Automatic Path Planning for Dual-Crane Lifting in Complex Environments Using a Prioritized Multiobjective PGA

Panpan Cai, Indhumathi Chandrasekaran, Jianmin Zheng, and Yiyu Cai

***Abstract*—Cooperative dual-crane lifting is an important but challenging process involved in heavy and critical lifting tasks. This paper considers the path planning for the cooperative dual-crane lifting. It aims to automatically generate optimal dual-crane lifting paths under multiple constraints, i.e., collision avoidance, coordination between the two cranes, and balance of the lifting target. Previous works often used oversimplified models for the dual-crane lifting system, the lifting environment, and the motion of the lifting target. They were thus limited to simple lifting cases and might even lead to unsafe paths in some cases. We develop a novel path planner for dual-crane lifting that can quickly produce optimized paths in complex 3-D environments. The planner has fully considered the kinematic structure of the lifting system. Therefore, it is able to robustly handle the nonlinear movement of the suspended target during lifting. The effectiveness and efficiency of the planner are enabled by three novel aspects: 1) a comprehensive and computationally efficient mathematical modeling of the lifting system; 2) a new multiobjective parallel genetic algorithm designed to solve the path planning problem; and 3) a new efficient approach to perform continuous collision detection for the dual-crane lifting target. The planner has been tested in complex industrial environments. The results show that the planner can generate dual-crane lifting paths that are easy for conductions and optimized in terms of costs for complex environments. Comparisons with two previous methods demonstrate the advantages of the planner, including safer paths, higher success rates, and the ability to handle general lifting cases.**

***Index Terms*—Continuous collision detection (CCD), dual-crane lifting, graphic processing unit (GPU) computing, multiobjective optimization, parallel genetic algorithm, robotic path planning.**

Manuscript received November 24, 2016; revised February 24, 2017; accepted May 19, 2017. Date of publication June 15, 2017; date of current version March 1, 2018. This work was supported by the BeingTogether Center, a collaboration between Nanyang Technological University, Singapore, and the University of North Carolina at Chapel Hill. The BeingTogether Center was supported by the National Research Foundation, Prime Minister's Office, Singapore, under its International Research Centers in Singapore Funding Initiative. Paper no. TII-16-1421. *(Corresponding author: Yiyu Cai.)*

P. Cai was with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798. She is now with the School of Computing, National University of Singapore, Singapore 119077 (e-mail: pcai2@e.ntu.edu.sg).

I. Chandrasekaran is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798 (e-mail: indhu_jaya@yahoo.com).

J. Zheng is with the School of Computer Engineering and the Institute for Media Innovation, Nanyang Technological University, Singapore 639798 (e-mail: ASJMZheng@ntu.edu.sg).

Y. Cai is with the School of Mechanical and Aerospace Engineering and the Institute for Media Innovation, Nanyang Technological University, Singapore 639798 (e-mail: myycai@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2715835

I. Introduction

HEAVY and critical lifting is frequently demanded during the construction, maintenance, and turn-around of industrial plants and work sites. Dual-crane lifting is often used in heavy lifting projects when large cranes with sufficient capacities to lift the load are either not available or prohibited due to space and budget restrictions. In this case, two cranes with relatively lower individual capacities are alternatively used as a cost-effective solution to share the load of the lifting target. Dual-crane lifting is also well used when erection of the lifting target is required. Cooperative dual-crane lifting for heavy and critical lifting is a highly complex task. During the operations, the major crane lifts the heavy target with the assistance of the second crane. The two cranes collaborate to move the lifting target toward the destination, while the balance of the target has to be maintained throughout the lifting process. Clearance must also be kept from obstacles such as equipment and structures in the work site. Moreover, the slings are supposed to keep in vertical directions as much as possible during the cooperative lifting operations in order to avoid undesired increase of sling tensions.

Studies have been conducted on developing simulations and intelligent algorithms to assist the lift planning process, which we call it computer-aided lift planning (CALP). Among many subproblems in CALP, automatic path planning of crane lifting is an important and probably the most challenging one, particularly when it is performed in complex 3-D environments. The path planning problem for dual-crane lifting involves three parts: the cooperative crane pair, the lifting target, and the industrial environment, which can be very complicated. Taking the geometry information of all the parts and kinematics of the lifting system as inputs, the path planning aims to produce safe paths with optimized costs (e.g., motion cost, energy cost, etc.) for dual-crane lifting given any start and end positions.

Path planning of dual-crane lifting has complexities in both the motion of the lifting system and the geometry of the environment. The kinematic structure of the dual-crane lifting system and the movement of the lifting target have been largely simplified in most of the prior arts [1]–[3]. Therefore, studies have been restricted in simple lifting cases. A practical planner would require comprehensive modeling of the system. The complexity of the environment, however, has seldom been discussed in previous works. In fact, complex and cluttered environments would pose huge challenges for both the effectiveness and efficiency of the path planning. Global optimization is another key requirement in path planning for dual-crane lifting. Optimization of lifting paths could substantially reduce the energy and time costs, lessen the probability of human errors, and improve easiness of lifting conductions. Yet existing approaches, including both single-crane and dual-crane works, tend to perform feasible, instead of optimal, path planning. This is probably because optimal planning would require drastically increased computation time compared to finding a feasible solution.

The purpose of this research is to develop a dual-crane path planner that can quickly produce highly optimized dual-crane lifting paths. The planner should be able to handle general lifting cases and complex environments. We achieve these goals by introducing a new solution, which extends the ideas in a previous work [4] for single-crane lifting. This solution has three novel aspects:

1) a comprehensive and computationally-efficient mathematical modeling of the lifting system;
2) a new multiobjective parallel genetic algorithm (GA) designed to efficiently solve the path planning problem;
3) a new efficient approach to perform continuous collision detection (CCD) for the dual-crane lifting target.

GAs [5] perform global optimization by mimicking the Darwinian evolutionary process. Standard GAs are designed for general optimization applications only and thus may not be able to efficiently solve practical problems of higher complexities. Recent developments on GAs have focused on redesigning them for practical industrial applications, such as multimodal manufacturing optimization problems [6] and path planning for unmanned air vehicles [7]. The idea of the recent work published in [4] was to redesign the genetic operators for the single-crane lifting problem and parallelize them following the master–slave framework. The work in [4] also developed an image-space collision detection (CD) approach based on a novel multilevel depth map representation to fully utilize the computational power of graphic processing units (GPUs). The resulting single-crane planner had strong optimization capability. The planner was made highly efficient by using an innovative way to map its functional components and subcomponents into the hierarchical structure of the CUDA architecture [8] in modern GPUs.

In this paper, we not only take the above-mentioned advantages of the optimization path planning solution from [4], but also introduce novel ideas to the problem formulations, designs of new genetic operators, and development of new CD subcomponents for the dual-crane problem, which is very different from the single-crane case. Our solution is not a trivial extension of the previous work. In fact, the dual-crane problem is substantially more complex than the single-crane scenario. The differences lie in three aspects: kinematics of the lifting system, degrees of freedom (DOFs) of the problem, and constraints of the optimization. Kinematics of the dual-crane lifting system has two important features that do not exist in the single-crane case: 1) the cranes, the lifting target, and the ground form a closed kinematic chain; and 2) the lifting target is suspended by two slings of individual cranes. On one hand, the closed kinematic chain makes it impossible to explicitly model the movement of the lifting target as in the single-crane case (for details, see Section III). On the other hand, the suspension system lets the forces, torques, and mass distributions affect the position and orientation of the lifting target (for details, see Section IV). Therefore, an imperative objective of this work is to handle simultaneously these two features of the dual-crane problem. Moreover, due to the additional DOFs involved, the dual-crane problem has a much larger solution space to explore, leading to significantly increased time for searching. Furthermore, additional constraints on the coordination of the two cranes have to be considered in the dual-crane problem. These constraints restrict the feasible space into narrow passages and make the planning even more challenging. As standard GAs perform randomized pseudosampling searches in the solution space [9], they are fairly weak in handling this type of high-DOF space with narrow passages.

Since the dual-crane problem has not been well addressed, this paper presents a novel approach to model, simulate, and plan dual-crane lifting. In terms of modeling, we analyze the kinematic structure, propose a new mathematical formulation of the path planning problem, and provide a simple yet accurate representation of the coordination constraints. In terms of simulation, balance of the suspended lifting target is investigated by concerning its equilibrium state under the effects of kinematics, forces, and torques. Based on these modeling and simulation, we develop an effective and efficient path planner for dual-crane lifting. The planer is based on our new multiobjective master-slave parallel genetic algorithms (MSPGA) with components designed using the lexicographic goal programming (LGP) strategy to assist the planning in narrow and complex high-dimensional spaces. LGP [10] is a classical method for general multiobjective optimization. Here, we incorporate its principle to design our multiobjective GA components. Furthermore, to handle the highly nonlinear motions of the dual-crane lifting target, a new type of swept volume (SV), triangle swept spheres (TSSs), is proposed and applied to perform CCD for lifting paths.

The rest of this paper is organized as follows. Section II reviews the prior arts. Section III gives an overall formulation of the problem. Section IV provides a comprehensive study on the suspension subsystems of dual-crane lifting. Section V describes the TSS-based CCD for the suspended lifting target. The LGP-enhanced planner, which utilizes the formulations and techniques in prior sections, is presented in Section VI. Section VII describes experimental results, comparisons, and discussions. Finally, Section VIII draws a conclusion and discusses the future work.

II. Literature Review

Earlier efforts on CALP include designing simulation systems like HeLPS and CLPS [11]–[13], which were designed to assist plant modeling, setup planning, and physical monitoring, mostly on single-crane lifting. Other works addressed subproblems of CALP. Investigations like [14] discussed the design of controllers for different types of cranes. Determination of crane locations was described and attempted as optimization problems in the work of [15] and [16]. Feasibility checking of crawler cranes walking paths were studied by Lei *et al.* [17], [18] by exploiting configuration spaces (C-space). Optimizing the layout of multiple tower cranes was also elaborated in [19] by using a hybrid particle bee algorithm.

In terms of lifting path planning, most of the research studies have been conducted on the single-crane problem. They applied heuristic depth first searches [20], bidirectional expanding trees [21], probabilistic road maps (PRMs) [3], A* searches [22], and bidirectional rapidly-exploring random tree (RRT) [23] into the single-crane lifting problem. Some of these works could quickly produce lifting paths for relatively simple environments. However, since they only use feasible planning algorithms, which do not aim to produce optimal solutions, the path qualities from these methods are limited. Optimal path planning for single-crane lifting was discussed in a recent work [4]. The single-crane planner in [4] was based on an MSPGA with genetic operators designed for the lifting problem. With properly designed components, the planner has inherited the global optimization capability from the standard GA. Moreover, the image-space CD and multilevel parallelization in GPU has made the single-crane planner highly efficient.

On the other hand, the CD strategy used in the path planning for single-crane lifting can be classified into two categories. The first category of methods exploited precomputed free spaces to speed up the runtime performance [3], [20], [22]. This category of methods had to sacrifice the preprocessing time to perform free space calculations. Another category of studies relied on the online CD strategy by performing CCD during the searches [4], [21], [23]–[25]. This type of methods do not require preprocessing and suit for applications in frequently changing environments causing information inheritance about the environment very difficult, if not impossible. However, for repeated tasks in same or slightly changed environments, this type of planning would waste much time in performing CCD, because shared environment information is not fully utilized. The hybrid CD strategy reported in [4] calculated a 2-D free space for the first two DOFs of the crane and used analytical SVs to handle the movements for the rest of the DOFs. This strategy has provided a balance between the preprocessing time and runtime calculations. It has also reduced the overall planning time for complex environments. In this paper, we take the advantage of the hybrid CD strategy and propose a new type of SV, which suits for the dual-crane problem uniquely.

In the literature, few efforts have been spent on automatic path planning for dual-crane lifting. The problem was initially explored by Sivakumar *et al.* [1] using discrete search algorithms such as hill climbing and A* performed in the C-space of the crane. They applied these algorithms to perform feasible planning. Their results showed that the scalability of these discrete search algorithms, in terms of the complexity of the lifting task, is very limited. The work was further extended in [2] using a serial GA to optimize candidate paths defined in the C-space for dual-crane lifting. This approach was able to provide highly optimized solutions, but was prohibited due to the expensive computation cost of the serial GA. These early works have provided initial ideas for formulating and solving the dual-crane problem. However, their formulations overly simplified the geometries and kinematics of the cranes, making their planners only valid for simple lifting cases. Recently, Chang *et al.* [3] utilized the PRMs in path planning of dual-crane lifting. Their algorithm constrained the tilt angle of the two slings and the lifting target to be strictly zero. Through this simplification, they developed an inverse kinematics path planning algorithm consisting of two stages: a PRM-based planning of the swinging and luffing movements and a rule-based optimization of the hoisting heights. Benefiting from the problem decomposition, their method was able to achieve near real-time performance in the given simple testing environments. Again, these simplifications of the kinematic structure made the method not suitable for general dual-crane lifting cases like erection of the target. These previous methods also lack a reliable CD mechanism to handle the highly nonlinear movements of the lifting target.

In our work, through comprehensive analysis, we provide a more accurate modeling of the dual-crane lifting system (see Sections III and IV). Based on the modeling, we design a robust, effective, and efficient dual-crane planner, which fully considers the kinematics of the system and is able to handle general lifting cases in complex environments. The planner can quickly find solutions under the tight constraints and optimize the paths toward high-quality solutions with low operation costs. In its CD subcomponent, our TSSs can effectively represent the bound to handle the nonlinear motion of the suspended lifting target.

III. Problem Formulation

In this section, we first analyze the overall structure and the DOFs of the dual-crane system and, then, give the mathematical formulation of the path planning problem for dual-crane lifting. This formulation is derived from that in [4] of the single-crane problem by introducing new variables and constraints. In this work, the complex industrial plants or construction sites that the cranes are working in are assumed to be static.

Because of the complex kinematic structure, accurate modeling of the dual-crane lifting system requires special treatments. Particularly, we divide the lifting system into two subsystems for discussions: the manipulation subsystem and the suspension subsystem. The manipulation subsystem contains the swinging and luffing components of the two individual cranes, each producing one open kinematic chain. The slings and the lifting target suspended from the two cranes via the slings form the suspension subsystem. The upper ends of the two slings are linked to pulleys at the tip of the booms. At the lower end, they are attached with the lifting target through hooks and rigging devices. Fig. 1 illustrates the overall structure of the dual-crane lifting system. Note that booms of cranes can only be extended during the time setting up the cranes for lifting tasks. During the lifting process, the lengths of the booms are fixed.

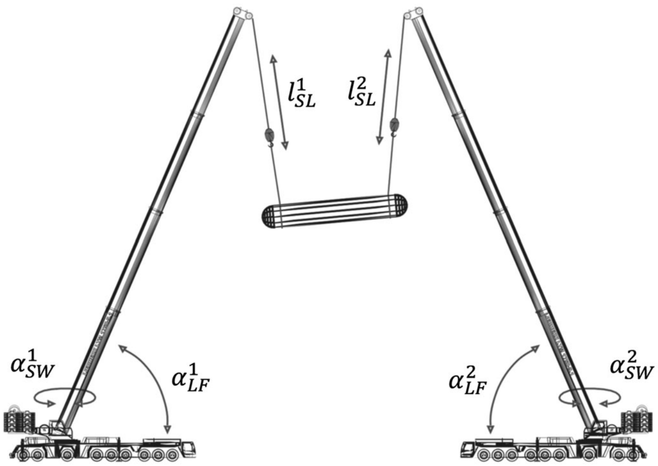

Fig. 1. Kinematics and DOFs of dual-crane lifting with manipulation variables annotated. α_{SW}^i: the swinging angle of the ith crane; α_{LF}^i: the luffing angle of the ith crane; l_{SL}^i: the sling length of the ith crane; The system involves many other parameters to be solved.

Intuitively, the manipulation subsystem has 2×2 DOFs considering the two terrain cranes. The four variables are denoted as $\alpha_{\text{LF}}^1, \alpha_{\text{SW}}^1, \alpha_{\text{LF}}^2$, and α_{SW}^2, representing the swinging and luffing angles of the cranes, respectively. The suspension subsystem is more complex. Generally, each sling in the suspension subsystem has three variables: sling length l, yaw angle θ, and tilt angle ϕ. The lifting target, however, also has five variables: the anchor position $A = (A_x, A_y, A_z)$, tilt angle φ, and yaw angle ϑ. Fortunately, these 11 variables are not independent. The analysis in Section IV will show that, under kinematics constraints and equilibrium conditions, the two sling lengths l_{SL}^1 and l_{SL}^2 can determine the rest of variables. Consequently, the suspension subsystem only has two independent variables or DOFs: l_{SL}^1 and l_{SL}^2. In total, the dual-crane lifting system has $2 \times 2 + 2 = 6$ DOFs. That is, the dual-crane lifting paths are defined in a 6-D C-space denoted as C_{dual} in this paper.

Now, it is ready to formulate the path planning problem of dual-crane lifting. Concluding from previous analyses, a dual-crane configuration can be defined as $c = (\alpha_{\text{LF}}^1, \alpha_{\text{SW}}^1, l_{\text{SL}}^1, \alpha_{\text{LF}}^2, \alpha_{\text{SW}}^2, l_{\text{SL}}^2)$, where α_{LF}^k, α_{SW}^k, l_{SL}^k $(k = 1, 2)$ stand for the luffing angle, swinging angle, and sling length of the kth crane, respectively. A dual-cane lifting path is represented as a string $s_{\text{dual}} = \{O_{\text{dual}}, E_{\text{dual}}\}$ with length L_s, where O_{dual} is the set of nodes (dual-crane configurations) $c_i,\ i = 0, 1, \ldots, L_s - 1$, and E_{dual} represents the set of edges (internal dual-crane paths between independent nodes) $e_j, j = 0, 1, \ldots, L_s - 2$. The edge e_j is determined by nodes c_j and c_{j+1} through linear interpolation of parameters. The task of the path planning is thus to find an optimal s_{dual}^* composed of c_j^*, which maximizes an evaluation function.

The two metric functions d and d' defined in [4] are accordingly adapted for C_{dual} here as

$$d(a, b) = \sum_{i=0}^{6} g(a_i - b_i); \quad d'(a, b) = \sum_{i=0}^{6} r_i |a_i - b_i| \tag{1}$$

TABLE I
PARAMETERS AND VARIABLES IN THE OBJECTIVE FUNCTION

Symbol	Expression
$F(s)$	The evaluation value of string s_{dual}
$d(s)$	The distance cost in string s_{dual}
c_j	The jth configuration in string s_{dual}
λ, λ'	Constant scaling factors

where

$$g(x) = \begin{cases} 1, & \text{if } x \neq 0, x \in R \\ 0, & \text{if } x = 0, x \in R. \end{cases} \tag{1.1}$$

Here, a and b denote, respectively, two dual-crane configurations in space C_{dual}, and $a_i, b_i\ (i = 1, \ldots, 6)$ are the unified representation of the six parameters of a and b. Generally, d represents the number of nonidentical parameters in a and b. d' measures the total number of weighted movement units of the two terrain cranes. The weight r_i indicates the energy cost of the crane when unit movement of the correspondent DOF is conducted. When r_i is set as 1 for all i, the function d' is equivalent to the $\mathbf{L^1}$ norm defined on 6-D vectors.

Now, the maximization problem for the path planning of dual-crane lifting can be written as (see Table I for explanations of the symbols):

$$\max \quad F(s_{\text{dual}}) = \lambda \left(1 + \frac{\lambda}{d(s_{\text{dual}}) + \lambda'(1 + sc(s_{\text{dual}}))}\right) \tag{2}$$

$$\text{s.t.} \quad n_{\text{node}}(s_{\text{dual}}) = 0 \tag{2.1}$$

$$n_{\text{edge}}(s_{\text{dual}}) = 0 \tag{2.2}$$

$$m_{\text{node}}(s_{\text{dual}}) = 0 \tag{2.3}$$

$$m_{\text{edge}}(s_{\text{dual}}) = 0 \tag{2.4}$$

$$\underline{B_{\text{dual}}} \leq c_i \leq \overline{B_{\text{dual}}},\ i = 0, 1, \ldots, L_s - 1 \tag{2.5}$$

$$\text{where} \quad c_i \in C_{\text{dual}},\ s_{\text{dual}} \in S_{\text{dual}},\ i = 0, \ldots, L_s - 1 \tag{2.6}$$

$$d(s_{\text{dual}}) = \sum_{i=0}^{L_s - 2} d'(c_i, c_{i+1}) \tag{2.7}$$

$$sc(s_{\text{dual}}) = \sum_{i=0}^{L_s - 2} d(c_i, c_{i+1}) \tag{2.8}$$

$$n_{\text{node}}(s_{\text{dual}}) = \sum_{i=0}^{L_s - 1} \delta(c_i),\ \delta(c_i) \in \{0, 1\} \tag{2.9}$$

$$n_{\text{edge}}(s_{\text{dual}}) = \sum_{i=0}^{L_s - 2} \delta(e_i),\ \delta(e_i) \in \{0, 1\} \tag{2.10}$$

$$m_{\text{node}}(s_{\text{dual}}) = \sum_{i=0}^{L_s - 1} \sigma(c_i),\ \sigma(c_i) \in \{0, 1\} \tag{2.11}$$

$$m_{\text{edge}}(s_{\text{dual}}) = \sum_{i=0}^{L_s - 2} \sigma(e_i),\ \sigma(e_i) \in \{0, 1\}. \tag{2.12}$$

The objective function includes the motion cost or energy cost $d(s_{\text{dual}})$, which is the sum of costs in all edges, and the operation switching cost $sc(s_{\text{dual}})$ of strings, which is the required number of changes in operation modes. The scaling factor λ' is used to adjust the weight of the two objectives. The weighted sum is then put in a reciprocal. While the cost items decrease in the optimization process, this reciprocal could create continuously increasing selection pressure in the population to assist the convergence of our GA.

Aside from the node collision constraint [see (2.1)], the edge collision constraint [see (2.2)], and the DOF limit constraint [see (2.5)], which also exist in the single-crane problem studied in [4], the dual-crane problem involves two additional hard constraints: *node coordination* constraint [see (2.3)], *edge coordination* constraint [see (2.4)]. To be specific, $\delta(c_i)$ and $\delta(e_i)$ represent the CD results of elements c_i and e_i in string s_{dual}. $\sigma(c_i)$ and $\sigma(e_i)$ represent the coordination test results of elements c_i and e_i in string s_{dual}. The two types of collision constraints require that there are no collision violations in the node configurations (discrete) or the edge paths (continuous). Details of the CD computations will be discussed in Section V. The two coordination constraints require the cranes to be well coordinated in each node and during the movement along the edges (continuous). Details about the coordination of cranes will be discussed in Section IV-B. $\underline{B_{\text{dual}}}$ and $\overline{B_{\text{dual}}}$ stand for the lower and upper bound values for the six DOFs in dual-crane lifting. They are empirically set as $(0^\circ, 0^\circ, 100\text{ cm}, 0^\circ, 0^\circ, 100\text{ cm})$ and $(82^\circ, 360^\circ, 7000\text{ cm}, 82^\circ, 360^\circ, 7000\text{ cm})$, which may vary for different heavy cranes. The optimal solution s^*_{dual} of the maximization problem is a dual-crane lifting path, which is well coordinated, collision free, optimized in energy cost, and easy to be conducted. The aim of the rest parts in this work is to develop an efficient path planning algorithm or path planner to solve this maximization problem.

IV. Suspension Subsystem

Before describing the path planner, we have to first study the unique component in dual-crane lifting, i.e., the suspension subsystem. This new structure brings the major difficulties for the simulation and planning of dual-crane lifting. There are two main problems to solve: 1) to determine the equilibrium state of the suspension subsystem for any given configuration; and 2) to give a simple but accurate model for the coordination constraints. Problem 1 is important for generating correct simulations of the dual-crane lifting process. This simulation can be used to validate output paths from the path planner. Problem 2 is critical for both the effectiveness and efficiency of the path planner.

A. Solving the Equilibrium State of the Suspension Subsystem

Unlike the manipulation subsystem, which can be directly solved using forward kinematics, the suspension subsystem is a complex structure affected by kinematics and physical factors such as forces, torques, and mass distributions. In order to solve the suspension subsystem, we base on several knowledge observed to tackle the problem: 1) the weight of the slings and rigging components are much smaller than the weight of the lifting target. In the heavy lifting context, they could be safely ignored without causing any problem; 2) while at the equilibrium state, all components in the subsystem, including the mass center of the lifting target, sling anchors, and rigging anchors must lie in a single vertical plane. Otherwise, the system would vibrate due to the influence of gravities. Therefore, the equilibrium position of the suspension subsystem can be analyzed in a 2-D plane.

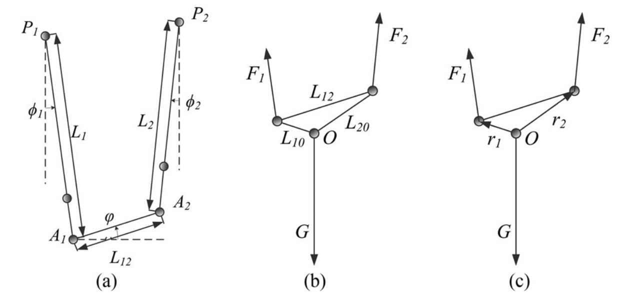

Fig. 2. Kinematics, forces, and moments of the suspension subsystem in dual-crane lifting.

The suspension subsystem can be seen as a closed kinematic chain of three links: two slings and a rigid lifting target. Fig. 2(a) shows its structure. Four rotational joints are connecting these links: sling anchors P_1, P_2, and attach anchors A_1, A_2. The mass center of the lifting target O is also introduced to the system to express the effect of gravity and the sling forces. Without losing generality, the lifting target can be abstracted as a triangle structure, as shown in Fig. 2(b). A group of parameters in the suspension subsystem are constants that can be known at any simulation time. These parameters include the lengths of the slings L_1 and L_2, the distance between the two attach anchors on the lifting target L_{12}, and the distances from attach anchors A_1, A_2 to O, denoted as L_{10} and L_{20}, respectively. The other group are seven unknown variables to be determined. They include ϕ_1 and ϕ_2 (angles between the slings and the z-axis), φ (tilt angle of the lifting target), F_1 and F_2 (tensions on the slings), as well as O. Among them, F_1 and F_2 can be represented with their magnitudes f_1 and f_2, since the directions are following the sling angles.

Three factors are determining the final state of the structure: geometric or kinematic constraints, the equilibrium of forces, and the equilibrium of torques. Geometric constraints include the closure of the kinematic loop and shape preservation of the lifting target triangle. The other two require that the resultant force and torque applied on O have to be zero for the lifting target to be balanced. These analyses lead to the following equations:

$$P_1 + L_1 \begin{pmatrix} \sin(\phi_1) \\ -\cos(\phi_1) \end{pmatrix} + L_{12} \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix} = P_2 + L_2 \begin{pmatrix} -\sin(\phi_2) \\ -\cos(\phi_2) \end{pmatrix} \quad (3)$$

$$||P_1 + L_1 \begin{pmatrix} \sin(\phi_1) \\ -\cos(\phi_1) \end{pmatrix} - O|| = L_{10} \quad (4)$$

$$||P_2 + L_2 \begin{pmatrix} -\sin(\phi_2) \\ -\cos(\phi_2) \end{pmatrix} - O|| = L_{20} \tag{5}$$

$$F_1 + F_2 = G \tag{6}$$

$$r_1 \times F_1 + r_2 \times F_2 = 0. \tag{7}$$

These equations are referred to as *equilibrium equations* in later part of texts. Among them, (3) represents the kinematic loop. It states that the attach anchor A_2 on the lifting target has to be aligned with the tip of the second sling. Equations (4) and (5) describe the shape of the triangle in the lifting target. They require the mass center to be in a fixed relative position in the lifting target, which is a rigid body. Equation (6) containing two dimensions represents the equilibrium of gravity force and the sling forces. Equation (7) states the equilibrium of moments on the mass center of the lifting target applied by the gravity and the sling forces. These nonlinear equations are solved using the "hybrids" solver in the GNU Scientific Library [26], which uses the finite difference approximation of Jacobian functions to guide the direction of searches.

In the proposed planner, this equilibrium solver is used to perform simulations for dual-crane lifting and monitor the safety factors such as sling angles and tensions.

B. Modeling the Coordination Constraints

The key of crane coordination is to maintain the sling tensions or forces in a tolerable range. Insufficient or overly loose coordination constraint may cause overloading for resulting lifting paths, while overconservative constraints will make it hard to find a solution. The design of coordination constraints have to consider the balance between these two factors. Moreover, coordination of the cranes needs to be frequently checked in the planning process. Therefore, the constraints should also be defined in a computationally efficient manner in order to ensure the efficiency of the path planner.

The forces or loads on the slings grow nonlinearly with the increase of tilt angles. Thus, in practice, a safety threshold, denoted as $\phi_{\max}$, is set to the tilt angle of the slings. When the absolute values of ϕ_1 and ϕ_2 are smaller than $\phi_{\max}$, the dual-crane configuration is called "coordinated" in the equilibrium state. Otherwise, the configuration is not properly coordinated. Coordinations of the cranes will be constantly checked during the path planning. Equations (2.11) and (2.12) in Section III imply such a coordination checking process.

There are two possible ways to check coordinations during the lifting path planning. An intuitive choice is to solve the equilibrium equations and check ϕ_1 and ϕ_2 against $\phi_{\max}$. This method is accurate but computationally prohibited due to the high computational load for solving the 7-D nonlinear equation system. The use of the numerical equilibrium solver would also make the planner unstable. Therefore, this subsection provides a way to approximate ϕ_1 and ϕ_2 with some stable (analytical) and easy-to-calculate values.

The key idea of the approximation is to introduce two estimated attach anchors B_1 and B_2 and analyze the possible deviations of the real attach anchors A_1 and A_2 from them. As

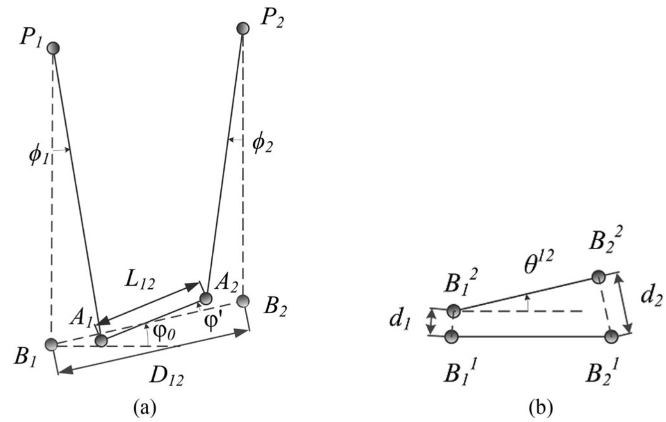

Fig. 3. Two types of coordination of the suspension subsystem. (a) Node coordination. (b) Edge coordination.

shown in Fig. 3(a), B_1 and B_2 stand, respectively, for the positions of A_1 and A_2 when both ϕ_1 and ϕ_2 are set to zero. Thus, there are two relationships for the deviations:

$$|A_1B_1| = L_1 \sin(\phi_1) = L_1\phi_1 + O(\phi_1^3) \tag{8}$$

$$|A_2B_2| = L_2 \sin(\phi_2) = L_2\phi_2 + O(\phi_2^3). \tag{9}$$

For small ϕ_1 and ϕ_2, the above equations are further written as

$$\begin{aligned} |A_1B_1| + |A_2B_2| = &\ L_1|\phi_1| + L_2|\phi_2| + O(|\phi_1|^3) + O(|\phi_2|^3) \\ \approx &\ L_1|\phi_1| + L_2|\phi_2|. \end{aligned} \tag{10}$$

We can thus use the value $L = L_1|\phi_1| + L_2|\phi_2|$ to represent the total deviations of the two attach anchors. We first use this value to construct an initial sufficient condition for the coordination of cranes:

$$L = L_1|\phi_1| + L_2|\phi_2| < \min(L_1, L_2)\phi_{\max}. \tag{11}$$

It is sufficient because we always have

$$\min(L_1, L_2)(|\phi_1| + |\phi_2|) < L. \tag{12}$$

Combining it with (11) shows that the condition ensures $|\phi_1| + |\phi_2| < \phi_{\max}$.

However, (11) is still dependent on ϕ_1 and ϕ_2, which can only be acquired through solving the equilibrium equations. Therefore, more steps are required to simplify the condition. Note that, in Fig. 3(a), when we translate A_1A_2 so that A_2 coincides with B_2, a triangle is formed by A_1, B_1, and B_2 with the length of the bottom edge to be approximately $|B_1A_1| + |A_2B_2|$ or L. By performing projections of A_1A_2 and the bottom edge on B_1B_2, we can get the following relationships:

$$Lcos(\phi_0) = D_{12} - L_{12}\cos(\phi') \tag{13}$$

$$Lcos(\phi_0) = D_{12} - \frac{D_{12}^2 + L_{12}^2 - L^2}{2D_{12}} \tag{14}$$

$$L = \frac{D_{12}}{2\cos(\phi_0)} - \frac{L_{12}^2}{2D_{12}\cos(\phi_0)} + \frac{L^2}{2D_{12}\cos(\phi_0)}. \tag{15}$$

Here, ϕ_0 is the tilt angle of B_1B_2 and ϕ' is the angle between A_1A_2 and B_1B_2 [see Fig. 3(a)]. L_{12} and D_{12} refer to the length of A_1A_2 and B_1B_2, respectively. One fact is that the direction of the lifting target, featured by A_1A_2, will never be tilted for more than 90°. Projecting B_1B_2 onto the bottom edge shows that it will always has

$$L < \ D_{12}\cos(\phi_0). \tag{16}$$

By substituting $D_{12}\cos(\phi_0)$ into the last item in (15), we get

$$L < \ \frac{D_{12}}{2\cos(\phi_0)} - \frac{L_{12}^2}{2D_{12}\cos(\phi_0)} + \frac{L}{2} \tag{17}$$

$$L < \ \frac{D_{12}^2 - L_{12}^2}{D_{12}\cos(\phi_0)}. \tag{18}$$

Finally, the value $e = |D_{12}^2 - L_{12}^2|/\left(D_{12}\cos(\phi_0)\right)$ is no longer dependent on the sling angles, but instead on three easy-to-calculate values D_{12}, L_{12}, and ϕ_0. This value e approximates the deviations of the attach anchors. In the rest of this paper, it is referred to as the *coordination error*.

Using (11) and (18), the final node coordination constraint is formulated as

$$\frac{|D_{12}^2 - L_{12}^2|}{D_{12}\cos(\phi_0)} < \min(L_1, L_2)\phi_{\max}. \tag{19}$$

When this inequality condition is met, it can be easily validated that $|\phi_1|$ and $|\phi_2|$ will be strictly smaller than $\phi_{\max}$. This indicates that the simplified constraint formulated in (19) is still a sufficient condition for coordination of cranes. In special cases where ϕ_0 is near 90°, which means that $\cos(\phi_0)$ is close to zero, the value $D_{12}\cos(\phi_0)$ can be used as the coordination error.

The second task is to formulate the coordination constraint for edges in lifting paths. This is necessary because, when the cranes are following the output path determined by the planner, it is also possible to violate the coordination when the cranes are moving from one node configuration to another. To make sure that coordinations are well preserved in the edges, we constrain the motion of the lifting target between neighboring nodes into short movements. The edge coordination constraint is defined as

$$||(B_1^1 - B_1^2)_{xy}|| + ||(B_2^1 - B_2^2)_{xy}|| \leqslant \lambda(L_1 + L_2)\phi_{\max} \tag{20}$$

$$|(B_1^2 - B_1^1)_z - (B_2^2 - B_2^1)_z| \leqslant \lambda'(L_1 + L_2)\phi_{\max}. \tag{21}$$

Here, B_1^1 and B_2^1 are the estimated attach anchors on the lifting target in time step 1, and B_1^2 and B_2^2 are the estimated attach anchors in the neighboring time step 2 [see Fig. 3(b)]. Equation (20) states that the total distance traveled by the two attach anchors [$d_1 + d_2$ as in Fig. 3(b)] has to be within a small threshold related to $\phi_{\max}$. The scaling factor λ is used to control the tightness of the threshold. Equation (21) constrains the vertical tilting of the lifting target [θ_{12} as in Fig. 3(b)] in a threshold related to $\phi_{\max}$. Synchronized hoisting that does not affect the coordination of the cranes is not constrained in these equations. This edge coordination constraint helps to maintain the sling angles within a tolerable range during the entire dual-crane lifting.

V. TSS-Based CD

CD is an inevitable subcomponent in path planning algorithms. In dual-crane lifting, difficulties of collision checking lie in two major aspects. The first difficulty is to handle complex and cluttered environments in industrial sites and plants. Complexity of the environment brings heavy computational loads to the CD subcomponents, thus seriously challenging the efficiency of the planner. This problem was solved in [4] with a novel image-space CD algorithm, which has been massively parallelized in GPUs. Cai *et al.* [4] has also introduced a hybrid C-space CD strategy to further improve the efficiency. The hybrid strategy maintains precalculated collision information for the movements of the single crane and performs online CCD for the lifting target. In this paper, we make use of these achievements to handle the complexity of industrial environments.

The second difficulty lies in the CCD of the lifting target. This is a problem particular to the dual-crane scenario. As mentioned earlier, one has to solve the nonlinear equilibrium equations to obtain the exact state of the lifting target. The need to use numeral solvers makes the accurate CD of the lifting target inefficient and unstable. So does the commonly used pseudo-continuous approaches [27], which basically consider discrete samples from continuous paths.

Therefore, it is more suitable to use SVs for CCD of the lifting target. An SV is a bounding shape, which covers the entire space swept by an object. Typical SVs include axis-aligned bounding boxes (AABBs) [21], [28] and line swept spheres (LSSs) [29]. They are usually calculated using interval arithmetics [21], Taylor models [28], or velocity bounds [29]. The resulting SVs from the first two approaches are overly conservative for lifting path planning. The latter is impractical because it is very difficult to obtain the velocity of the lifting target. Here, we introduce a new type of SV, namely TSSs. Generally, a TSS is a volume generated by a sphere sweeping through a triangle. It can also be defined as the volume dilated from a triangular core primitive with a uniform radius. By fitting with the kinematic structure, the TSSs can tightly bound the trajectory of the dual-crane lifting target. Since we avoid to solve the equilibrium equations, the state of the lifting target will not be exactly known during the CD process. The uniquely designed TSSs for the planner can also handle this uncertainty of the motions.

Now, we can apply TSSs into the dual-crane lifting problem. Having observed that lifting targets usually have cylindric or capsule-like shapes, we first abstract them as LSSs that are generated by uniformly dilating the line segment between the two attach anchors. The radius of the LSS is the sum of two parts: the size of the lifting target, and the uncertainty threshold of the attach anchors. The first item is intuitive. The second item is based on the fact that the real attach anchors can only move around the estimated one within a small sphere. We define the radius of sphere as the *uncertainty threshold*. The parameters of the LSS represent some state of the lifting target. During the conduction of paths, an edge element basically transforms an initial LSS into another one. The core line segments in the two LSSs are not necessarily in the same plane. Therefore, we discretize the motion of the core lines into two triangles. Two

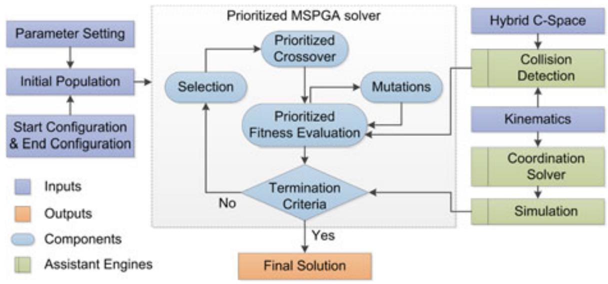

Fig. 4. Workflow of the proposed path planner for dual-crane lifting.

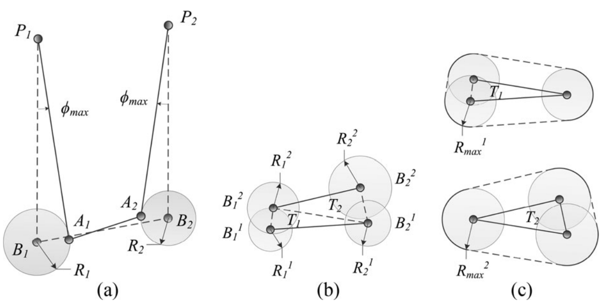

Fig. 5. CCD of the lifting targets. (a) Swinging threshold of the attach anchors. (b) Triangles approximating the swept path of the lifting target during neighboring steps. (c) Volume generated by the CCD triangles with the dilation factor.

TSSs are accordingly constructed by inheriting the radii from the LSSs. We introduce this method for constructing the TSSs in detail below.

For a given node configurations, we first build the LSS for the lifting target using the line segment B_1B_2 with a length of d_{12}. Identical with the denotations in Section IV-B, B_1, B_2, and d_{12} are, respectively, the two estimated attach anchors and the distance between them. The size r of the lifting target is calculated from its bounding box, such that the capsule with radius r covers the whole geometry. Next is to determine the uncertainty threshold. As the sling angles are restricted within a threshold $\phi_{\max}$ by the coordination constraints, real positions of attach anchors A_1 and A_2 will only vary within the neighborhood of B_1 and B_2 [see Fig. 5(a)] with a range proportional to $\sin(\phi_{\max})$. The final radii R_1 and R_2 on the two ends of the LSS can be thus written as

$$R_1 = L_1 \sin(\phi_{\max}) + r \tag{22}$$

$$R_2 = L_2 \sin(\phi_{\max}) + r. \tag{23}$$

Since $\phi_{\max}$ is a small angle, the equations can be simplified as

$$R_1 \approx L_1 \phi_{\max} + r \tag{24}$$

$$R_2 \approx L_2 \phi_{\max} + r. \tag{25}$$

For any two configurations c^1 and c^2, the estimated attach anchors B_1^1, B_2^1 in c^1 and B_1^2, B_2^2 in c^2 form two triangles $B_1^1B_1^2B_2^1$ and $B_1^2B_2^1B_2^2$ [see Fig. 5(b)]. The triangles also inherit the radii R_1^1, R_2^1, R_1^2, and R_2^2 from the LSSs. As a result, two TSSs are constructed: T_1 with vertices B_1^1, B_1^2, B_2^1 and radius $R_{T1} = \max(R_1^1, R_1^2, R_2^1)$, and T_2 with vertices B_1^2, B_2^1, B_2^2 and radius $R_{T2} = \max(R_1^2, R_2^1, R_2^2)$ [see Fig. 5(c)].

Algorithm 1: Pseudocode of the CCD for the Lifting Target.

1: **for all** String s^i_{dual} in the population P_{dual} **do**
2: **for all** Edge e_j in string s^i_{dual} **do**
3: $B_1^1 \leftarrow Crane1ForwardKinematics(c_j)$;
4: $B_1^2 \leftarrow Crane1ForwardKinematics(c_{j+1})$;
5: $B_2^1 \leftarrow Crane2ForwardKinematics(c_j)$;
6: $B_2^2 \leftarrow Crane2ForwardKinematics(c_{j+1})$;
7: $T_1 \leftarrow (B_1^1, B_1^2, B_2^1, \max(R_1^1, R_2^1, R_1^2))$;
8: $T_2 \leftarrow (B_1^2, B_2^1, B_2^2, \max(R_2^1, R_1^2, R_2^2))$;
9: $\text{AABB}_{T1} \leftarrow \text{Bound}_{x-y}(T_1)$;
10: $\text{AABB}_{T2} \leftarrow \text{Bound}_{x-y}(T_2)$;
11: **for all** pixel entry p covered by AABB_{T1} **do**
12: $d \leftarrow \text{Proximity}(p, B_1^1B_1^2B_2^1)$;
13: **if** $d <= R_{T1}$ **then**
14: $\delta(e_j) \leftarrow 1$; Report collision;
15: **end if**
16: **end for**
17: **for all** pixel entry q covered by AABB_{T2} **do**
18: $d \leftarrow \text{Proximity}(q, B_1^2B_2^1B_2^2)$;
19: **if** $d <= R_{T2}$ **then**
20: $\delta(e_j) \leftarrow 1$; Report collision;
21: **end if**
22: **end for**
23: **if** No collision reported **then**
24: $\delta(e_j) \leftarrow 0$;
25: **end if**
26: **end for**
27: $n_{\text{edge}}(s^i_{\text{dual}}) \leftarrow \sum_{j=0}^{L_s-2} \delta(e_j)$;
28: **end for**

The CCD of the lifting target has now been converted to the collision checking between the TSSs and points in the multilevel depth maps [4], [30]. Algorithm 1 describes the detailed workflow of the TSS-based CCD algorithm. For any edge e_j in a path, the estimated attach anchors at its two end configurations c_j and c_{j+1} are first computed using forward kinematics (lines 3–6 in Algorithm 1). Then, the two TSSs are constructed following the above-mentioned method (lines 7 and 8). Afterwards, regions on the xy plane that are covered by the TSSs are invoked for pixelwise collision checking (lines 9 and 10). For each point in the multilevel depth map of the plant (lines 11 and 17), the algorithm computes its distance from the core triangles (lines 12 and 18). If the distance is smaller than the radius of the TSS, the algorithm reports collision (lines 13, 14, 19, and 20). If no such case happens in all covered pixels, the algorithm considers the edge as collision free (lines 23 and 24). The collision checking results for all the edges are summed up to be a part of the edge collision violation number in the fitness function (see Section VI-C) (line 27). The whole CD process is conducted with GPU kernels in a massively parallel way.

VI. A Dual-Crane Path Planner Using Prioritized Optimization

In this section, we present the path planner for the dual-crane lifting problem. The fundamental idea underlying the planner is that the objectives and constraints of the optimization problem described in Section III can all be defined as goals to be sequentially optimized. For example, the coordination constraints are equivalent to optimizing the number of coordinated nodes and edges in a path. In this way, the problem can be treated as an unconstrained multiobjective optimization problem with a sequence of goals, which can be solved using GAs.

Importantly, some of these goals have to be strictly fulfilled (the constraints), while others are more soft (the objectives). Moreover, the coordination constraints of the problem are very tight and easily violated. Standard multiobjective GAs that consider all goals as soft objectives are not able to solve this type of problem. Therefore, we introduce a lexicographical approach. We define priorities for the goals and use the LGP strategy to handle them. Doing so, the constraints and objectives can be solved or optimized one by one, till a feasible and optimal lifting path is finally obtained. We assign the priorities based on two principles below. First, constraints should have higher priorities than objectives. This is to avoid wasting time in optimizing the cost of infeasible solutions. Second, among the multiple constraints, we assign the priorities according to their tightness and try to solve the loose ones first. For instance, constraints for nodes are given higher priorities because they are easier to be satisfied than the constraints for edges. Coordination constraints are assigned with higher priorities than collision constraints because the mutation operators are helping to achieve them. Following these principles, the priorities are given from high to low as: node coordination, edge coordination, collision avoidance for nodes, collision avoidance for edges, and, finally, the path costs.

Inheriting the MSPGA framework in [4], we assign the functional components of the planner, i.e., selection, fitness evaluation, crossover, and mutations into the GPU for parallel computations. The iterations and the flow in each iteration are controlled by the central processing unit. In the following parts, we will describe in detail how these problem-specific components are designed and how the LGP strategy is embedded into these components.

A. Framework of the Planner

The framework of the proposed path planner is illustrated in Fig. 4. Its kernel is a prioritized MSPGA solver designed using the above ideas. In this solver, dual-crane lifting paths are abstracted as linear chromosomes. Each gene in the chromosomes represents a node configuration on the path, while two neighboring genes implicitly denote an edge of the path. Taking the geometries, kinematics, and lifting task specifications as inputs, the solver constructs an initial population of chromosomes and evolves it to optimize the goals. Optimal path planning is thus performed through the evolutionary iterations. The solver has four major components: fitness evaluation, selection, crossover, and mutations. The lexicographically defined fitness function evaluates the "fitness" values of chromosomes or candidate paths. It uses two subcomponents of the planner. The first subcomponent is the coordination checking engine, which evaluates the coordination constraints introduced in Section IV-B. The other is the CD engine that applies the techniques to be introduced in Section V. According to the fitness values, the selection operator generates a mating pool for reproduction. For each chromosome, the chance of being selected into the mating pool is proportional to its fitness. Afterwards, a multiobjective crossover operator is performed for candidates in the mating pool to produce offsprings. These offsprings are then further processed by the customized mutation operators. The resulting paths replace the original population as a new generation to be processed in the next iteration. As in [4], elitism strategy is also applied here. The above procedure is iteratively performed till a final solution or path is obtained. Due to the complexity of the problem, it is impossible to know the exact optimal path. Therefore, the simulation engine is exploited for users to determine whether the solution is satisfactory. The engine has incorporated the equilibrium solver developed in Section IV-A to accurately simulate the motions. If the output path is not optimal, the planner will continue the iterations to further optimize the path. This simulation engine also helps to validate the safety factors in output paths. In the following sections, we will describe the above-mentioned key components in detail.

TABLE II
INITIALIZATION STRATEGY FOR THE SEEDS IN THE DUAL-CRANE POPULATION

Parameter	Strategy
α^1_{SW} & α^2_{SW}	Interpolate start and end swinging angles
α^1_{LF} & α^2_{LF}	Interpolate start and end luffing angles
l^1_{HS} & l^2_{HS}	Randomly choose hoisting height

B. Initialization With Seeds

Paths in the initial population are often randomly generated in the initialization process. However, given the large and complex solution space of the dual-crane path planning problem, random initialization can hardly provide useful genes that could evolve to form a feasible or an optimal solution. This would largely constrain the performance of GA-based planners. Here, we solve this issue by introducing "seeds" into the initial population. The "seeds" are candidate paths constructed by disturbing the straight line from the start configuration to the goal in the C-space of dual-crane lifting. In detail, swinging angles and luffing angles of the "seeds" are generated through linearly interpolating the values in the start & end configurations. To perform disturbance, hoisting values are determined by choosing a random height for the lifting target. The sling lengths are acquired through subtracting the maximum hoisting value of the cranes (related to luffing angle) with the selected height. Table II summarizes this strategy. Although these "seeds" are usually not feasible solutions, they are more likely to provide good building blocks for feasible or even optimal solutions through mutations.

C. Lexicographic Fitness Function

The fitness function used in the planner combines the objective function described in (2) and the hard constraints of

TABLE III
PARAMETERS AND VARIABLES USED IN THE FITNESS FUNCTION

Symbol	Expression
s^i_{dual}	The ith dual-crane string in the population
m_i	The violation number of coordination constraints in s^i_{dual}
n_i	The violation number of collision constraints in s^i_{dual}
d_i	The path cost of s^i_{dual}
mo_i	The coordination violation number of the nodes in s^i_{dual}
me_i	The coordination violation number of the edges in s^i_{dual}
no_i	The violation number of node collision in string s^i_{dual}
nc_i	The violation number of edge collision for the cranes in string s^i_{dual}
nt_i	The violation number of edge collision for the lifting target in string s^i_{dual}
L_p	The size of population

the optimization problem described in (2.1)–(2.4). It is the first component in the planner that has incorporated the LGP strategy. This fitness function is designed as a piecewise continuous function coarsely reflecting the previously defined goal priorities. Effectively, it divides the evolving process of each candidate path into three stages: coordination, collision avoidance, and cost optimization. Each stage has its own range of fitness value, so that candidates in different stages can be differed.

Specifically, the fitness value for a given chromosome s^i_{dual} in the population P_{dual} is defined as (see Table III for explanations of the symbols):

$$f(s^i_{\text{dual}}) = \begin{cases} \lambda_1/m_i, & \text{if } m_i > 0 \\ \lambda_1(1 + 1/n_i), & \text{if } m_i = 0, n_i > 0 \\ \lambda_1(2 + \lambda_1/d_i), & \text{if } m_i = 0, n_i = 0 \end{cases} \tag{26}$$

where

$$m_i = mo_i + me_i \tag{27}$$

$$n_i = no_i + nc_i + nt_i \tag{28}$$

$$i = 0, 1, 2, \ldots, L_p - 1. \tag{29}$$

The case $m_i > 0$ indicates that the cranes are not well coordinated at m_i nodes or edge in the string s^i_{dual}. When $n_i > 0$, it means there are totally n_i nodes and edges in string s^i_{dual} colliding with the environment. d_i includes the motion cost and operation switching cost as defined in Section III. Following the objective function, all the three cases use reciprocal forms to increase the selection pressure.

It can be seen that the function has three disjoint value ranges. Its value lies in $(0, \lambda_1]$ when $m_i > 0$ and within $(\lambda_1, 2\lambda_1]$ when $m_i = 0$ and $n_i > 0$. For cases when $m_i = 0$ and $n_i = 0$, which means that s^i_{dual} is a feasible solution, the fitness values are larger than $2\lambda_1$. In this way, paths violating higher priority constraints will surely have lower fitness values, leading to less survival chances. This means that the planner biases its optimization process to solve higher priority goals.

D. Multiobjective Genetic Operators

The proposed planner uses two types of genetic operators: crossover and mutation. Both are used to reproduce new candidate paths. Ideally, the crossover operator should combine good elements from a selected pair of parents and thus create better offsprings. The mutation operators that alter partial information in paths should be able to explore new possibilities of feasible solutions. Their designs should be highly problem specific. Otherwise, the operators may cause the loss of the existing good genes and building blocks.

In the single-crane planner designed by Cai *et al.* [4], the genewise crossover tends to eliminate invalid genes. When genes are both valid or invalid, it would make a random choice. Unfortunately, in the dual-crane scenario, as the coordination constraints are very tight, almost all genes and edges are invalid in starting generations, which means that the crossover would become a purely random operator in the stage. In this case, the performance of the planner will be largely degraded. To solve this problem, we consider each constraint separately in the dual-crane planner and construct a multiobjective crossover operator.

This idea is implemented by using the LGP strategy. Algorithm 2 illustrates the described crossover operator. The crossover happens with a rate r_c for each pair of parent chromosomes in the mating pool. To determine each location c_i in the offspring, this operator takes the candidate genes from the corresponding position of the parents (line 3 in Algorithm 2). Comparisons between the parent genes are done in a lexicographic way following the priorities of the goals: node coordination (lines 4–9), edge coordination (lines 10–15), node collision (lines 16–21), edge collision (lines 22–27), and, finally, the local motion cost (lines 28–33). Starting from the highest priority, penalty values are assigned to the two parent genes if they violate the goal or constraint (lines 4, 10, 16, and 22). Then, the offspring chooses the one with a lower value (lines 5, 6, 11, 12, 17, 18, 23, and 24). If the two parent genes have the same value, the operator continues to evaluate the penalties for next goal (lines 8, 14, 20, 26, and 32), till the better one is identified. In cases when the genes have identical penalties for all the goals, the planner chooses either one randomly (lines 34 and 35). This genewise crossover strategy helps the planner eliminate configurations that violate higher priority constraints. This way, it gradually directs the search toward the narrow feasible space of the multiconstraint problem. The function $ChooseBetter$ used in Algorithm 2 chooses one from c^1_j and c^2_j according to their penalties g_1 and g_2.

Three types of mutations are used in the proposed planner: bitwise, smoothing, and coordination mutations. Among them, the bitwise and smoothing mutations are extended versions of those in [4]. The coordination mutation is uniquely designed for the dual-crane problem.

The bitwise mutation disturbs bits (parameters in genes) in selected chromosomes with a given scale to perform local refinements. For the start and end configuration in each chromosome, it only alters the hoisting height (sling lengths of the cranes). Mutation scales of well-coordinated or collision-free

Algorithm 2: Pseudocode of the Crossover Operator in the Proposed Path Planner for Dual-Crane Lifting.

for all Offspring s^i_{dual} subject to crossover **do**
  for all Gene c_j in string s^i_{dual} **do**
    Load parent genes c^1_j and c^2_j from the mating pool;
    Penalty $g_1 \leftarrow \sigma(c^1_j)$, penalty $g_2 \leftarrow \sigma(c^2_j)$;
    if $g_1 \neq g_2$ **then**
      $c_j \leftarrow ChooseBetter(c^1_j, g_1, c^2_j, g_2)$;
    else
      Remain undetermined;
    end if
    $g_1 \leftarrow \sigma(e^1_{j-1}) + \sigma(e^1_j)$, $g_2 \leftarrow \sigma(e^2_{j-1}) + \sigma(e^2_j)$;
    if $g_1 \neq g_2$ **then**
      $c_j \leftarrow ChooseBetter(c^1_j, g_1, c^2_j, g_2)$;
    else
      Remain undetermined;
    end if
    Penalty $g_1 \leftarrow \delta(c^1_j)$, penalty $g_2 \leftarrow \delta(c^2_j)$;
    if $g_1 \neq g_2$ **then**
      $c_j \leftarrow ChooseBetter(c^1_j, g_1, c^2_j, g_2)$;
    else
      Remain undetermined;
    end if
    $g_1 \leftarrow \delta(e^1_{j-1}) + \delta(e^1_j)$, $g_2 \leftarrow \delta(e^2_{j-1}) + \delta(e^2_j)$;
    if $g_1 \neq g_2$ **then**
      $c_j \leftarrow ChooseBetter(c^1_j, g_1, c^2_j, g_2)$;
    else
      Remain undetermined;
    end if
    $g_1 \leftarrow d(c^1_j, c^1_{j+1})$, $g_2 \leftarrow d(c^2_j, c^2_{j+1})$;
    if $g_1 \neq g_2$ **then**
      $c_j \leftarrow ChooseBetter(c^1_j, g_1, c^2_j, g_2)$;
    else
      Remain undetermined;
    end if
    if Undetermined **then**
      $c_j \leftarrow RandomChoose(c^1_j, c^2_j)$;
    end if
  end for
end for
return s^i_{dual}

genes are kept small in order to preserve the good features. Invalid genes are disturbed largely to explore the unknown solution space. Table V shows the details of the bitwise mutation operator.

Algorithm 3 illustrates the procedure of the smoothing mutation. Its purpose is to smoothen candidate paths and help the edges achieve coordination. The smoothing mutation picks genes from invalid chromosomes with a certain chance (lines 2–4 in Algorithm 3) and replaces some of its parameters with random convex combinations of the values from neighboring genes (lines 5–7). This process is performed only when the neighboring genes are collision free.

TABLE IV
PARAMETERS AND VARIABLES IN ADAPTIVE MUTATION RATES

Symbol	Expression
$r_m(s_{\text{dual}})$	The mutation rate of string s_{dual}
$\overline{r_m}$	The basic mutation rate
$\overline{f}$	The average fitness value in the population
$f(s_{\text{dual}})$	The fitness value of string s_{dual}

TABLE V
BITWISE MUTATION OPERATOR

Configuration	Case	Strategy
c_1 & c_{L_s-2}	$\delta(c_i) + \delta(e_i) = 0$	Alter hoisting height in smaller scale
	$\delta(c_i) + \delta(e_i) > 0$	Alter hoisting height in larger scale
$c_2 \sim c_{L_s-3}$	$\sigma(c_i) + \sigma(e_i) = 0$ or $\delta(c_i) + \delta(e_i) = 0$	Alter any configuration parameter in smaller scale
	$\sigma(c_i) + \sigma(e_i) > 0$ and $\delta(c_i) + \delta(e_i) > 0$	Alter any configuration parameter in larger scale

Algorithm 3: Pseudocode of the Smoothing Mutation Strategy in the Proposed Path Planner for Dual-Crane Lifting.

1: **for all** String s^i_{dual} in the population P_{dual} **do**
2:   **if** $n_i = 0$ or $m_i > 0$ **then**
3:     **for all** Gene c_j in string s^i_{dual} **do**
4:       **if** $Rand(0,1) < r_m$ **then**
5:         Randomly choose variable a^j_k for mutation;
6:         $\lambda \leftarrow Rand(0,1)$;
7:         $a^j_k \leftarrow \lambda a^j_{k-1} + (1-\lambda) a^j_{k+1}$;
8:       **end if**
9:     **end for**
10:   **end if**
11: **end for**
12: **return** s^i_{dual}

In the initial population, usually very few or none of the genes are well coordinated. Many of the them are actually very far from the coordinated configurations. Apparently, it is unreasonable to rely on the bitwise mutation for producing well-coordinated genes. Therefore, we introduce a new coordination mutation operator into the path planner, which replaces the original invalid configurations by recalculated well-coordinated ones. Algorithm 4 gives details on the coordination mutation. For any selected gene, the operator uses forward kinematics to calculate the estimated attach anchors (line 5 in Algorithm 4) in order to determine the position and orientation of the lifting target (lines 6 and 7). Using the lifting target as a reference, the operator computes the correct configurations of the two cranes using inverse kinematics (lines 8–10). This new dual-crane configuration then serves as the mutated gene. By doing so, the coordination mutation can continuously supply well-coordinated genes for the population.

Algorithm 4: Pseudocode of the Coordination Mutation Strategy in the Proposed Path Planner for Dual-Crane Lifting.

for all s^i_{dual} subject to mutation **do**
  if $m_i > 0$ **then**
    for all c_j in s^i_{dual} **do**
      if $\sigma(c_j) = 1$ **then**
        $B_1, B_2 \leftarrow FK(c_j)$;
        $Dir \leftarrow Normalize(B_2 - B_1)$;
        $L \leftarrow ||B_2 - B_1||$;
        $d \leftarrow (L - L_{12})/2$;
        $B'_1 \leftarrow B_1 + d(Dir), B'_2 \leftarrow B_2 - d(Dir)$;
        $c_j \leftarrow (IK(B'_1), IK(B'_2))$;
      end if
    end for
  end if
end for
return s^i_{dual}

Similar as in [4], the planner uses adaptive rates in all the three types of mutations. The mutation rate r_m for a chromosome is calculated by (see Table IV for the symbols)

$$r_m(s_{\text{dual}}) = \begin{cases} \min\left(\overline{r_m} + (\overline{f} - f(s_{\text{dual}}))/\overline{f}, 1\right), & \text{if } f(s_{\text{dual}}) < \overline{f} \\ \overline{r_m}, & \text{if } f(s_{\text{dual}}) \geq \overline{f}. \end{cases} \tag{30}$$

In this adaptive scheme, low-quality paths are assigned with higher mutation rates for active explorations of the solution space. High-quality paths are subject to lower mutation rates in order to preserve their good features.

The described initialization strategy, fitness function, and genetic operators constitute the kernel of the proposed path planner. They make the planner being able to efficiently search in the large and complex solution space, and quickly find safe dual-crane lifting paths that satisfy all the constraints and are optimized in costs.

VII. Results and Analysis

A. Path Planning in Complex Industrial Environments

In order to verify that the proposed method supports efficient path planning for dual-crane lifting in complex environments, Experiment 1 is conducted on a complex industrial plant. The plant model contains 376 205 triangles and 274 108 vertices including a great number of complex piping structures and various equipment. The experiment lifting case moves a 10-m-long lifting target from the ground into the gap between two sets of complex piping structures. The lifting target is required to bypass a set of obstacles that are up to 23 m high and rotate the lifting target horizontally for over 90°.

The dual-crane lifting path generated by the proposed planner is illustrated in Fig. 6. The trajectory of the lifting target is

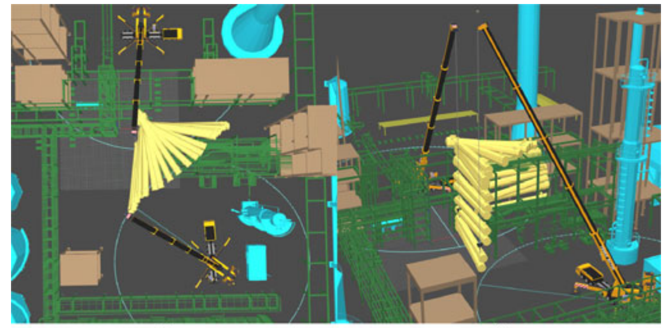
(a) (b)

Fig. 6. Dual-crane lifting path generated in Experiment 1. (a) Top view. (b) Side view.

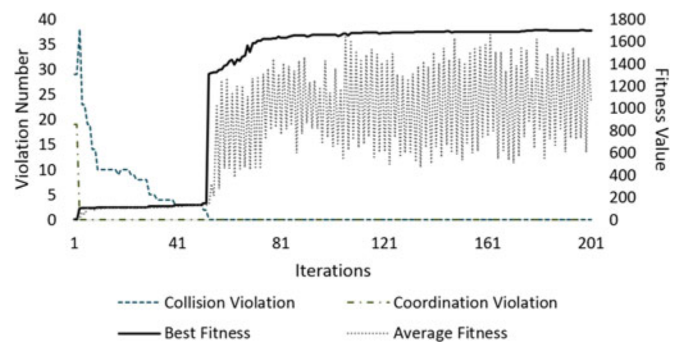

Fig. 7. Fitness convergence trend in Experiment 1. Left vertical axis: labels for collision and coordination violation numbers. Right vertical axis: labels for best and average fitness values.

displayed in yellow. The lifting path consists of 26 steps. The target is first hoisted to the height of the obstacle and slowly rotated to the destination orientation with intensive coordinations. Finally, the lifting target is lowered into the gap between the structures. This solution is achieved within 200 iterations, which consumes 2 s of execution time. As shown in the fitness value trend plotted in Fig. 7, the planner has found a feasible solution near the 50th iteration and come to convergence near the 100th iteration. No violation of collision or coordination happens during the movements. For this specific lifting case, the success rate of the planner is 98% for 50 trials.

B. Performance Comparison With a GA-Based Method

To verify the improvements brought by our planner, we compare our algorithm with the GA-based planner proposed in Ali's work [2]. Experiment 2 is conducted in the benchmark environment containing three cuboid obstacles used in [2]. The lifting target of over 10 m long is lifted to undergo a parallel movement from one side of the obstacles to another. 200 runs of trials are conducted with both the proposed method and Ali's method. To conduct a fair comparison, we also use the proposed parallel CD engine in the implementation of Ali's method. Moreover, we parallelized the original serial components in Ali's method to enjoy the GPU acceleration. Performance of the methods, in terms of success rates and solution qualities with different numbers of iterations, is shown in Figs. 8 and 9. The success

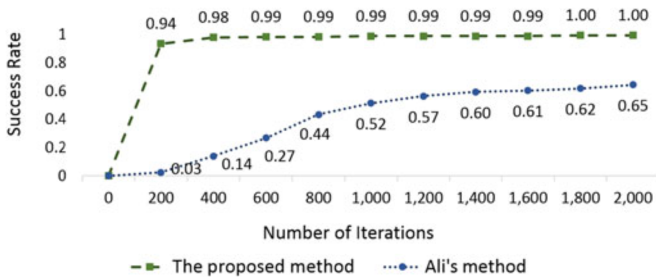

Fig. 8. Comparison of the success rate using Ali's method and the proposed method under different numbers of iterations.

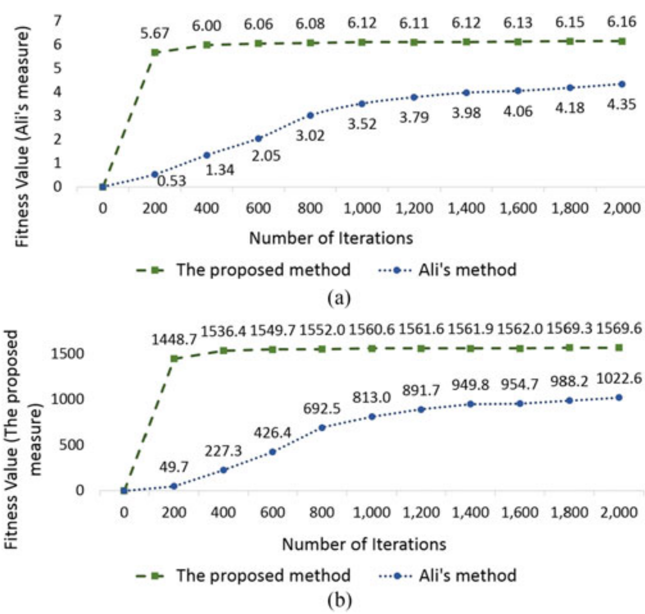

Fig. 9. Comparison of the solution qualities using Ali's method and the proposed method under different numbers of iterations (Experiment 2). (a) Using Ali's measure. (b) Using the proposed measure.

rate of the proposed method in Experiment 2 reaches 94% in the first 200 iterations, while that of Ali's method is still 3%. With the increase of the iteration number, the success rate of the proposed method approaches 100%. With 2000 iterations, the proposed method is able to achieve a success rate of 99.5%, about 54% higher than that of Ali's method. Fig. 9(a) and (b) illustrates the solution qualities measured in both the proposed fitness function and Ali's fitness function. The trend is similar to that of the success rates. At finishing the first 200 runs, the ratio of the average fitness value in the proposed method with that in Ali's method is nearly 11 when both measured with Ali's fitness function. The ratio is around 29 when measuring with the proposed fitness function. At the end of 2000 iterations, the average fitness value of the best chromosome achieved by the proposed method is 41–54% higher than that obtained by Ali's method. In this specific lifting case, the average number of iterations required for finding a feasible solution using the proposed method is 61 iterations (see Table VI), far smaller than the 743 iterations required by Ali's method. The time required to complete one iteration using the proposed method is slightly higher

TABLE VI
COMPARISON WITH ALI'S METHOD ON THE NUMBER OF ITERATIONS REQUIRED FOR FINDING FEASIBLE SOLUTION AND THE EXECUTION TIME FOR EACH ITERATION (EXPERIMENT 2)

	Number of iterations	Time per iteration (ms)
The proposed method	61	8.28
Ali's method (GPU parallelized & using the proposed parallel CD)	743	7.64

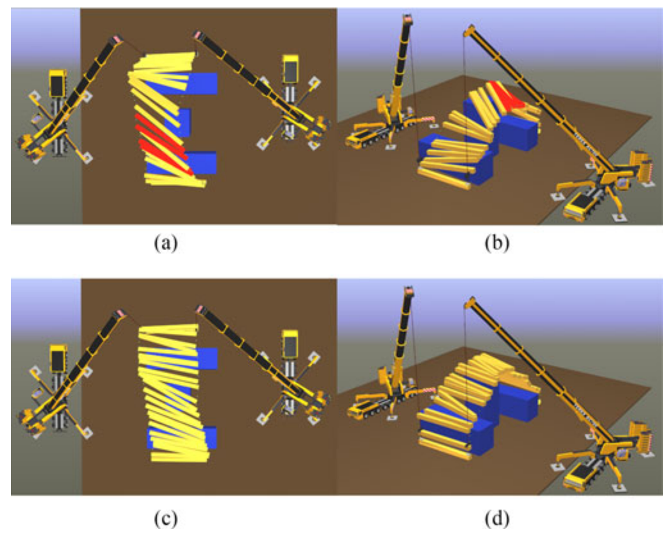

Fig. 10. Sample path generated with the method of [2]: (a) top view and (b) side view. And the path generated with the proposed method: (c) top view and (d) side view.

than using Ali's method (this is the result with GPU parallelization and the proposed CD engine. The original planning time stated in [2] was 12 min). This is because the proposed method considers more constraints such as edge coordination.

The lack of consideration for internal movements between path steps in Ali's method leads to possible violations of constraints in the output paths. As an example, two dual-crane lifting paths output from Ali's method and the proposed method are shown in Fig. 10. The path generated with Ali's method, even though looks smoother and shorter than that in Fig. 10(c) and (d), violates the coordination constraint in five sections of internal movements (highlighted in red color).

In order to produce a clearer view of the difference, Experiment 3 is conducted to compare the sling angles, the resulting extra loads, tilting of the lifting target, and the coordination errors $e = |D_{12}^2 - L_{12}^2| / (D_{12}\cos(\phi_0))$ between paths produced by the two methods. Ten successful runs of executions are tested for both the proposed method and Ali's method. Tables VII and VIII show the maximum and average values, respectively. The maximum sling angle of the cranes during conduction of the paths generated with Ali's method is three times larger than that for the paths generated with the proposed method. The tilting of slings causes up to maximum 3.28% more load shared on one of the cooperative cranes, which leads to higher

TABLE VII
COMPARISON ON THE BALANCING PROPERTIES IN THE SAMPLE PATHS OUTPUT BY ALI'S METHOD AND THE PROPOSED METHOD IN EXPERIMENT 3 (MAXIMUM VALUES)

	Maximum sling angle (degree)	Maximum extra load (%)	Maximum tilt angle of the lifting target (degree)	Maximum coordination error (cm)
The proposed method	1.82	1.26	7.14	266.68
Ali's method	5.38	3.28	18.4779	980.05

TABLE VIII
COMPARISON ON THE BALANCING PROPERTIES IN THE SAMPLE PATHS OUTPUT BY ALI'S METHOD AND THE PROPOSED METHOD IN EXPERIMENT 3 (AVERAGE VALUES)

	Average sling angle (degree)	Average extra load (%)	Average tilt angle of the lifting target (degree)	Average coordination error (cm)
The proposed method	1.69	0.84	5.31	240.30
Ali's method	3.56	2.31	14.72	533.19

overloading risks. This risk may also be measured by the coordination errors, which is 72.8% smaller in the paths generated with the proposed method. On the other hand, the average tilting of the lifting target in the paths generated with the proposed method is maintained around 5°, nearly 64% reduction compared with the paths generated with Ali's method.

The above results show that the proposed method has greatly improved the convergence of the GA searches and is able to achieve higher quality solutions within much shorter time. The results also validate that the proposed method is able to output safe paths that are well coordinated for the whole conduction process.

C. Comparison With a PRM-Based Method

The effectiveness of the proposed algorithm is also compared with a previous method [3] (Experiment 4) based on the state-of-the-art PRM approach, which samples configurations in the C-space to construct road maps. Their method has simplified the problem by constraining both the sling angles and the tilt angle of the lifting target to be zero so that the dual-crane lifting system can be solved by inverse kinematics. An optimal path generated in the proposed algorithm is shown in Fig. 11 together with two paths presented in Chang's paper [3]. Both paths from [3] let the lifting target follow linear motions interpolating keyframe locations in the Euclidean space, one of which climbed along the obstacles. The other path traveled around the main obstacle. The underlying problem of the paths is brought by the fact that the tips of crane booms who are driving the lifting target through cables are only allowed to move on a spherical surface. Linear motions of the end effector, when projected back to the polar system of the configuration space, would be nonlinear arcs. Comparatively, the path generated by the proposed algorithm appears to be slightly irregular in the Euclidean space. But, when the path is drawn in the C-space, it reveals a linear, smooth, and axis-aligned pattern, which is much easier for the cranes and human operators to track. On the other hand, Chang's algorithm relied on zero tilt angle of slings and the lifting target.

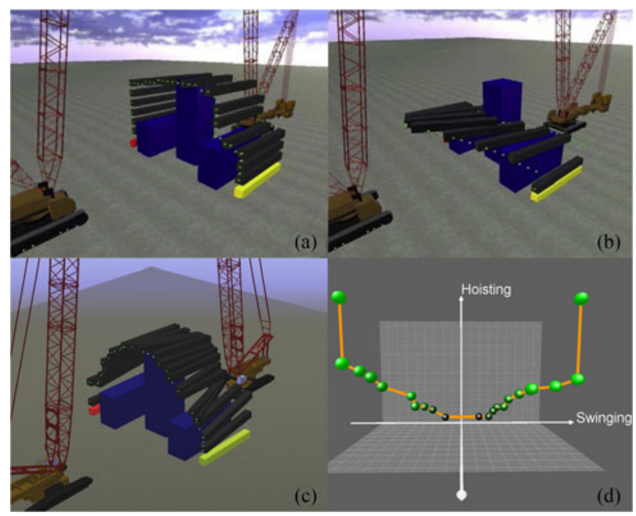

Fig. 11. Comparison of two paths generated by (a) and (b) the method of [3] and (c) the proposed method. (d) is the C-space path of the major crane conducting the lifting path shown in (c). Lighter green stands for smaller luffing angle.

This constraint reduced the complexity of the planning process by reducing two DOFs for the PRM searches. However, This assumption will be invalid when the cranes are asked to fulfill erection tasks requiring tilting of the lifting target, which are frequently demanded in industrial applications. Consequently, the proposed GA-based planner performing forward kinematics path planning is able to produce optimized paths in a more smooth and nature way. By accepting tolerable sling angles and allowing the tilting of the lifting target, it also suits for more types of demands.

D. Effect of the String Length Setting

The string length of chromosomes in the population is an important parameter to be set for each planning task. Here, we have conducted Experiment 5 to illustrate the effect of the string length setting. We measure the success rates and planning

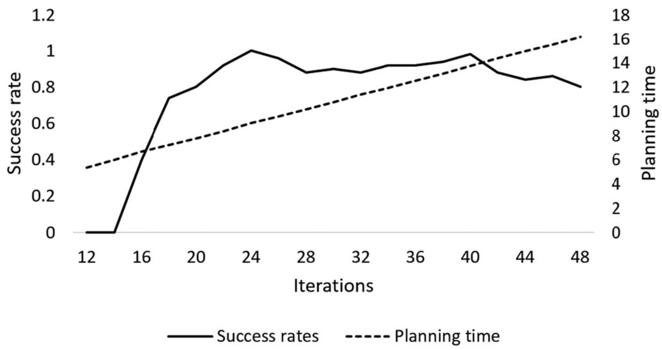

Fig. 12. Success rates and planning time of the planner (with 200 iterations) under different string length settings.

time of the planner for the same lifting case as explained in Experiment 1 under different string length settings. These results are plotted in Fig. 12.

The success rate curve in Fig. 12 shows that the algorithm is quite robust regarding the string length setting. The planner can achieve reasonable success rates (close to or higher than 80%) under a wide range of string length settings (from 18 to 48 as tested). The optimal success rate of the planner is achieved around 24–26 as the string length for this specific case, which explains the setting used in Experiment 1. When the string length is set to be too small (12–14 for this case), the planner is not able to produce a feasible solution (success rate is 0). This is because large-stride movements along the edges in these settings cannot satisfy the edge coordination constraint. When the length is set to be relatively larger (>26), the success rate of the planner decreases slowly, as it leads to more complex model of paths that require longer time to optimize. Another nature effect of choosing larger string lengths is the linear increase of the calculation time, as indicated by the planning time curve in Fig. 12.

The quality of lifting paths output by the planner is also influenced by the string length setting. Fig. 13 shows the lifting paths generated using different string lengths (16, 24, 32, and 40). All the four cases generate similar results, but the paths generated using longer strings are less optimized within the given 200 iterations (as they lift the target higher than necessary). However, the coordination error e decreases significantly when using longer strings to represent paths.

In conclusion, choosing the string length setting requires an overall consideration of planning time, success rate, solution quality, and safety factors. Using smaller but acceptable string lengths make the planner faster, by trading off the safety factors. On the other hand, using longer strings improves the safety factors, but requires more iterations and planning time to achieve optimal solutions. In practice, one can follow an empirical equation to set the string length:

$$L_s = \frac{4D_{\max}}{\lambda(L^1_{\min} + L^2_{\min})\phi_{\max}}. \tag{31}$$

Here, $D_{\max}$ is the maximum XY distance traveled by the two attach anchors on the lifting target. $L^1_{\min}$ and $L^2_{\min}$ stand for the minimum sling lengths of the two cranes as required by the lifting task. Other notations are identical as defined in (21). Equation (31) calculates the string length setting according to the tightness of the edge coordination constraint. It provides an initial value of the string length, while the user can fine-tune it according to requirements of specific lifting tasks.

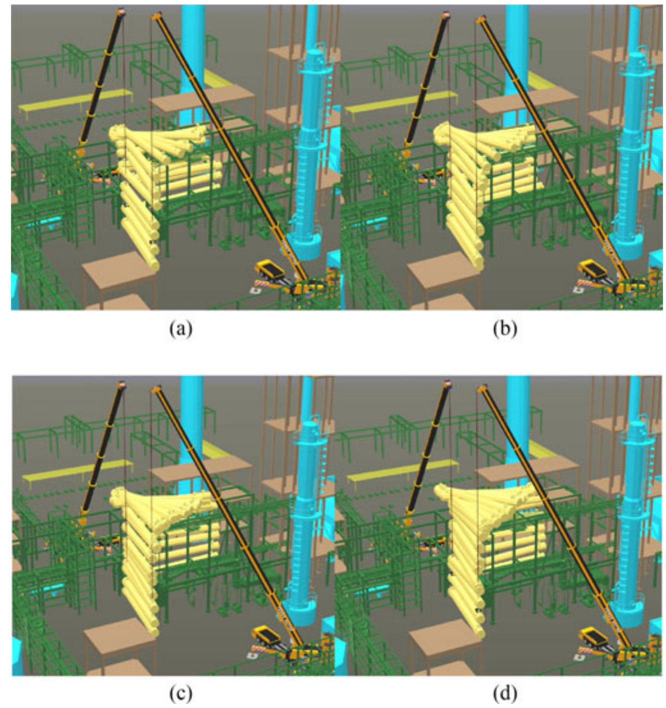

Fig. 13. Lifting path output by the planner (with 200 iterations) under different string length settings (L_s). e represents the coordination errors of the paths. (a) $L_s = 16$, $e = 427.9$. (b) $L_s = 24$, $e = 353.7$. (c) $L_s = 32$, $e = 302.9$. (d) $L_s = 40$, $e = 270.7$.

VIII. Conclusion

This paper proposes a new solution to perform automatic path planning for dual-crane lifting. The solution defines the path planning as a multiobjective multiconstraint optimization problem and solves it using a path planner with a novel prioritized multiobjective GA to perform effective and efficient global optimization. Different from previous approaches, both the formulation of the optimization problem and the path planner are developed upon comprehensive analyses on the structure of the dual-crane lifting system. The coordination checking and CD subcomponents of the path planner reflect the kinematic features of the dual-crane lifting system and are made computationally efficient in the meantime. As a result, the path planner is able to effectively compute safe, optimal, and smooth dual-crane lifting paths, within a short time, for complex environments and general lifting cases. We successfully tested the path planner in a complex industrial environment and validated its advantages through comparisons with two previous methods.

This research is conducted under the assumption that dual-crane lifting can be generally regarded to be pseudostatic throughout the whole process, especially for the cranes.

In practice, when physical factors such as vibrations of booms and wind conditions participate in the process, the planner has to be enhanced by including these factors. Therefore, the future direction of the research is to take into consideration of such dynamics information, for example, speeds and accelerations, as well as swinging of the lifting target due to the influence of winds.

Dynamic environment is another aspect not considered in this paper. The proposed method is discussed under the assumption that the environment is static, i.e., not changing over time. Fortunately, for dynamic environments, it is possible to utilize information from the offline planning to perform fast online re-planning, when local changes in the environment have been detected. For example, the replanning can be achieved by using the GA population obtained at the last termination as the initial population for the new GA search for online replanning. If the environment is only partially changed, it is expected that the mechanism may quickly generate new suitable paths. However, a potential problem of this mechanism is that the existing population might have been conquered by several elite paths and is thus not able to provide enough diversity for future searches. In this case, we can generate new random chromosomes and replace a portion of the redundant elite paths in the population. In order to make these random seeds survive, the selection pressure may need to be reduced in the starting phases.

References

[1] P. Sivakumar, K. Varghese, and N. R. Babu, "Automated path planning of cooperative crane lifts using heuristic search," *J. Comput. Civil Eng.*, vol. 17, no. 3, pp. 197–207, 2003.

[2] M. A. D. Ali, N. R. Babu, and K. Varghese, "Collision free path planning of cooperative crane manipulators using genetic algorithm," *J. Comput. Civil Eng.*, vol. 19, no. 2, pp. 182–193, 2005.

[3] Y.-C. Chang, W.-H. Hung, and S.-C. Kang, "A fast path planning method for single and dual crane erections," *Autom. Construction*, vol. 22, pp. 468–480, 2012.

[4] P. Cai, Y. Cai, I. Chandrasekaran, and J. Zheng, "Parallel genetic algorithm based automatic path planning for crane lifting in complex environments," *Autom. Construction*, vol. 62, pp. 133–147, 2016. http://dx.doi.org/10.1016/j.autcon.2015.09.007.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[6] C. H. Chen, T. K. Liu, and J. H. Chou, "A novel crowding genetic algorithm and its applications to manufacturing robots," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1705–1716, Aug. 2014.

[7] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[8] *NVIDIA CUDA C Programming Guide*, Version 3.2, NVIDIA Corporation, Santa Clara, CA, USA, 2010.

[9] J. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–72, 1992.

[10] C. Romero, "Naive prioritization and redundancy in LGP1 (in collaboration with Francisco Amador)," in *Handbook of Critical Issues in Goal Programming*. Amsterdam, The Netherlands: Pergamon, 1991, ch. 4.

[11] W. Hornaday, C. Haas, J. O'Connor, and J. Wen, "Computer-aided planning for heavy lifts," *J. Construction Eng. Manage.*, vol. 119, no. 3, pp. 498–515, 1993.

[12] K.-L. Lin and C. T. Haas, "An interactive planning environment for critical operations," *J. Construction Eng. Manage.*, vol. 122, no. 3, pp. 212–222, 1996.

[13] K. Varghese, P. Dharwadkar, J. Wolfhope, and J. T. O'Connor, "A heavy lift planning system for crane lifts," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 12, no. 1, pp. 31–42, 1997.

[14] C. Y. Chang, "Adaptive fuzzy controller of the overhead cranes with nonlinear disturbance," *IEEE Trans. Ind. Informat.*, vol. 3, no. 2, pp. 164–172, May 2007.

[15] H. Safouhi, M. Mouattamid, U. Hermann, and A. Hendi, "An algorithm for the calculation of feasible mobile crane position areas," *Autom. Construction*, vol. 20, no. 4, pp. 360–367, 2011.

[16] J. Olearczyk, M. Al-Hussein, and A. Bouferguène, "Evolution of the crane selection and on-site utilization process for modular construction multilifts," *Autom. Construction*, vol. 43, pp. 59–72, 2014.

[17] Z. Lei, H. Taghaddos, U. Hermann, and M. Al-Hussein, "A methodology for mobile crane lift path checking in heavy industrial projects," *Autom. Construction*, vol. 31, pp. 41–53, 2013.

[18] Z. Lei, S. Han, A. Bouferguène, H. Taghaddos, U. Hermann, and M. Al-Hussein, "Algorithm for mobile crane walking path planning in congested industrial plants," *J. Construction Eng. Manage.*, 2014.

[19] L.-C. Lien and M.-Y. Cheng, "Particle bee algorithm for tower crane layout with material quantity supply and demand optimization," *Autom. Construction*, vol. 45, pp. 25–32, 2014.

[20] H. Reddy and K. Varghese, "Automated path planning for mobile crane lifts," *Comput. Aided Civ. Infrastruct. Eng.*, vol. 17, no. 6, pp. 439–448, 2002.

[21] S. Redon, Y. J. Kim, M. C. Lin, and D. Manocha, "Fast continuous collision detection for articulated models," in *Proc. 9th ACM Symp. Solid Model. Appl.*, 2004, pp. 145–156.

[22] J. Olearczyk, A. Bouferguene, M. Al-Hussein, and U. R. Hermann, "Automating motion trajectory of crane-lifted loads," *Autom. Construction*, vol. 45, pp. 178–186, 2014.

[23] Y. Lin, D. Wu, X. Wang, X. Wang, and S. Gao, "Lift path planning for a nonholonomic crawler crane," *Autom. Construction*, vol. 44, pp. 12–24, 2014.

[24] S. Kang and E. Miranda, "Planning and visualization for automated robotic crane erection processes in construction," *Autom. Construction*, vol. 15, no. 4, pp. 398–414, 2006.

[25] P. Cai, Y. Cai, I. Chandrasekaran, and J. Zheng, "A GPU-enabled parallel genetic algorithm for path planning of robotic operators," in *GPU Computing and Applications*, Y. Cai and S. See, Eds. New York, NY, USA: Springer, 2015, pp. 1–13.

[26] B. Gough, *GNU Scientific Library Reference Manual*, 3rd ed. Surrey, U.K.: Network Theory Ltd., 2009.

[27] M. Held, J. T. Klosowski, and J. S. B. Mitchell, "Collision detection for fly-throughs in virtual environments," in *Proc. 12th Annu. Symp. Comput. Geom.*, 1996, pp. 513–514.

[28] X. Zhang, S. Redon, M. Lee, and Y. J. Kim, "Continuous collision detection for articulated models using Taylor models and temporal culling," *ACM Trans. Graph.*, vol. 26, Jul. 2007, Art. no. 15.

[29] J. A. Corrales, F. A. Candelas, and F. Torres, "Safe human-robot interaction based on dynamic sphere-swept line bounding volumes," *Robot. Comput.-Integr. Manuf.*, vol. 27, pp. 177–185, Feb. 2011.

[30] P. Cai *et al.*, "Collision detection using axis aligned bounding boxes," in *Simulations, Serious Games and Their Applications (ser. Gaming Media and Social Effects)*, Y. Cai and S. L. Goei, Eds. Singapore: Springer, 2014, pp. 1–14.

Panpan Cai received the Ph.D. degree in mechanical engineering from Nanyang Technological University, Singapore, in 2017.

She is a Research Fellow with the School of Computing, National University of Singapore. Her research interests include industrial automation, motion planning, and artificial intelligence.

Indhumathi Chandrasekaran received the Ph.D. degree in mechanical engineering from Nanyang Technological University, Singapore, in 2012.

She is a Research Fellow with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. Her research interests include medical image processing, serious games, virtual reality, and engineering applications.

Jianmin Zheng received the B.S. and Ph.D. degrees in applied mathematics from Zhejiang University, Hangzhou, China, in 1986 and 1992, respectively.

He is an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published more than 150 technical papers in international journals and conferences. He has served on the program committee of many international conferences. His recent research focuses on T-spline technologies, digital geometric processing, virtual reality, visualization, simulation, reality computing, and applications.

Dr. Zheng is an Associate Editor of *The Visual Computer*.

Yiyu Cai received the Ph.D. degree in mechanical engineering from National University of Singapore, in 1996.

He is a faculty member with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. His research interests include computer-aided engineering, virtual reality, and artificial intelligence for medical, engineering, and educational applications.