Action Sequence and Position Planning for Crane Lifting based on CS and AStar

Wan Da Guo, Xin Wang, Bo Jiao and Ru Ming Teng

December 4, 2019

Action Sequence and Position Planning for Crane Lifting based on CS and AStar

1st Wan Da Guo
School of Mechanical Engineering of
Dalian University of Technology
DaLian, China
460608972@qq.com

2nd Xin Wang
School of Mechanical Engineering of
Dalian University of Technology
DaLian, China
wangxbd21@163.com

3rd Bo Jiao
DaLian YiLiYa Construction Machinery
Co..LTD.
DaLian, China
tianfu275@163.com

4th Ru Min Teng
School of Mechanical Engineering of
Dalian University of Technology
DaLian, China
tengrumin@163.com

Abstract- Based on Cuckoo Search (CS) and A*, a combine algorithm is presented for planning feasible crane's action sequence and position. Firstly, A* algorithm is improved according to the characteristic of crane's action as A*-circle algorithm for planning object's path. And the path can be transformed easily into crane's action sequence. Secondly, for finding the best crane's position, the feasible area consisting of crane's position points is defined. And the quality of the action sequence searched by the A*-circle is used as the objective function of CS. Finally, after iterations, the best crane position within the feasible position area can be found by CS. The smaller the number of crane action change and the shorter the moving distance of the object, the higher the quality of action sequence. When the crane is at the best position found by CS, a high quality actions sequence can be found by A*-circle algorithm. In order to verify the effectiveness of the combined algorithm, a simulation is performed in PreLifting software.

Keywords-Crane, Action Sequence, CS, A*-circle, Crane's Position

I. Introduction

The action sequence planning of crawler crane is essential for a heavy lifting project. Lots of previous works had been done in path planning. Based on Rapidly-exploring Random Tree (RRT) [1-2], a path planning algorithm was used to search crane's path. RRT can avoid spatial modeling and solve path planning problems with high-dimensional space and complex constraints efficiently. Considering the low exploring inefficiency of basic RRT due to single dimensional growing, Bi-directional RRT was presented to improve the convergence speed [3]. Yu-Cheng Chang proposed a method using the probabilistic road map (PRM) to search a collision-free path in the configuration space [4]. An Jianqi proposed a path planning algorithm of automobile crane based on two-dimensional mapping of three-dimensional object [5]. Wang X proposed an improved ant colony approach for crane path planning, which took into full account of not only the shortest path but also crane operation safety [6]. Lei, Z proposed an algorithm for mobile crane path planning, which considered the typical site constraints, the geometry of lifted modules, and crane configuration [7-8]. The Cuckoo Search (CS) was a meta-heuristic algorithm proposed by Yang and Deb [9]. And this algorithm was often used to solve optimization problems.

However these algorithms do not fully consider the special DOFs of the crane, so it is difficult to translate crane's path into action sequence. And crane's position is not fully considered too.

So, in this paper, CS and A* are improved to solve path planning problem considering both crane's DOFs and position.

II. Problem Formulations

The crane action sequence planning problem can be summarized as follow: finding a high-quality crane action sequence to move the object from the picking position to the placing position, and avoid crane and object collide with obstacles. Therefore, the problem can be formulated as following:

$$P = (LP, En, q_{picking}, q_{place}, f) \qquad (1)$$

Where,

LP: Crane's position;

En: Work environment include obstacles;

$q_{picking}, q_{place}$: Object's start position and target position;

f: The function of action planning

III. Searching Crane's Position

In nature, some species of cuckoos produce eggs in the host's nest. If a host bird realizes that eggs are not its own, it will abandon eggs or re-establish a new nest elsewhere. Based on cuckoos' behaviors, Yang and Deb proposed CS algorithm in 2009. In this algorithm, Levy Flight was used to generate the next generation of nests after iteration. Yang and Deb proposed the following three idealization rules to describe Multi-objective Cuckoo Search [10].

- Each cuckoo lays K eggs at a time, and dumps them in a randomly chosen nest. Egg k corresponds to the solution to the *k*th objective.
- The best nests with high quality of eggs (solutions) will carry over to the next generations.
- Each nest will be abandoned with a probability p_a, and a new nest with K eggs will be built, according to the similarities/ differences of the eggs. Some random mixing can be used to generate diversity.

Since the crane's high-quality position should ensure that $q_{picking}$ and $q_{placing}$ are within the working range of crane, and the position cannot be located in obstacle region. As shown in Fig. 1, R_{max} and R_{min} are maximum and minimum working range of crane respectively.

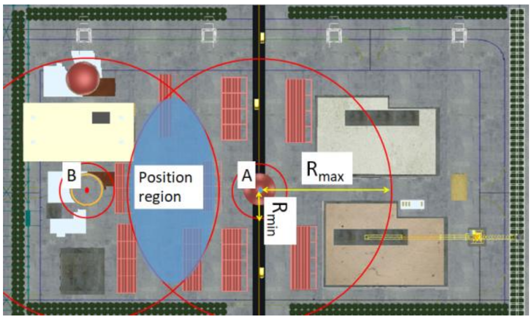


Fig. 1. Crane's Position Region

For finding crane's best position, we modify the first and add fourth rules to incorporate crane work environment:

- Each cuckoo lays K eggs at a time, and dumps them in a randomly chosen nest in crane's position region.
- If the host's nest is within the obstacle region, the nest can be considered unsuitable for the survival of the cuckoo, and the nest is set to be an inferior nest.

The search of crane's position can greatly affect the quality of the action sequence, so in this paper, the quality of the action sequence is used as objective function of CS for sorting crane's position. The calculation of the action sequence is based on A*-circle algorithm. The optimization goal of the action sequence is as follow: the path length is the shortest and the number of action change is the minimum. When a crane’s position (nest) is found, the corresponding action sequence is calculated by A*-circle. After that, the length of the path (PathL) and the number of action change (Action_N) of action sequence are calculated and saved. Finally, positions are sorted and the highest quality position is saved. The algorithm generates a group of new positions $X^{(t+1)}$ after iteration by using equation (2). For example, for Cuckoo i, a new position is generated by applying Levy Flight.

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus \mathrm{levy}(\gamma) \tag{2}$$

Where, a>0 is step length, usually a=1. Essentially, Equation (2) is the stochastic equation for a random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the equation (2)) and the transition probability (the second term).

To summarize, the pseudo code of the modified Multi-objective Cuckoo Search algorithm is as follow:

```
Initialize Obstacle_area and Position_area
Generate an initial population of n host nests $x_i$ at Position_area
While (t<MaxGeneration) or (stop criterion)
    Get a cuckoo (say i) randomly by Levy Flight
    Plan an action sequence by A*-circle, and calculate the value of PathL and Action_N.
    Choose a nest among n (say j) randomly
    If j in obstacle_area
        PathL (j) =INF
        Action_N (j) =INF
    End
    If new solutions of nest j dominate those of nest i
        Replace nest i by the new solution set of nest j
    End
    Abandon a fraction ($p_a$) of worse nests
    And generate new nest by Levy Flight
    Sort and find the current Pareto optimal solutions
    End
Postprocess results and visualization
```

IV. Calculating Crane’s Action Sequence

After Multi-objective CS finding a crane’s position, A*-circle is used to search a group of corresponding action sequence. The A* algorithm is a grid-based heuristic search algorithm. The cost f(n) is usually used to determine which node to search for next.

$$f(\mathrm{n}) = \mathrm{g}(\mathrm{n}) + h(\mathrm{n}) \tag{3}$$

Where, n is current node. g(n) is the path length cost from the start node to n. h(n) is the estimated cost from n to the target node.

In the process of crane path planning, the smoothness of the path must be guaranteed, so the direction change cost dir(n) is added to the function. In addition, it is also necessary to consider the cost lift(n) generated by the lifting movement of crane’s hook in space, so the improved cost f(n) is shown in Equation (4):

$$f(\mathrm{n}) = \mathrm{g}(\mathrm{n}) + h(\mathrm{n}) + \mathrm{dir}(\mathrm{n}) + \mathrm{lift}(\mathrm{n}) \tag{4}$$

In order to avoid collision, the algorithm expands the size of obstacles according to the three-dimensional size of the object. After that, the meshing of the work area is performed. The original A* algorithm uses a rectangular grid. However the path found by A* is difficult to be converted into the action sequence of crane. Because of the special action characteristics of crane, when the crane is at the position, the actions that can be performed include slewing, lifting and Luffing. When the crane's boom is slewing, the trajectory of the object is an arc rather than line. So a fan-shaped grid is used to mesh the work area. First, a set of concentric circles centering on the crane position are drawn, and then a set of lines along the radial direction of circles are drawn too, so that circles and lines are combined into a fan mesh. Thus, the radial line of the grid is defined as one step of crane luffing, and the arc line is defined as one step of crane slewing. Through this fan-shaped mesh division, the path can be automatically combined with the crane action sequence to achieve mutual transformation. We name it as A*-circle algorithm. The A*-circle algorithm is shown in Fig. 2.

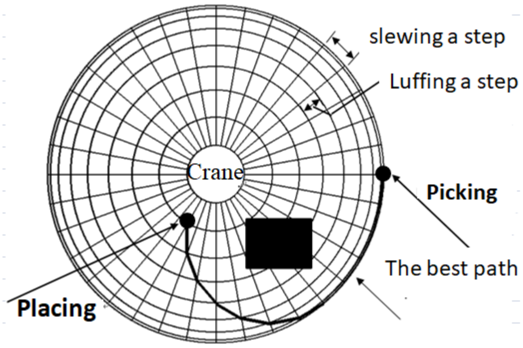
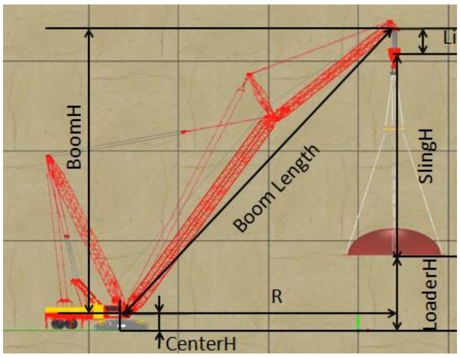


Fig. 2. The grid of A*-circle algorithm

In the A*-circle algorithm, the concept of Pseudo-3D is applied to convert the height of the obstacle into the cost of the path. And the height of the object is determined by the height of the apex of boom and the position of the hook. Therefore, when judging whether the object can cross the obstacle, we have to calculate the maximum height (ObstacleHMax) at which crane can lift the object. This height is determined by the following variables: height of the apex of boom (BoomH), the length of suspension tool (SlingH) and the height of the center of boom slewing (CenterH), as shown in Fig. 3.



LiftH—The length of sling; LoaderH—The height of object; BoomLength—The length of boom; R—Radius of slewing;

Fig. 3. Crane's Parameters

ObstacleHMax is calculated using the following Equation:

$$\mathrm{ObstacleHMax} = \mathrm{BoomH} + \mathrm{CenterH} - \mathrm{SlingH} \quad (5)$$

And BoomH can be expressed as follows:

$$\mathrm{BoomH(i)}^2 = \mathrm{BoomLength}^2 - \mathrm{R(i)}^2 \quad (6)$$

Where, R(i) is the radius of slewing when the object is at i row.

ObstacleHMax of each row is saved in a one-dimensional array ObstacleMaxArray. In the process of searching for the object's path, it is necessary to compare the obstacle height (ObstacleH) with ObstacleMaxArray(i) at i row. There are two outcomes:

(1) ObstacleH> ObstacleMaxArray(i), so lift(i)=+∞;

(2)ObstacleH<=ObstacleMaxArray(i) and ObstacleH> LoaderH, lift(i) is calculated using the following equation:

$$\mathrm{lift(i)} = \alpha(\mathrm{ObstacleH} - \mathrm{LoaderH}) + \mathrm{difBoomH} \quad (7)$$

Where, α is coefficient. When the object is moved toward the center of the circle, difBoomH is $\mathrm{BoomH(i)} - \mathrm{BoomH(i-1)}$, or else difBoomH is $\mathrm{BoomH(i)} - \mathrm{BoomH(i+1)}$.

The flow chart of the whole algorithm is shown in Fig. 4.

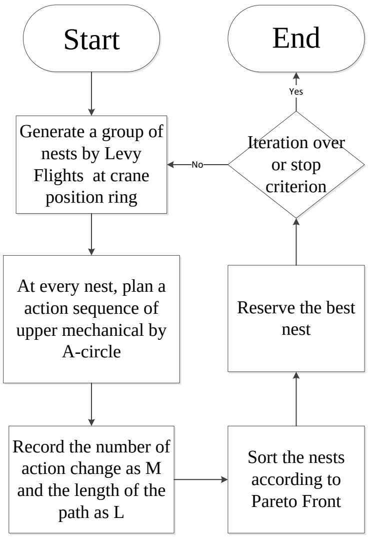


Fig. 4. Flow chart represented for the proposed algorithm

V. Case Study

The effectiveness of the crane's action sequence planning algorithm is verified by a simulation experiment.

The crane's work environment is shown in Fig. 1. The weight of the object is 20t and its radius and height is 25.8m and 7.5m respectively. The height of picking point A and placing point B is 0m and 13m respectively. The Information of crane is show in Table I.

Table I

Crane's Information

Brand	DEMAG
Model	CC8800-1
Boom Type	Main Boom+ Fixed Jib+ Super Lifting
Main Boom Length	72m
Jib Length	48m
Working Range	28m~70m

In the simulation, the number of cuckoos is 20. The longitudinal lines of the grid present that the boom luffs 500mm, and the arc line presents that the boom pivots 1° .

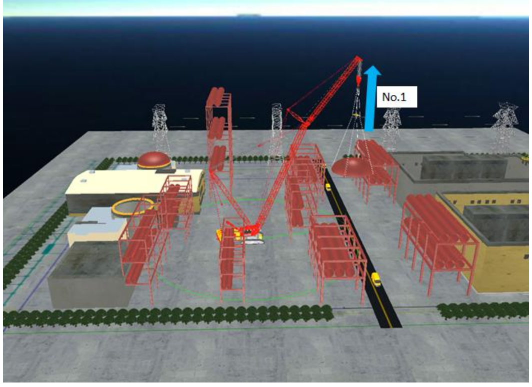
After iterating 100 times, the crane's action sequence is shown in Table II.

Table II

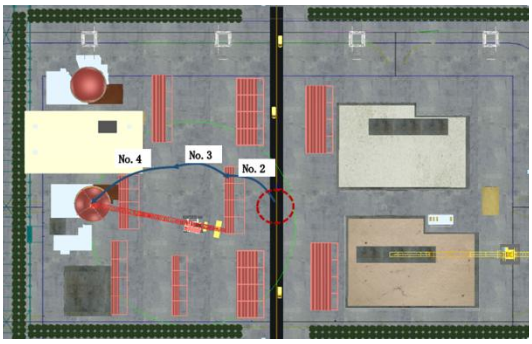
Crane's action sequence

Sequence Action	No.1	No.2	No.3	No.4
Luffing	—	—	—	—
Slewing	—	—	+51 °	—
Lifting	+39.5m	—	—	—
Slewing + Luffing	—	+40 °, -20m	—	+64 °, +31.5m

We can calculate the length of path which is 189.3m, by action sequence and grid's size. The coordinates of the position is (-63028.7，-26032.3). The total duration: 319.8s, in the position region shown in Fig. I, 269 positions are found, and the average time to search for action sequence at each position is 1.19 s. The path of the crane performing the action sequence is shown in Fig. 5.



(a) Lifting 39.5m



(b) Top View

Fig. 5. Crane's Action Sequence

VI Conclusions

In this paper, a combined algorithm based on improved Multi-objective CS and A* algorithm is proposed to solve the problem of crane action sequence planning. The crane action sequence planning is divided into two parts: position planning and action sequence planning. The Multi-objective CS is used to search crane position, and the action sequence under the position is searched according to A*-circle algorithm. Then, the quality of the action sequence is used to sort position, and the best action sequence is obtained after iterations. Simulation results demonstrate the effectiveness of the algorithm.

VII. References

1. Lin, Y., et al. (2012). Path Planning for Crawler Crane Using RRT*, Berlin, Heidelberg, Springer Berlin Heidelberg.
2. Lin, Y., et al. (2016). "Lift Path Planning without Prior Picking/Placing Configurations: Using Crane Location Areas." Journal of Computing in Civil Engineering 30(1): 04014109.
3. X. Wang et al., "Path Planning for Crane Lifting Based on Bi-Directional RRT", Advanced Materials Research, Vols. 446-449, pp. 3820-3823, 2012
4. Y.C. Chang, W.H. Hung, S.C. Kang, A fast path planning method for single anddual crane erections, Autom. Constr. 22 (2012) 468–480,
5. Jianqi An, Min Wu, Jinhua She, Takao Terano,Re-optimization strategy for truck crane lift-path planning,Automation in Construction,Volume 90,2018,Pages 146-155,ISSN 0926-5805.
6. Wang, X., et al. (2011). Collision-free path planning for mobile cranes based on ant colony algorithm. Materials, Mechatronics and Automation, Trans Tech Publications Ltd.
7. Lei, Z., et al. (2013). "A methodology for mobile crane lift path checking in heavy industrial projects." Automation in Construction 31: 41-53.
8. Lei, Z., et al. (2015). "Algorithm for mobile crane walking path planning in congested industrial plants." Journal of Construction Engineering and Management 141(2).
9. Yang, X.-S. And S. Deb (2013). "Multiobjective cuckoo search for design optimization." Computers and Operations Research 40(6): 1616-1624.