COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 01/06 pp 1−17
Volume 2, Number 1, March 2022
DOI: 10.23919/CSMS.2021.0028

Trajectory Predictions with Details in a Robotic Twin-Crane System

Ning Zhao, Gabriel Lodewijks*, Zhuorui Fu, Yu Sun, and Yue Sun

Abstract: Nowadays, more automated or robotic twin-crane systems (RTCSs) are employed in ports and factories to improve material handling efficiency. In a twin-crane system, cranes must travel with a minimum safety distance between them to prevent interference. The crane trajectory prediction is critical to interference handling and crane scheduling. Current trajectory predictions lack accuracy because many details are simplified. To enhance accuracy and lessen the trajectory prediction time, a trajectory prediction approach with details (crane acceleration/deceleration, different crane velocities when loading/unloading, and trolley movement) is proposed in this paper. Simulations on different details and their combinations are conducted on a container terminal case study. According to the simulation results, the accuracy of the trajectory prediction can be improved by 20%. The proposed trajectory prediction approach is helpful for building a digital twin of RTCSs and enhancing crane scheduling.

Key words: twin-crane systems; trajectory prediction; acceleration; velocity; interference

1 Introduction

Gantry cranes are one of the most commonly used pieces of equipment in material handling systems to transport heavy and bulky containers or piece goods. Nowadays, to improve the efficiency of the lifting process, more container terminals and factories are being equipped with automated or robotic twin-crane systems (RTCSs). In a twin-crane system, two cranes run on a common runway. They cannot pass each other and must keep a minimum safety distance between them. Crane interferences seriously impact the working efficiency of a twin-crane system[1]. Therefore, the crane trajectory prediction is a practical and valuable problem to address. Many studies about crane scheduling have focused on this problem, and excellent results have been achieved using a simulation-optimization framework[2]. In most of these simulation studies, cranes are assumed to travel with a constant velocity. However, some details on crane movements, such as crane acceleration or deceleration (CA), trolley movements (TMs), and different crane velocities corresponding with loading and unloading (VC), are normally neglected. The inclusion of these detailed movements makes the theoretical crane travel different compared to the crane movement prediction assuming a constant crane velocity. The difference may result in inaccurate trajectory predictions and worsen the performance of RTCSs. In this study, we investigate how much these detailed movements influence the crane trajectory prediction and rank them by order of importance. We developed a new trajectory prediction approach with consideration of detailed movements. Consequently, cranes may be scheduled to escape interference with an accurate trajectory prediction. This method will be helpful for the design of a digital twin (DT) of an RTCS and the enhancement of crane scheduling.

This paper is organized as follows: In Section 2, we present a thorough review of the relevant literature. In Section 3, we describe the problem and define

• Ning Zhao, Zhuorui Fu, Yu Sun, and Yue Sun are with the School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: nickzhao@me.ustb.edu.cn.
• Gabriel Lodewijks is with the School of Aviation, University of New South Wales, Sydney 2052, Australia. E-mail: g.lodewijks@unsw.edu.au.
∗ To whom correspondence should be addressed.
Manuscript received: 2021-09-23; revised: 2021-11-08; accepted: 2021-11-22

notations. In Section 4, we analyze the detailed movements and present a trajectory prediction approach. In Section 5, we present a case study of a container terminal and investigate the impacts of different detail movements. In Section 6, we draw conclusions based on the case study.

2 Literature Review

Crane trajectory prediction is the precondition of efficient crane scheduling in an RTCS. Crane scheduling is well studied in terms of task sequencing and task assignments[2]. From a literature review, it became clear that most papers studied this problem with the objective to minimize the overall task completion time or makespan. For this purpose, operations research based approaches have been widely used to solve the crane scheduling problem, such as branch-and-bound[3, 4], branch-and-cut[5, 6], dynamic programming[7−11], branch-and-price[12], and alternating direction method of multipliers[13]. To match with operations research based approaches, the crane trajectory prediction is simplified in the aforementioned studies. Crane movements are assumed as travels with a constant velocity, whereas CA and VC are ignored. Peterson et al.[3] are the only some that considered TMs, which makes trajectory prediction practical.

Most scheduling problems are NP-hard problems, including the crane scheduling problem. Thus, besides the operation research based approaches, heuristic algorithms are widely used to study the scheduling problem. Many research works have considered the optimization of algorithms to improve efficiency and effectiveness. For instance, Cao et al.[14] proposed a comprehensive learning particle swarm optimizer (CLPSO) embedded with local search to enhance its performance. He et al.[15] proposed a discrete multi-objective firework algorithm to address the multi-objective flow-shop scheduling problem with sequence-dependent setup times. Li et al.[16] proposed two many-objective evolutionary algorithms to examine the energy-efficient job-shop scheduling problem with limited workers. Luo et al.[17] studied the organization of production based on suborders in real time under the constraints of smart contracts. A real-time edge scheduling model and a real-time edge adjustment method were proposed to solve this problem. He et al.[18] studied an energy-efficient job-shop scheduling problem with sequence-dependent setup times and proposed a multi-objective optimization framework based on the finite element method and an adaptive local search strategy to solve the problem.

In the crane scheduling problem, genetic algorithms (GAs)[1, 19−32] and heuristic methods[33−38] are the most commonly used approaches. Zhang and Rose[1] and Al-Dhaheri et al.[19] used a GA together with a simulation approach to solve the crane scheduling problem and to obtain an optimal integrated crane schedule. A GA was also used to study the quay crane scheduling problem[20−22, 28] and its extended problems, such as berth allocation and quay crane assignment problem[23], quay crane assignment and scheduling problem[25], quay crane scheduling problem with draft and trim constraints[29], and integrated quay crane and yard truck scheduling problem[31]. Other problems using GAs include the scheduling problems of multiple vehicles/cranes along a common lane[24, 26, 27, 30] and the yard crane scheduling problem with a non-interference constraint in a container terminal[32]. The results of the previously mentioned studies show that GAs manage to provide practical solutions and significant time savings for scheduling problems. For the heuristic methods, Carlo and Vis[34] examined sequencing dynamic storage systems with multiple lifts and shuttles, which bear resemblances to the study of twin-crane scheduling on a common lane. They proposed an integrated heuristic look-ahead strategy to assign requests and priority rules to handle interferences. Furthermore, Carlo and Martínez-Acevedo[35] studied the priority rules for scheduling a twin-crane system in container ports. They assumed that the crane movement determines the crane conflict and that there is an exchange point where one crane can start a request and leave it to the other crane to complete it. Clearly, this exchange point simplifies the interference situation and lags the working efficiency with the repetition of loading and unloading. Following Carlo’s work, Gharehgozli et al.[36] studied port logistics. The conflict location was determined according to the horizontal movement of the crane and the loading/unloading time. A heuristic algorithm was used to study the difference between the presence and absence of a handshake area and the size, location, and number of handshake areas by avoiding collisions and pursuing the shortest completion time. Other heuristics are also used to study the scheduling problem[33, 37, 38]. For example, Chen et al.[37] built a heuristic decomposition framework enhanced by tabu search to study the quay crane scheduling problem at an indented berth. Li et al.[38] considered non-conflict constraints between cranes, station-capacity constraints, and jobs with inaccurate release times and different temporal scheduling objectives. A heuristic method for a multi-crane-scheduling problem was presented to minimize the total cost. Among the above studies, Moccia et al.[6] considered the safety distance to be zero, whereas others assumed a certain safety distance. Hakam et al.[26] considered the velocity as infinite, whereas other studies assumed cranes to travel with constant velocities. Although TMs are considered in some studies, CA and VC are still neglected during trajectory prediction.

The aforementioned papers made great achievements on crane scheduling both in theory and practice. From these papers, we have made an overview in Table 1. As shown in Table 1, the scheduling method attracted most of the research interests, whereas detailed movements were neglected. Among them, TMs were considered by several studies, whereas CA and VC were totally neglected. Besides RTCSs, similar studies on shuttle-based storage and retrieval systems[39] and robotic cells with multiple dual gripper robots[40, 41] arose. Due to different structures, CA was considered in these studies, whereas VC and TMs were neglected.

From the viewpoint of complexity, TMs could be calculated with a constant velocity similar to crane movements. By contrast, CA and VC must be calculated in a dynamic way. For this reason, CA and VC calculations are highly suitable for the DT technique. The concept of a DT was first introduced and later elaborated by Grieves[42] in his product lifecycle management classes. Since then, DTs have become a noticeable issue. Recently, the industry has been one of the most popular areas for DT applications[43] and some industry systems are studied with a DT technique. In 2019, Tao et al.[44] integrated DT research in the industry and analyzed current challenges and future developments. In 2019, Fang et al.[45] proposed the combination of DT techniques with dynamic job-shop scheduling to achieve real-time and precise scheduling. A DT-enhanced shop-floor scheduling was presented by Zhang et al.[46]. All the research results showed that applying DT technology requires an accurate mapping to physical systems in a virtual space. For a physical system, the research on the control problem of mechanical systems is precise and practicable and has been verified in physical crane systems[47]. Thus, for the virtual crane system, precise conditions are needed to be considered to accurately reflect the real-time state of a physical crane system. For this reason, TMs, CA, and VC are strongly dynamic and will be helpful for building DTs of twin-crane systems. However, as shown in Table 1, only a few studies are concerned with TMs, CA, and VC. Thus, in this study, the trajectory prediction approach is examined with consideration of the above-mentioned details. The objective of this paper is to present an

Table 1 Overview of studies considering crane movements.

Publication	Method	TM	CA	VC
Zhang and Rose (2013)[1]	GA	×	×	×
Legato and Trunfio (2014)[4]	BB	×	×	×
Peterson et al. (2014)[3]	BB, BP	√	×	×
Moccia et al. (2006)[6]	BC	×	×	×
Cheng et al. (2015)[5]	BC	×	×	×
Kasm and Diabat (2019)[12]	BP	×	×	×
Chen et al. (2020)[13]	ADMM	×	×	×
Briskorn and Angeloudis (2016)[8]	DP	×	×	×
Boysen et al. (2015)[7]	DP, DEC	×	×	×
Park et al. (2010)[11]	DP, H	×	×	×
Kung et al. (2014)[10]	DP	×	×	×
Aron et al. (2010)[9]	DP	×	×	×
Al-Dhaheri et al. (2016)[19]	GA	×	×	×
Chung and Choy (2012)[22]	GA	×	×	×
Chung and Chan (2013)[21]	GA, H	×	×	×
Kayeshgar et al. (2012)[28]	GA	×	×	×
Chang et al. (2017)[20]	GA	×	×	×
Wu and Ma (2017)[29]	GA	×	×	×
Fu et al. (2014)[25]	H, GA	×	×	×
Correcher and Alvarez-Valdes (2017)[23]	GA	×	×	×
Emde and Boysen (2014)[24]	GA	×	×	×
Hakam et al. (2012)[26]	GA	×	×	×
Hu et al. (2016)[27]	GA	×	×	×
Choe et al. (2012)[30]	GA	×	×	×
Skaf et al. (2021)[31]	GA	×	×	×
Carlo and Martínez-Acevedo (2015)[35]	PR	√	×	×
Carlo and Vis (2012)[34]	PR, H	√	×	×
Gharehgozli et al. (2017)[36]	PR	√	×	×
Ge et al. (2012)[33]	H	×	×	×
Chen et al. (2011)[37]	H	×	×	×
Li et al. (2020)[38]	H	×	×	×
Zhao et al. (2018)[39]	H	×	√	×
Dawande et al. (2002)[40]	H	×	√	×
Geismar et al. (2008)[41]	H	×	√	×

Note: GA — Genetic Algorithm, BB — Branch-and-Bound, BP — Branch-and-Price, BC — Branch-and-Cut, DP — Dynamic Programming, ADMM — Alternating Direction Method of Multipliers, DEC — Decomposition, H — Heuristic, PR — Priority Rules, √ — The factor is considered in the study, × — The factor is not considered in the study.

accurate trajectory prediction approach. This approach is dynamic and will be helpful for building the DTs of RTCSs and enhancing crane scheduling.

3 Problem Descriptions and Notations

Typically, a crane moves horizontally over a bridge along parallel overhead runways (X axis), and a trolley moves along the bridge (Y axis) (see Fig. 1a). The materials are lifted up and down by a hoist integrated into the trolley (Z axis) and can be transported to any position covered by the parallel runways.

The studied twin-crane scheduling problem can be illustrated in the top view of a multi-crane system shown in Fig. 1a. Assume there are a series of transport tasks waiting for the cranes, where each task has a definite origin and destination. The crane selects a task and conducts movements with the trolley in the X axis and Y axis, respectively. An evasion movement will be conducted by the crane when interference occurs. The crane scheduling objective is to minimize the overall completion time of all tasks, which is also called the makespan. To illustrate the connection between trajectory prediction and crane scheduling, the notations used in this study are as follows:

n: number of tasks;

i: i-th task;

j: j-th crane, in this paper, ($j = 1$ (crane on the left), 2 (crane on the right));

n_j: number of tasks allocated to crane j;

$x_{i,j}$: $x_{i,j} = 1$ if the i-th task is allocated to crane j; $x_{i,j} = 0$ otherwise;

s: safety distance;

$tA_{j,i}$: avoidance time of crane j with task i;

$tL_{j,i}$: travel time of crane j from origin to loading location with task i;

$tU_{j,i}$: travel time of crane j from loading location to unloading location with task i;

t_a: time point to predict a crane's trajectory during simulation;

S_X: crane's travel distance from the origin to destination in the X axis;

S_Y: trolley's travel distance from origin to destination in the Y axis;

t_X: crane's travel time from the origin to destination in X axis;

t_Y: trolley's travel time from origin to destination in Y axis;

$O_{j,i,X}$: origin of the i-th task in the X axis of the j-th crane;

$O_{j,i,Y}$: oirigin of the i-th task in the Y axis of the j-th crane;

$T^{i,X}_{j,s,O}$: arrival time of the corresponding crane with the i-th task at $O_{i,x}$ of the j-th crane;

$T^{i,X}_{j,f,O}$: loading complete time of the corresponding crane with the i-th task at $O_{i,x}$ of the j-th crane;

$D_{i,j,X}$: destination of the i-th task in the X axis of crane j;

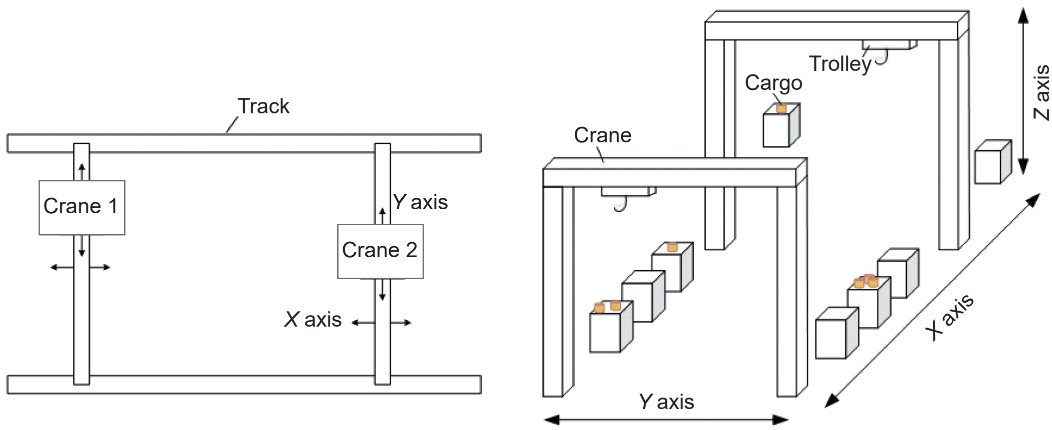
$D_{i,j,Y}$: destination of the i-th task in the Y axis of crane j;

$T^{i,X}_{j,s,D}$: arrival time of the j-th crane with the i-th task at $D_{i,x}$;

$T^{i,X}_{j,f,D}$: unloading completion time of the j-th crane with the i-th task at $D_{i,x}$;

$T_{collide,j}$: collision time of the j-th crane to the other one;

$T_{j,s,a}$: start time of the j-th crane to conduct avoidance;



(a) Top view of twin-crane system (b) Sketch map of twin-crane system

Fig. 1 Overview of a twin-crane system.

$T_{j,f,a}$: completion time of the j-th crane to conduct avoidance;

$T_{j,i}$: completion time of the j-th crane with the i-th task;

ΔT: time interval between two consecutive time points to predict the trajectory;

$P_{j,X,t}$: current location of the j-th crane in the X axis at time t;

$P_{j,Y,t}$: current location of the trolley on the j-th crane in the Y axis at time t;

$P_{j,X,a}$: location of the j-th crane in the X axis when it starts to conduct avoidance;

V_{Xmax}: maximum velocity of the loaded crane;

V_{Ymax}: maximum velocity of the loaded trolley;

V'_{Xmax}: maximum velocity of the unloaded crane;

V'_{Ymax}: maximum velocity of the unloaded trolley;

$V_{j,a,b}$: average velocity of the j-th crane moving from position a to b;

a_X: acceleration/deceleration of the crane;

a_Y: acceleration/deceleration of the trolley;

l_t: loading/unloading time;

C_j: completion time of all tasks allocated to crane j.

Typically, Eq. (1) denotes the allocation of tasks.

$$x_{i,j} = \begin{cases} 1, \text{task } i \text{ allocated to crane } j; \\ 0, \text{task } i \text{ not allocated to crane } j \end{cases} \quad (1)$$

Equation (2) denotes that every task has to be allocated to one crane.

$$\sum_{i=1}^{n}\sum_{j=1}^{2} x_{i,j} = \sum_{j=1}^{2}\sum_{i=1}^{n} x_{i,j} = n \quad (2)$$

With the constraint of Eq. (2), the objective of the crane scheduling is always to minimize the makespan of all tasks allocated to all cranes, which is denoted by Formula (3). The scheduling includes task allocation and sequencing. Clearly, a different scheme results in a different makespan.

$$min\{max(C_1, C_2)\} \quad (3)$$

In Formula (3), C_j can be calculated as $max\ T_{j,n_j}$, and T_{j,n_j} can be calculated with Eq. (4):

$$T_{j,n_j} = T_{j,n_{j-1}} + tA_{j,n_{j-1}} + tL_{j,n_j} + tU_{j,n_j} + 2lt \quad (4)$$

where $tA_{j,n_{j-1}}$is equal to

$$tA_{j,n_j} = T_{j,f,a} - T_{j,s,a} \quad (5)$$

Equation (4) shows that the completion time of the last task is equal to the summation of the completion time of the former task and the total processing time of the last task. Equation (4) denotes a recursive procedure, and every task is constrained by the corresponding former task. This condition indicates that a small deviation in the prediction of the processing time may accumulate in a large deviation with the increase of the task number. For this reason, the use of DTs is helpful to correct deviations with the monitoring of physical RTCSs. Furthermore, the CA, VC, and TMs make contributions to tL_{j,n_j} and tU_{j,n_j}. However, tA_{j,n_j} is dynamically constrained by the position of other cranes. Equation (5) denotes that tA_{j,n_j} depends on the trajectory prediction whereas trajectory prediction depends on tA_{j,n_j}. Consequently, Eq. (4) is a strong dynamic procedure, and it is difficult to predict the trajectory and makespan with a static formulation. For this reason, a simulation-based trajectory prediction approach is presented.

4 Simulation-Based Trajectory Prediction

As discussed in Section 3, twin-crane scheduling can be regarded as an optimization problem with objective functions. However, based on the strong dynamic characteristics denoted in Eq. (4), crane trajectories cannot be easily evaluated and predicted using equations or continuous simulations. Hence, to make an accurate prediction by taking dynamic details into consideration, a discrete event simulation is employed. To simplify this problem, a single-crane trajectory prediction without dynamic characteristics is studied first. Then, a twin-crane trajectory prediction with dynamic characteristics is examined.

4.1 Single-crane trajectory prediction with CA and VC

Typically, a crane or trolley starts the travel with acceleration and stops with deceleration. As mentioned in the literature review, cranes and TMs are always simplified by assuming that a travel with a constant velocity is equal to the maximum velocity V_{max}. The effect of this simplification is shown in Fig. 2a: The trajectory is a curve when acceleration and deceleration are considered, whereas it is a straight line when acceleration and deceleration are neglected. Moreover, the crane's travel time, including acceleration and deceleration, is slightly longer than the travel time with constant velocity over the same travel distance. Therefore, there are tiny deviations in the processing time if the travel time is predicted with a constant velocity V_{max}.

Loaded crane travels with a low velocity, whereas unloaded crane travels with a high velocity. As shown

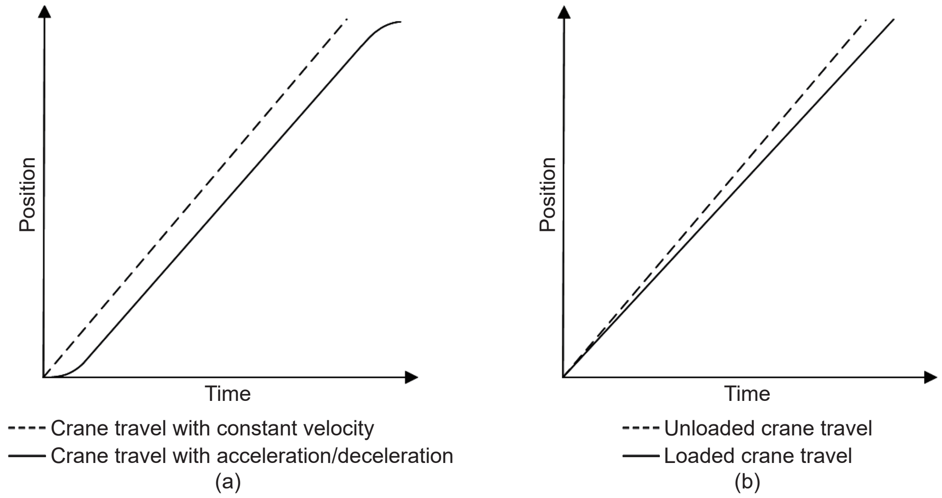


Fig. 2 Example of trajectory of acceleration.

in Fig. 2b, there are tiny deviations in the processing time between the loaded travel and unloaded travel. The travel trajectory will have different slopes to denote the different travel velocities. For this reason, tiny deviations will accumulate to a large deviation if the CA and VC are not considered in the trajectory prediction. Unpredicted interference may happen and lag the total completion time.

The accelerating and decelerating procedures of a single crane and trolley can be regarded as the same. Previous research studied the influence and necessity of considering acceleration and deceleration in different travel distances[39]. Consequently, the travel time of a single crane can be predicted with the maximum crane velocity V_{Xmax}, maximum trolley velocity V_{Ymax}, travel distance of crane S_X and trolley S_Y, and accelerations of crane a_X and trolley a_Y. The crane's travel time t_X and trolley's travel time t_Y can be calculated as follows:

$$t_X = \begin{cases} \dfrac{S_X}{V_{Xmax}} + \dfrac{V_{Xmax}}{a}, & S_X \geqslant \dfrac{V_{Xmax}^2}{a_X}; \\ 2\sqrt{\dfrac{S_X}{a_X}}, & S_X < \dfrac{V_{Xmax}^2}{a_X} \end{cases} \tag{6}$$

$$t_Y = \begin{cases} \dfrac{S_Y}{V_{Ymax}} + \dfrac{V_{Ymax}}{a}, & S_Y \geqslant \dfrac{V_{Ymax}^2}{a_Y}; \\ 2\sqrt{\dfrac{S_Y}{a_Y}}, & S_Y < \dfrac{V_{Ymax}^2}{a_Y} \end{cases} \tag{7}$$

With consideration of VC, the loaded crane travels with a small V_{Xmax} and V_{Ymax}, while unloaded crane travels with a great V'_{Xmax} and V'_{Ymax}. Thus, considering the CA and VC, the travel time of a loaded crane can be predicted with V_{Xmax}, V_{Ymax}, S_X, and S_Y. Meanwhile, the travel time of an unloaded crane can be predicted with V'_{Xmax}, V'_{Ymax}, S_X, and S_Y.

4.2 Single-crane trajectory prediction with TMs

First, the dynamics of a set of cranes and trolleys is studied. Typically, the transportation of cranes and trolleys can be considered in four parts without interference consideration: travel to the origin, loading at the origin, travel to the destination, and unloading at the destination. The necessary condition of loading/unloading is the arrival at the location of the crane and trolley. For this reason, the travel time is the maximum of t_X and t_Y. Consequently, the travel time can be defined as

$$t = max(t_X, t_Y) \tag{8}$$

Figure 3 shows an example of a single-crane trajectory. In Fig. 3, P, P_1 and P_2 are a dwell point, the origin, and the destination in the X direction, respectively. As shown in Fig. 3, the crane's trajectory from A to B is relatively sharp as compared to the trajectory from C to D. This condition can be attributed to two reasons: First, V_{Xmax} can be reached during the

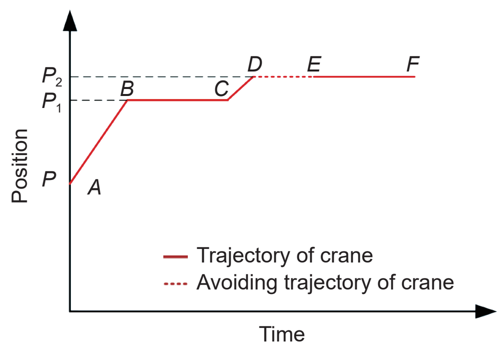


Fig. 3 Example of trajectory of one crane with idle time.

travel from A to B, but it is not reached during the travel from C to D. Therefore, the crane shows a lower average velocity from C to D than that from A to B. Second, the crane is unloaded during the travel from A to B, whereas it is loaded during the travel from C to D, and a loaded travel results in a lower V_{Xmax} than that on an unloaded travel. Based on Eq. (8), a crane idle time arises when t_X is less than t_Y. For this reason, the crane's idle time is indicated by segment D to E in Fig. 3, which is caused by the waiting time for the trolley.

With the aforementioned analysis, the trajectories of one crane can be predicted with the procedure shown in Fig. 4. During the prediction, each task is divided into two travel parts and two loading/unloading parts. In Fig. 4, T_1 and T_3 can be calculated with Eq. (6), and T_2 and T_4 can be calculated with Eq. (7). The crane idle time is predicted with $(T_2 - T_1)$ and $(T_4 - T_3)$.

4.3 Twin-crane trajectory prediction with interference handling

With the single-crane trajectory prediction procedure, a twin-crane trajectory can be predicted, and interference can be determined by the intersection of two trajectories. Different from former studies, a crane idle time creates a long stay somewhere in the pathway and

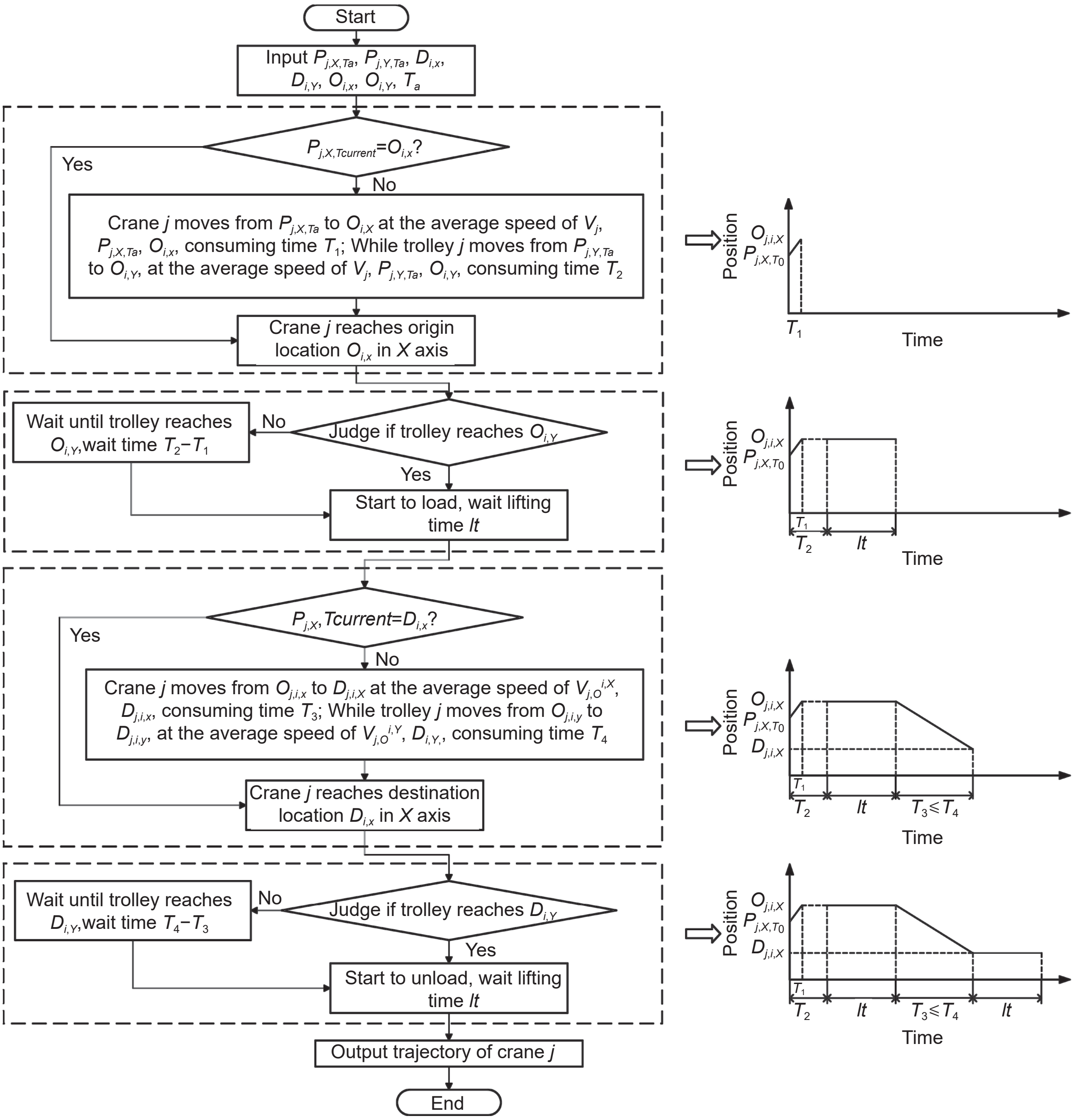


Fig. 4 Procedure of a single-crane trajectory prediction.

may create new intersections in twin-crane trajectories. Figure 5a shows an example of a twin-crane trajectory without idle time and interference. Figure 5b shows a new interference created by the crane idle time. Figure 5 illustrates the dynamics with the consideration of the movement of the crane and trolley. Meanwhile, it indicates the necessity of movement integration.

Due to the dynamic behavior of interference, it is critical to perform an interference detection with a twin-crane trajectory prediction. However, two questions need to be answered before interference can be detected:

(1) When should the interference detection be conducted?

(2) How long should the interference detection cover?

In this study, the moment when one crane will start a new travel task is selected as the interference detection time and denoted as t_a. The reason is that one travel task is divided into two travel parts and two loading/unloading parts. The trajectories of the four parts can be accurately predicted, and interference detection can be conducted. For this reason, the detection interval is the duration of a travel task. However, it is important to emphasize that each crane should conduct its own interference detection and complete travel, respectively. Therefore, the detection interval is the minimum detection interval of the two cranes, which is denoted as ΔT in Section 3.

With t_a and ΔT, interference handling can be conducted based on interference detection. The procedure of interference handling is illustrated in Fig. 6. As shown in Fig. 6, the simulation forwards a theoretical ΔT and detects interference in that period. The interference avoidance algorithm will be activated if interference is detected. Meanwhile the simulation will forward a factual ΔT if no interference is detected. Twin cranes will acquire their next allocated tasks separately and repeat the interference detection procedure again. The crane will move to the boundary of its working area to escape interference when it completes all its allocated tasks.

Figure 7 shows an example of interference handling. As shown in Fig. 7a, interference is detected in interval ΔT_1. To avoid a collision, crane 1 conducts the avoidance and lags its task travel. The simulation forwards ΔT_1 and detects interference in ΔT_2. The simulation directly forwards ΔT_2 because no interference is detected. However, interference is detected in ΔT_3 because the new task of the left crane still interferes with crane 2. Crane 2 conducts avoidance this time, and ΔT_3 is obtained due to the complete time of crane 2. The simulation keeps the detection interference and conducts avoidance until all tasks are completed. The final twin-crane trajectories are shown in Fig. 7b.

As shown in Fig. 7, how to conduct avoidance affects the subsequent trajectory and interference. As shown in Boysen and colleagues' work (2017)[2], two main steps are performed to conduct avoidance:

(1) Selecting which crane should conduct avoidance;

(2) Deciding how the selected crane can conduct avoidance.

Because the objective of this paper is not crane scheduling, we only employ heuristics in the two steps. As mentioned before, the cranes' loading and unloading statuses are considered in the travel profiles of a single crane. It is more difficult for a loaded crane to conduct avoidance than for an unloaded crane. This

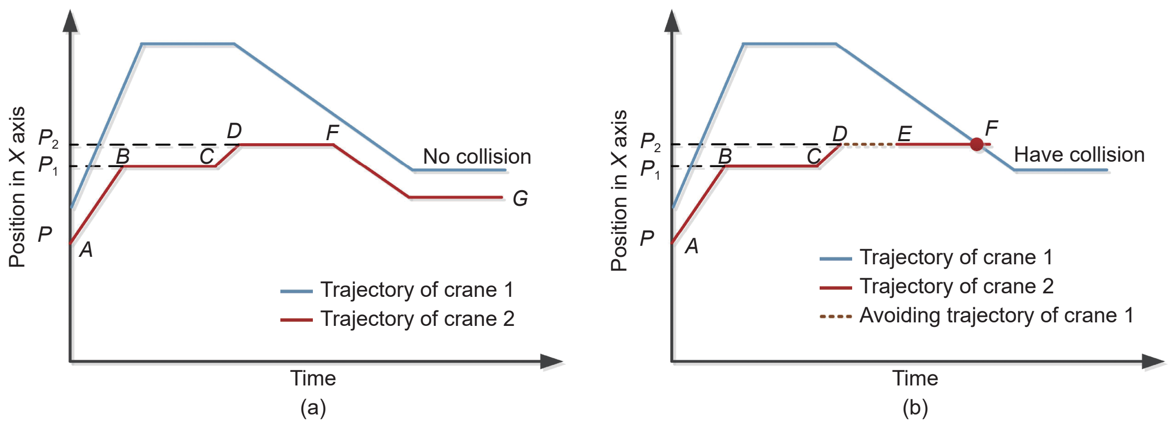


Fig. 5 Example of a twin-crane interference with idle time.

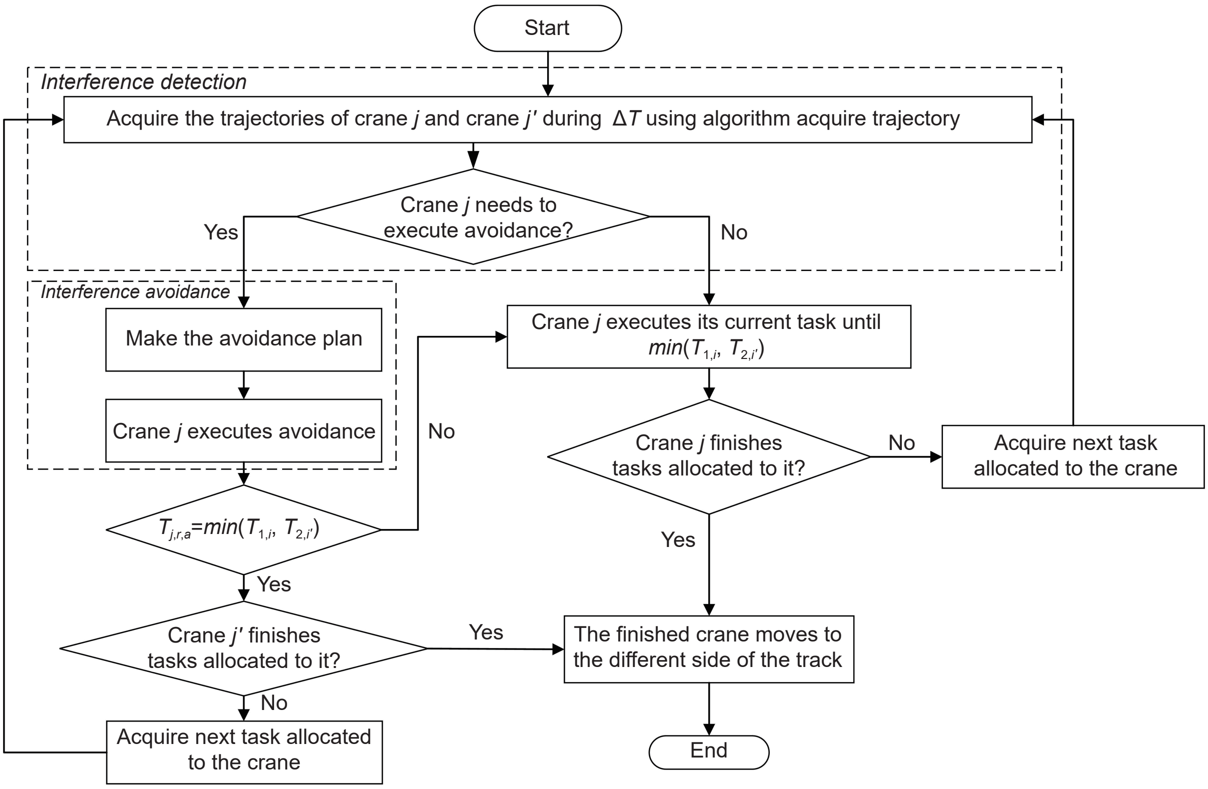


Fig. 6 Interference handling procedure.

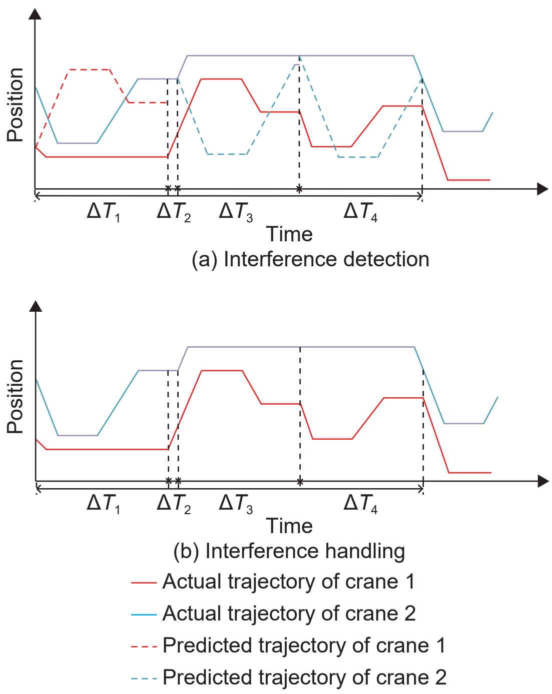


Fig. 7 Example to handle interference.

is caused by the higher inertia of the loaded crane. Therefore, conducting avoidance with the loaded crane may lead to an unnecessary waste of time and energy. For this reason, we select the unloaded crane to conduct avoidance. Another situation is that the two cranes are either loaded or unloaded. In this situation, we select the crane with less travel distance to conduct avoidance.

When deciding how the selected crane can conduct avoidance, two situations that result in different trajectories are considered. First, the avoiding crane must conduct avoidance travel (see Fig. 8). In this situation, the avoiding crane needs to move from the origin to an avoiding location apart from the priority crane by a safety distances. In Fig. 8, the avoiding location and avoiding time are calculated with the following three steps.

Step 1. The interference detection during the period of $T^{i,x}_{2,f,o}$ involves the arrival time of crane j with the i-th task at $O_{i,x}$. The interference is detected at $T_{collide,1}$.

Step 2. Because the destination of the i-th task in the X axis of crane 2 is greater than the origin of the i-th task in the X axis, i.e., $D_{i,2,x} > O_{i,2,x}$, for the first crane, the avoiding location $P_{1,X,a} = D_{i,2,x+s}$.

Step 3. According to the avoiding location, the avoiding time is $T^{i,x}_{2,f,D}$, which is the unloading completion time of crane 2 with the i-th task at $D_{i,x}$.

The other situation is where the avoiding crane does not need to travel but just waits at the origin (see Fig. 9).

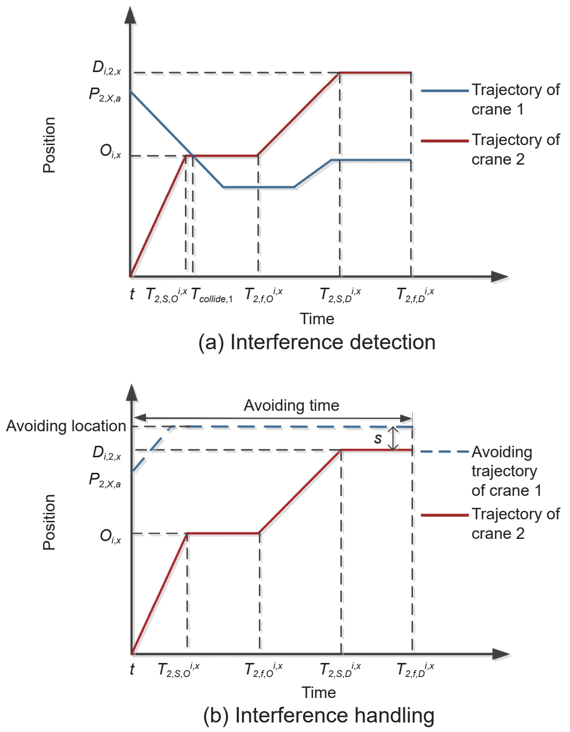


Fig. 8 Avoiding crane with avoidance travel.

In this situation, the avoiding time is the only parameter that affects the performance. There are two interference detections in this situation. The first detection is illustrated in Fig. 9a and the second detection is illustrated in Fig. 9b. The motivation for the second detection is to shorten the avoiding time. As shown in Fig. 9b, crane 1 can travel in parallel with crane 2 at $T^{i,X}_{2,f,O}$. Once the interference is detected, crane 1 keeps waiting until it is collision-free with crane 2 at $T^{i,X}_{2,f,D}$. In Fig. 9, the avoiding time can be calculated with the following steps:

Step 1. The first interference detection is called during the period of $T^{i,X}_{2,f,O}$, and the interference is detected at $T_{collide,1}$.

Step 2. The avoiding time is presumed as $T^{i,X}_{2,f,O}$, and the second interference detection is called between $T^{i,X}_{2,f,O}$ and $T^{i,X}_{2,f,D}$, where the interference is detected again.

Step 3. The avoiding time is presumed as $T^{i,X}_{2,f,D}$, and the avoiding crane is switched as the priority crane.

With the aforementioned situations, the interference is handled, and interference-free trajectories are obtained. Consequently, twin-crane trajectories can be predicted with interference handling.

In addition, when a crane stops for avoidance, its trolley could have simultaneously moved to the desired position of the next task to improve efficiency. Thus, in this situation, when the avoidance crane begins traveling, the travel time $t = max(t_X, max(t_Y - T_{2,f,D^{i,X}}, 0))$.

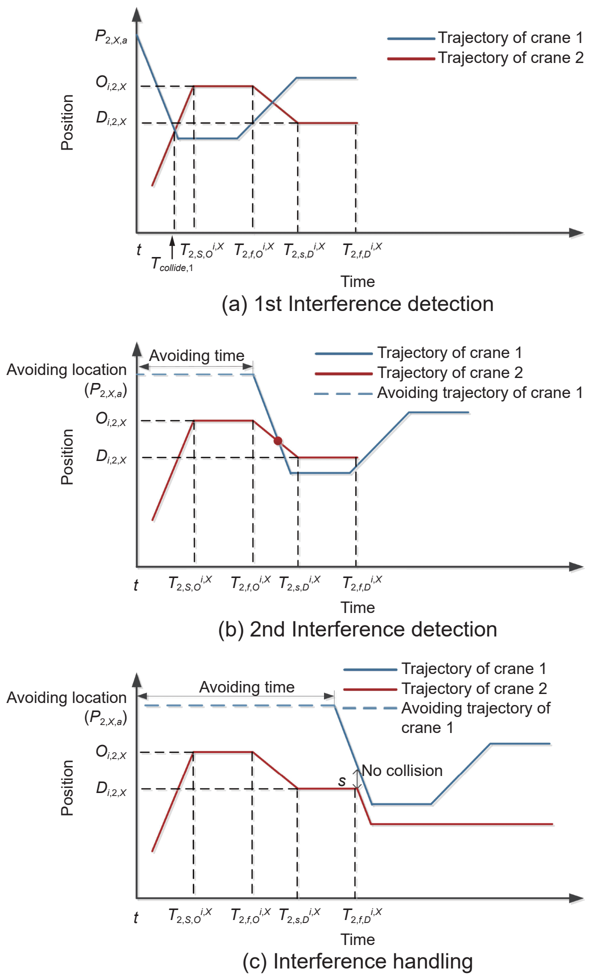


Fig. 9 Avoiding crane without avoidance travel.

5 Experiment

In a real twin-crane system, it is impossible to neglect the effect of acceleration/deceleration, TMs, and different velocities of a loaded/unloaded crane. Therefore, to examine the performance of aforementioned studies, a trajectory prediction simulation model was developed using the simulation software Tecnomatix Plant Simulation. A real crane system in a container yard is shown in Fig. 10, which consists of a crane, track, stack, and cargo. The simulation model was built using the layout of this container terminal. Three experiments with a twin-crane system in a real container terminal are presented. The interface of the simulation model in this case is shown in Fig. 11. All the experiments were conducted

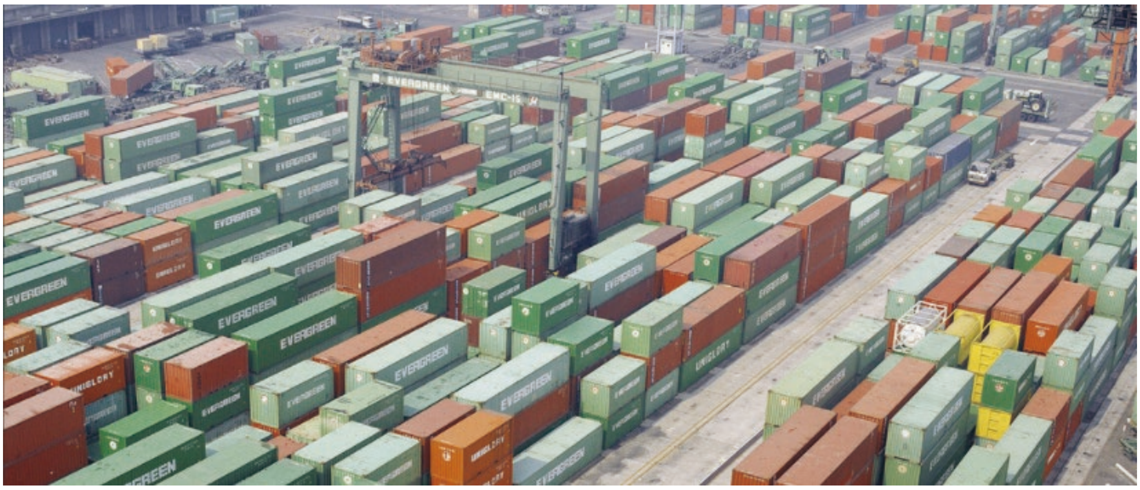


Fig. 10 Crane system in a container yard.

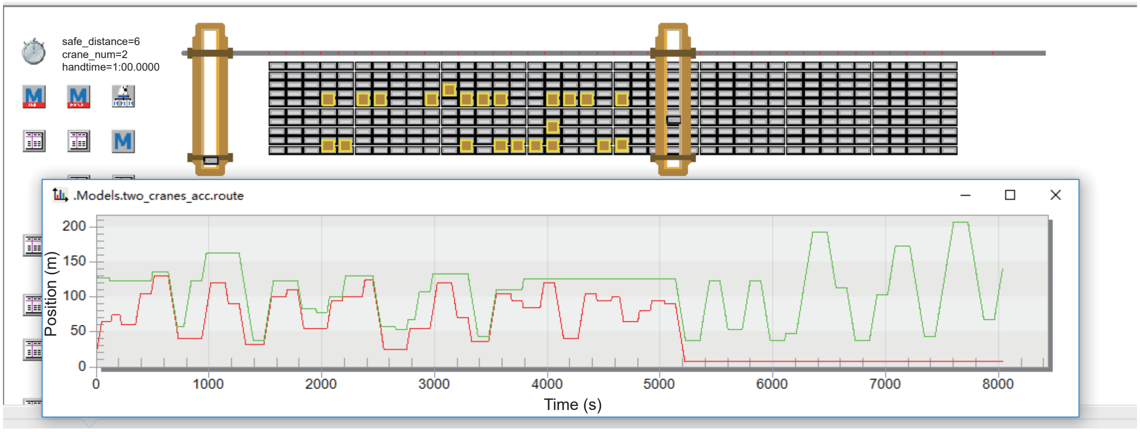


Fig. 11 Interface of the developed simulation model.

on a laptop with CPU Intel i5-4210H(2.9 GHz).

To investigate which crane movement details (CA, VC, and TMs) dominate the performance of a twin-crane system, we further developed eight parameterized models:

(1) Neglecting all detailed movements (NDM model);

(2) Considering trolley movements (CTM model);

(3) Considering crane acceleration/deceleration (CCA model);

(4) Considering different crane velocities when loading/unloading (CVC model);

(5) Considering trolley movements and crane acceleration/deceleration (CTM&CCA model);

(6) Considering trolley movements and different crane velocities when loading/unloading (CTM&CVC model);

(7) Considering crane acceleration/deceleration and different velocities when loading/unloading (CCA& CVC model);

(8) Considering all details (CDM model).

The parameters of the simulation models are given in Table 2, which are the same as thouse used in a real container terminal. The level of detail increases from models 1 to 8 evidently. In the eight models, the NDM model neglects all details, while the CDM model considers all details studied in this paper. As such, the CDM model is more realistic compared to the other models. For this reason, we take the CDM model as the benchmark and use the following formula to calculate the deviation of the others:

$$dev = [(makespan\ of\ the\ other\ model - makespan\ of\ CDM\ model)/(makespan\ of\ CDM\ model)] \times 100\% \quad (9)$$

5.1 Experiment 1: Performance in different detail levels

To compare the performance of the eight models, we assumed a scheduling scenario with 20 tasks, as shown

Table 2 Parameter table of different models.

Model	a_x (m/s^2)	V'_{Xmax} (m/s)	V_{Xmax} (m/s)	a_y (m/s^2)	V'_{Ymax} (m/s)	V_{Ymax} (m/s)
NDM	∞	1	1	∞	∞	∞
CTM	∞	1	1	∞	0.4	0.4
CCA	1	1	1	∞	∞	∞
CVC	∞	1.2	0.6	∞	∞	∞
CTM&CCA	1	1	1	0.5	0.4	0.4
CTM&CVC	∞	1.2	0.6	∞	0.5	0.3
CCA&CVC	1	1.2	0.6	∞	∞	∞
CDM	1	1.2	0.6	0.5	0.5	0.3

in Table 3. “ Original position” means the storage position where crane loads the container, “Destination” means the storage position where crane unloads the container, and “ Allocated crane” means the task is allocated to a specified crane. Three types of tasks are randomly distributed in this scenario. Type “in” denotes the crane travel to the input position and the carrying of the container to the corresponding destination. Type “out” denotes the crane travel to the container location and the carrying of the container to the output position. Type “move” denotes the crane travel to the old location of a container and carry the container to the new location. Each task has been allocated to one crane and is assumed to be the result of crane scheduling. The layout of the yard of this twin-crane system is shown in Fig. 12.

The model runs 50 000 times faster than the real-time performance of the real terminal. After the simulation, the crane trajectories corresponding with each model are shown in Fig. 13 and the makespan and calculation time are listed in Table 4.

Figure 13 shows the actual crane movements. As shown in Fig. 13, the shape of the trajectories is almost the same, whereas the makespans are different. CDM is taken as the benchmark because it considers all details. As shown in Table 4, there is a 25.03% difference in makespans between the NDM and CDM models. This finding indicates that crane scheduling using an NDM scenario would perform 25.03% less than the crane scheduling using a CDM scenario. Moreover, the CTM model results in a makespan closest to the makespan of CDM, whereas NDM, CCA, and CVC show significant deviations. This finding shows that TMs are the most important crane movement detail. Meanwhile, CTM&CCA and CTM&CVC result in makespans that are also close to the makespan of CDM.

Table 3 Task list with 20 tasks.

Task number	Original position	Destination	Type	Allocated crane
1	123	Exit	Out	2
2	Entrance	234	In	1
3	Entrance	302	In	1
4	205	120	Move	1
5	Entrance	167	In	1
6	150	155	Move	1
7	130	Exit	Out	2
8	188	87	Move	2
9	160	135	Move	1
10	168	Exit	Out	2
11	208	155	Move	2
12	180	85	Move	1
13	162	165	Move	1
14	Entrance	80	In	1
15	167	Exit	Out	2
16	195	106	Move	2
17	144	120	Move	1
18	Entrance	168	In	1
19	115	160	Move	1
20	210	120	Move	1

In addition, Table 4 shows the computation efficiency of each model. NDM and CCA reduce the computation time by approximately 30% compared to CDM. This indicates the advantage of NDM on computation efficiency. However, CTM only reduces the computation time by 1.95% compared to CDM. CTM&CCA and CTM&CVC have computation times very close to those of CDM. Thus, although TMs are the most important crane movement detail, they contribute less in terms of computation efficiency.

5.2 Experiment 2: Performance in different task ratios

To check the generality of the results from experiment 1, six groups of scenarios with different task ratios were

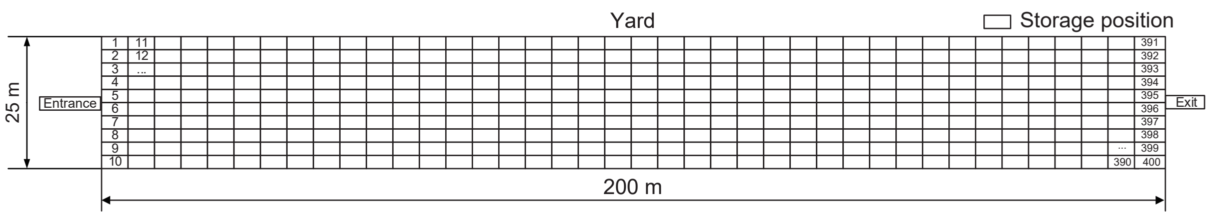


Fig. 12 Layout of the yard of the case study.

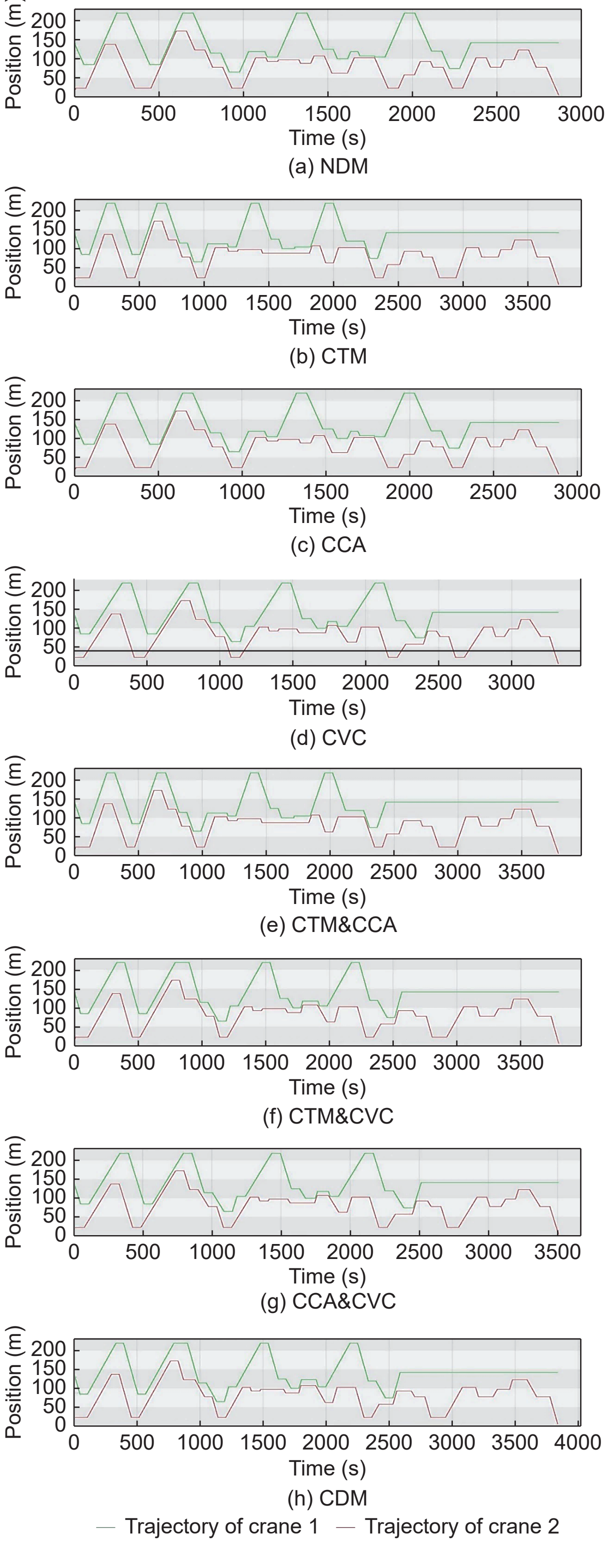


Fig. 13 Trajectories computed by the eight different detail models.

studied. Each group contains ten scenarios with randomly generated task locations. Simulations were conducted with these groups of scenarios, and the simulation results are shown in Table 5.

As shown in Table 5, the average deviation between the NDM and CDM models are more than 20%. Hence, a large deviation exists in the results of the NDM model, which proves the necessity of crane movement detail consideration. Among these details, CTM offers more than 90% accuracy, and TMs are the most important crane movement detail. The combination of crane movement details always shows more accuracy than using a single-crane movement detail. CTM&CVC offers more than 99% accuracy, whereas CTM&CCA offers more than 90% accuracy. Hence, CA is the less important crane movement detail. By contrast, CTM&CVC only reduces the computation time by 0.78%, and CTM makes an 11.62% difference as compared to CDM. Hence, TMs are the most time-consuming crane movement detail, whereas CA is the least one. The conclusions on the model performance differ when considering accuracy or efficiency.

5.3 Experiment 3: Performance in different layouts

To check the efficiency of the proposed model, experiments with different container terminal layouts were performed. Ten groups of scenarios with different layouts were designed. Five groups have a different number of columns compared to the original layout, and five others have a different number of rows. Each group contains five scenarios with randomly generated task locations.

The average deviation of makespans in a layout with different column numbers is shown in Table 6. The max deviation, min deviation, and average deviation are shown in Fig. 14.

The average deviation of the makespans in a layout with different row numbers is shown in Table 7. The max deviation, min deviation, and average deviation are shown in Fig. 15.

The results of the experiments show that the CDM model proposed in this paper still has the best efficiency in different layouts. The completion time considerably varies with different layouts, whereas the deviation is similar. For the NDM model, there is approximately a 25% deviation compared to the CDM model. The CTM and CVC models show a similar accuracy, but the CVC model has a larger bias. The CCA model shows the least accuracy regardless of the layout. The CTM&CVC model can provide high accuracy. Therefore, to obtain high accuracy, it is

Table 4 Completion of the makespans of eight models.

Model	Makespan	Deviation (%)	Calculation time (×10^{-7} s)	Deviation (%)
NDM	47 min 45 s	−25.03	6.632	−33.39
CTM	1 h 2 min 29 s	−1.91	8.678	−1.95
CCA	48 min 8 s	−24.42	6.686	−32.31
CVC	55 min 12 s	−13.34	7.667	−15.39
CTM&CCA	1 h 3 min 3 s	−1.02	8.756	−1.03
CTM&CVC	1 h 3 min 16 s	−0.69	8.786	−0.69
CCA&CVC	58 min 28 s	−8.22	8.12	−8.96
CDM	1 h 3 min 42 s	0.00	8.847	0.00

Table 5 Statistical table of the makespans.

Task group	NDM	CTM	CCA	CVC	CTM&CCA	CTM&CVC	CCA&CVC	CDM
1	47 min 45 s	1 h 2 min 29 s	48 min 8 s	55 min 12 s	1 h 3 min 3 s	1 h 3 min 16 s	58 min 28	1 h 3 min 42 s
2	1 h 26 s	1 h 19 min 33 s	1 h 51 s	1 h 12 min 48 s	1 h 20 min 17 s	1 h 24 min 31 s	1 h 16 min 54 s	1 h 25 min 17 s
3	1 h 7 min 17 s	1 h 18 min 8 s	1 h 7 min 45 s	1 h 18 min 14 s	1 h 18 min 52 s	1 h 28 min 33 s	1 h 21 min 28 s	1 h 29 min 9 s
4	53 min 24 s	1 h 4 min 40 s	53 min 49 s	1 h 2 min 5 s	1 h 5 min 22 s	1 h 8 min 59 s	1 h 4 min 20 s	1 h 9 min 32 s
5	1 h 4 min 50 s	1 h 25 min 51 s	1 h 5 min 23 s	1 h 15 min 37 s	1 h 26 min 57 s	1 h 37 min 44 s	1 h 22 min 41 s	1 h 38 min 51 s
6	50 min 26 s	53 min 44 s	50 min 48 s	53 min 7 s	54 min 11 s	57 min 40 s	55 min 46 s	58 min 7 s
7	1 h 21 s	1 h 7 min 6 s	1 h 53 s	1 h 11 min 14 s	1 h 7 min 47 s	1 h 25 min 22 s	1 h 16 min 9 s	1 h 26 min 15 s
8	57 min 50 s	1 h 7 min 36 s	58 min 14 s	1 h 4 min 5 s	1 h 8 min 10 s	1 h 11 min 26 s	1 h 6 min 30 s	1 h 11 min 10 s
9	1 h 6 min 7 s	1 h 32 min 59 s	1 h 6 min 37 s	1 h 15 min 2 s	1 h 33 min 54 s	1 h 42 min 30 s	1 h 217 s	1 h 43 min 23 s
10	1 h 16 min 28 s	1 h 35 min 39 s	1 h 17 min 1 s	1 h 29 min 16 s	1 h 36 min 40 s	1 h 48 min 14 s	1 h 35 min 36 s	1 h 49 min 14 s
Avg dev of makespan	−24.03%	−8.94%	−23.53%	−14.19%	−8.14%	−0.67%	−9.79%	0.00%
Avg dev of calculation time	−37.98%	−11.62%	−36.90%	−19.81%	−10.52%	−0.78%	−13.08%	0.00%

Table 6 Statistics table of deviation in different column numbers (row = 10).

Column	Deviation(%)							
	NDM	CTM	CCA	CVC	CTM&CCA	CTM&CVC	CCA&CVC	CDM
20	−25.31	−9.57	−22.84	−16.32	−8.29	−1.56	−11.04	0.00
30	−25.56	−12.19	−24.93	−13.44	−11.08	−1.74	−10.47	0.00
40	−25.78	−10.67	−25.26	−14.19	−9.69	−0.97	−8.38	0.00
50	−27.28	−11.93	−26.61	−13.76	−11.37	−1.16	−8.72	0.00
60	−25.64	−15.66	−25.22	−10.50	−15.08	−0.71	−6.84	0.00
Min	−25.31	−9.57	−22.84	−10.50	−8.29	−0.71	−6.84	0.00
Max	−27.28	−15.66	−26.61	−16.32	−15.08	−1.74	−11.04	0.00
Average	−25.92	−12.00	−24.97	−13.64	−11.10	−1.23	−9.09	0.00

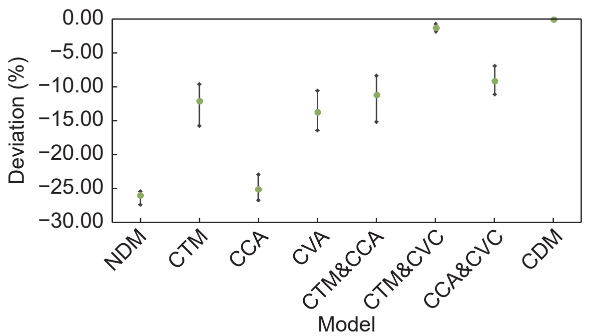


Fig. 14 Makespan deviation in different column numbers of different models.

necessary to consider the trolley's movements and the speed difference of the crane in the loaded/unloaded state when conducting research related to the prediction and conflict resolution of the crane trajectory.

6 Conclusion

In this study, we propose a novel trajectory prediction approach with crane movement details. The objective is to offer accurate trajectory predictions of the movements of twin-crane systems. The details include

Table 7 Statistics table of deviations in different row numbers (column = 40).

Row	Deviation (%)							
	NDM	CTM	CCA	CVC	CTM&CCA	CTM&CVC	CCA&CVC	CDM
6	−25.29	−11.97	−24.73	−14.19	−11.32	−1.12	−9.19	0.00
8	−25.47	−12.21	−24.93	−10.17	−11.36	−0.83	−8.01	0.00
10	−25.78	−10.67	−25.26	−14.19	−9.69	−0.97	−8.38	0.00
12	−26.50	−12.24	−25.97	−17.76	−11.32	−1.06	−12.36	0.00
14	−25.50	−11.39	−25.01	−13.77	−10.48	−1.15	−9.14	0.00
Min	−25.29	−10.67	−24.73	−10.17	−9.69	−0.83	−8.01	0.00
Max	−26.50	−12.24	−25.97	−17.76	−11.36	−1.15	−12.36	0.00
Average	−25.71	−11.70	−25.18	−14.02	−10.83	−1.02	−9.42	0.00

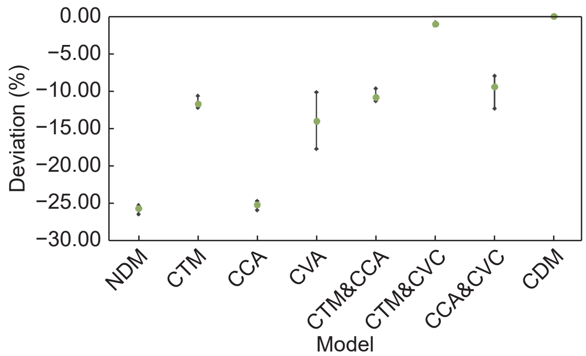


Fig. 15 Makespan deviation in different row numbers of different models.

TMs, CA, and different crane velocities corresponding with loading and unloading (VC). A single-crane trajectory prediction approach with CA, VC, and TMs was studied. Consequently, a twin-crane trajectory prediction approach with interference handling was studied. Based on the theoretical studies, we developed a simulation model and presented a case study on the container terminal. According to the simulation results, the proposed trajectory prediction performs 20% better in terms of accuracy compared to the traditional approach without crane movement details.

In the future, we would like to further our study by evaluating the dynamic energy consumption of a twin-crane system, which is another interesting problem and may help to develop an energy-efficient crane scheduling approach.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 52075036).

References

[1] T. Zhang and O. Rose, Simulation-based overhead-crane scheduling for a manufacturing plant, in *Proc. the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, Washington, DC, USA, 2013, pp. 2633–2642.

[2] N. Boysen, D. Briskorn, and F. Meisel, A generalized classification scheme for crane scheduling with interference, *European Journal of Operational Research*, vol. 258, no. 1, pp. 343–357, 2017.

[3] B. Peterson, I. Harjunkoski, S. Hoda, and J. N. Hooker, Scheduling multiple factory cranes on a common track, *Computers & Operations Research*, vol. 48, pp. 102–112, 2014.

[4] P. Legato and R. Trunfio, A local branching-based algorithm for the quay crane scheduling problem under unidirectional schedules, *4OR*: *A Quarterly Journal of Operations Research*, vol. 12, pp. 123–156, 2014.

[5] X. Cheng, L. X. Tang, and P. M. Pardalos, A branch-and-cut algorithm for factory crane scheduling problem, *Journal of Global Optimization*, vol. 63, pp. 729–755, 2015.

[6] L. Moccia, J. F. Cordeau, M. Gaudioso, and G. Laporte, A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal, *Naval Research Logistics*, vol. 53, no. 1, pp. 45–59, 2006.

[7] N. Boysen, D. Briskorn, and S. Emde, A decomposition heuristic for the twin robots scheduling problem, *Naval Research Logistics*, vol. 62, no. 1, pp. 16–22, 2015.

[8] D. Briskorn and P. Angeloudis, Scheduling co-operating stacking cranes with predetermined container sequences, *Discrete Applied Mathematics*, vol. 201, pp. 70–85, 2016.

[9] I. Aron, L. Genç-Kaya, I. Harjunkoski, S. Hoda, and J. N. Hooker, Factory crane scheduling by dynamic programming, http://public.tepper.cmu.edu/jnh/dpcrane ICS2post.pdf, 2010.

[10] Y. H. Kung, Y. Kobayashi, T. Higashi, M. Sugi, and J. Ota, Order scheduling of multiple stacker cranes on common rails in an automated storage/retrieval system, *International Journal of Production Research*, vol. 52, no. 4, pp. 1171–1187, 2014.

[11] T. Park, R. Choe, S. M. Ok, and K. R. Ryu, Real-time scheduling for twin RMGs in an automated container yard, *Or Spectrum*, vol. 32, pp. 593–615, 2010.

[12] O. A. Kasm and A. Diabat, The quay crane scheduling

problem with non-crossing and safety clearance constraints: An exact solution approach, *Computer & Operations Research*, vol. 107, pp. 189–199, 2019.

[13] X. C. Chen, S. W. He, Y. X. Zhang, L. Tong, P. Shang, and X. S. Zhou, Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework, *Transportation Research Part C Emerging Technologies*, vol. 114, pp. 241–271, 2020.

[14] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions, *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 718–731, 2019.

[15] L. J. He, W. F. Li, Y. Zhang, and Y. L. Cao, A discrete multi-objective fireworks algorithm for flowshop scheduling with sequence-dependent setup times, *Swarm and Evolutionary Computation*, vol. 51, no. 1, p. 100575, 2019.

[16] W. Li, L. He, and Y. Cao, Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers, *IEEE Transactions on Cybernetics*, doi: 10.1109/TCYB.2021.3069184.

[17] Y. Luo, W. Li, W. Yang, and G. Fortino, A real-time edge scheduling and adjustment framework for highly customizable factories, *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5625–5634, 2021.

[18] L. J. He, R. Chiong, W. F. Li, S. Dhakal, Y. L. Cao, and Y. Zhang, Multiobjective optimization of energy-efficient JOB-shop scheduling with dynamic reference point-based fuzzy relative entropy, *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 600–610, 2022.

[19] N. Al-Dhaheri, A. Jebali, and A. Diabat, A simulation-based genetic algorithm approach for the quay crane scheduling under uncertainty, *Simulation Modelling Practice and Theory*, vol. 66, pp. 122–138, 2016.

[20] D. F. Chang, T. Fang, and Y. Q. Fan, Dynamic rolling strategy for multi-vessel quay crane scheduling, *Advanced Engineering Informatics*, vol. 34, pp. 60–69, 2017.

[21] S. H. Chung and F. T. S. Chan, A workload balancing genetic algorithm for the quay crane scheduling problem, *International Journal of Production Research*, vol. 51, no. 16, pp. 4820–4834, 2013.

[22] S. H. Chung and K. L. Choy, A modified genetic algorithm for quay crane scheduling operations, *Expert Systems with Applications*, vol. 39, no. 4, pp. 4213–4221, 2012.

[23] J. F. Correcher and R. Alvarez-Valdes, A biased random-key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem, *Expert Systems with Applications*, vol. 89, no. C, pp. 112–128, 2017.

[24] S. Emde and N. Boysen, One-dimensional vehicle scheduling with a front-end depot and non-crossing constraints, *Or Spectrum*, vol. 36, no. 2, pp. 381–400, 2014.

[25] Y. M. Fu, A. Diabat, and I. T. Tsai, A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach, *Expert Systems with Applications*, vol. 41, no. 15, pp. 6959–6965, 2014.

[26] M. H. Hakam, W. D. Solvang, and T. Hammervoll, A genetic algorithm approach for quay crane scheduling with noninterference constraints at Narvik container terminal, *International Journal of Logistics Research and Applications*, vol. 15, no. 4, pp. 269–281, 2012.

[27] Z. H. Hu, J. B. Sheu, and J. X. Luo, Sequencing twin automated stacking cranes in a block at automated container terminal, *Transportation Research Part C*: *Emerging Technologies*, vol. 69, pp. 208–227, 2016.

[28] N. Kayeshgar, N. Huynh, and S. K. Rahimian, An efficient genetic algorithm for solving the quay crane scheduling problem, *Expert Systems with Applications*, vol. 39, no. 18, pp. 13108–13117, 2012.

[29] L. X. Wu and W. M. Ma, Quay crane scheduling with draft and trim constraints, *Transportation Research Part E*: *Logistics and Transportation Review*, vol. 97, pp. 38–68, 2017.

[30] R. Choe, H. Yuan, Y. Yang, and K. R. Ryu, Real-time scheduling of twin stacking cranes in an automated container terminal using a genetic algorithm, in *Proc. the 27th Annual ACM Symposium on Applied Computing*, Trento, Italy, 2012, pp. 238–243.

[31] A. Skaf, S. Lamrous, Z. Hammoudan, and M. -A. Manier, Integrated quay crane and yard truck scheduling problem at port of Tripoli-Lebanon, *Computer and Industrial Engineering*, vol. 159, no. C, p. 107448, 2021.

[32] W. C. Ng, K. L. Mak, and W. S. Tsang, Scheduling yard crane in a port container terminal using genetic algorithm, *International Journal of Industrial Engineering*, vol. 13, no. 3, pp. 246–253, 2006.

[33] P. Ge, J. Wang, M. Z. Jin, J. Y. Ren, and H. F. Gao, An efficient heuristic algorithm for overhead cranes scheduling operations in workshop, *Applied Mathematics & Information Sciences*, vol. 6, no. 3, pp. 1087–1094, 2012.

[34] H. J. Carlo and I. F. A. Vis, Sequencing dynamic storage systems with multiple lifts and shuttles, *International Journal of Production Economics*, vol. 140, no. 2, pp. 844–853, 2012.

[35] H. J. Carlo and F. L. Martínez-Acevedo, Priority rules for twin automated stacking cranes that collaborate, *Computers & Industrial Engineering*, vol. 89, pp. 23–33, 2015.

[36] A. H. Gharehgozli, F. G. Vernooij, and N. Zaerpour, A simulation study of the performance of twin automated stacking cranes at a seaport container terminal, *European Journal of Operational Research*, vol. 261, no. 1, pp. 108–128, 2017.

[37] J. H. Chen, D. H. Lee, and J. X. Cao, Heuristics for quay crane scheduling at indented berth, *Transportation Research Part E*: *Logistics and Transportation Review*, vol. 47, no. 6, pp. 1005–1020, 2011.

[38] J. Li, A. J. Xu, and X. S. Zang, Simulation-based solution for a dynamic multi-crane-scheduling problem in a

steelmaking shop, *International Journal of Production Research*, vol. 58, no. 22, pp. 6970–6984, 2020.
[39] N. Zhao, L. Luo, and G. Lodewijks, Scheduling two lifts on a common rail considering acceleration and deceleration in a shuttle based storage and retrieval system, *Computers & Industrial Engineering*, vol. 124, pp. 48–57, 2018.
[40] M. Dawande, C. Sriskandarajah, and S. Sethi, On throughput maximization in constant travel-time robotic cells, *Manufacturing & Service Operations Management*, vol. 4, no. 4, pp. 296–312, 2002.
[41] H. N. Geismar, M. Pinedo, and C. Sriskandarajah, Robotic cells with parallel machines and multiple dual gripper robots: A comparative overview, *IIE Transactions*, vol. 40, no. 12, pp. 1211–1227, 2008.
[42] M. Grieves, *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management*. Cocoa Beach, FL, USA: Space Coast Press, 2011.
[43] T. H. J. Uhlemann, C. Lehmann, and R. Steinhilper, The digital twin: Realizing the cyber-physical production system for industry 4.0, *Procedia CIRP*, vol. 61, pp. 335–340, 2017.
[44] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, Digital twin in industry: State-of-the-art, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
[45] Y. L. Fang, C. Peng, P. Lou, Z. D. Zhou, J. M. Hu, and J. W. Yan, Digital-twin-based job shop scheduling toward smart manufacturing, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6425–6435, 2019.
[46] M. Zhang, F. Tao, and A. Y. C. Nee, Digital twin enhanced dynamic job-shop scheduling, *Journal of Manufacturing Systems*, vol. 58, pp. 146–156, 2021.
[47] T. Yang, N. Sun, and Y. Fang, Adaptive fuzzy control for a class of MIMO underactuated systems with plant uncertainties and actuator deadzones: Design and experiments, *IEEE Transactions on Cybernetics*, doi: 10.1109/TCYB.2021.3050475.

Ning Zhao received the BS and MS degrees in mechanical engineering from Shandong University, Jinan, China, in 1999 and 2002, respectively, and the PhD degree in mechanical engineering from Beijing Institute of Technology, Beijing, China. He is now a professor in University of Science and Technology Beijing, Beijing, China. His research interests involve simulation modelling and scheduling of logistics system, factory planning and optimization, and digital twin application in manufacturing and logistics system.

Gabriel Lodewijks received the BS degree in transport engineering and logistics from University of Twente, the Netherlands in 1990, the MS degree in transport engineering and logistics from Delft University of Technology, the Netherland in 1992, and the PhD degree in dynamics of transportation systems from Delft University of Technology, the Netherland in 1996. He has been a full professor and the head of School of Aviation of the University of New South Wales, Sydney, Australia, since 2017. He also works as a visiting/guest/chair professor at the University of Witwatersrand, South Africa, Wuhan University of Technology, University of Science and Technology Beijing, China University of Mining and Technology, all in China, and Newcastle University, Australia. He has over 300 publications including 2 books and 140+ journal papers. His research interests are optimization of maintenance, repair, and overhaul processes, automation of air cargo handling systems, tracking and tracing of equipment, components, and people at airports and in aviation related companies, optimization of gate processes and baggage handling procedures to reduce the turnaround time of aircraft, maintaining safety and security in airport logistic processes, and the improvement of passenger experience by streamlining airport logistics.

Zhuorui Fu received the BS degree from University of Science and Technology Beijing, China in 2020. He is currently pursuing the PhD degree at School of Mechanical Engineering in University of Science and Technology Beijing. His main research interests include system simulation and optimization, intelligent manufacturing, and digital twin.

Yu Sun received the MS degree from University of Science and Technology Beijing, China in 2021. Her main research interests include system simulation, scheduling, and optimization.

Yue Sun received the MS degree from University of Science and Technology Beijing, China in 2019. Her main research interests include system simulation, scheduling, and optimization.