords –**

**Tower crane; Path planning; Particle swarm optimization (PSO); Modular integrated construction (MiC)**

## 1 Introduction

Modular integrated construction (MiC) represents the highest level of off-site construction [1]. In a typical high-rise MiC project, multiple heavy and large-sized prefabricated volumetric modules are manufactured first

in factory followed by installation on site. Hoisting those hefty units relies on the stringent deployment of tower cranes, which occupies most of crane-lift tasks [2]. However, it is challenging to get them installed efficiently and safely from confined storage yards to the target position, especially in high-density and congested metropolises such as Hong Kong. Neitzel et al. [3] mentioned that more than 30% of construction and maintenance casualties resulted from cranes. Tam and Fung [4] pointed out that the lack of transferable skills and the fatigue of crane operators are the critical reasons leading to accidents. In high-rise MiC projects, crane practitioners are required to have higher professional skills and should pay closer attention to collision identification. Nevertheless, it is easy to cause time delay and safety hazards when the hoisting executions depend on operators' subjective behaviors.

Computer-aided path planning has become a popular research topic to assist crane drivers in identifying the optimal crane-lift path. However, the literature reveals that existing approaches are highly homogeneous. Genetic algorithms [5-8], for instance, gain the most popularity in metaheuristics. The existing solutions also seldom consider the velocities of the mechanism transmission and the strategies of crane operation, which resulted in impractical lifting trajectory with redundant turns and twists [9]. In addition, other algorithms with high performance such as particle swarm optimization (PSO) have not attracted researchers' attention in the field of path planning. Moreover, few studies conduct path planning for high-rise MiC considering the modular-specific features such as large-sized and heavy modules.

To address the challenges and fill the knowledge gaps, this paper aims to propose a modified PSO algorithm to automatically conduct crane-lift path planning. Based on the existing PSO algorithm, a new fitness function is devised with three significant indicators i.e., overall hoisting time, total movement distance and the collision penalty mechanism. Also, three auxiliary engines, i.e., crane configuration engine, model regeneration engine and collision detection engine are used for PSO execution.

Following this introduction section, Section 2 reviews the crane-lift path planning methods, the published metaheuristics and PSO related research. Section 3 sheds light on the proposed system architecture and its development. Section 4 verifies the proposed

method using a real-life MiC project. The last two sections are presented for discussion and conclusions, respectively.

## 2 Literature Review

### 2.1 Overview of Crane-lift Path Planning Algorithms

Finding a collision-free crane-lift path is a scorching research topic among various countries and regions [5-7,10], like India [6], Singapore [7,8], Canada [10], etc. It normally takes two steps to achieve this goal. The first is to transform the work space to the configuration space, in which the information from environmental obstacles and crane's particulars can be well extracted. The other is to screen out or design a proper algorithm to match specific crane-lift scenarios. Researchers have invested tremendous time and energy in exploring adaptable, robust and highly productive algorithms for crane-lift path planning. Those algorithms can be divided into three categories, namely, node-based methods, sampling-based methods, and meta-heuristic methods.

Node-based methods, including Dijkstra's and A\*, have a powerful performance in finding a high-quality and collision-avoidance lifting path. Nevertheless, their shortcoming is obvious for the high time complexity, particularly in complicated job sites [11].

With respect to sampling-based methods, rapidly exploring random tree with its variants has been widely adopted. However, the lifting trajectory is prone to more turns and twists, enhancing crane drivers' manipulation complexity [10], although there have been some improvements recently [9].

Meta-heuristic methods contribute a lot to crane-lift research. Genetic algorithms (GAs), ones of the most classical metaheuristics, have attracted huge attention over the past two decades. Researchers from Indian Institute of Technology first started GAs' application to lifting path planning. They initially designed a simple GA model in 2D configuration space for two degrees of freedom (DOFs) crane manipulators [5]. Thereafter, Ali et al. [6] used GA for the dual-crane path planning. Cai et al. [7] improved the fitness function and altered the evolutionary strategies based on Ali's research. To find the erection paths efficiently, their algorithm was implemented via parallel computing on graphic processing units, and then they applied it into four DOFs tower cranes [8]. Although GAs have been widely used among various disciplines, it should not be treated as the only option and it cannot always perform best, compared to other EAs. Wang et al. [12] tried to use ant colony optimization (ACO) algorithm for three DOFs mobile cranes, but their research was at the preliminary stage and the lifting trajectory was hard to manipulate.

### 2.2 Particle Swarm Optimization (PSO)

Another classical evolutionary algorithm is PSO, which was coined by Kennedy and Eberhart [13] in 1995, inspired by the school of birds and fish. Each particle in the particle swarm represents a possible solution to a problem. Through the simple behavior of individual particles and the information interaction within the swarm, problem-solving intelligence is realized. It has been demonstrated and verified that PSO can perform better in a cheaper, simpler, and faster way than other intelligent algorithms [13,14].

In respect to the construction industry, Zhang and Xing [15] utilized PSO together with a fuzzy-integrated approach to find the best solution for addressing the time-cost-quality tradeoff problem. PSO served as a useful tool for construction layout planning, like unequal-area design [14]. It also presents superb performance for crane optimization issues, for instance, the fault diagnosis of mobile crane [16], and the design optimization of tower crane hoisting system [17]. However, there is a paucity of related research into PSO within the context of tower crane lifting path optimization.

## 3 System Architecture of Crane-lift Path Planning

### 3.1 Overview

Given that the conventional construction practice is rather risky, time-consuming, and laborious when conducting crane-lift issues, this paper proposes a modified PSO algorithm to find a collision free crane-lift path automatically, thereby accelerating intelligence and automation in MiC featured projects.

The system architecture builds on two categories of modeling environments and an output illustrated in Figure 1.

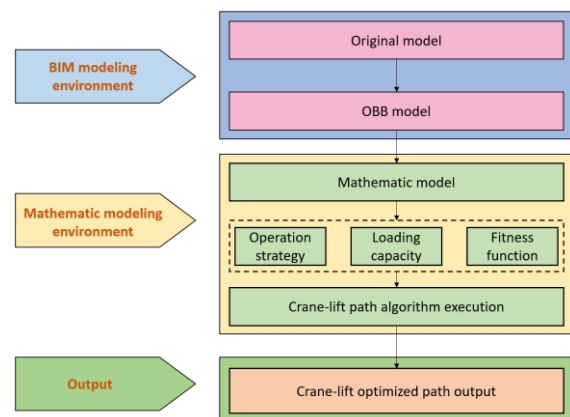


Figure 1. The system architecture of crane-lift path planning

One is the building information modeling (BIM) environment comprising of an original BIM model and an oriented bounding box (OBB) model. Another one is the mathematic modeling environment that gives importance to crane-lift path planning. There are three critical components in mathematical modeling environment: (1) tower crane operation strategy; (2) loading capacity; and (3) fitness function. The output provides the best solution space.

### 3.2 The Development of BIM Modeling Environment

The BIM modeling environment provides all the necessary information of MiC projects.

First, a delicate BIM model is built using various module families in Revit (Figure 2(a)). Second, to degrade computing complexity of the follow-up path planning issues, the original model is simplified as an OBB model (Figure 2(b)), which can perform more accurately than axis-aligned bounding box or bounding sphere algorithms in collision detection trials.

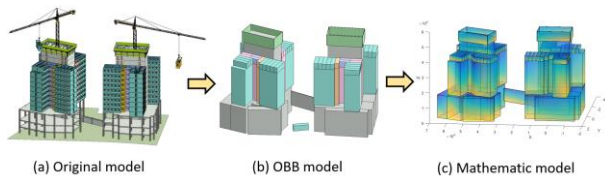


Figure 2. Project modeling process

Since Revit's dedicated project file with a .RVT extension is not compatible with most software, it needs to be converted to a more general type. In this paper, the .RVT file is transformed to a file with an .OBJ extension containing vertices and polygons of each component. By reading the .OBJ file in MATLAB, the mathematical model can be generated (Figure 2(c)).

### 3.3 The Development of Mathematic Modeling Environment

The mathematic modeling environment plays a paramount role in finding a collision-free and high-quality solution.

#### 3.3.1 Problem formulation

##### Assumptions

After conducting a critical literature review, the authors adopted the assumptions of Cai et al. [7], due to the fact that they drew upon the expertise of professionals and scholars.

##### Degree of Freedom (DOF)

The transmission mechanism of a hammerhead tower crane comprises three DOFs, i.e., jib slewing, trolley

movement, and sling hoisting. In addition to the actuated system, the self-rotation angle of the lifting frame is another non-negligible DOF, altered manually by building workers. Therefore, the tower crane is specified to have four DOFs.

For the convenience of research, the work space is converted to configuration space (C space). In C space, the nodes in the path are called configurations. The set of configurations in the solution space can be written as  $\mathbf{X}$ . As a result, the configurations in the  $i$ th path is  $\mathbf{X}^i = (X_{i1}, X_{i2}, X_{i3}, X_{i4})$ , where  $X_{ij}$  is a N-by-1 vector,  $i \in [1, M]$ ,  $j \in [1, 4]$ ,  $M$  denotes the number of paths and  $N$  denotes the number of configurations in each path. The number of columns of  $\mathbf{X}^i$  represents the degrees of freedom, corresponding to the self-rotation angle of the lifting frame (equivalent to the rotation angle of the module, denoted as Rotation), the slewing angle of the jib (denoted as Slewing), the range of trolley movement (denoted as Movement), and the height of the hook (denoted as Hoisting).

Those nodes are connected by abstract edges (denoted as  $\mathbf{E}$ , where  $\mathbf{E}^i \in \mathbf{E}$ ). In fact, the operation of the tower crane from one configuration to another is the edge.

After giving the above definitions of nodes and edges, any path of the tower crane can be expressed as a sling, marked as  $s \subset \mathbf{S}$ .  $\mathbf{S}$  represents the solution space of all paths, whose expression is ruled by

$$\mathbf{S} = \mathbf{X} \cup \mathbf{E} \quad (1)$$

#### Tower Crane Operation Strategy for MiC projects

As discussed above,  $(\mathbf{X}^i)^1$  and  $(\mathbf{X}^i)^N$  represent the start configuration and the end configuration in the  $i$ th path, calculated in advance. In actual crane-lift tasks, workers often manually rotate the module to a proper pose initially, and then the hook goes up. Therefore, for  $(\mathbf{E}^i)^1$ , the first edge, randomly alter Rotation and Hoisting of each sling, while other DOFs are aligned with  $(\mathbf{X}^i)^1$ . When the lifting task is about to end, the hook will slowly descend with the hoisted module. Riggers, in the end, will rotate the module again, placing them to the demand point from a relatively fixed height. Therefore, this paper proposes a pseudo end configuration concept, that is, the end configuration of the planned path is always  $h$  higher than the actual one. In the case of this paper, the height of the module is 3150mm, so  $h$  is assumed as 4000mm. As a result, only stochastically change Rotation and Movement for  $(\mathbf{E}^i)^{N-1}$  and  $(\mathbf{E}^i)^{N-2}$  respectively. The purpose of not putting these two variables in one edge is to ensure their strict sequence. The parameters of in-between edges vary randomly.

#### Loading Capacity and Working Radius

Most of the existing research on tower crane lifting issues has neglected the correlation between modular

weight and working radius of the trolley. In fact, it is challenging to have a unified mathematical expression for multiple of cranes, because the mechanical principles, product materials, etc. of cranes normally differ from each other. To solve the problem, tower crane manufacturers give importance to pre-loading experiments, measuring a series of scattered working radius under corresponding capacities, and providing experimental tables for users to inquire. However, it is not easy to immediately receive the correct working radius once the value of load capacity cannot be found in the table. This paper leverages the cubic spline interpolation method to pre-process those discrete data. Before the algorithm is executed, the maximum working radius will be automatically calculated according to the weight of lifted modules. Figure 3, for example, exhibits the relationship of load capacity and the corresponding working radius of the 50-meter jib of the SANY SYT315 T7530-16 model. When the input value of module weight is 11.85 tonnes, the algorithm will automatically calculate its working radius as 29.29 meters. With a minimum working radius of 3.5 meters, the range of the trolley movement is determined to be 3.5 to 29.29 meters.

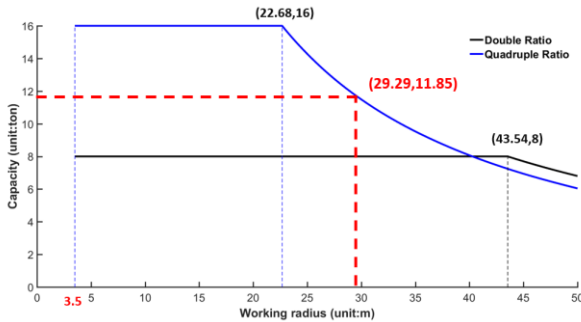


Figure 3. Load capacity diagram of 50m jib (SYT315 T7530-16)

### 3.3.2 The proposed path planning algorithm

#### Introduction and Definition of PSO

For PSO, each path  $s$  can be a representative of the particle in  $C$  space with  $N \times 4$  dimensions, where  $N$  represents the number of configurations in a lifting path. As a result, the position for the  $i$ th particle is denoted as  $\mathbf{X}^i = (\mathbf{X}_{i1}, \mathbf{X}_{i2}, \mathbf{X}_{i3}, \mathbf{X}_{i4})$  identified above, and the velocity for the  $i$ th particle is denoted as  $\mathbf{V}^i = (\mathbf{V}_{i1}, \mathbf{V}_{i2}, \mathbf{V}_{i3}, \mathbf{V}_{i4})$ . The best history positions of each particle and the whole particle swarm (i.e., the population) are logged and represented as  $\mathbf{P}^i = (\mathbf{P}_{i1}, \mathbf{P}_{i2}, \mathbf{P}_{i3}, \mathbf{P}_{i4})$  and  $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4)$  respectively. Additionally, the range of variables is written as 1) position:  $\mathbf{X} \in [-\mathbf{X}_{min}, \mathbf{X}_{max}]$ ; 2) velocity:  $\mathbf{V} \in [-\mathbf{V}_{min}, \mathbf{V}_{max}]$ , where  $\mathbf{X}_{min}$  and  $\mathbf{X}_{max}$  denote the minimum and maximum values among various DOFs of tower crane;  $\mathbf{V}_{min}$  and  $\mathbf{V}_{max}$  correspond to the minimum and maximum

velocities of particles.

The evolution of GA relies on its exclusive genetic operators, such as reproduction, crossover and mutation, while the updating process in PSO is quite different. For the  $i$ th particle, in the  $k$ th iteration, its position and velocity are updated as Equation (2)-(4):

$$\omega_k = \omega_{k-1} \omega_{damp} \quad (2)$$

$$\mathbf{V}_k^i = \omega_k \mathbf{V}_{k-1}^i + c_1 \mathbf{r}_1 (\mathbf{P}_i - \mathbf{X}_{k-1}^i) + c_2 \mathbf{r}_2 (\mathbf{G} - \mathbf{X}_{k-1}^i) \quad (3)$$

$$\mathbf{X}_k^i = \mathbf{X}_{k-1}^i + \mathbf{V}_k^i \quad (4)$$

where  $\omega_k$  is the inertial weight decreasing with time and balancing the results of global and local optimum, in cooperation with  $\omega_{damp}$ , the damping ratio of inertial weight denoted as 0.99 over the iterations;  $c_1$  and  $c_2$  correspond to two positive acceleration constants;  $\mathbf{r}_1$  and  $\mathbf{r}_2$  represent two N-by-4 matrices, where the values are uniformly distributed random numbers in the range [0, 1].

The traduction of PSO in a search space mainly encompasses three characteristics: 1) inertial behavior, the first part of Equation (3), in which particles keep existing velocity flying; 2) personal cognition, the second part of Equation (3), in which the velocity of each particle is randomly increased to the history best solution; 3) social perception, the last part of Equation (3), in which the velocity of each particle is randomly reached the history optimum position of the whole population.

#### Crane-lift Path Generation Mechanism

Figure 4 shows the proposed lifting path generation mechanism, containing four sections, i.e., input, PSO path planner, auxiliary engines and output.

The input information includes confirming the initial population size, the maximum number of iterations, the number of configurations, and various coefficients' values. During the path planner execution, it is necessary to calculate the fitness function value to evaluate the particle's personal and global best outputs and update the particle swarm position and velocity. When PSO reaches the maximum generation, it stops running, performing postprocessing analysis accordingly. Also, the auxiliary engines play a vital role over the iterations. The model regeneration engine assists in creating the mathematical model in MATLAB; then, the crane configuration engine will import the required crane particulars to PSO. In the algorithm iteration process, the collision detection engine will be called all the time to detect constraint points.

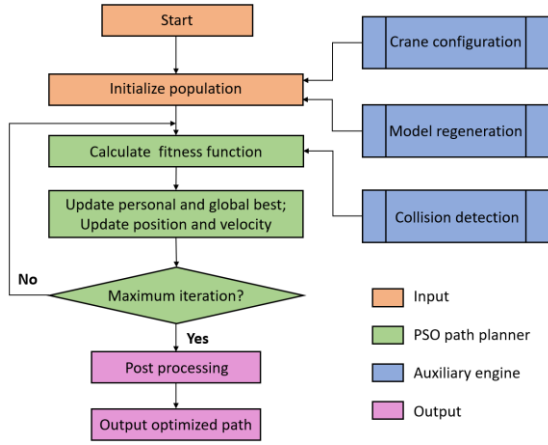


Figure 4. The proposed path generation mechanism

### Fitness Function

The modified fitness functions contain three metrics. To receive a more realistic result, the metric  $r_1$  is written as:

$$r_1(\mathbf{m}, \mathbf{n}) = \sum_{i=1}^4 \rho_i |m_i - n_i| \quad (5)$$

where  $\mathbf{m}$  and  $\mathbf{n}$  are two adjacent configurations in C space;  $m_i$  and  $n_i$  correspond to each DOF of  $\mathbf{m}$  and  $\mathbf{n}$ ; a velocity influencing factor  $\rho_i$  is devised,  $\rho_i = \frac{1}{v_i}$ ,  $\mathbf{v} = (v_1, v_2, v_3, v_4) = (v_{lf}, v_{jb}, v_{ld}, v_{sl})$ , representing the velocities of module rotation, jib slewing, module movement and sling hoisting respectively. The last three parameters are the tower crane's transmission mechanism provided by vendors in advance. In contrast, the first parameter is determined by riggers' operational practice, which means that it is hard to give a specific data range. Its research is out of the scope of the current research, temporarily assumed as the same value of Slewing after the interview with crane operators. Taking SANY SYT315 as an example,  $v_{jb} = 0.75\text{r/min}$  (i.e.  $0.079\text{rad/s}$ ),  $v_{ld} = 100\text{m/min}$  (i.e.  $1666.66\text{mm/s}$ ) and  $v_{sl} = v_{lf} = 80\text{m/min}$  (i.e.  $1333.33\text{mm/s}$ ). As a result, the metric  $r_1$  can be utilized to calculate exactly lifting time.

The metric  $r_2$  is used for computing the movement distance, in accordance with crane's operation strategy, which is ruled by

$$r_2(\mathbf{m}, \mathbf{n}) = \begin{cases} |m_2 - n_2| \cdot |m_3| + \sum_{i=3}^4 |m_i - n_i| & \text{if } m_4 < n_4 \\ |m_2 - n_2| \cdot |n_3| + \sum_{i=3}^4 |m_i - n_i| & \text{if } m_4 > n_4 \end{cases} \quad (6)$$

where  $m_4$  and  $n_4$  represent Hoisting. If  $m_4 < n_4$ ,

this means the position of configuration  $\mathbf{m}$  is lower than  $\mathbf{n}$ ; otherwise, higher.

The metric  $r_3$  is devised for collision detection, which is denoted as

$$r_3((\mathbf{E}^i)^j) = \begin{cases} 1 & \text{condition 1} \\ 0 & \text{condition 2} \end{cases} \quad (7)$$

where the condition 1 means the lifted module collides with obstacles and the condition 2 means the lifting process is collision avoidance.  $(\mathbf{E}^i)^j$  represents the  $j$ th edge in the  $i$ th path. This paper adopted OBB algorithm for crane-lift collision detection.

Based on the above three metrics, a modified fitness function is determined as follows:

$$F(s) = at(s) + \beta d(s) + \gamma c(s) \quad (8)$$

where,

$$\gamma \gg \alpha, \beta, t(s) = \sum_{j=1}^{N-1} r_1((\mathbf{X}^i)^j, (\mathbf{X}^i)^{j+1}) \\ d(s) = \sum_{j=1}^{N-1} r_2((\mathbf{X}^i)^j, (\mathbf{X}^i)^{j+1}), c(s) = \sum_{j=1}^{N-1} r_3((\mathbf{E}^i)^j).$$

Therefore, the issue of finding a collision-free lifting path in C space is transformed to calculate the minimum value of  $F(s)$ .  $\alpha$  and  $\beta$  are two influencing factors for weighting hoisting time and the movement distance of the tower crane, respectively. To some extent, the first metric is to guarantee that the time is as short as possible. The second metric is to ensure that the movement of the tower crane is reduced as much as possible during the installation process. There are two purposes for this. One is to reduce energy consumption and save costs; the other is to reduce the redundancy of the algorithm and enhance the operational space for tower crane operators.

Furthermore, the influencing factor  $\gamma$  of the third metric is much larger than the first two, which ensures that sufficient punishment is imposed to accelerate the elimination of the collision points in the event of a collision.

## 4 Case Study

A real-life MiC project was selected to demonstrate the proposed method and to test the performance of the revised PSO algorithm. The case project is a student residence with more than 900 steel-framed modules, and the project is currently under construction.

### 4.1 Parameter initialization

The initialization data of the tower crane and PSO are shown in Table 1 and Table 2.



Table 1 Tower crane specifications (SANY: SYT315 T7530-16)

Parameter	Input value
$X_1$	(-1.3089, 2.6275, 27988, 0)
$X_{end}$	(-3.0618, 0.8924, 17678, 37500)
$X_{min}$	(-3.1416, -3.1416, 3500, 0)
$X_{max}$	(3.1416, 3.1416, <u>29294</u> , 50000)
$v_{jb}$	0.06
$v_{lf}$	0.06
$v_{ld}$	1000
$v_{sl}$	500
$W_m$	<u>11.85</u>

Note: The underlined fonts indicate that precomputation is required for calculating the maximum working radius.

Table 2 PSO parameters

Parameter	Input value
$N$	6
$M$	50
$\omega$	1
$\omega_{damp}$	0.99
$c_1$	1.5
$c_2$	2
$\alpha$	1
$\beta$	0.01
$\gamma$	10000

Table 3 Experiment results among ten scenarios

Scenario	G	$G_s$	$N_l$	$R_{con}$	$N_g$	$R_{best}$	$C_{max}$	$C_{min}$	$C_{ave}$
1	30	-	-	-	1	3.33%	1027.1385	955.1588	970.9793
2	50	-	-	-	1	3.33%	981.9662	952.9122	959.4187
3	80	-	-	-	1	3.33%	1017.2713	952.8614	958.4079
4	100	87	14	14.00%	1	3.33%	978.0376	952.8613	963.03
5	150	92	59	39.33%	3	10.00%	1017.2608	952.8613	960.1404
6	180	90.7	90.3	50.17%	10	33.33%	1017.2608	952.8613	955.8471
7	200	94.6364	106.3636	53.18%	11	36.67%	1017.2608	952.8613	959.2038
8	250	111.9565	139.0435	55.62%	23	76.67%	1017.2608	952.8613	960.5113
9	300	108.7692	192.2308	64.08%	26	86.67%	978.0366	952.8613	957.8964
10	350	109.1852	241.8148	69.09%	27	90.00%	978.0366	952.8613	955.3788

## 4.2 Analysis and Postprocessing

To verify the applicability of the proposed algorithm, this paper sets the population size to 50, and the independent variable is the evolutionary generation, i.e., maximum iteration, ranging from 30 to 350. This study determined ten different scenarios according to the above interval and ran each one independently 30 times. Table 3 recorded the simulation results, where G means the maximum evolutionary generation in each scenario;  $N_g$  represents the number of minimum global cost over 30 runs;  $G_s$  represents the start generation of the minimum cost (if  $N_g$  is larger than one, calculate the average start generation);  $N_l = G - G_s$ , represents the average number of minimum local cost over the generations;  $R_{con} = N_l/G$  represents the convergence rate;  $R_{best} = N_g/30$  represents the occurrence rate of the global minimum cost over the runs;  $C_{min}$ ,  $C_{max}$ ,  $C_{ave}$  represent the minimum, maximum and average cost of all the minima over 30 runs.

The proposed algorithm shows better explore-exploit trade-off according to the experiment results. When the population has only evolved for 30 generations, i.e., Scenario 1, it is relatively close to convergence. Among the 30 optimal costs in this scenario, the minimum, maximum, and average values are only 0.2411%, 7.7952%, and 1.9014% larger than the global optimum whose value is 952.8613 (Table 3), respectively.  $N_l$ ,  $R_{con}$  and  $R_{best}$  increase steadily with the increase of evolutionary generation, indicating that the algorithm is accelerating convergence, as shown in Figure 5(a) and (b). Additionally, the convergence solution of the algorithm remains unchanged when the generations are more than 100, indicating that the proposed algorithm is highly stable, as shown in Figure 5(c). When the maximum iteration reaches 300, the convergence solutions of 30 independent runs have up to 26 times, and the trend of change is shown in Figure 5(b).

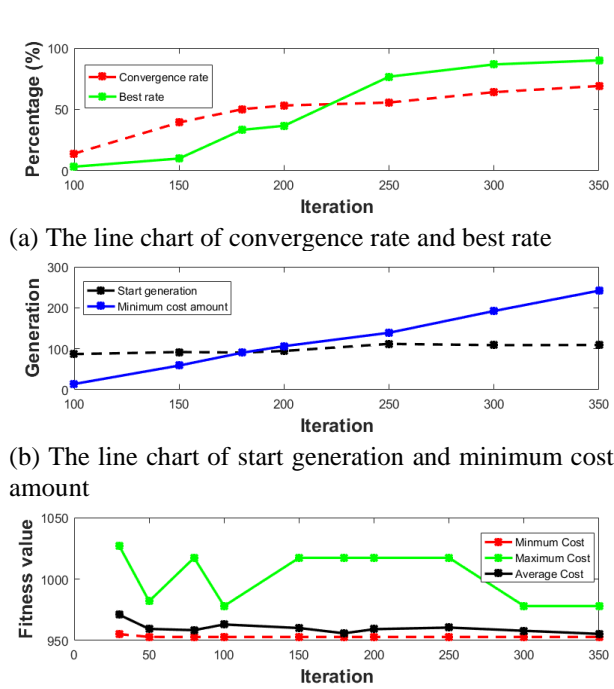


Figure 5. Data visualization among various evolutionary generations

Figure 6 plots the crane-lift path trajectory of Scenario 9. The crane-lift trajectory is visualized in mathematic modeling environment, whose lifting path is smooth and easy to operate. Plus, Figure 7 depicts the convergence curve of Scenario 9, which manifests PSO reaches its optimal value quickly. Therefore, the modified PSO can obtain a better solution space with only a small population size and evolutionary generation.

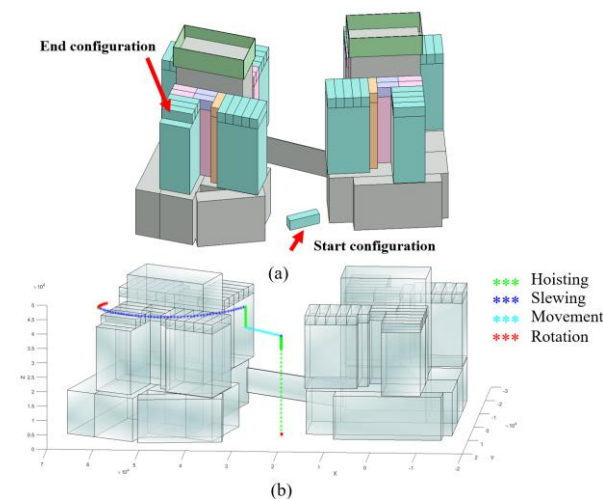


Figure 6. Crane-lift path trajectory visualization in mathematic modeling environment

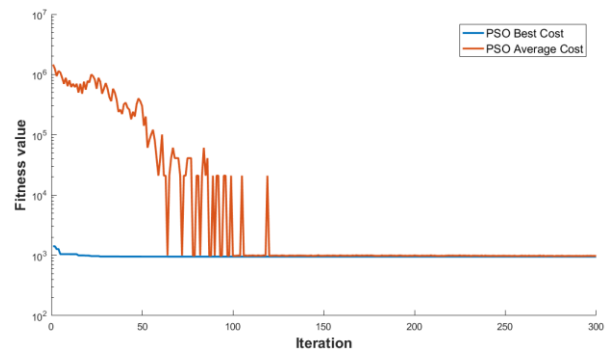


Figure 7. PSO convergence curve of Scenario 9

## 5 Discussion

For the sake of a most awesome optimization performance, a well-designed fitness function is treated as a paramount issue. Cai et al. [7] once improved an evaluation function, which was first created and exhibited by Ali et al. [6] in 2005. In Cai's new function, they introduced a metric that indicates the total variations of all configurations. Although they realized the impact of scaling factors on final performance over the iteration loop, those coefficients are determined basically on their own knowledge and experience. Therefore, in the similar metric of this paper, the undetermined factors are equal to the reciprocal of the actual velocity of corresponding DOF, so that an accurate minimum hoisting time can be acquired.

In relation to the convergence characteristics of PSO, it outperforms GA, while the time complexity is unchanged. Cai [7] used GA for crane-lift path planning. The convergence condition of their algorithm is that the population size is 100 and the evolutionary generation is 400, but the corresponding numbers are 50 and 100, respectively in the proposed algorithm.

Furthermore, the path smoothness is a significant benchmark for verifying solution quality. Nevertheless, most of the existing lifting trajectories are redundant. For example, Wang et al. [12] used ACO for mobile crane path planning, but the visualization result illustrated that there were multiple turns and twists in the trajectory. In comparison, the lifting trajectory is extremely smooth in the developed architecture (see Figure 6).

## 6 Conclusions

This paper has proposed a modified PSO algorithm to automatically conduct crane-lift path planning for high-rise MiC. This research is the first of its kind to introduce PSO for crane-lift path planning, and is an initial and successful attempt of planning crane-lift path in MiC projects. The paper concludes that the proposed algorithm can generate a collision-free crane-lift path with a small population size and a few iterations for

module installation. Also, the lifting trajectory is found to be smooth and easy-to-operate.

Practically, this paper should assist construction planners and crane operators in identifying an optimal crane-lift path, thus facilitating safe and efficient MiC project delivery. Scientifically, the paper contributes to the theoretical development of smart and automated technologies and algorithms in construction. In the future, the research will give importance to multiple-crane path planning in the dynamic environment.

### Acknowledgements

This work is supported by the Research Impact Fund of the HK Research Grants Council (Project No. HKU R7027-18) and the Strategic Public Policy Research Funding Scheme from the Policy Innovation and Co-ordination Office of the Government of the HK Special Administrative Region (Project No. S2019.A8.013.19S).

### References

- [1] W. Pan, C.K. Hon. Briefing: Modular integrated construction for high-rise buildings. *Proceedings of the Institution of Civil Engineers - Municipal Engineer*, 173 (2): 64-68, 2020.
- [2] Z. Zhang, W. Pan. Lift planning and optimization in construction: A thirty-year review. *Automation in Construction*, 118: 2020.
- [3] R.L. Neitzel, N.S. Seixas, K.K. Ren. A review of crane safety in the construction industry. *Appl Occup Environ Hyg*, 16 (12): 1106-1117, 2001.
- [4] V.W.Y. Tam, I.W.H. Fung. Tower crane safety in the construction industry: A Hong Kong study. *Safety Science*, 49 (2): 208-215, 2011.
- [5] P. Sivakumar, K. Varghese, N.R. Babu. Path Planning of Cooperative Construction Manipulators Using Genetic Algorithms. *2000 Proceedings of the 17th ISARC*: 1-6, 2000.
- [6] M.S.A.D. Ali, N.R. Babu, K. Varghese. Collision Free Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm. *Journal of Computing in Civil Engineering*, 19 (2): 182-193, 2005.
- [7] P. Cai, Y. Cai, I. Chandrasekaran, J. Zheng. Parallel genetic algorithm based automatic path planning for crane lifting in complex environments. *Automation in Construction*, 62: 133-147, 2016.
- [8] S. Dutta, Y. Cai, L. Huang, J. Zheng. Automatic re-planning of lifting paths for robotized tower cranes in dynamic BIM environments. *Automation in Construction*, 110: 2020.
- [9] S. Hu, Y. Fang, H. Guo. A practicality and safety-oriented approach for path planning in crane lifts. *Automation in Construction*, 127: 2021.
- [10] C. Zhang, A. Hammad. Improving lifting motion planning and re-planning of cranes with consideration for safety and efficiency. *Advanced Engineering Informatics*, 26 (2): 396-410, 2012.
- [11] A.R. Soltani, H. Tawfik, J.Y. Goulermas, T. Fernando. Path planning in construction sites: performance evaluation of the Dijkstra, A\*, and GA search algorithms. *Advanced Engineering Informatics*, 16 (4): 291-303, 2002.
- [12] X. Wang, Y.Y. Zhang, D. Wu, S.D. Gao. Collision-Free Path Planning for Mobile Cranes Based on Ant Colony Algorithm. *Key Engineering Materials*, 467-469: 1108-1115, 2011.
- [13] J. Kennedy, R. Eberhart. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks: 1942-1948*, Perth, Australia, 1995.
- [14] H. Zhang, J.Y. Wang. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134: 739-748, 2008.
- [15] H. Zhang, F. Xing. Fuzzy-multi-objective particle swarm optimization for time-cost-quality tradeoff in construction. *Automation in Construction*, 19 (8): 1067-1075, 2010.
- [16] A.L. Wenliao Du, Pengfei Ye, Chengliang Liu. Fault diagnosis of plunger pump in truck crane based on relevance vector machine with particle swarm optimization algorithm. *Shock and Vibration*, 20: 781-792, 2013.
- [17] S.C. Yuan, C.F. Wang, H. Xue, F. An, D.H. Wang. The Optimized Design of Tower Crane Hoisting System Based on the Improved Particle Swarm Optimization Algorithm. *Applied Mechanics and Materials*, 548-549: 444-448, 2014.