Computers & Industrial Engineering 125 (2018) 649–657

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Research on coupling scheduling of quay crane dispatch and configuration in the container terminal

Chengji Liang[a], Lubin Fan[b,*], Dehong Xu[c], Yi Ding[a], Mitsuo Gen[d]

[a] Logistics Research Center, Shanghai Maritime University, Shanghai, China
[b] Shanghai International Port Group Co., Ltd, Shanghai, China
[c] Logistics Institute, Xi'an International University, Xi'an, Shaanxi, China
[d] Fuzzy Logic Systems Institute, Tokyo University of Science, Tokyo, Japan

ARTICLE INFO

Keywords:
Container terminal
Quay crane
Coupling scheduling
Task dispatch
Quantity configuration

ABSTRACT

This paper studies the quay crane scheduling problem (QCSP) from two aspects: the task dispatch and quantity configuration of quay cranes. Quay crane task dispatch problem is mainly concerned with the study of handling sequence of quay cranes, while the quay crane configuration problem is used to determine how many quay cranes should be configured to the vessels. Through the analysis of the relationship between the two problems, a coupling model is established. The optimal number of quay cranes and corresponding scheduling scheme can be obtained by solving the coupling model. Compared to the objective and approach of QCSP found in literature, this paper takes the optimal number of quay cranes into account and designs a loop iteration method to solve the coupling model.

1. Introduction

In recent years, with the development of economy and trade in China, the demand of shipping market is increasing. Container terminals' logistics pressure is getting bigger and bigger, thus there is an urgent need to establish a smooth and efficient container transportation system. Because of the scarcity of berth resources, container terminals try to improve the utilization efficiency of berth by improving loading and unloading efficiency and reducing the berthing time of vessels. Quay crane (QC) is an important equipment in container terminal which provides loading and unloading operations for containers vessels. At the same time, QC is a kind of expensive equipment, since its running and maintenance costs are high. Therefore how to finish the loading and unloading operations with less time and less number of QCs is the key to the problem of the QC scheduling problem.

Efficient planning of seaside operations has a direct impact on the dwell time of the vessels (Türkoğulları, kın, Aras, & Altınel, 2016), so many literatures have carried out extensive research into this problem. According to Bierwirth and Meisel (2010), the seaside operations planning in container terminal involves three parts, namely berth allocation problem (BAP), quay crane assignment problem (QCAP) and the quay crane scheduling problem (QCSP). BAP addresses the assignment of quay space and service time to vessels that need to be loaded and unloaded. QCAP researches the assignment of QCs to vessels and QCSP researches the determination of work plans for the QCs.

Berths and QCs are important resources in container terminal. They are basically interrelated, so many literatures believe that BAP and QCAP should be studied together when seeking to optimize quayside operation plan. In Yang, Wang, and Li (2012), BAP and QCAP were proposed, which got the idea from the fact that berth allocation and QC assignment is coupled. Meanwhile, a formulation is conducted by using the vessel handling time as a coupling variable. In Elwany, Ali, and Abouelseoud (2013), an integrated heuristics-based solution methodology is proposed that tackles both BAP and QCAP simultaneously. Hu, Hu, and Du (2014) also researched the two problems which considered a vessel's fuel consumption and emissions. Li, Sheu, and Gao (2015) considered the restricted coverage range of QCs and the available QCs-to-vessel workloads, and analyzed the relationship between berth positions of vessels and QCs coverage range. Karam and Eltawil (2016) addressed the integration of the BAP, the QCAP and the specific quay crane assignment problem (SQCAP). The SQCAP follows the QCAP and determines the specific quay cranes to serve each vessel.

In the QCAP, according to Karam and Eltawil (2016) and actual operation of container terminal, the number of QCs is usually determined by the number of containers to be handled, the handling time of vessel and the average productivity (container moves/hour) of QCs. That is to say, the number of containers and the handling time limits are usually known, we can easily get the number of containers needed to be

* Corresponding author at: Shanghai International Port Group Co., Ltd, SIPG Tower 9F, 358 East Daming Road, Shanghai 200080, China.
E-mail address: helloflb@yeah.net (L. Fan).

https://doi.org/10.1016/j.cie.2018.05.004

Available online 05 May 2018
0360-8352/

handled per hour, and then the QC quantity can be calculated through the number of containers to be handled per hour divided by the productivity of QC. In actual operation of container terminal, they also use this simple method when making the QC scheduling. Actually there is no way to know whether the quantity of equipment is optimal. It's also able to complete all tasks on time even if decrease one or two QCs. And sometimes increasing one or two QCs will not accomplish all tasks ahead of time. There is no method to test the optimal quantity of QCs in a certain period of time, and this point provides a motivation of this research.

Besides BAP and QCAP, quayside planning also includes QCSP. The basic QCSP has been intensively studied and many solving methods have been proposed. Al-Dhaheri and Diabat (2015) developed a novel simple formulation to solve the QCSP by minimizing the differences between the container loads stacked over a number of bays and by maintaining a balanced load across the bays. Kenan and Diabat (2015) addressed a branch-and-price algorithm to effectively solve the QCSP with very large instances. Chen, Lee, and Goh (2014) focused on the study of a special strategy for the cluster-based QCs scheduling problem that forces QCs to move unidirectionally during the scheduling. Liu, Zheng, and Zhangb (2016) focused on the rescheduling problem dealing with the disruption that a quay crane breakdowns unexpectedly in the middle of the execution of a planned schedule, which aimed to reschedule the system with the objective to minimize their negative deviation from the originally planned schedule. The problem of scheduling QCs with non-crossing constraints was studied by Zhang, Zhang, Chen, Chen, and Chen (2017), wherein QCs cannot cross over each other because they are on the same track. Al-Dhaheri, Jebali, and Diabat (2016) proposed a novel MIP model for QCSP that takes into account vessel stability constraints. Meanwhile, several practical features of the problem such as preemption, non-crossing, safety margin, QC traveling time and QC initial position were considered in the proposed model. Lee, Liu, and Chu (2014) researched the QC double-cycling problem (QCDCP), and they proved the QCDCP can be formulated as a two-machine flow shop scheduling problem with series-parallel precedence constraints. Legato, Trunfio, and Meisel (2012) proposed a rich model for QCSP which covers practical issues like ready time, due dates for cranes, safety requirements and precedence relations among container groups. Nguyen, Zhang, Johnston, and Tan (2013) developed a hybrid evolutionary computation method for QCSP and they developed a new priority-based schedule construction procedure to generate QC schedules. Lu, Han, Xi, and Erera (2012) studied the QCSP based on single container operation. Türkoğulları et al. (2016) researched the integration of BAP, QCAP and QCSP. They first dealt with berthing positions of the vessels and the assigned numbers of cranes in each time period, and then obtained the optimal crane scheduling plan based on crane assignment. Diabat and Theodorou (2014) presented a formulation for the QCs assignment and scheduling Problem (QCASP), which take into account the crane positioning conditions. They also developed a Genetic Algorithm to solve the QCASP. Ji, Guo, Zhu, and Yang (2015) developed a mathematical model to integrate the loading sequence and the rehandling strategy under the circumstance of parallel operation of multi-QCs, which aimed to minimize the number of rehandles and obtain the optimal loading sequence. The uncertainty is naturally inevitable when planned schedules are executed in terminal system and has its influence on the actual performance of terminal operation. So Han, Lu, and Xi (2010) addressed berth and QCSP in a simultaneous way under uncertainties of vessel arrival time and container handling time.

The objective function of QCSP is always to minimize the makespan of the schedule according to Legato et al. (2012), Chen et al. (2014) and Al-Dhaheri et al. (2016), while Hu et al. (2014) and He, Huang, Yan, and Wang (2015) built the QCSP models to minimize the energy consumption. Since berth is a kind of scant resource, the container terminals are inclined to provide the fastest service to the vessels in order to increase the throughput of containers. Therefore, we take the minimization of makespan as an objective of QC scheduling in the paper.

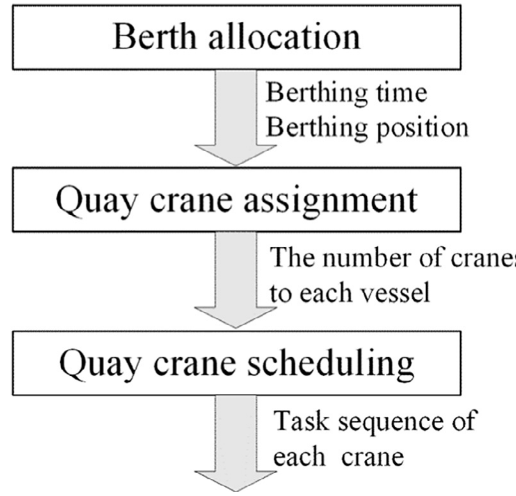

Fig. 1. Procedure of seaside operations planning.

QC is the key equipment in container terminal. Increasing the number of QCs will lead to different QC scheduling. For example, if we increase one QC to the vessel, it means we need to add a new handling route, we also need at least two or three trucks to serve for the new route, and even need to increase yard cranes. So irrational QC quantity to be assigned to vessels will result in increased costs. The increased costs mainly include: fuel cost, electric charge, maintenance costs of equipment, and labor costs of drivers. Moreover, increasing the number of QCs will increase fuel emissions, the container terminal need to pay more for the environmental governance. On the contrary, if we can use less QCs to handle containers within allowable time limit, the operation costs of container terminal will be cut down. Base on this point, we take the optimization of QC quantity as another objective of QC scheduling in this paper.

According to Bierwirth and Meisel (2010) and Yang et al. (2012), as well as the actual planning process of container terminal, the procedure for seaside operations planning is shown in Fig. 1.

(1) Berth allocation planning determines the berthing time and berthing position.
(2) Quay crane assignment planning determines the number of quay cranes to be assigned to each vessel.
(3) Quay crane scheduling planning makes determination of task sequences for each quay crane.

In the actual operations of the terminal, we believe the equipment quantity and the task dispatch scheme are in coupling relationship. Appropriate task dispatch scheme can not only improve the efficiency of equipment handling, but also improve the utilization rate of equipment. The task dispatch scheme and the equipment quantity are mutually influenced, as shown as Fig. 2. So we can't simply take the idea of one-way coordination which determines the equipment quantity first, and then calculates the optimal equipment dispatch scheme. The optimal scheme of the whole scheduling is not the optimal equipment quantity or the optimal task dispatch scheme, but the integration of both.

Based on the above analysis, this paper studies the QC scheduling problem from two aspects: QC task dispatch and QC quantity configuration, by using coupling method. The QC task dispatch problem is used to research the task sequence of each QC. And the QC quantity configuration problem mainly study how many QC should be configured for the tasks. According to this, the QCSP is transformed into a two-objective optimization problem with optimal QC task dispatch

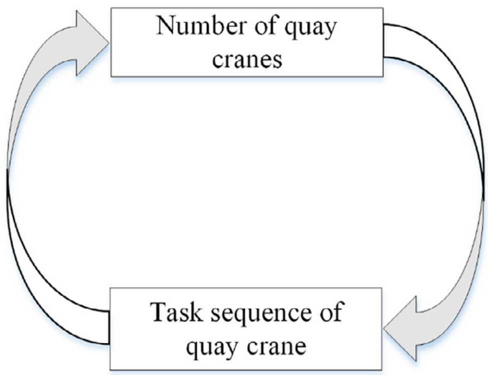

Fig. 2. Relationship between the number of QCs and it's task sequence.

scheme and optimal QC quantity.

The rest of this paper is organized as follow. Section 2 gives more details about QCs work plan and the framework of this paper. Section 3 and section 4 present the QC dispatch model and QC configuration model respectively. Section 5 is coordination algorithm for the two models above. Section 6 is the numerical experiment and improvement of QC configuration model, and we also made a cost-benefit analysis for QC handling is this section, which proved the importance of reducing QC quantity in cost saving. Section 7 is the conclusion.

2. Problem description

The container terminal assigns multiple QCs to vessels according to the number of container tasks, thus the makespan of all tasks depends on the makespan of the last QC. This paper proposes a dispatch model (Model 1) which aims to minimize the makespan of the last QC to finish its tasks, and a configuration model (Model 2) which aims to minimize the number of QCs. These models are built to optimize the number of QCs and scheduling of QCs.

Four assumptions were considered, including:

1. Berth allocation plan, the number of ships, and the number of tasks are all known;
2. The types of tasks at a certain period include loading and unloading;
3. All containers to be operated are 20 feet and the operation time of one container is set as a fixed value;
4. QCs are not allowed to cross each other and we keep the safety distance between QCs by forcing the QCs to move unidirectionally.

In the two sub-models, time windows of tasks are set as public variables. At the beginning of iteration, an initial number of QCs is set as the input of Model 1. Other inputs include handling type (loading/unloading) and location (bay) of each task. These two inputs can be got from stowage plan of each container vessel. The outputs of Model 1 are the earliest operation time of tasks handled by each QC, and the time windows of tasks can be obtained according to these outputs. Then time windows are set as inputs of Model 2, we can get the minimum number of QCs and the operation time of each task by the calculation of Model 2. We use the control coordinator to judge whether the results meet the constraints or not. If yes, we set the minimum number of QCs as the input of Model 1 and start a new round of iteration. Through the coupling of dispatch and configuration of QC, the optimal number of QCs and the corresponding scheduling plan can be obtained. The framework of coupling model is shown as Fig. 3.

3. QC dispatch model (Model 1)

In the actual scheduling, QC dispatch has two steps. The first step is to make decisions on which bays to be handed by each QC. This decision-making process should consider the problem of interference between QCs. QCs can't cross each other, and one bay can only has one QC in operation. In order to reduce the number of moves, QCs move to the next bay when all tasks are completed in a bay. The second step is to determine the handling sequence of QC within a bay. The dual-cycle operations is a more convenient and efficient way for QCs handling, which significantly reduce the occurrence of empty travel of the container spreader. So the best way to deal with the operation sequence of QC in a bay is dual-cycle operations.The set of unloading tasks is denoted by U; the set of loading tasks is denoted by L; the union of loading and unloading tasks is denoted by J,$J = U \cup L$, indexed by i and$j$; the set of all container operation tasks is denoted by $\bar{J}$,$\bar{J} = \{I,F\} \cup J$, which $I \in \{0\}$ represents a virtual initial task and $F \in \{J+1\}$ represents a virtual final task; the set of QCs is denoted by K, indexed by k. The following notation is considered.

- n the number of containers that need to be operated, $i,j \in J,i,j = 1,...,n$;
- m the number of QCs, $k \in K,k = 1,...,m$;
- b_i bay number, a QC lifts the container task i from bay b_i or stacks it on bay b_i;
- qt_{ij}^k the time interval between task i and task j operated by the QC k, starts from the moment that QC spreader arrives at the position of task i for handling task i to the moment that spreader arrives at the position of task j for handling task j;
- d_{ij}^k the idle time of spreader between task i and task j operated by the QCk,if task i and task j are loading container and unloading container respectively, $d_{ij}^k = 0$;
- c_{ij}^k the moving time of QC k from the bay of task i to the bay of task$j$,if task i and task j stack on the same bay, $c_{ij}^k = 0$;
- hq_i^k the time taken by QC k to handle task i;
- qr_i^k the moment that QC k begins to handle the task i;
- Q_J a constant represents that all QCs must complete all tasks before the moment of Q_J, namely the time limit of handling operation. In actual operation of terminal, it is usually determined by the handling time of vessel in berth plan;
- M a constant that represents a sufficiently large number.

Decision variables used in this model are as follows.

- x_{ijk} 1 if task j is operated directly after task i by the QC k and 0 otherwise
- y_{ik} 1 if task i is operated by the QC k and 0 otherwise.

The time interval qt_{ij}^k between task i and task j operated by the QC k, is determined by the types of tasks, such as that shown in Fig. 4, there are four kinds of sequential relationship between tasks, namely: (a) Unloading → Unloading, (b) Loading → Loading, (c) Unloading → Loading, (d) Loading → Unloading.

For case (a) and (b), the calculating method of qt_{ij}^k is shown in Eq. (1), while for case (c) and (d), the calculating method is shown in Eq. (2).

Note that if task i and task j stack on the same bay, $c_{ij}^k = 0$.

$$qt_{ij}^k = hq_i^k + c_{ij}^k + d_{ij}^k \tag{1}$$

$$qt_{ij}^k = hq_i^k + c_{ij}^k \tag{2}$$

In Kim and Park (2004), they developed a basis model for QCSP, we referred to their work and formulated QC dispatch problem as follows.

Model 1:

$$\min\max \sum_{j=1}^{n} \sum_{i=1}^{n} qt_{ij}^k \cdot x_{ijk} \tag{3}$$

s.t.

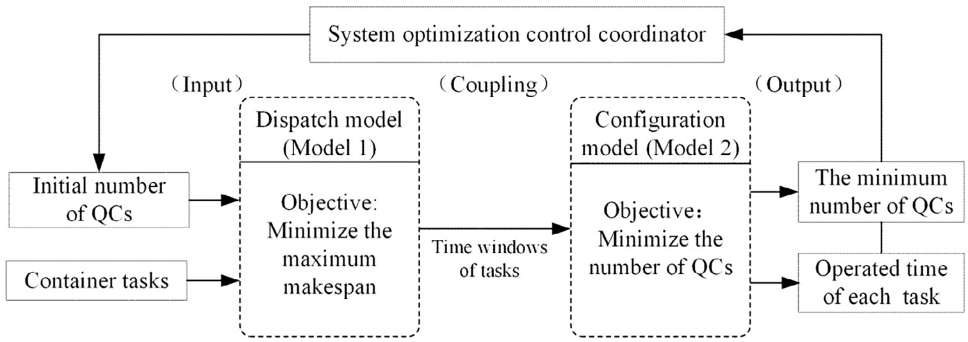

Fig. 3. Framework of dispatch and configuration coupling model for QCs.

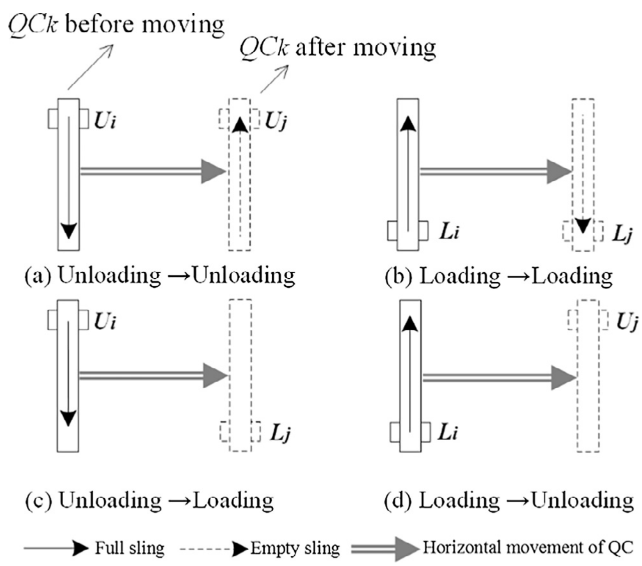

Fig. 4. QCs' moving trace between two types of tasks.

$$\sum_{j=1}^{n}\sum_{i=1}^{n} qt_{ij}^{k}\cdot x_{ijk} \leqslant Q_J \ \forall\ k\in K \tag{4}$$

$$\sum_{k=1}^{m} y_{ik} = 1 \quad \forall\ i\in J \tag{5}$$

$$\sum_{j=1}^{n} x_{ijk} = \sum_{j=1}^{n} x_{jik} \quad \forall\ i\in J;k\in K \tag{6}$$

$$\sum_{j=1}^{n} x_{ijk} = y_{ik} \quad \forall\ i\in J;k\in K \tag{7}$$

$$\sum_{i=1}^{n} x_{ijk} = y_{jk} \quad \forall\ j\in J;k\in K \tag{8}$$

$$\sum_{k=1}^{m} y_{Ik} = m \tag{9}$$

$$\sum_{k=1}^{m} y_{Fk} = m \tag{10}$$

$$qr_i^k + qt_{ij}^k + hq_j^k - qr_j^k \leqslant M(1-x_{ijk}) \quad \forall\ i,j\in J\ ;\forall\ k\in K \tag{11}$$

$$k\cdot\sum_{k=1}^{m} y_{ik} \leqslant l\cdot\sum_{l=1}^{m} y_{jl} ifb_i \leqslant b_j \quad \forall\ i,j\in J\ ;\forall\ k,l\in K \tag{12}$$

$$qt_{Ij}^k + hq_j - qr_j \leqslant M(1-x_{Ijk}) \quad \forall\ j\in J\ ;\forall\ k\in K \tag{13}$$

$$qr_j + qt_{jF}^k - \sum_{j=1}^{n}\sum_{i=1}^{n} qt_{ij}^k\cdot x_{ijk} \leqslant M(1-x_{jFk}) \quad \forall\ j\in J\ ;\forall\ k\in K \tag{14}$$

$$x_{ijk}\in\{0,1\} \quad \forall\ i,j\in J;k\in K \tag{15}$$

$$y_{ij}\in\{0,1\} \quad \forall\ i,j\in J \tag{16}$$

The objective function (3) minimizes the maximum completion time of QCs, namely minimize the makespan of all tasks. Constraint (4) indicates all QCs must complete their tasks before the moment of Q_J .Constraints (5) and (6) imply the fact that each container task can only be handled by a single QC, and each container task has only one previous task and one subsequent task. Constraints (7) and (8) define the relationship between the two decision variables. Constraints (9) and (10) state that each QC must complete the virtual initial task and the virtual final task. Constraint (11) defines the completion time of each task, and eliminates the appearance of the sub task. Constraint (12) ensures that the QCs does not cross each other. Constraints (13) and (14) define the start time and end time of each QC. Constraints (15) and (16) define the integrality of the binary decision variables.

4. Configuration Model for QCs (Model 2)

In this section, a configuration model is proposed to optimize the number of QCs. We take the time window of each task into consideration, and try to reduce the number of QCs by adjusting the actual operation time of each take in its time window.Model 1 can calculate the actual start time qr_i of the QC to handle the container task i, which does not take into account the truck, and it will finish the operation of all tasks at the fastest speed. Therefore, qr_i can be regarded as the earliest available time of the task i, as shown in Eq. (17), which determines the left boundary of the time window of task i, namely Et_i. Eq. (18) aims at to calculate the right boundary of the time window of task$i$,which is denoted by Lt_i. It should be noted that Q_k is the time that QC k completes its all container tasks, and Q_J is the moment that all QCs must complete their all tasks before, namely the deadline of handling. As shown in Fig. 5, we assume that a QC has 4 container tasks to operate. Through the Model 1, it is concluded that the QC complete the 4 tasks at the fastest speed at the moment Q_k. So we get the earliest possible time of each task Et_i. If the QC must complete these 4 tasks before the moment of Q_J, the QC at least to start the 1st task at the moment $Q_J - Q_k$, to start the 2nd task at the moment $Q_J-(Q_k-qr_2)$, and so on. We define this moment as the latest available time of each task, namely Lt_i.

$$Et_i = qr_i \tag{17}$$

$$Lt_i = Q_J-(Q_k-qr_i) \tag{18}$$

As shown in Fig. 6, by selecting the starting time for each task, we can obtain a new scheduling with different operation sequence of QCs, even different number of QCs.In order to improve the speed of solving process, this paper transforms the continuous problem into discrete

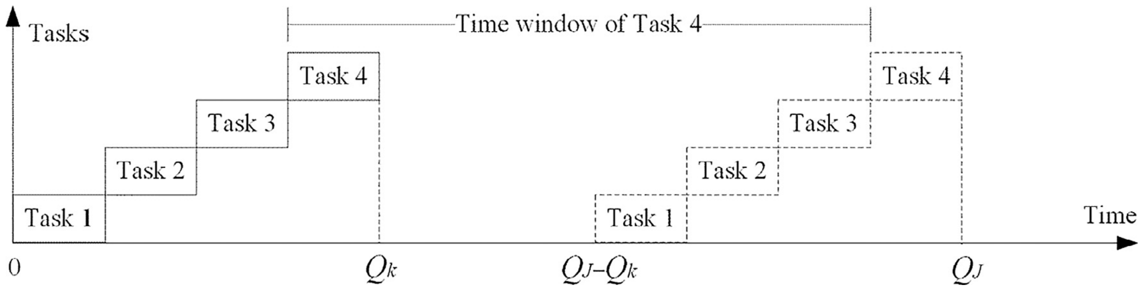

Fig. 5. Computation of time windows for tasks.

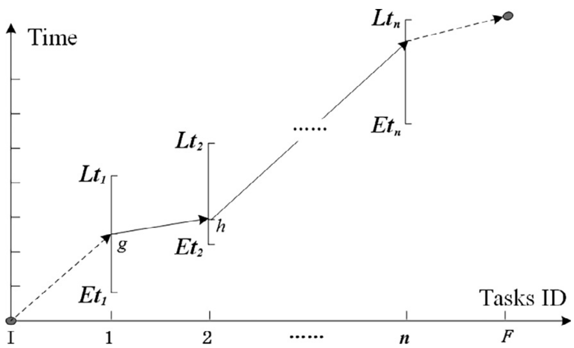

Fig. 6. Time windows for tasks of QCs.

problem by dividing each time window into equal parts. The set of equidistant points is denoted by W, indexed by g and h, which are both equidistant points, Eq. (19) represents the method to calculate the corresponding time for equidistant point g. The set of unloading tasks is denoted by U; the set of loading tasks is denoted by L; the union of loading and unloading tasks is denoted by $J, J = U \cup L$, indexed by i and j; the set of all container tasks is denoted by $\bar{J} = \{I,F\} \cup J$, which $I \in \{0\}$ represents a virtual initial task and $F \in \{J+1\}$ represents a virtual final task; the set of container ships is denoted by S. The following notation is considered.

$$st_{ig} = Et_i + (Lt_i - Et_i)(g-1)/w \tag{19}$$

w	The number of equal parts in each time window;
u	The number of QCs for all operation tasks;
$u_{\max}$	The maximum number of available QCs;
u_s	The number of QCs used for ship s;
b_i	The bay of task i;
$[Et_i, Lt_i]$	The time window of task i;
st_{ig}	The corresponding time of equidistant point g in the time window of task i;
qt_{ij}	The time interval between task i and j;

In this model, a binary variable Z_{igjh} is considered. It is equal to 1 if a QC handles task i at moment g and then to handle task j directly at the moment h, and 0 otherwise. The model is formulated as Model 2.In Model 2, the objective function (20) is to minimize the number of QCs which used for operating all tasks. Constraints (21) and (22) assure the virtual initial task and the virtual final task should both be finished by each QC. Constraint (23) defines the flow balance. Constraint (24) and Constraint (25) state that each task only has one pre-task and one subsequent task. Constraint (26) states that the bay of task i and its subsequent task j should be same or adjacent to each other. Constraint (27) guarantees that the moving time of QC between task i and its subsequent task j should not be smaller than the fixed time interval. Constraint (28) defines the handle sequence of task i and task j. Constraint (29) specifies the domain of decision variable.

Model 2:

$$\min\ u \tag{20}$$

$$\text{s.t.} \quad \sum_{h=1}^{m}\sum_{j=1}^{n+1}\sum_{g=1}^{m} Z_{Igjh} = u \tag{21}$$

$$\sum_{h=1}^{m}\sum_{g=1}^{m}\sum_{i=0}^{n} Z_{igFh} = u \tag{22}$$

$$\sum_{g=1}^{m}\sum_{i=0}^{n} Z_{igab} = \sum_{h=1}^{m}\sum_{j=1}^{n+1} Z_{abjh} \quad \forall\ a \in J; b \in \{1,2,\cdots,m\} \tag{23}$$

$$\sum_{h=1}^{m}\sum_{g=1}^{m}\sum_{i=0}^{n} Z_{igjh} = 1 \quad \forall\ j \in J \tag{24}$$

$$\sum_{h=1}^{m}\sum_{j=1}^{n+1}\sum_{g=1}^{m} Z_{igjh} = 1 \quad \forall\ i \in J \tag{25}$$

$$Z_{igjh}|b_i - b_j| \leqslant 1\ \forall\ i,j \in J; \quad \forall\ g,h \in \{1,2,\cdots,m\} \tag{26}$$

$$|st_{jh} - st_{ig}| \geqslant Z_{igjh}\cdot qt_{ij}\ \forall\ i,j \in \bar{J}; \quad \forall\ g,h \in \{1,2,\cdots,m\} \tag{27}$$

$$st_{jh} \geqslant st_{ig}\cdot Z_{igjh} \quad \forall\ i,j \in \bar{J}; \forall\ g,h \in \{1,2,\cdots,m\} \tag{28}$$

$$Z_{igjh} \in \{0,1\} \quad \forall\ i,j \in \bar{J}; \forall\ g,h \in \{1,2,\cdots,m\} \tag{29}$$

5. Coordination algorithm for the two models

In this paper, we put forward the terminate condition of iterative calculation and the systemic coordination controller to control the coupling process between the two sub-models.

5.1. Setting of coordinator

The number of QCs which input to Model 1 is denoted by u_1. After solving the Model 1, the time windows Et_i, Lt_i can calculated by the above method. Put the time windows into Model 2, we get the minimum number of QCs, which is denoted by u_2. The configuration number of the QCs should not be greater than the available number of QCs, which is denoted by $u_{\max}$, as indicated by the constraint (30).

$$u_2 \leqslant u_{\max} \tag{30}$$

After solving the Model 2, we need to replace u_1 with u_2, and input it again into Model 1 for calculation. If the u_1 and u_2 are equal, as shown in the Eq. (31), it imply that the optimal number of QCs for QCSP is obtained. Model 2 is also able to get a dispatch scheme and the processing moment st_{ig} of task i. For all tasks, if the Eq. (32) is true, it indicates that the QCs complete all tasks with the fastest speed, but this situation cannot always happen. This is because u_1 and u_2 are not equal, the dispatch scheme of Model 1 and the dispatch scheme of Model 2 are certainly different.

If the calculation results of Model 1 and Model 2 meet the Eqs. (31) and (32), stop the iteration calculation. That is to say, at the time, we not only get the best QCs dispatch scheme, but also get the optimal QCs

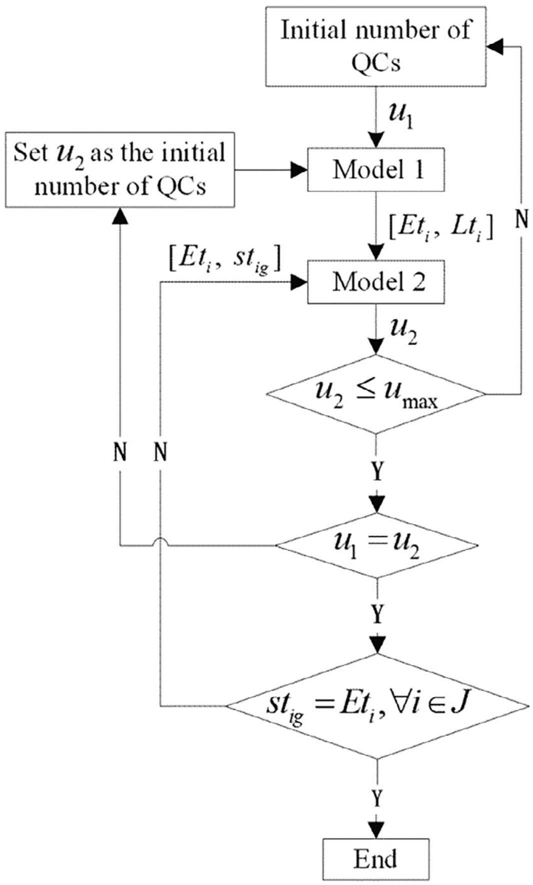

Fig. 7. Flow chart of loop iteration.

configuration quantity.

$$u_1 = u_2 \tag{31}$$

$$st_{ig} = Et_i\ ,\forall\ i \in J \tag{32}$$

Eqs. (31) and (32) can achieve the two-way coordination of the QCs dispatch sub-model and QCs configuration sub-model. Therefore, this two equations are used as the systemic coordination controller of QCs scheduling problem.

5.2. Iterative calculation process

The specific process of iterative calculation is shown in Fig. 7.Step 1: Set an initial number of QCs, and input it into the Model 1, calculate the qr_i and translate it into time windows$[Et_i,Lt_i]$,and then put $[Et_i,Lt_i]$ into Model 2;

Step 2: According to Model 2, get the minimum number of QCs u_2 , then to check whether $u_2$meets the constraint (30), if meets, jump to Step 3; otherwise, jump to Step 1 and set an initial number of QCs again.

Step 3: Check whether u_2 meets the constraint (31), if meets, jump to Step 4; otherwise, jump to Step 1 and set u_2 as the initial number of QCs.

Step 4: Check whether st_{ig} meets the constraint (32), if meets, stop the iterative calculation; otherwise, adjust the time window to $[Et_i,st_{ig}]$, continue to calculate and shorten the time window constantly until it meets the constraint (32).

6. Numerical experiment and model improvement

6.1. Parameters

An instance is designed to verify the above models. The number of vessels waiting for handling is set to 3, and the number of container tasks at a certain period is set to 60. Those containers are distributed in the different bays of three vessels and their types are shown as table 1. As a note, U represents unloading and L represents loading. As Fig. 8 shown, we suppose that there are 6 bays in each vessel.

Table 1
Container stowage plan.

ID	Type	Bay	ID	Type	Bay	ID	Type	Bay	ID	Type	Bay
1	U	1	16	U	5	31	L	9	46	U	13
2	L	1	17	L	6	32	U	9	47	U	13
3	L	1	18	L	6	33	U	10	48	L	14
4	U	1	19	L	6	34	L	10	49	L	14
5	L	2	20	U	6	35	L	10	50	U	14
6	U	2	21	U	6	36	U	11	51	U	15
7	U	2	22	U	7	37	U	11	52	L	15
8	L	3	23	U	7	38	L	11	53	L	16
9	L	3	24	U	7	39	L	11	54	L	16
10	U	3	25	L	7	40	U	11	55	U	16
11	U	3	26	L	8	41	L	12	56	U	17
12	U	4	27	L	8	42	L	12	57	U	17
13	L	4	28	U	8	43	U	12	58	L	17
14	L	5	29	U	9	44	U	12	59	U	18
15	L	5	30	L	9	45	L	13	60	L	18

The available number of QCs $u_{\max}$ is set to 9. The operation time hq_i for each task is 1.2 min/TEU. The idle time d_{ij} of spreaders between two tasks of the same type is set to 0.8 min. The traveling time c_{ij} of is 1.5 min when a QC moved to adjacent bay. The time limit Q_J is assumed to 30 min. And the time window of each task is divided into 4 parts.

6.2. Numerical results

On the basis of the above setting, the models are formulated and solved in a CPLEX 12.2(http://www-01.ibm.com/) environment on a personal computer with 2.10 GHz CPU and 4.00 GB RAM. The results of initial dispatch plan for QCs are shown as Table 2.

The maximum makespan of QCs for the initial dispatch plan is 12.3 min. According to the calculation method of time windows, we can get the initial time windows of all tasks by using the starting operation time of tasks which comes from Model 1. The time windows are shown as Table 3.

In order to solve the minimum number of QCs and corresponding dispatch plan, we set time windows of tasks as input of Model 2.

However, the result is not as good as expected. There are two main issues for Model 2:

(1) After solving the minimum number of QCs, we find that tasks on the same bay are operated by two QCs. But in practice, the tasks on the same bay can only be operated by one QC;
(2) Some QCs begin to operate tasks on another bay without finishing tasks on current bay. In fact, QCs should finish tasks on current bay and then to start to handle new task on another bay.

It's too complex to consider the relationship between bays and QCs in the existing models. An improvement is made on Model 2 to solve above problems which set all tasks on the same bay as one task group.

6.3. Improved configuration Model for QCs (Model 2+)

Parts of dispatch plan solved by Model 1 are shown as Fig. 9. It will take least time if we take the method of dual cycle loading/unloading in quayside operation. This method avoids empty traveling of spreaders, improves the handling efficiency, and transfers uncertain operation time to certain operation time when a QC handles a bay. In other words, if the number and type of tasks on a bay are known, the operation time of this bay can be obtained from Model 1. Thus, when study the

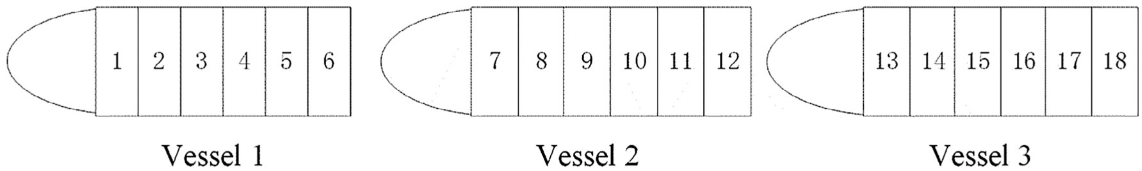

Fig. 8. Distribution of bays on vessels.

Table 2
Initial dispatch plan for QCs.

QC	Sequence of tasks	Makespan (min)
1	1 → 2→4 → 3→6 → 5→7	9.9
2	11 → 8→10 → 9→12 → 13	8.7
3	14 → 16→15 → 17→20 → 19→21 → 18	11.9
4	22 → 25→24 → 23→27 → 28→26	10.7
5	30 → 29→31 → 32→35 → 33→34	9.9
6	36 → 38→37 → 39→40 → 42→44 → 41→43	12.3
7	46 → 45→47 → 49→50 → 48	8.7
8	52 → 51→54 → 55→53	7.5
9	56 → 58→57 → 60→59	7.5

Table 3
Initial time windows of tasks.

Task	Et_i	Lt_i	Task	Et_i	Lt_i	Task	Et_i	Lt_i	Task	Et_i	Lt_i
1	0	20.1	16	1.2	19.3	31	2.4	22.5	46	0	21.3
2	1.2	21.3	17	5.9	24	32	3.6	23.7	47	2.4	23.7
3	3.6	23.7	18	10.7	28.8	33	7.5	27.6	48	7.5	28.8
4	2.4	22.5	19	8.3	26.4	34	8.7	28.8	49	5.1	26.4
5	7.5	27.6	20	7.1	25.2	35	6.3	26.4	50	6.3	27.6
6	6.3	26.4	21	9.5	27.6	36	0	17.7	51	1.2	23.7
7	8.7	28.8	22	0	19.3	37	2.4	20.1	52	0	22.5
8	1.2	22.5	23	4.4	23.7	38	1.2	18.9	53	6.3	28.8
9	3.6	24.9	24	2.4	21.7	39	3.6	21.3	54	3.9	26.4
10	2.4	23.7	25	1.2	20.5	40	4.8	22.5	55	5.1	27.6
11	0	21.3	26	9.5	28.8	41	9.9	27.6	56	0	22.5
12	6.3	27.6	27	7.1	26.4	42	7.5	25.2	57	2.4	24.9
13	7.5	28.8	28	8.3	27.6	43	11.1	28.8	58	1.2	23.7
14	0	18.1	29	1.2	21.3	44	8.7	26.4	59	6.3	28.8
15	2.4	20.5	30	0	20.1	45	1.2	22.5	60	5.1	27.6

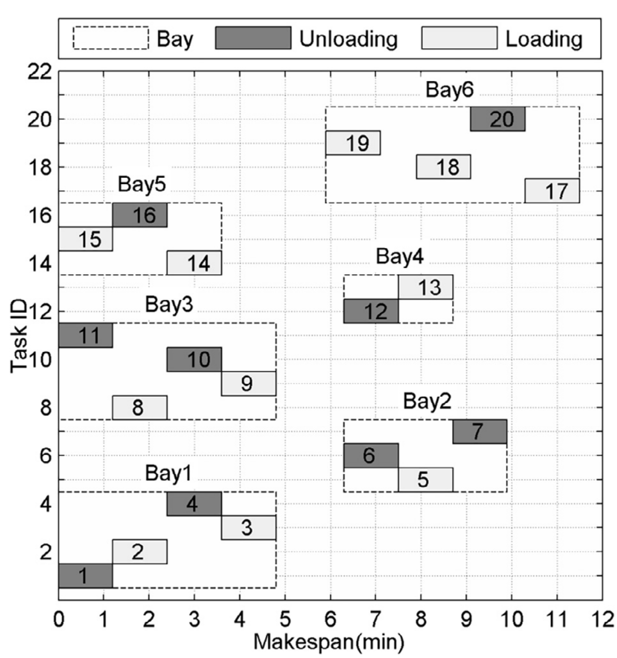

Fig. 9. One bay be regarded as one task group.

configuration problem of QCs, we regard the tasks on the same bay as one task. It avoids the two problems described above.The set of bays is denoted by$J$,indexed by i and j; the set of all operation bays is denoted by $\bar{J}$,$\bar{J} = \{I,F\} \cup J$, which $I \in \{0\}$ represents a virtual initial bay and $F \in \{J+1\}$ represents a virtual final bay; the set of equidistant points is denoted by W, indexed by g and h. The following notation is considered.

w	The number of equal parts of each time window;
u	The number of QCs for all bays;
$u_{\max}$	The maximum number of available QCs;
$[Et_i,Lt_i]$	The time window of bay i;
st_{ig}	The corresponding time of equidistant point g in the time window of bay i;
qt_{ij}	The time interval between bay i and j;

In this model, a binary variable Z_{igjh} is considered. It is equal to 1 if a QC handles bay i at the moment g and then to handle task j directly at the moment$h$,and 0 otherwise. The model is formulated as Model 2+.

Model 2+:

$$\min u \tag{33}$$

$$\text{s.t.} \quad \sum_{h=1}^{m}\sum_{j=1}^{n+1}\sum_{g=1}^{m} Z_{Igjh} = u \tag{34}$$

$$\sum_{h=1}^{m}\sum_{g=1}^{m}\sum_{i=0}^{n} Z_{igFh} = u \tag{35}$$

$$\sum_{g=1}^{m}\sum_{i=0}^{n} Z_{igab} = \sum_{h=1}^{m}\sum_{j=1}^{n+1} Z_{abjh} \quad \forall\ a \in J;b \in \{1,2,\cdots,w\} \tag{36}$$

$$\sum_{h=1}^{m}\sum_{g=1}^{m}\sum_{i=0}^{n} Z_{igjh} = 1\ \forall\ j \in J \tag{37}$$

$$\sum_{h=1}^{m}\sum_{j=1}^{n+1}\sum_{g=1}^{m} Z_{igjh} = 1 \quad \forall\ i \in J \tag{38}$$

$$Z_{igjh}\cdot|i-j| = 1 \quad \forall\ i,j \in J\ ;\forall\ g,h \in \{1,2,\cdots,w\} \tag{39}$$

$$|st_{jh}-st_{ig}| \geqslant Z_{igjh}\cdot qt_{ij}\ \forall\ i,j \in \bar{J}\ ; \quad \forall\ g,h \in \{1,2,\cdots,w\} \tag{40}$$

$$st_{jh} \geqslant st_{ig}\cdot Z_{igjh} \quad \forall\ i,j \in \bar{J}\ ;\forall\ g,h \in \{1,2,\cdots,w\} \tag{41}$$

$$Z_{igjh} \in \{0,1\} \quad \forall\ i,j \in \bar{J}\ ;\forall\ g,h \in \{1,2,\cdots,w\} \tag{42}$$

The objective function (33) is to minimize the number of QCs which used for operating all bays. Constraints (34) and (35) assure the virtual initial and final bays should be finished by each QC. Constraint (36) defines the flow balance. Constraint (37) and (38) state that each bay task can only has one pre-task and one subsequent task. Constraint (39) assures bay i and its subsequent bay j should be adjacent. Constraint (40) guarantees that the moving time of QC between bay i and its subsequent bay js hould not be smaller than the fixed time interval. Constraint (41) defines the handle sequence of bay i and bay j. Constraint (42) specifies the domain of decision variable.

On the basis of the above models, we set the time windows of bays operated by these 9 QCs resulting from Model 1 as the input of Model 2+, and the results of Model 2+ show the optimal number of QCs is 6 and it means 6 QCs can finish all tasks within the allowable time frame. We set 6 QCs as the input of Model 1 to start a new iteration, and then we obtain the time window and the makespan of each bay, which are shown as Table 4.

Input all the data in Table 4 into Model 2+ and we find the optimal number of QCs is still 6. Then we carry out the iterative calculation by continuously adjusting the right edge of time windows in Model 2+. Finally, the number of QCs stabilizes at 6. The optimal dispatch plan is shown as Fig. 10, the maximum makespan is 17 min.

Table 4
Time window and makespan of each bay.

Bay	Et_i	Lt_i	Makespan (min)	Bay	Et_i	Lt_i	Makespan (min)
1	0	13.8	4.8	10	0	12.6	3.6
2	6.3	20.1	3.6	11	5.1	17.7	6
3	11.4	25.2	4.8	12	12.6	25.2	4.8
4	0	14.2	2.4	13	0	17.4	3.6
5	3.9	18.1	3.6	14	5.1	22.5	3.6
6	9.8	24	6	15	10.2	27.6	2.4
7	0	13	5.6	16	0	17.4	3.6
8	7.1	20.1	3.6	17	5.1	22.5	3.6
9	12.2	25.2	4.8	18	10.2	27.6	2.4

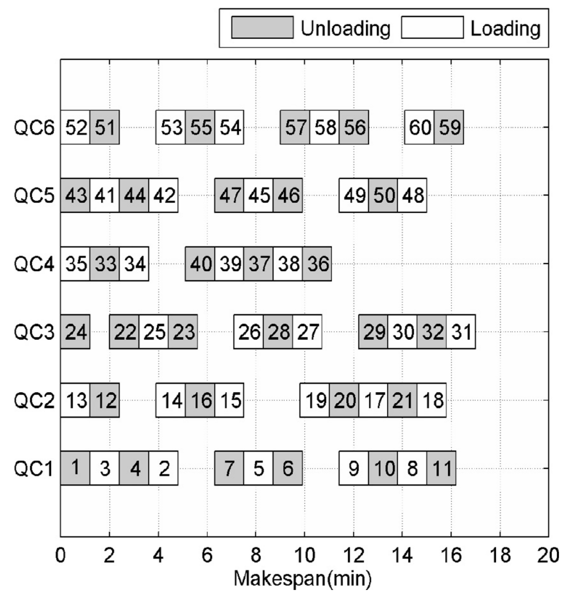

Fig. 10. The optimal scheduling plan of QCs.

6.4. Results analysis

As shown in Table 5, for the examples with different numbers of tasks (or bays), we can always obtain the optimal number of QCs and corresponding scheduling scheme through the above models. We set the value of Q_J (namely the time limit of handling operation) to 30 above. In the same time limit, the optimal number of QCs is still 3 when increasing the number of tasks from 32 to 44. That is to say, there is no need to increase the QC quantity sometimes when more tasks needed to be handled, because the same amount of QCs is enough to complete all tasks on time. But, sometimes it would be better to increase the QC quantity when more tasks needed to be handled in order to ensure all tasks can be completed in the time limits. By calculation, the coupling model of the paper is good at making the decision that when to increase the QC quantity and when not to.

The value of Q_J determines the right boundary of time window of each task, so the time windows of tasks will be changed if we adjust the value of Q_J. As shown in Fig. 11, we can obtain the optimal number of QCs that actually required within various time limits by the calculation of coupling model.

When the reserved handling time of vessel changes, for example, the actual arrival time of vessel is later than the planned arrival time, handling time of vessel will be less. In this situation, the coupling model can easily provide a QC scheduling plan with an optimal number of QCs according to the number of tasks and the changed time limit of handling. Almost all production processes in container terminals are driven by plan, every operation process is closed linked with the next one, and the scheduling system is full of variability. Thus, we believe the scheduling method of the paper has more application scenarios in complex system.

Table 5
Optimal number of QCs and makespan for different numbers of tasks/bays.

Number of tasks	Number of bays	Optimal number of QCs	Makespan (min)
60	18	6	17
58	17	6	17
55	16	5	17.7
52	15	5	17.4
50	14	4	20.1
47	13	4	18.9
44	12	3	22.5
40	11	3	21.3
35	10	3	18.9
32	9	3	17
28	8	2	22.1
25	7	2	20.1
21	6	2	16.2

6.5. Cost-benefit analysis of QC handling

By reducing the number of QCs, the equipment costs of terminal will decrease. But in the meantime, decreasing the makespan by increasing the number of QCs will also reduce the handling costs of terminal. So which way is more effective in reducing total costs? We made a cost-benefit analysis for the trade-off between equipment quantity and makespan.

The costs of QC handling include five aspects: a) Labour costs; b) Electricity fees; c) Maintenance costs; d) Depreciation costs; e) Lubricant costs.

Taking into account all above costs and according to the time-driven activity-based costing (TDABC) method[1], the activity cost driver rate of one QC is 1367yuan/hour.[2] So the total costs caused by QC handling can be solved by the following formula.

$$\text{Total QC handling costs} = \text{The number of QCs}*\text{Makespan} *\text{Activity cost driver rate} \tag{43}$$

By the calculation of Model 1of this paper, we can obtain the minimum handling makespan for different numbers of QCs, which are shown as Table 6.

We can easily obtain the total handling costs through formula (43), and the results are shown in Fig. 12.

Fig. 12 shows that increasing the number of QCs results in increasing the total costs, even though the makespan is dropping.

On the other hand, the income of QC handling comes mainly from handling charges, and it is charged pursuant to the number of containers. In other words, when handling the same number of containers, increasing the number of QCs will not increase the handling income. So from the point of view of costs and benefits, decreasing the number of

[1] More details about TDABC, see: Kaplan, Robert S. and Anderson, Steven R., Time-Driven Activity-Based Costing (November 2003). Available at SSRN: https://ssrn.com/abstract=485443orhttp://dx.doi.org/10.2139/ssrn.4854431.

[2] The data was based on the cost accounting (December 2014) of Shanghai Shengdong International Container Terminal.

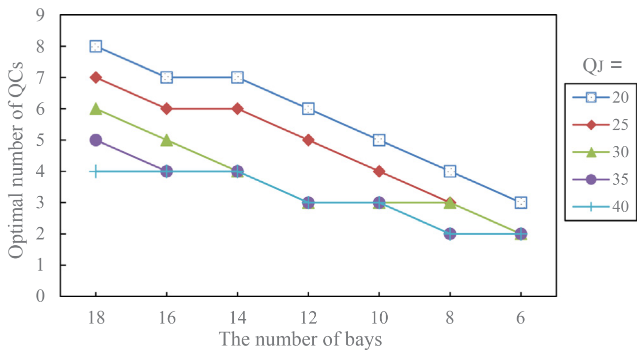

Fig. 11. Optimal number of QCs for different values of Q_J and different numbers of bays.

Table 6
Makespan for different numbers of QCs.

Number of QCs	3	4	5	6	7	8	9
Makespan (h)	0.53	0.4	0.33	0.28	0.25	0.22	0.21

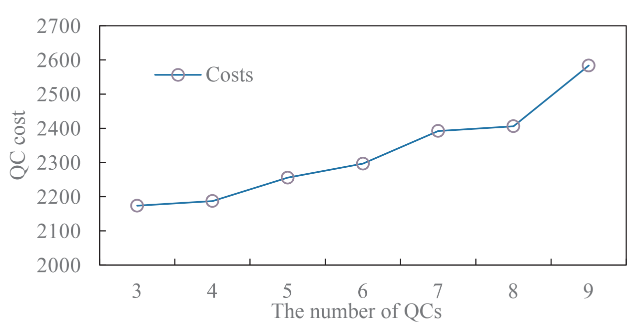

Fig. 12. Total costs of QCs handling.

QCs is more effective than decreasing makespan in reducing the total costs of terminal.

The primary improvement of the paper is we introduced a new methodology to solve the optimal quantity of equipment. Within a certain time limit, the optimal number of equipment will always exist. Besides, the cost-benefit analysis of QC handling denotes that too much use of equipment leads to waste. So if we can use less QCs to handle containers within allowable time limits, the operation costs of terminal will be cut down.

7. Conclusion

The coupling model of this paper mainly solves two problems by loop iteration method: how the QCs be scheduled and how many QCs assigned to vessels is reasonable. The reason why we build the QC dispatch model first is that we need an initial QC scheduling plan to reckon the handling time of each task. QC dispatch model aims to schedule QCs to finish all tasks with the fastest speed, so the handling time of each task form dispatch model can be regard as the earliest handling time. All operations in container terminal have their deadlines, so we can easily figure out the latest handling time of each task. After grasping those two kinds of handling time (time windows), QC configuration model provide a new methodology to solve the optimal QC quantity in the time windows.

Compared to the approach of QCSP found in literature, the proposed approach of this paper decomposes the complex QC scheduling system into two simple subsystems: task dispatch subsystem and quantity configuration subsystem, and then integrates them by using multi-disciplinary variable coupling design optimization (MVCDO) method. We believe that the relation of the two subsystems is coupling. So the coupling model of QC dispatch and configuration is established in the paper.

Compare to the objective of traditional QCSP which aims to minimize the makespan of handling operation, this paper takes the optimal number of QCs into account. Besides, the method which according to the time windows to study the quantity of QCs is also an innovation which is total different form the literature. For the further research, we will use this method to solve other equipment's scheduling problems in container terminal.

Acknowledgement

The authors would like to thank all anonymous referees for their careful review and helpful suggestions. This research is supported by National Natural Science Foundation of China (71471110, 71301101, and 61540045); Science and Technology Commission of Shanghai Municipality (14170501500, 16DZ1201402); Leading Academic Discipline Project of Shanghai Municipal Education Commission (J50604) and Social Science Foundation of Shaanxi Province (2015D060).

References

Al-Dhaheri, N., & Diabat, A. (2015). The quay crane scheduling problem. *Journal of Manufacturing Systems, 36*, 87–94.
Al-Dhaheri, N., Jebali, A., & Diabat, A. (2016). The quay crane scheduling problem with nonzero crane repositioning time and vessel stability constraints. *Computers & Industrial Engineering, 94*, 230–244.
Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research, 202*, 615–627.
Chen, J. H., Lee, D.-H., & Goh, M. (2014). An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem. *European Journal of Operational Research, 232*, 198–208.
Diabat, A., & Theodorou, E. (2014). An integrated quay crane assignment and scheduling problem. *Computers & Industrial Engineering, 73*, 115–123.
Elwany, M. H., Ali, I., & Abouelseoud, Y. (2013). A heuristics-based solution to the continuous berth allocation and crane assignment problem. *Alexandria Engineering Journal, 52*, 671–677.
Han, X.-L., Lu, Z.-Q., & Xi, L.-F. (2010). A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research, 207*, 1327–1340.
He, J., Huang, Y., Yan, W., & Wang, S. (2015). Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption. *Expert Systems with Applications, 42*, 2464–2487.
Hu, Q.-M., Hu, Z.-H., & Du, Y. (2014). Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels. *Computers & Industrial Engineering, 70*, 1–10.
Ji, M., Guo, W., Zhu, H., & Yang, Y. (2015). Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals. *Transportation Research Part E, 80*, 1–19.
Karam, A., & Eltawil, A. B. (2016). Functional integration approach for the berth allocation, quay crane assignment and specific quay crane assignment problems. *Computers & Industrial Engineering, 102*, 458–466.
Kenan, N., & Diabat, A. (2015). A branch-and-price algorithm to solve a quay crane scheduling problem. *Procedia Computer Science, 61*, 527–532.
Kim, K. H., & Park, Y.-M. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research, 156*, 752–768.
Lee, C.-Y., Liu, M., & Chu, C. (2014). Optimal algorithm for the general quay crane double-cycling problem. *Transportation Science, 49*(4), 957–967.
Legato, P., Trunfio, R., & Meisel, F. (2012). Modeling and solving rich quay crane scheduling problems. *Computers & Operations Research, 39*(9), 2063–2078.
Li, F., Sheu, J.-B., & Gao, Z.-Y. (2015). Solving the continuous berth allocation and specific quay crane assignment problems with quay crane coverage range. *Transportation Science, 49*(4), 968–989.
Liu, C., Zheng, L., & Zhangb, C. (2016). Behavior perception-based disruption models for berth allocation and quay crane assignment problems. *Computers & Industrial Engineering, 97*, 258–275.
Lu, Z., Han, X., Xi, L., & Erera, A. L. (2012). A heuristic for the quay crane scheduling problem based on contiguous bay crane operations. *Computers & OperationsResearch, 39*, 2915–2928.
Nguyen, S., Zhang, M., Johnston, M., & Tan, K. C. (2013). Hybrid evolutionary computation methods for quay crane scheduling problems. *Computers & Operations Research, 40*(8), 2083–2093.
Türkoğulları, Y. B., kın, ZCTs, Aras, N., & Altınel, I. K. (2016). Optimal berth allocation, time-variant quay crane assignment and scheduling with crane setups in container terminals. *European Journal of Operational Research, 254*, 985–1001.
Yang, C., Wang, X., & Li, Z. (2012). An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. *Computers & Industrial Engineering, 63*, 243–253.
Zhang, A., Zhang, W., Chen, Y., Chen, G., & Chen, X. (2017). Approximate the scheduling of quay cranes with non-crossing constraints. *European Journal of Operational Research, 258*, 820–828.