Mathematical Biosciences and Engineering

MBE, 20(3): 5993–6015.
DOI: 10.3934/mbe.2023259
Received: 02 November 2022
Revised: 18 December 2022
Accepted: 28 December 2022
Published: 18 January 2023

http://www.aimspress.com/journal/MBE

Research article

Service scheduling optimization for multiple tower cranes considering the interval time of the cross-tasks

Jing Yin*, Jiahao Li, Yifan Fang and Ahui Yang

Department of Industrial Engineering, School of Mechanical-Electronic and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

* **Correspondence:** Email: yinjing@bucea.edu.cn; Tel: +8613811835463.

Abstract: The key issues that have always affected the production yield of the construction industry are delays and cost overruns, especially when dealing with large-scale projects and super-high buildings in which multiple tower cranes with overlapping areas are often deployed because of urgent due date and limited space. The service scheduling of tower cranes, which act as the crucial site equipment for lifting and transporting materials, is one of the main problems not only related to the construction progress and project cost but also affecting equipment health, and it may bring security risks. The current work presents a multi-objective optimization model for a multiple tower cranes service scheduling problem (MCSSP) with overlapping areas while achieving maximum interval time of cross-tasks and minimum makespan. For the solving procedure, NSGA-II is employed with double-layer chromosome coding and simultaneous coevolutionary strategy design, which can obtain a satisfactory solution through effectively allocating tasks within overlapping areas to each crane and then prioritizing all the assigned tasks. The makespan was minimized, and stable operation of tower cranes without collision was achieved by maximizing the cross-tasks interval time. A case study of the megaproject Daxing International Airport in China has been conducted to evaluate the proposed model and algorithm. The computational results illustrated the Pareto front and its non-dominant relationship. The Pareto optimal solution outperforms the results of the single objective classical genetic algorithm in terms of overall performance of makespan and interval time of cross-tasks. It also can be seen that significant improvement in the time interval of cross-tasks can be achieved at the cost of a tiny increase in makespan, which means effective avoidance of the tower cranes entering the overlapping area at the same time. This can help eliminate collision, interference and frequent start-up and braking of tower cranes, leading to safe, stable and efficient operation on the construction site.

Keywords: multiple tower cranes; service scheduling; multi-objective optimization; NSGA-II; overlapping areas

1. Introduction

All the time, the key issues that have been affecting the production yield of the construction industry are delays and cost overruns, especially when dealing with large-scale projects and super-high buildings with high resource requirements and construction difficulties [1], such as the Winter Olympic Games venues, International Airport and emergency building, in which multiple tower cranes (TC) with overlapping area are often deployed because of urgent due date and limited space. As the key special equipment for construction material lifting and transportation, the operation efficiency of tower cranes not only directly affects the construction progress and project cost but also affects equipment health and brings security risks [2]. In recent years, with the promotion of construction industrialization represented by prefabricated buildings, it comes with an even higher challenge for operation management of tower cranes in terms of incremental costs and safety issues [3]. Different from ordinary materials, prefabricated components have strict time windows, and incremental costs come into being if it cannot be hoisted on time [4]. Studies have shown that the main reasons are the shortage of scheduling and planning and the confusion of onsite management [5]. Meanwhile, with the rapid development of intelligent construction and unmanned tower cranes, only relying on the operator's experience to manage tower crane operations can no longer meet the practical need [2,6]. Therefore, it is particularly important to form efficient scheduling models and corresponding algorithms.

This research mainly focused on service scheduling of multiple tower cranes with overlapping areas, which is composed of allocation of tasks within overlapping areas to each crane and prioritization of all the assigned tasks of each crane. Because of the dynamic uncertainties of the construction site, the operating time of the tower crane may change with the climate and the condition of staff and equipment. Then, the tasks cannot be completed on time as planned, and hence collision may occur. In this case, emergency braking achieved by a limit switch and monitoring software will interrupt the smooth operation of tower cranes and bring security risks on site. Therefore, on the basis of collision avoidance, the time interval between tower cranes entering the overlapping area is taken as one of the objective functions. A multi-objective optimization model for multiple tower cranes service scheduling was proposed. To achieve maximum interval time of cross-tasks and minimum makespan, a solving algorithm based on NSGA-II is designed with double-layer chromosome and simultaneous coevolutionary strategy, which can obtain a satisfactory solution to improve the stability and effectiveness of TC operation.

The remainder of the paper is organized as follows. Section 2 summarizes the existing research related to tower cranes. With the problem description in Section 3, mainly defining the hook travel time calculation and identifying the conflicting tasks, a multi-objective optimization model is developed to not only minimize the construction period but also reduce safety risks. By double-layer chromosome coding and simultaneous coevolutionary strategy designing, the solving algorithm is implemented in Section 4. In Section 5, different scale cases are generated with the engineering data of Beijing Daxing International Airport, and the Pareto optimal solutions are obtained by the proposed model and algorithm. Finally, Section 6 concludes the paper.

2. Literature review

So far, the research related to tower cranes mainly includes three aspects: 1) layout planning, which involves production selection and location optimization; 2) monitoring software and conflict detection system; and 3) service scheduling, which means operation optimization.

In the preplanning, it is very important to select the appropriate equipment and installation location for construction activities. TC type selection is a complex decision process involving multiple decision criteria and multi-criteria decision methods such as the fuzzy logic method [7], hierarchical analysis [8] and neural network methods in combination with project technical parameters [9]. For the location and loading point optimization, different mathematical modeling and computer simulations have been adopted, such as Monte Carlo-based simulation of multiple crane operations [10], genetic algorithm-based optimization for minimizing hook travel time [11], discrete-event [12] and agent-based simulation [13] considering potential conflicts, mixed integer programming for minimizing operating time and cost [10,14], combining genetic algorithm (GA) and building information modeling (BIM) to plan the optimal location of TCs and facilities [15]. The latest research has proposed a two-step framework for tower crane layout planning (TCLP) based on a fuzzy integration technique to solve both TC selection and layout problems simultaneously [16].

After the selection and location of TCs, TC path monitoring and conflict detection derived from construction safety attracted extensive international attention. Frameworks, algorithms and methods for path planning have been proposed for path planning and synchronized operations of multiple TCs to avoid collisions, such as the incremental coordination method and probabilistic roadmap method [6,17–18]. Visualization tools in conjunction with the specific problem have been applied for operation modeling [19] and operator visibility modeling [20]. To improve operational safety, robots for lifting operations have also been explored [21].

In 2014, Zavichi first proposed the Tower Crane Service Sequencing Problem (TCSSP) and transferred it into the Traveling Salesman Problem (TSP) while achieving shortest operation time [22,23]. Boysen et al. [24] deeply analyzed and classified the tower crane conflict problem and proposed a classification scheme for the crane scheduling problem with tower crane interference. Huang et al. [25] proposed an optimization model for crane installation location, hook movement sequence and tower crane service plan, and they converted it into binary mixed integer linear programming (BMILP). Kim et al. [26] proposed a framework for tower crane scheduling based on a structural matrix (DSM) sequence feature. Ahmed Younes et al. [27] proposed that tower crane layout should be combined with tower crane operation efficiency and designed an agent simulation model to quantitatively evaluate the impact of conflict on tower crane operation time and cost. Hattab et al. [28] explored the optimization problem of tower crane operation with overlapping areas in high-rise building construction and provided a reliable and unconstrained daily operation plan of tower cranes based on a predictive plan. During the three-dimensional movement of the tower crane, the collision is checked, and the task allocation of the overlapping area is optimized in real time, which effectively improves the working efficiency of the crane. Tarhini et al. [29] established an integer linear programming model based on the two-TCs scheduling problem and introduced a heuristic method to improve the model's processability, solution quality and calculation time. Huang et al. [30] transformed the multi-TCs service scheduling problem with overlapping areas into a mixed integer programming model and optimized it based on the following three aspects: request task allocation in overlapping areas, supply location selection of requested tasks and arrangement of operation sequence of each TC.

Unlike tower crane monitoring, which uses automatic sensing devices to achieve collision avoidance, service scheduling of tower cranes is to optimize the operation process at the task level. Through reasonable allocation and sorting of material request tasks, it can realize the stability and effectiveness of tower crane operation. Related achievements have been emerging in the past five years, but there have been few published studies on the service scheduling problem of multiple cranes with overlapping areas (MCSSP). Therefore, to meet the practical need for enhancing the efficiency and safety of construction operations, it is particularly necessary to establish efficient scheduling models and specific algorithms and to conduct useful theoretical exploration and engineering applications.

3. Mathematical model

3.1. Problem description

Given a set of lifting requirements and a tower crane site layout, scheduling optimization of TC operation can be achieved by determining the movement between tower crane locations, material requirements and supply locations. Assume that two tower cranes have overlapping areas (see Figure 1) and that cranes 1 and 2 are assigned tasks NCR^1 and NCR^2, respectively, from the non-overlapping area and have a collective task CR in the overlapping area. The decision maker needs to first assign the overlapping tasks CR to cranes 1 and 2 and then classify the tasks for each crane in a rational way. If the task set CR is divided into CR^1 and CR^2 and then assigned to cranes 1 and 2, respectively, the task sorting combinations for the two cranes are therefore $(NCR^1+CR^1)!$ and $(NCR^2+CR^2)!$ In order to minimize the makespan and maximize the cross-tasks interval time for the tower cranes, the operator needs to find the best task scheduling in all these combinations.

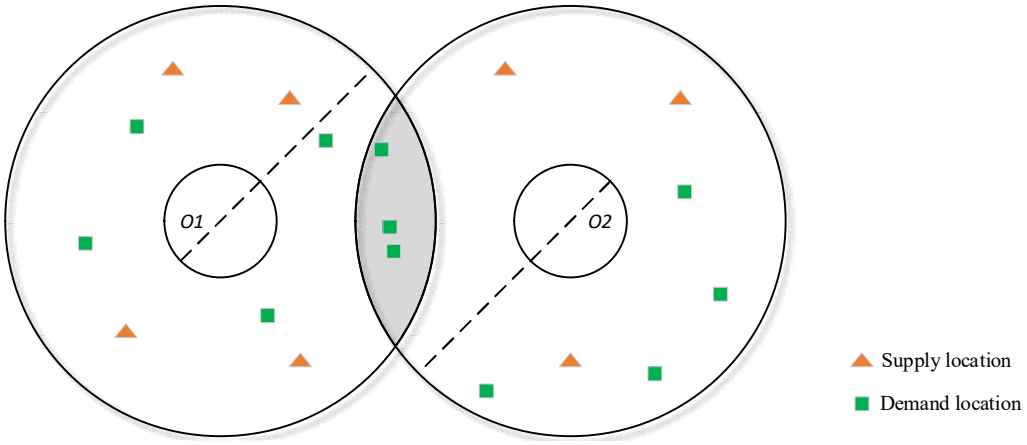


Figure 1. Two tower cranes with overlapping areas.

The simplifications of the optimization model in this research are as follows:
1) There is no priority constraint between requesting tasks.
2) Only one type of material is allowed to be lifted at one time for each tower crane.
3) The default initial location of each tower crane for each task is at the demand position of the previous task.

3.2. Travel time calculation formulation

The calculation of hook travel time is the basis and premise to solve the MCSSP. In this paper, the Cartesian analysis model established by Zhang et al. is used to analyze the tower crane hook travel time. Based on the position of the tower crane, the coordinate system is established, as the tower crane position is $O=(x_o,y_o)$, the material supply is $A=(x_A,y_A)$, and the material demand is $B=(x_B,y_B)$, as seen in Figure 2. Table 1 summarizes the corresponding parameters appearing in Eqs (1)–(5).

Table 1. Travel time calculation related symbols.

Symbol	Expression
i	Material supply location i;
j	Material demand location j;
x	Position along the x-axis;
y	Position along the y-axis;
z	Position along the z-axis;
V_r	Radial velocity of the hook;
V_a	Angular velocity of the hook;
V_v	Vertical velocity of the hook;
$T_r^{(i,j)}$	Radial hook travel time from i to j;
$T_a^{(i,j)}$	Tangent hook travel time from i to j;
$T_h^{(i,j)}$	Horizontal hook travel time from i to j;
$T_v^{(i,j)}$	Vertical hook travel time from i to j;
$T_T^{(i,j)}$	The total hook travel time from i to j;
λ	Degree of coordination of hook movement in radial and tangential directions in horizontal plane ranging between 0.0 and 1.0 (Where 0.0 stands for full simultaneous movement and 1.0 for full consecutive movement);
η	Degree of coordination of hook movement in vertical and horizontal planes ranging between 0.0 and 1.0 (where 0.0 stands for full simultaneous movement and 1.0 for a full consecutive moment);
μ	Degree of obstacle blocking hook movement from an initial hook location to the material supply location i when TC sets up at location k ranging between 1.0 and ∞ (where 1.0 represents normal operation without obstacle, and ∞ represents difficult operation with the most number of obstacles);

3.2.1. Radial travel time and angular travel time

The radial travel time ($T_r^{(i,j)}$) of the hook is expressed by Eq (1), and the tangent travel time ($T_a^{(i,j)}$) is expressed by Eq (2).

$$T_r^{(i,j)} = \frac{\left|\sqrt{(x_B^2+y_B^2)}-\sqrt{(x_A^2+y_A^2)}\right|}{V_r} \tag{1}$$

$$T_a{}^{(i,j)} = \frac{|tan^{(-1)}(\frac{y_B}{x_B}) - tan^{(-1)}(\frac{y_A}{x_A})|}{V_a} \tag{2}$$

3.2.2. Horizontal travel time

According to the radial and angular travel time synthesis, hook travel time in the horizontal direction, consider the introduction of the λ parameter: the reaction operator's ability to move the hook simultaneously in the radial and angular directions, i.e., the degree of overlap between the radial movement and the tangential movement. λ is a continuous positive number, taking values in the range $0 \leq \lambda \leq 1$. The smaller the value of λ is, the higher the simultaneity of the reaction to move the hook in the radial and tangential directions. The hook horizontal travel time can be calculated using Eq (3).

$$T_h{}^{(i,j)} = max\left\{T_r{}^{(i,j)}, T_a{}^{(i,j)}\right\} + \lambda \cdot min\{T_r{}^{(i,j)}, T_a{}^{(i,j)}\} \tag{3}$$

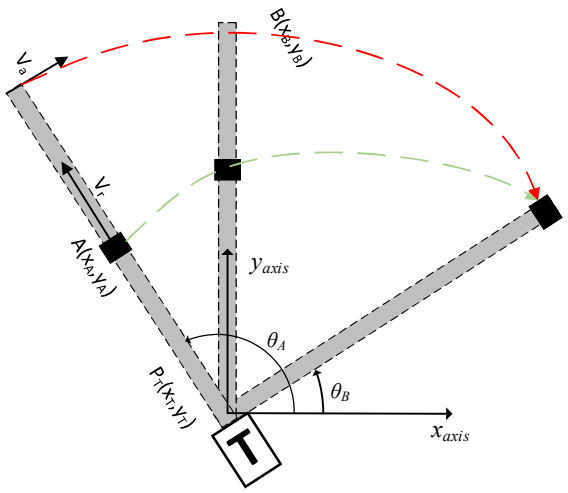


Figure 2. Coordinate system of tower crane's horizontal motion.

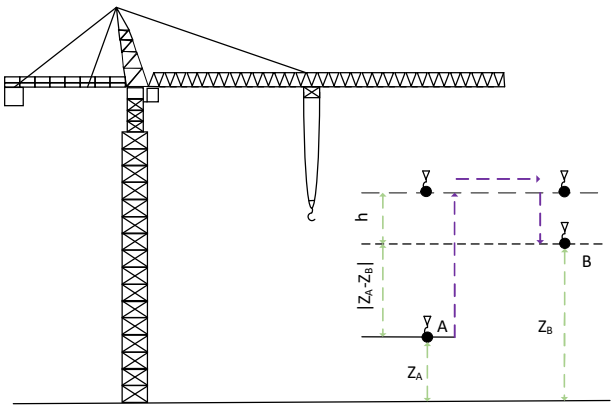


Figure 3. Coordinate system of tower crane's vertical motion.

3.2.3. Vertical travel time

As shown in Figure 3, the vertical travel time ($T_v^{(i,j)}$) of the hook is expressed by Eq (4).

$$T_v^{(i,j)} = \frac{|Z_B - Z_A| + 2h}{V_v} \tag{4}$$

where Z_A, Z_B are the coordinates of point A and point B in Z (height) direction, respectively, and h is the minimum value of lifting height.

3.2.4. Total travel time

The total travel time ($T_T^{(i,j)}$) of the hook is expressed by Eq (5).

$$T_T^{(i,j)} = \mu \cdot \left(max\{T_h^{(i,j)}, T_v^{(i,j)}\} + \eta \cdot min\{T_h^{(i,j)}, T_v^{(i,j)}\} \right) \tag{5}$$

where η ($0 \le \eta \le 1$) and λ, respectively, represent the horizontal and vertical synergistic movement ability of the operator. The parameter μ ($1 \le \mu \le \infty$) describes uncontrollable conditions onsite. The larger the value of μ is, the more adverse the construction environment.

3.3. Time calculation of task entry/exit overlapping area

Taking two tower cranes as an example, the judgment basis of two tower cranes entering the overlapping area is discussed, and the relevant description and explanation are carried out. Table 2 summarizes the corresponding parameters appearing in Eqs (6)–(15).

Table 2. Entry and exit time calculation for task overlapping areas related symbols.

Symbol	Expression
α	The azimuth of O;
β	Half of the angle between two intersection points and O;
$O_1(x_{o1},y_{o1})$	Tower 1 coordinates;
$O_2(x_{o2},y_{o2})$	Tower 2 coordinates;
O_1O_2	The distance between the two tower cranes;
FL_1	Tower 1 forearm length;
FL_2	Tower 2 forearm length;
$[\theta_1,\theta_2]$	Angle range of overlapping area;
γ	The operation angle of the tower crane forearm;

The plane coordinate system of the overlapping area of the two tower cranes is shown in Figure 4. According to the distance calculation Eq (6), the distance between the two towers is O_1O_2.

$$O_1O_2 = \sqrt{(x_{o1}+x_{o2})^2 + (y_{o1}-y_{o2})^2} \tag{6}$$

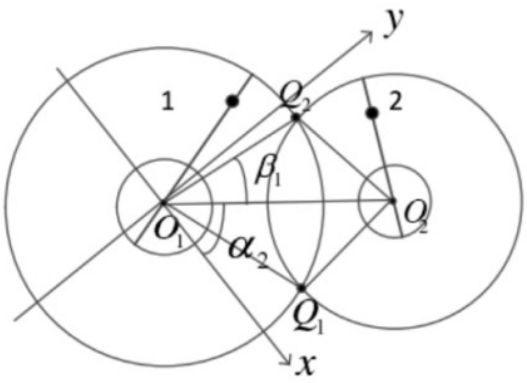


Figure 4. Diagram of tower position.

According to Eq (7) of the cosine theorem, β_1 can be calculated as follows:

$$\beta_1=arccos(\frac{FL_1^2+(O_1O_2)^2-FL_2^2}{2\times FL_1\times O_1O_2}) \tag{7}$$

α_2 is the azimuth angle of O_2, $Q_1(x_{q1}, y_{q1})$ and $Q_2(x_{q2}, y_{q2})$ are the intersections of two circles. The calculation is based on the above β_1 and α_2. Then, the coordinate values are calculated as Eqs (8)–(11):

$$x_{q1}=FL_1\times cos(\alpha_2-\beta_1) \tag{8}$$

$$y_{q1}=FL_1\times sin(\alpha_2-\beta_1) \tag{9}$$

$$x_{q2}=FL_1\times cos(\alpha_2+\beta_1) \tag{10}$$

$$y_{q2}=FL_1\times sin(\alpha_2+\beta_1) \tag{11}$$

The range of tower crane 1 entering the overlapping area is $[\theta_1^1, \theta_2^1]$, and then the values of θ_1^1 and θ_2^1 are

$$\theta_1^1=\alpha_2-\beta_1 \tag{12}$$

$$\theta_2^1=\alpha_2-\beta_1 \tag{13}$$

The range of tower crane 2 entering the overlapping area is $[\theta_1^2, \theta_2^2]$, and the specific values are calculated from Eq (14) to (15):

$$\theta_1^2=max\{arctan2\left(\frac{y_{q1}-y_{o2}}{x_{q1}-x_{o2}}\right),\ arctan2\left(\frac{y_{q2}-y_{o2}}{x_{q2}-x_{o2}}\right)\} \tag{14}$$

$$\theta_2^2=min\{arctan2\left(\frac{y_{q1}-y_{o2}}{x_{q1}-x_{o2}}\right),\ arctan2\left(\frac{y_{q2}-y_{o2}}{x_{q2}-x_{o2}}\right)\} \tag{15}$$

γ_1, γ_2 are, respectively, the operation angles of the TC 1 forearm and TC 2 forearm.

1) If $\gamma_1 \notin [\theta_1^1, \theta_2^1]$ or $\gamma_2 \notin [\theta_1^2, \theta_2^2]$, there will be no collision during the work process. Else, there is conflict risk.

2) Conflict identification. If conflict risk exists, when scheduling solution generated by chromosome decoding, the time parameters of all cross-tasks are calculated according to Eqs (1)–(5), including CT_{ibegin}, CT_{iend} and $CT_{iterval}$.

3) Conflict resolution. If there is $CT_{iterval} < 0$, postpone the CT_{ibegin} of the task late into the overlapping area; make it equal to CT_{iend} of the task early in the overlapping area.

3.4. Cross-tasks interval time calculation

In this paper, considering the makespan f_1 and the cross-tasks interval time f_2, a multi-objective optimization model is established. Table 3 summarizes the corresponding parameters appearing in Eqs (16)–(23).

Table 3. Objective function calculation related symbols.

Symbol	Expression
m	Number of tower crane;
M	Total number of tower cranes;
N	Extremely positive numbers;
CT	Cross-task;
C_r	Time of finish for task r;
H_r	Collection of loadable tower codes for task r;
K_m	The task k of tower m;
CT_{ibegin}	Cross-task i start time;
CT_{iend}	Cross-task i end time;
$CT_{iterval}$	Time interval between cross-tasks;
T_{total}	Maximum makespan;
$T_{loading}$	Material loading time;
$T_{unloading}$	Material unloading time;
R_m	Tower m has a total of tasks R;
P_{mr}	Tower m processing time for task r;
D_{km}	The task k of tower m where materials are demanded;
S_{km}	The task k of tower m where materials are supplied;
S_{mlr}	Preparation time when order r is assigned to tower
P_T	The position of the tower crane hook;
R	Total number of lifting tasks, $\{R_m \mid m = 1,2 \ldots M\}$;
r	Lifting tasks r;
T_m	Operation time of tower crane m;
T_T	Total time of crane hook travel for the task being hoisted;
y_{mrk}	The decision variable, $y_{mrk} = 1$ means that task r is scheduled for operation on tower m at position k, otherwise, r is not scheduled for operation on tower m;

The calculation of cross-tasks interval time can be divided into two cases:

1) When there is no cross-task assigned to tower crane 1 and tower crane 2 or only one tower crane has cross-task, there will be no cross operation when the two towers operate.

2) There are m conflicting jobs CT_i on tower crane 1, the beginning time of the task is CT_{ibegin}, and the end time is CT_{iend}. There are n conflicting jobs CT_j on tower crane 2, the beginning time of the task is CT_{ibegin}, and the end time is CT_{iend}. The minimum interval time is calculated by all possible conflicting jobs on tower crane 1 and possible conflicting jobs on tower crane 2. The specific calculation process is the following:

(i) If the end time of a cross-task $CT_i\,(1 \le i \le m)$ on tower 1 is less than the begin time of a possible cross-task $CT_j\,(1 \le j \le n)$ on tower 2, then $CT_{interval\_ji} = CT_{jbegin} - CT_{iend}$;

(ii) If the end time of a possible cross-task $CT_i(1 \le i \le m)$ on tower 1 is longer than the beginning time of a possible cross-task $CT_j\,(1 \le j \le n)$ on tower 2, then $CT_{interval\_ji} = CT_{ibegin} - CT_{jend}$.

In summary, the objective function and constraints are obtained as shown in Eqs (16)–(23).

$$f_1 = Min(T_{total}) = Min(max\{T_1, T_2, T_3 \ldots T_M\}) \tag{16}$$

$$T_m = T_{T(Om,S_{1m})} + T_{\text{loading}} + T_{T(S_{1m},D_{1m})} + T_{\text{unloading}},\ R_m = 1$$

$$T_m = T_{T(Om,S_{1m})} + T_{loading} + T_{T(S_{1m},D_{1m})} + T_{unloading} + \sum_{k=2}^{n_m}(T_{T(D_{k-1m},S_{km})} + T_{loading} + T_{T(S_{km},D_{km})} + T_{unloading}),\ R_m \neq 1 \tag{17}$$

$$f_2 = Max(CT_{interval}) = Max(\sum_{i=1}^{m} min\ (T_{i\text{interval}\_i\delta}, T_{i\text{interval}\_(\delta+1)i} + \sum_{j=1}^{n} min\ (T_{i\text{interval}\_j\varepsilon}, T_{i\text{interval}\_(\varepsilon+1)i}\ )) \tag{18}$$

S.T.

$$\sum_{r=1}^{R} y_{mr(k+1)} \le N \sum_{r=1}^{R} y_{mrk},\ m = 1, 2, \ldots, M,\ k = 1, 2, \ldots, R\text{-}1 \tag{19}$$

$$C_r - P_{mr} - S_{mlr} - C_l + N(2 - y_{mr(k+1)} - y_{mlk}) \ge 0 \quad m \ne l,\ r, l = 1,2,\ldots,R,\ \forall m \in H_r \cap H_l,\ k = 1, 2, \ldots, n\text{-}1 \tag{20}$$

$$C_r - \sum_{m \in H_r} \sum_{k=1}^{r} y_{mrk} P_{mr} \ge 0,\ m = 1, 2, \ldots, M \tag{21}$$

$$y_{mrk} = 0,\ r,k = 1,2,\ldots,R,\ \forall m \in \{1,2,\ldots M\} \setminus H_r \tag{22}$$

$$y_{mrk} \in \{0,1\},\ r, k = 1,2,\ldots,R,\ \forall m \in H_r \tag{23}$$

The objective functions (16) and (17) represents the minimum makespan, and the objective function (18) represents the maximum cross-tasks interval time, where δ is the task on tower 2 that enters the overlapping area before task i, and ε is the task on tower 1 that enters the overlapping area before task j. Constraint (19) means that each tower is allowed to schedule at most one order.

Constraint (19) represents the continuity of machine position numbers, and if position k does not schedule orders, subsequent locations do not allow orders to be scheduled. Constraint (20) requires that a tower crane is allowed to work on only one task at a time, that there is a sequence between tasks and that the transfer time between tower cranes depends on the limitation of the tower crane's working range and does not allow the task to be assigned to an unfinished tower crane. Constraint (21) limits the completion time of the task. Constraint (22) limits the range of available towers, and the task is not allowed to be arranged on the tower crane that cannot be hoisted. Constraint (23) is for variable assignment.

4. Solution algorithm

In this paper, a multi-objective optimization algorithm based on NSGA-II (TCSC-NSGA-II) is designed for solving the model proposed above, which is developed on Matlab, and the computational process is shown as Figure 5.

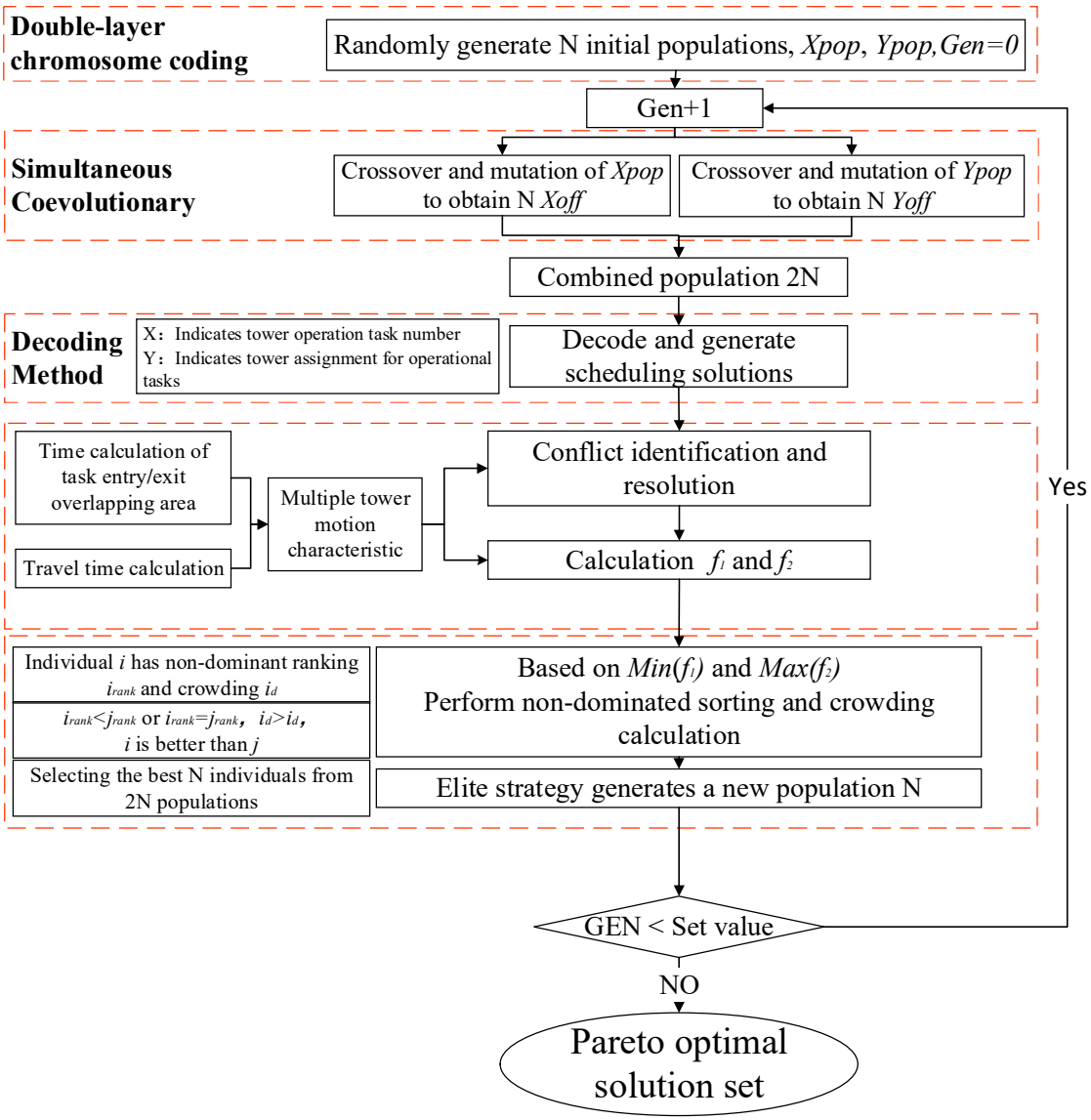


Figure 5. Multi-objective optimization based on TCSC-NSGA-II.

The key steps of double-layer chromosome coding, genetic operator design and TCSC-NSGA-II algorithm design are explained in the following sections.

4.1. Overview of the NSGA-II algorithm

The existence of multiple objectives in a problem will, in principle, result in a set of optimal solutions (often called Pareto-optimal solutions) rather than one optimal solution. In the absence of any further information, it cannot be said that one of the Pareto-optimal solutions is better than another. This requires the user to find as many Pareto-optimal solutions as possible. Classical optimization methods (a priori methods) suggest transforming multi-objective optimization problems into single-objective optimization problems, emphasizing one particular Pareto-optimal solution at a time. When this approach is used to find multiple solutions, it must be applied multiple times.

The Non-Dominated Sorting Genetic Algorithm (NSGA) is an a posteriori intelligent algorithm derived in the context of multi-objective problems. The main difference between NSGA and GA is in the operation of selection, of which the former stratifies individuals according to the value of the objective function, dominated and non-dominated, and then selects according to the stratification by an iterative process of looping to obtain a relatively good solution for each objective function value [31]. Based on this, Deb improved the NSGA by adding an elite strategy for selection after stratification, namely NSGA-II. After a large number of practical applications, NSGA-II has been proved to outperform NSGA in three advantages: the introduction of the crowding degree comparison operator, the non-dominant order crowding degree comparison operator and the elite strategy. Using the concept of non-dominant order Pareto optimal solutions to classify individuals in the population, the method retains individuals with better characteristics in the population and is able to find much better spread of solutions and better convergence near the true Pareto-optimal front [32]. NSGA-II is currently the most popular multi-objective problem solving method for production scheduling, logistics planning, process parameter optimization, etc. [33–35].

4.2. Double-layer chromosome coding

As the MCSS problem mainly deals with two problems, the overlapping area task assignment and task sequencing of each crane, double-layer chromosomes (X and Y) are employed in this research. Chromosome $\boldsymbol{X} = \{X_1, X_2, \ldots, X_r, \ldots, X_R\}$ represents the execution order of all tasks, where the r indicates the rth scheduled task. Chromosome $\boldsymbol{Y} = \{Y_1, Y_2, \ldots, Y_s, \ldots, Y_S\}$ represents the corresponding cranes related to assigned tasks X, and in order to limit the search space to the range of available tower cranes, let crane Y_s belong to set H_s, where the s indicates the number of the task which is the value of X_r.

The following is an example to illustrate this coding strategy. Consider an example of 5 tasks and a total of 3 cranes, where $H_1 = \{1, 2\}$, $H_2 = \{2, 3\}$, $H_3 = \{1, 2, 3\}$, $H_4 = \{2\}$, $H_5 = \{1, 2, 3\}$. Given the chromosome encoding [X, Y], where $X = [5, 3, 1, 2, 4]$, $Y = [1, 2, 3, 1, 1]$. According to the sequence of chromosome X, Task 5 ($X_1 = 5$) is scheduled first assigned to Crane 1 ($H_5(Y_5) = 1$); thereafter, Task 3 ($X_2 = 3$) is assigned to Crane 2 ($H_3(Y_3) = 3$); Tasks 1 and 2 are assigned to Cranes 1 and 3 in that order, immediately after Tasks 5 and 3; Task 4 is assigned to Crane 2. In summary, the schedule for Cranes 1, 2 and 3 is {{5, 1}, {4}, {3, 2}}.

This encoding method can ensure the independence between chromosome X and Y, guaranteeing that the infeasible solutions are excluded in the algorithm search process. On the basis of this encoding

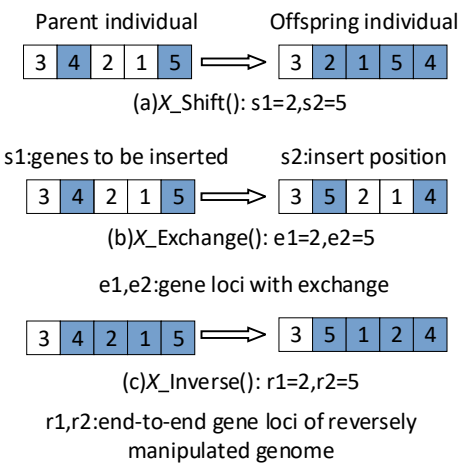
method and the available crane range of each task, the service scheduling of each crane can be obtained.

4.3. Synchronous evolution genetic operator design

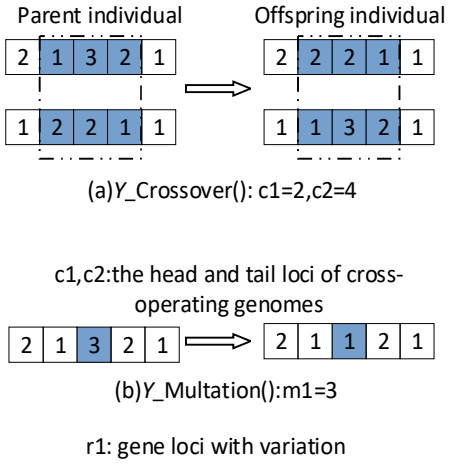
Due to the difference in constraints, the genetic operators for chromosome X and Y are designed separately to ensure their feasibility.

For genes on chromosome X, a value representing a specific task can only occur once. Crossovers and variants of traditional genetic operators may produce infeasible solutions by producing the same value for different genes. Using recombination operators instead of traditional crossover operators, partial genetic algorithms can achieve population diversity by changing the position of genes. In this way, each individual of the generated offspring corresponds to only one parent, so there is no infeasible solution. In this study, three genetic recombination operators were used to populate the effective offspring: gene transposition, gene exchange and gene inversion, as shown in Figure 6(a).

For the Y chromosome, the recombination operator is not suitable because each gene has its own specific range. In this case, a viable solution to obtain populated offspring can be obtained by using the traditional crossover operator, as shown in Figure 6(b). If the parents Y_1 and Y_2 are valid, then swapping the values of the same gene position for Y_1 and Y_2 is also valid for each offspring. After crossover, a simple mutation is used, as shown in Figure 6(b). It is only necessary to restrict the mutation of the gene Y_s to H_s.



(a) Partheno-genetic operators of chromosome X



(b) Crossover and mutation operator of chromosome Y

Figure 6. Genetic operators of TCSC-NSGA-II algorithm.

4.4. TCSC-NSGA-II algorithm design

The operation steps of the TCSC-NSGA-II algorithm are as follows:

1) An initial population of size N is generated.

2) Generation of offspring with population size N is performed by parental crossover and mutation.

3) The N initial populations and the obtained N new populations are merged. Then, the non-dominated sorting stratification and crowding degree are calculated by calculating the objective function value of each individual, and the crowding degree is calculated as follows: (i) non-dominated

stratification of individuals, the number of ranks recorded as i_{rank}.; (ii) decreasing individual ranking according to crowding value, assigning their endpoint crowding to ∞ while calculating the crowding of the remaining individuals recorded as i_d, as shown in Eq (24).

$$i_{\mathrm{d}}=\sum_{j=1}^{m}\left(\left|f_j^{i+1}-f_j^{i-1}\right|\right) \tag{24}$$

where f_j^{i+1} is the j target at point i + 1, and f_j^{i-1} is the j target at point i - 1.

4) N superior individuals are selected as new parents based on the results of non-dominated sorting stratification and crowding calculation. The superiority comparison method is as follows: i) when the number of tiers of i is greater than j, then individual i is superior to j, so $i_{rank} > j_{rank}$; ii) if i and j have the same rank and the crowdedness of individual i is greater than that of individual j, then individual i is superior to j, so $i_{rank} = j_{rank}$ and $i_d > j_d$.

5) The process is repeated until the predefined maximum number of iterations is reached.

4.5. Algorithm procedures

The pseudo-codes of TCSC-NSGA-II algorithm are shown in Table 4.

Table 4. Pseudo-codes of TCSC-NSGA-II algorithm.

Algorithm： TCSC-NSGA-II pseudo-code	
Input:	Initial population, crossover probability, mutation probability, fitness function, number of iterations
output:	Pareto optimal solution；
1.	Chromosomes→scheduling// Conflict identification and resolution
2.	def *m* (f_1,f_2);
3.	f_1 = compute f_1 (scheduling);
4.	f_2 = compute f_2 (scheduling);
5.	return m(f_1,f_2)
6.	def InitialisePopulationX (Task);
7.	PopulationX = random.shuffle(Task);
8.	return PopulationX
9.	def InitialisePopulationY (PopulationX, FeasTower);
10.	Index = 0;
11.	For Item in len(PopulationX)
12.	PopulationY[Index]=FeasTower[Index][random.randrange(len(FeasTower[Index]))]
13.	Index = Index + 1;
14.	return PopulationY
15.	return PopulationY
16.	def CrossTask (Population);
17.	return Parent1, Parent2
18.	def TournamentSelection(Population);
19.	return Parent1, Parent2

Continued on next page

Algorithm：TCSC-NSGA-II pseudo-code

```
def Reproduction(Parent1, Parent2);
    Offspring1.crossover (Parent1.x, Parent2.x);
    Offspring2.crossover (Parent1.y, Parent2.y);
    Offspring1.mutation (Parent1.x, Parent2.x);
    Offspring2.mutation (Parent1.y, Parent2.y);
    return Offspring1, Offspring2
def fast-non-dominated-sort(P);
    for each p∈P
      S_p = Ø;
      n_p =0;
    for each q∈P
      if (p < q)then
        S_p = S_p ∪ {q};
      else if (p > q)
        n_p = n_p + 1;
      if n_p = 0 then
        p_rank = 1;
        F_1 = F_1 ∪ {p};
        i = 1;
    while F_i ≠ Ø
        Q = Ø;
    For each p ∈ F_i
    For each q ∈ S_p
        n_q = n_q - 1;
    if n_q = 0 then
        q_rank = i + 1;
        Q = Q ∪ {q};
        i = i + 1;
        F_i = Q;
    Def crowding-distance-assignment (I);
      l = |I|;
    For each i,set I[i]_distance = 0
      I = sort（I, m）;
      I[1]_distance = I[l]_distance = ∞;
    For i = 2 to （l-1）
      I[i]_distance = I[i]_distance + (I[i + 1].m - I[i - 1].m)/ (f_m^max - f_m^min);
      R_t = P_t ∪ Q_t;
      F = fast-non-dominated-sort (R_t);
      P_t+1 = Ø and i = 1;
      Until |P_t+1| + |F_i| ≤ N;
      def Crowding-distance-assignment （F_i）
        P_t+1 = P_t+1 ∪ F_i;
```

Continued on next page

Algorithm：TCSC-NSGA-II pseudo-code	
62.	$I = i + 1$;
63.	Sort (F_i,u);
64.	$P_{t+1} = P_{t+1} \cup F_i[1:(N\text{-}\lvert P_{t+1}\rvert)]$;
65.	Q_{t+1} = Make-new-pop (P_{t+1});
66.	$t = t + 1$;
67.	return Scheduling set

5. Experiment results and analysis

5.1. Case background

This case study focuses on the construction of the bottom part of Daxing Airport Project Region 1. Due to the regional similarity, the tower crane service scheduling in Region 1 has certain guiding significance for other regions. The main work of a tower crane is to lift formwork, steel bars, steel cages and fresh concrete. In Figure 7, Cranes 1 and 2 in Daxing Airport Region 1 are taken as examples to optimize the task scheduling of tower crane operation. These two tower cranes have the operation area of overlapping areas, but the operations between them are unconstrained.

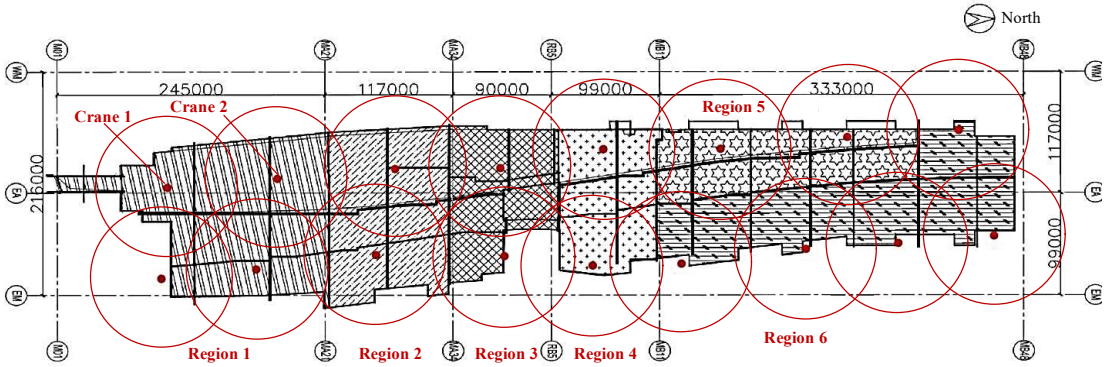


Figure 7. Schematic diagram of Daxing Airport.

5.2. Input model parameters

Tables 5–7 give the material supply position coordinates, material demand position coordinates and two tower crane position coordinates. The vertical velocity is V_v = 136 m / min, the radial velocity is V_r = 60 m / min, the rotary angular velocity is V_a = 0.5 arc / min, and the lifting capacity of the tower crane is 30. In the case of good weather conditions, there is no obstacle in the hook travel path between the supply position and the demand position under normal operation conditions, and there is no additional delay, so set $\mu = 1$. In order to evaluate the ability of tower crane operators on site, parameters λ and η are set to 1.0 and 0.25, respectively. Material loading and unloading time is set to 1.0 time units (minutes).

Table 5. Place of supply.

Material supply	position coordinate			Material supply type
	x	*y*	*z*	
1	10	36	0	2, 4
2	10	70	0	1, 2, 3
3	36	86	0	2, 3
4	70	86	0	1, 4

Table 6. Material demand.

Material demand	Material demand coordinates		
	x	*y*	*z*
1	36	36	19
2	60	36	19
3	82	36	19
4	82	60	19
5	24	48	30
6	48	48	26
7	72	48	26
8	24	72	30
9	48	72	34
10	72	72	30

Table 7. Tower position coordinates.

Tower crane	Position coordinates of tower crane		
	x	*y*	*z*
Cr1	63	55	70
Cr2	30	66	70

MATLAB R2018a software is applied to solve the problem. The operating environment is the i7 –8565U core of CPU @ 1.80 GHz, 1.99 GHz, and the parameters such as population number, maximum iteration number, crossover probability and mutation probability are constant. Each task randomly generates the demanded material type and material demand place No., and 10, 50, 70 and 100 tasks are set, respectively. The number of chromosomes is set to 100. The non-dominated sorting genetic algorithm is implemented to solve the problem.

5.3. TCSC-NSGA-II multi-objective optimized results

In this study, the objective is to obtain the maximum cross-tasks interval time while ensuring a relatively short duration of makespan to make the construction process smoother and safer. From the results, it can be seen that the obtained non-dominated solution has some practical significance. In the past, the shortest makespan was excessively pursued, which led to frequent collision and interference, while a very small increase in completion time would obviously improve the interval of cross-tasks if

viewed from the current perspective. This means a small reduction in operation efficiency can bring a significant increase in safety.

In order to show the Pareto front more intuitively, the Pareto front graphs of 50 tasks and 100 tasks are obtained with the objective function f_1 as the ordinate and the objective function f_2 as the abscissa. As shown in Figures 8 and 9, each point in the graph is the optimal solution. The optimized Pareto front image is tilted above the right, indicating that the solutions obtained in the figure are non-dominated. The results are consistent with the definition of the non-dominated solution set mentioned above, indicating that the obtained optimal solution is reliable.

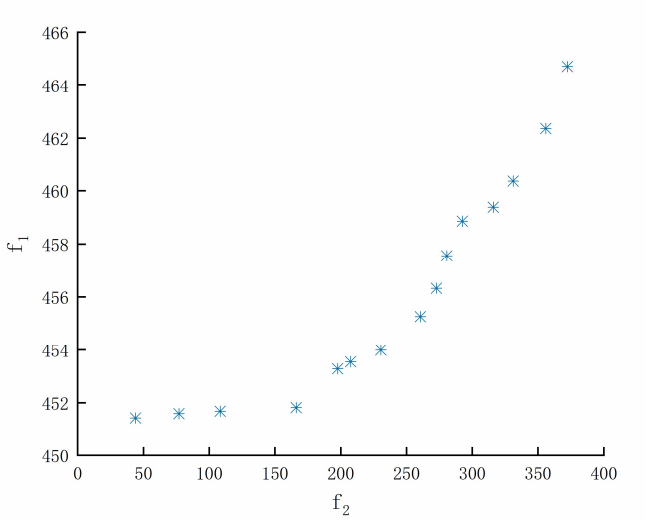


Figure 8. 50 tasks on the Pareto front.

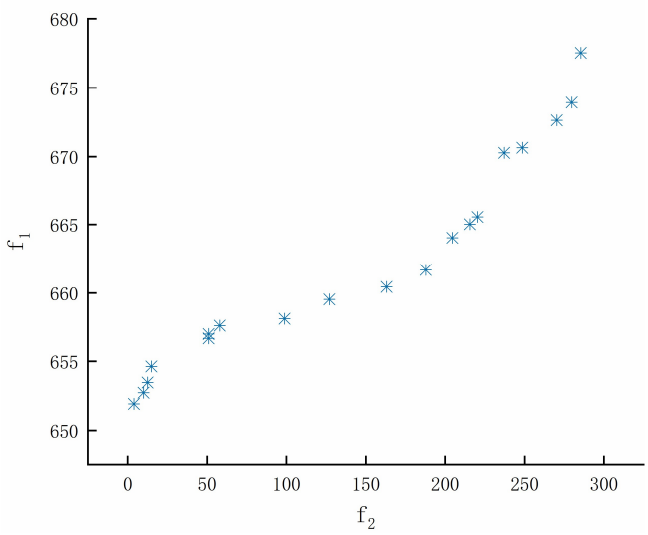


Figure 9. 100 tasks on the Pareto front.

5.4. Comparative analysis with classical genetic algorithm

In order to further verify the rationality and effectiveness of the proposed multi-objective model

and algorithm, this section will use the classical genetic algorithm to solve the two objective functions respectively.

In order to explore the reasonable and effective Pareto optimal solution, the specific operation is as follows: The genetic algorithm is used to optimize f_1 and f_2 respectively, and the corresponding optimal value is obtained. The relevant parameters are consistent with the previous design. First, using a genetic algorithm to optimize f_1, 10, 50, 70, and 100 tasks are generated, with each group running 30 independently. Figure 10 shows the change of the objective function f_1 in the 500 iterations of the computing. It can be seen that with the continuous search of the genetic algorithm, the objective function value decreases and finally tends to be stable, indicating that the single objective genetic algorithm is effective. The value of f_2 also tends to be stable after 500 iterations, and the convergence trend is shown in Figure 11.

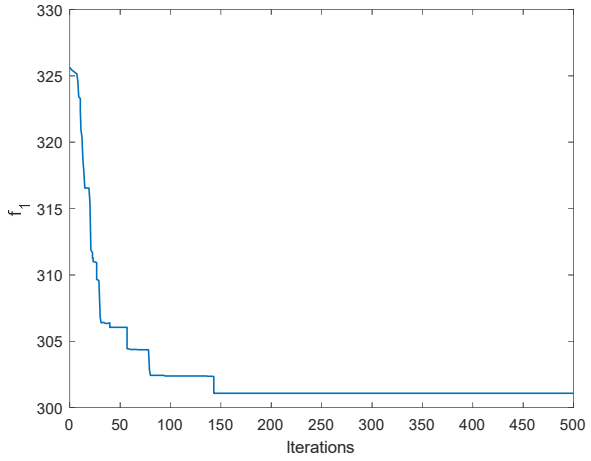


Figure 10. f_1 Optimization convergence curve.

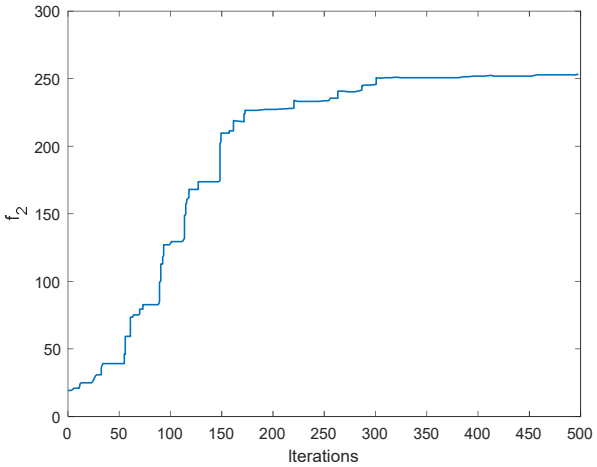


Figure 11. f_2 Optimization convergence curve.

Tables 8 and 9 give the optimal solutions of f_1 and f_2 under 10, 50, 70, 100 different-size tasks. It can be seen from Table 8 that when the optimization is with only f_1, there may be a negative number

of f_2, which means tower crane 1 and tower crane 2 enter the overlapping area at the same time. Although there are usually monitoring and sensing devices installed on the equipment which can make it brake, this will inevitably lead to a sudden change in speed of TC, resulting in waste of time and even safety accidents. At the same time, it can also be seen that when the optimization is carried out with f_1 as the objective, the best value is better than that of f_1 in the Pareto optimal solution set, but the value of f_2 is much worse. It can also be seen from Table 9 that when only the objective function f_2 is optimized, the best value is better than the maximum value of f_2 in the Pareto optimal solution set, but the value of f_1 may increase at different degrees. The results can also show an interesting phenomenon that significant improvement of f_2 can be achieved at the cost of tiny increase in f_1, which means effective avoidance of the tower cranes entering the overlapping area at the same time.

Table 8. The objective function value of f_1 is optimized by single objective.

n	f_1	f_2
10	67.87	2.79
50	339.68	-11.92
70	534.72	-12.35
100	673.63	-13.49

Table 9. The objective function value of f_2 is optimized by single objective.

n	f_1	f_2
10	100.74	130.73
50	454.16	141.56
70	637.63	292.66
100	872.83	151.50

Based on the above results and analysis, it can be concluded that the results obtained by the dual-objective optimization based on TCSC-NSGA-II in this paper are reasonable. When the non-dominated genetic algorithm is used to optimize the two objective functions at the same time, the chromosomes entering the overlapping area at the same time are eliminated, which ensures f_2 as non-negative. Meanwhile, the value of the objective function f_1 obtained is close to the optimal value. Hence, it shows that the service scheduling optimization of multi-objective tower crane based on non-dominated genetic algorithm is reasonable and effective.

6. Conclusions

The multi-crane service scheduling problem is one of the major problems related to tower crane operation that has received limited attention in terms of efficiency and safety. In this paper, a mathematical model of the multi-objective tower service scheduling problem was developed considering the conflicting task intervals. Double-layer chromosome and simultaneous coevolutionary strategy were designed and implemented with the NSGA-II computational procedure. The algorithm proposed in this paper is not only an improvement in efficiency compared to previous optimization methods but also is able to handle the large-scale tasks with reasonable solution quality and efficiency,

which is difficult to solve with integer linear programming and mixed integer linear programming. In addition, the TCSC-NSGA-II method introduced as an a posteriori method is less dependent on background knowledge than traditional a priori methods, and selection is made only after candidate solutions are obtained, so the diversity of solutions is better than traditional methods.

A case study of the International Airport, Daxing, China, was conducted to verify the proposed model and algorithm in the paper. In terms of makespan of tower cranes and the time interval of cross-tasks, the obtained optimization solution was proved to be effective and can meet with the practical requirement of the engineering project. The current work can be applied directly to enhance operation management of tower cranes on the construction site.

In future research, it would be useful to remove as much simplification as possible from the proposed optimization model to make it more practical for use in the field. Additionally, it would be beneficial to incorporate dynamic features that align with actual construction requirements. For example, uncertainty regarding the arrival time of tasks and the sequential constraints of prefabricated components should be taken into account, and appropriate algorithms should be designed to address these issues, as they are often present in real construction scenarios.

Acknowledgments

This research was funded by the Beijing Social Science Fund General Project (No. 19GLB011).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. B. Flyvbjerg, Over budget, over time, over and over again: managing major projects, in *The Oxford Handbook of Project Management*, Oxford: Oxford University Press, Britain, (2011), 321–344. https://doi.org/10.1093/oxfordhb/9780199563142.003.0014
2. A. Tork, *A Real-time Crane Service Scheduling Decision Support System (css-dss) For Construction Tower Cranes*, Electronic Theses and Dissertations, University of Central Florida, 2013. http://stars.library.ucf.edu/etd/2799
3. M. Hussein, T. Zayed, Crane operations and planning in modular integrated construction: Mixed review of literature, *Autom. Constr.*, **122** (2021), 103466. https://doi.org/10.1016/j.autcon.2020.103466
4. Y. Zhao, C. F. Cao, Z. S. Liu, A framework for prefabricated component hoisting management systems based on digital twin technology, *Buildings*, **12** (2022), 167–174. https://doi.org/10.3390/buildings12030276
5. A. Khalili, D. K. Chua, Integrated prefabrication configuration and component grouping for resource optimization of precast production, *J. Constr. Eng. Manage.*, **140** (2014), 4013052. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000798
6. S. C. Kang, E. Miranda, Planning and visualization for automated robotic crane erection processes in construction, *Autom. Constr.*, **15** (2006), 398–414. https://doi.org/10.1016/j.autcon.2005.06.008

7. A. Shapira, M. Goldenberg, AHP-based equipment selection model for construction projects, *J. Constr. Eng. Manage.*, **131** (2005), 1263–1273. https://doi.org/10.1061/(ASCE)0733-9364(2005)131:12(1263)
8. A. Shapira, M. Goldenberg, "Soft" considerations in equipment selection for building construction projects, *J. Constr. Eng. Manage.*, **133** (2007), 749–760. https://doi.org/10.1061/(ASCE)0733-9364(2007)133:10(749)
9. C. M. Tam, T. K. Tong, GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction, *Constr. Manage. Econ.*, **21** (2003), 257–266. https://doi.org/10.1080/0144619032000049665
10. P. Zhang, F. C. Harris, P. O. Olomolaiye, G. D. Holt, Location optimization for a group of tower cranes, *J. Constr. Eng. Manage.*, **125** (1999), 115–122. https://doi.org/10.1061/(ASCE)0733-9364(1999)125:2(115)
11. K. Alkriz, J. C. Mangin, A new model for optimizing the location of cranes and construction facilities using genetic algorithms, in *Proceedings 21st Annual ARCOM Conference,* UK, (2005), 981–991.
12. D. Briskorn, M. Dienstknecht, Mixed-integer programming models for tower crane selection and positioning with respect to mutual interference, *Eur. J. Oper. Res.*, **273** (2018), 160–174. https://doi.org/10.1016/j.ejor.2018.07.033
13. A. Younes, M. Marzouk, Tower cranes layout planning using agent-based simulation considering activity conflicts, *Autom. Constr.*, **93** (2018), 348–360. https://doi.org/10.1016/j.autcon.2018.05.030
14. C. Huang, C. K. Wong, C. M. Tam, Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming, *Autom. Constr.*, **20** (2011), 571–580. https://doi.org/10.1016/j.autcon.2010.11.023
15. Y. S. Ji, F. Leite, Automated tower crane planning: leveraging 4-dimensional BIM and rule-based checking, *Autom. Constr.*, **93** (2018), 78–90. https://doi.org/10.1016/j.autcon.2018.05.003
16. Z. Q. Zhang, W. Pan, Multi-criteria decision analysis for tower crane layout planning in high-rise modular integrated construction, *Autom. Constr.*, **127** (2021). https://doi.org/10.1016/j.autcon.2021.103709
17. Y. Fang, B. Ma, P. Wang, X. Zhang, A motion planning-based adaptive control method for an underactuated crane system, *Control Syst. Technol. IEEE Trans.*, **20** (2012), 241–248. https://doi.org/10.1109/TCST.2011.2107910
18. J. J. Cruz, F. Leonardi, Minimum-time anti-swing motion planning of cranes using linear programming, *Opt. Control Appl. Methods*, **34** (2013), 191–201. https://doi.org/10.1002/oca.2016
19. M. Al-Hussein, N. M. Athar, H. Yu, H. Kim, Integrating 3D visualization and simulation for tower crane operations on construction sites, *Autom. Constr.*, **15** (2006), 554–562. https://doi.org/10.1016/j.autcon.2005.07.007
20. T. Cheng, J. Teizer, Modeling tower crane operator visibility to minimize the risk of limited situational awareness, *J. Comput. Civ. Eng.*, **28** (2014), 4014004. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000282
21. S. C. Kang, E. Miranda, Computational methods for coordinating multiple construction cranes, *J. Comput. Civ. Eng.*, **22** (2008), 252–263. https://doi.org/10.1061/(ASCE)0887-3801(2008)22:4(252)

22. A. Zavichi, K. Madani, P. Xanthopoulos, A. A. Oloufa, Enhanced crane operations in construction using service request optimization, *Autom. Constr.*, **47** (2014), 69–77. https://doi.org/10.1016/j.autcon.2014.07.011
23. A. Zavichi, A. H. Behzadan, A real time decision support system for enhanced crane operations in construction and manufacturing, in *2011 ASCE International Workshop on Computing in Civil Engineering Miami*, Florida, (2011), 586–593. https://doi.org/10.1061/41182(416)72
24. B. Nils, B. Dirk, M. Frank, A generalized classification scheme for crane scheduling with interference, *Eur. J. Oper. Res.*, **258** (2017), 343–357. https://doi.org/10.1016/j.ejor.2016.08.041
25. C. Huang, C. K. Wong, Optimization of crane setup location and servicing schedule for urgent material requests with non-homogeneous and non-fixed material supply, *Autom. Constr.*, **89** (2018), 183–198. https://doi.org/10.1016/j.autcon.2018.01.015
26. K. Seungho, K. Sangyong, L. Dongoun, Sequential dependency structure matrix based framework for leveling of a tower crane lifting plan, *Can. J. Civ. Eng.*, **45** (2018), 516–525. https://doi.org/10.1139/cjce-2017-0177
27. Y. Ahmed, M. Mohamed, Tower cranes layout planning using agent-based simulation considering activity conflicts, *Autom. Constr.*, **93** (2018), 348–360. https://doi.org/10.1016/j.autcon.2018.05.030
28. A. H. Malak, Z. Emile, Crane overlap and operational flexibility: balancing utilization, duration, and safety, *Constr. Innovation*, **18** (2018), 43–63. https://doi.org/10.1108/CI-11-2016-0062
29. H. Tarhini, B. Maddah, F. Hamzeh, The traveling salesman puts-on a hard hat -tower crane scheduling in construction projects, *Eur. J. Oper. Res.*, **292** (2020), 327–338. https://doi.org/10.1016/j.ejor.2020.10.029
30. C. Huang, W. J. Li, W. S. Lu, F. Xue, M. Liu, Z. S. Liu, Optimization of multiple-crane service schedules in overlapping areas through consideration of transportation efficiency and operational safety, *Autom. Constr.*, **127** (2021). https://doi.org/10.1016/j.autcon.2021.103716
31. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi objective genetic algorithm NSGA-II, *IEEE Trans. Evol. Comput.,* **6** (2002), 182–197. https://doi.org/10.1109/4235.996017
32. K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, *IEEE Trans. Evol. Comput.,* **18** (2014), 577–601. https://doi.org/10.1109/TEVC.2013.2281535
33. H. Zhang, J. Li, M. N. Hong, Y. Man, Z. He, Cost optimal production-scheduling model based on VNS-NSGA-II hybrid algorithm—study on tissue paper mill, *Processes*, **10** (2022), 2072–2072. https://doi.org/10.3390/PR10102072
34. W. K. Fang, Z. L. Guan, P. Y. Su, D. Luo, L. S. Ding, L. Yue, Multi-objective material logistics planning with discrete split deliveries using a hybrid NSGA-II Algorithm, *Mathematics*, **10** (2022), 2871–2871. https://doi.org/10.3390/math10162871
35. X. K. Li, F. H. Yan, J. Ma, Z. Z. Chen, X. Y. Wen, Y. Cao, RBF and NSGA-II based EDM process parameters optimization with multiple constraints, *Math. Biosci. Eng.*, **16** (2019), 5788–5803. https://doi.org/10.3934/mbe.2019289.