An Approach Integrating Simulation and Q-learning Algorithm for Operation Scheduling in Container Terminals

Qingcheng ZENG
Associate Professor
Transportation Management College
Dalian Maritime University
1, Linghai Road, Dalian
116026 China
Fax: +86-411-84726756
E-mail:zqcheng2000@hotmail.com

Zhongzhen YANG
Professor
Transportation Management College
Dalian Maritime University
1, Linghai Road, Dalian
116026 China
Fax: +86-411-84726756
E-mail:yangzhongzhen@263.net

Abstract: In this paper, a method integrating Q-learning algorithm and simulation technique is proposed to optimize the operation scheduling in container terminals. Firstly Q-learning algorithms for yard cranes and yard trailers are designed to obtain the optimal scheduling strategy of yard cranes and yard trailers. Then Q-learning is combined with simulation to develop an integrating scheduling model includes all stages of operation process. In this method, simulation model is used to construct the system environment, Q-learning algorithm is used to learn the optimal dispatching rules for equipments, and the optimal scheduling scheme is obtained by the interaction of Q-learning algorithm and simulation environment. Finally, numerical tests are used to illustrate the validity of the proposed method.

Key Words: *Container terminals, scheduling, simulation, Q- learning algorithm*

1. INTRODUCTION

With the rapid increase of container volume, how to improve the operation efficiency is one of the most important issues for container terminals. For most container terminals, there are mainly three types of equipments involved in the loading and unloading, i.e., quay cranes, yard trailers and yard cranes. Upon a ship's arrival, quay cranes unload containers from or load containers onto the ship, and yard trailers move containers from quayside to storage yard and vice versa. At the storage yard, yard cranes perform the loading and unloading for yard trailers.

The operation scheduling in terminals have the features of multi-objectives, uncertainty, and complexity, which has been proved a NP-hard problem. It is deemed unable to obtain optimal solutions for large-scale problems. Hence, heuristic algorithms are widely used to obtain near-optimal solutions efficiently. However, because of the numerous constraints, it is difficult to evaluate a scheduling scheme in the process of heuristic algorithms. Meanwhile, although many constraints are considered in above scheduling model, it is too complex to model all the constraints analytically. Also uncertainty is difficult to tackle by analytical model alone.

To tackle the complex constraints and the stochastic factors, simulation is used in scheduling problem of container terminals. Researches developed simulation models for the problem of operation scheduling in container terminals. Simulation model can be used to evaluate the scheduling scheme, however, as a test and validation tool, it can only evaluate a given design, not providing more assistant decision making function.

Therefore, in this paper, Q-learning algorithm (one of the reinforcement learning algorithms) is used to obtain the self-adaptability and dynamic scheduling rules for yard cranes in different states. Also it is integrated with simulation, simulation model is used to construct the

system environment, and optimal scheduling scheme is obtained by the interaction of Q-learning algorithm and simulation environment.

This paper is organized as follows. In Section 2, a brief review of previous works is given. Q-learning algorithms for scheduling of yard cranes and yard trailers are designed in Section 3. The framework for method integrating Q-learning algorithm and simulation is developed in section 4. Numerical examples are used to test the performance of the proposed method in Section 5. And conclusions are given in Section 6.

2. LITERATURE REVIEW

Issues related to container terminal operations have gained attention and have been extensively studied recently due to the increased importance of container transport systems. The researches on operation scheduling problem in container terminals can be divided into two types, namely mathematical optimization models and simulation models.

Due to the complexity of container terminal operation, it is difficult to optimize the whole system with a single analytical model, therefore, generally the operation system in container terminal are divided into several sub-processes and each sub-process is optimized respectively. Researchers developed mathematical optimization models for different sub-processes of the container terminal operation system, e.g. quay crane scheduling model (Daganzo, 1989; Kim, 2004; Goodchild and Daganzo, 2007; Lee et al, 2008), YCs allocation and scheduling model (Zhang et al, 2002; Linna et al, 2003; Kim et al,2003, Ng W.C, 2005; Lee et al,2007), storage optimization (Kim and Park, 2003; Zhang et al, 2003), and Yard vehicles routing model (Liu and Ioannou, 2002; Vis, 2005; Nishimura et al, 2005)etc.

The operation efficiency of container terminals depends on the coordination of different sub-processes. To improve the coordination and efficiency of operation in container terminals, some researchers carried out study on the cooperation or harmonization among several activities. E.g. Bish (2003) provided models and algorithms to integrate several sub-processes; the problem is (1) to determine a storage location for each unloaded container, (2) to dispatch vehicles to containers, and (3) to schedule the loading and unloading operations on the cranes, so as to minimize the maximum time it takes to serve a given set of ships. Chen et al (2007) developed an integrated model to optimize the whole loading and unloading process. Lau and Zhao (2008) constructed an operation model for an all-automatic container terminal.

The scheduling problem of container terminals involves numerous variables and constraints. Therefore, when tackling complexion of model and computation, especially, considering the uncertain and stochastic factors, analytic models often confront either the problem that model is too simplify, or the problem that the computation is too complex. Therefore, simulation is widely used in scheduling problem of container terminals recently.

Shabayek and Yeung (2002) developed an application of a simulation model using Witness software to simulate Kwai Chung container terminals. Won and Yong (1999) proposed a simulation model for container terminal system analysis. The simulation model was developed using an object-oriented approach, and using SIMPLE++, object-oriented simulation software. Maurizio et al. (2006) outlined a container terminal simulation model and gave components architecture that was implemented with Java.

Simulation model can be used to evaluate the scheduling scheme, however, as a test and validation tool, it can only evaluate a given design, not providing more assistant decision making function. Recently, simulation optimization method is proposed to overcome these limitations. Combining the simulation analysis and the optimal decision-making mechanism, the simulation optimization method can not only enhance intelligent decision-making of the simulation, but also build the complex system model easily that is more difficult by traditional optimization methods. However, the main disadvantage of the simulation optimization is the long computation time.

To solve scheduling optimization model, mainly two kinds of method are used presently. The first is to search optimal scheme in solution space of combinatorial optimization problem, and most of the above researches belong to this method. Heuristic algorithms are widely used to obtain near-optimal solutions efficiently. However, with the increase of problem scale, the computation efficiency decreases greatly. The second kind method is to obtain optimal scheduling rules given the initial and objective states. Reinforcement learning belongs to this method. It first observes the environment change from one state to another state caused by the action of agents; then calculate the value function; and find the optimal scheduling rule by the learning process of agents. Although reinforcement learning can not ensure to obtain optimal scheduling scheme, it can reduce the calculation complex greatly, and obtain relatively rational scheduling rules. Thus it has received more and more attention in scheduling problem.

E.g. Aydin and Öztemel (2000) proposed a dynamic scheduling system based on agent, and reinforcement learning was used to train the agents to obtain the optimal scheduling strategy. Wang and Usher (2005) applied Q-learning algorithm to obtain optimal assignment strategy for single machine. Although reinforcement learning has been proved an efficient method to solve scheduling problem, it has not been well used to container terminals. In this paper, we will use reinforcement learning to reduce the computation complexity of operation scheduling in container terminals. And, we will integrate simulation with Reinforcement learning. Reinforcement learning is used to reduce the computation complexity; simulation is used to tackle complex constraints and obtain the evaluation of each scheduling scheme.

3. Q-LEARNING ALGORITHMS FOR OPERATION SCHEDULING IN CONTAINER TERMINALS

Container terminal operations can be divided into two parts: loading outbound containers and unloading inbound ones. E.g. the process of loading outbound containers involves three stages: yard cranes pick up the desired containers from yard blocks and load them onto the yard trailers, then yard trailers transport the containers to quay cranes, finally the quay cranes load the containers onto the vessels. The objective of operation scheduling in container terminals is (1) to assign each operation to a machine (2) to sequence the assigned operations on each machine, thus to minimize the makespan of the loading or unloading operations.

To obtain the optimal scheduling scheme, firstly, Q-learning algorithms for yard cranes and yard trailers are designed to obtain the optimal scheduling strategy of yard cranes and yard trailers. And then Q-learning is combined with simulation to develop an integrating scheduling model includes all stages of operation process.

3.1 Q-learning algorithms

Reinforcement learning is a kind of unsupervised machine learning technique. It deals with the problem how an autonomous agent can learn to select proper actions through interacting with its system environment. Each time after an agent performs an action, the environment's response (as indicated by its new state) is used by the agent to reward or penalize its action. The objective is to develop a decision-making policy on selecting the appropriate action rule for each agent. By reinforcement learning, the optimal assigning rules for each agent can be obtained.

Q-learning algorithm is one of the most widely used reinforcement learning algorithms. It was proposed by Watkin in 1989. The objective of this algorithm is to learn the state-action pair value $Q(s,a)$, which represents the long-term expected reward for each pair of state and action (denoted by s and a respectively). $Q(s,a)$ can be denoted by the following equation:

$$Q(s_t,a_t) = (1-\alpha)Q(s_t,a_t) + \alpha(r + \gamma V^*) \tag{1}$$

Where, $Q(s_t,a_t)$ is the expected value to execute action a_t at state s_t; r is the immediate reward to execute action a_t; α is the step-size parameter which influences the learning rate. γ is discount-rate parameter ($0 \le \gamma \le 1$), which impacts the present value of future rewards. V^* is the maximal value of Q under state s_{t+1}:

$$V^* = \max_b Q(s_{t+1},b) \tag{2}$$

At each state, the probability to implement certain action can be calculated by the following equation.

$$p(a_i / s_t) = \frac{\{1/Q(s_t,a_i)\}}{\sum_j \{1/Q(s_t,a_j)\}} \tag{3}$$

In the first iteration of the Q-learning algorithm, the probabilities to select all the possible actions will be the same. However, with the iteration repeats, action with a smaller estimate of $Q(s,a)$ has a higher probability to be selected as the next action. By the iterations, the optimal scheduling rule can be obtained.

Scheduling decision of container terminals can be divided into several inter-related stages, and decisions should be made in each stage. The decisions in each stage depend on current state, and also influence their successor states. The decisions of all stages form a dynamic sequence, whose objective is to optimize the whole scheduling process. The research object of reinforcement learning is decision problems with the characteristic of dynamic, multi-stage, and real-time. The objective is to obtain a scheduling strategy, thus the expected accumulative rewards of all states can be maximized.

By using reinforcement learning to operation scheduling in container terminals, each equipment can be regarded as an autonomous agent. When the system state or environment is changed, agent can make a decision according to real-time state, namely determine the dispatching rule to select next operation task, e.g. yard crane or yard trailer can select the optimal scheduling strategy according to real-time state of operation system. Existing studies indicate that agent can select proper dispatching rules from a set of given rules, which illustrates the feasibility and validity of reinforcement learning in scheduling problem.

3.2 Reinforcement learning for yard crane scheduling

In container terminals, yard storage is place for temporary storage of import and export containers to facilitate the loading and unloading operations. In storage yard, yard cranes processes the loading or unloading of yard trailers and the movement or rehandles of containers. The goal of yard crane scheduling is to decrease the waiting of yard trailers by optimizing the operation sequence of yard cranes.

Let $w_i, i = 1,2,\ldots,n$ denote the time that a yard trailer arrives at storage yard to wait for yard crane to loading or unloading the container. $d_{ij}, i = 1,2,\ldots,n$, $j = 1,2,\ldots,n$ is the time needed for a yard crane to move from storage location i to j, or vice versa. t_i is the time that a yard crane finishes the operation of container i. If the operation of container j is immediately precedes container i by yard crane, $x_{ij} = 1$; and 0, otherwise.

For container i, the waiting time in storage yard can be denoted as $t_i - h_i - w_i$. Thus, the model for yard crane scheduling can be formulated as:

$$\text{Min} \sum_{i=1}^{n} (t_i - h_i - w_i) \tag{4}$$

$$\text{s.t. } t_i \geq w_i + h_i,\ i = 2,3,\cdots,n \tag{5}$$

$$t_j - t_i \geq d_{ij} + h_j - (1 - x_{ij})M,\ i, j = 2,3,\cdots,n, \text{and } i \neq j \tag{6}$$

$$x_{ij} + x_{ji} = 1,\ i, j = 2,3,\cdots,n, \text{and } i \neq j \tag{7}$$

$$x_{ij} = 0 or 1,\ i, j = 2,3,\cdots,n, \tag{8}$$

The objective function (4) is to minimize the total waiting time of yard trailers. Constraints (5) denote the relation of start, operation, and completion time of each task. Constraints (6) denote the relation of each operation task with its predecessor task. Constraints (7) ensure each operation task has at most one predecessor or successor task. Constraints (8) are binary constraints for decision variables.

Generally, dynamic programming method is used to solve the model. However, with the increase of problem scale, the computation time increases rapidly. Thus Q-learning algorithm is used to solve the model.

The process of Q-learning algorithm for yard crane scheduling problem is:

Step0: Initialize the value of Q. For all the state s and action a, let $Q(s,a) = 1$, $n = 1$.

Step1: Obtain the current state s, if $n > N$, the algorithm is end; and go to Step2, otherwise.

Step2: Select the action according to current state. The probability to implement certain action can be calculated by equation (3).

Step3: Implement the selected action, and obtain the immediate rewards r and the next state. The objective of our model is to minimize the waiting time of yard trailers,

therefore, r denote the time penalty to implement certain action, namely the change of yard trailer waiting time. r can be calculated by equation(9). Where, n_j^1, n_j^0 are number of yard trailers to wait for yard cranes at time t^1, t^0 in the j the bay. B is the number of bay in a block.

$$r = \sum_{j=1}^{B} \sum_{t=1}^{n_j} \max(0, t^1 - d_{ij}) - \sum_{j=1}^{B} \sum_{t=1}^{n_j^0} \max(0, t^0 - d_{ij}) \tag{9}$$

Step4: Update Q function according to equation (10).

$$Q(s_0, a) = (1-\alpha) Q(s_0, a) + \alpha [r + \gamma \min_b Q(s_1, b)] \tag{10}$$

Step5: Update the system state, let $s_0 = s_1$, $n = n + 1$.

Step6: If the stop criterion is reached, stop the algorithm, and go to Step1 otherwise.

To apply Q-learning algorithm, the states and policies table should be designed. In yard crane scheduling, the state is determined according to the waiting time of yard trailers and the expected mean service time (EMPT) of yard cranes (Table1), where, m denotes multiple. Three actions (rules) are used to assign yard cranes to a yard trailer, namely first come first served (FCFS), yard cranes makes uni-directional travel to select yard trailers need serving (UT), and select the nearest yard trailer to serve firstly (NT). Thus the table for $Q(s, a)$ can be denoted as Table 1.

Table 1 State policy of Q-learning algorithm for yard crane scheduling

State	State criteria	FCFS	UT	NT
Dummy state	No waiting yard trailer	0	0	0
Dummy state	One waiting yard trailer	0	0	0
1	0≤AVT <m*EMPT	Q(1,1)	Q(1,2)	Q(1,3)
2	m*EMPT≤AVT <2m*EMPT	Q(2,1)	Q(2,2)	Q(2,3)
3	2m*EMPT≤AVT <3m*EMPT	Q(3,1)	Q(3,2)	Q(3,3)
4	3m*EMPT≤AVT <4m*EMPT	Q(4,1)	Q(4,2)	Q(4,3)
5	4m*EMPT≤AVT <5m*EMPT	Q(5,1)	Q(5,2)	Q(5,3)
6	5m*EMPT≤AVT	Q(6,1)	Q(6,2)	Q(6,3)

AVT: average waiting time of yard trailers

The value of α influences the learning efficiency, it can be constant or change with the learning process. In this paper, we suppose $\alpha = 0.1$. The value of γ is between 0 to 1, if it is close to zero, only the immediate penalty will be considered when selecting an action; if it is close to 1, the immediate penalty has a small weight relative to succeeding cumulative penalty. Because this paper is attempted to minimize the total penalty in the long run, γ is set to be 0.9.

Numerical tests are used to indicate the validity of Q-learning algorithm for yard cranes. Supposed that the block that yard crane operated is composed of 40 bays. The arrival interval of yard trailers follows exponential distribution whose mean value is 4 minutes. The operation efficiency of yard cranes follows normal distribution whose expected value is 2 minutes/TEU. The move speed of yard cranes is 7 seconds/bay. The stop criterion is 10,000 iterations. Results indicate that we can obtain stable scheduling results; and the waiting time of yard trailers can reach convergence.

Furthermore, numerical tests are used to compare Q-learning algorithm and other three scheduling rules e.g. FCFS, NT, UT. Fig.1 shows the results of the four methods for different arrival interval of yard trailers. From the results, we can find that FCFS is the worst rule of three rules (FCFS, NT, UT), and the UT is the best one. When the arrival interval of yard trailers are short (e.g. 3 or 4 minutes), Q-learning is not the best methods comparing to other three rules, but when the arrival interval of yard trailers are long (e.g. 5 or 6 minutes), Q-learning is the best one. This indicates that with the increase of arrival interval of yard trailers, Q-learning algorithm becomes more efficient comparing to other three rules.

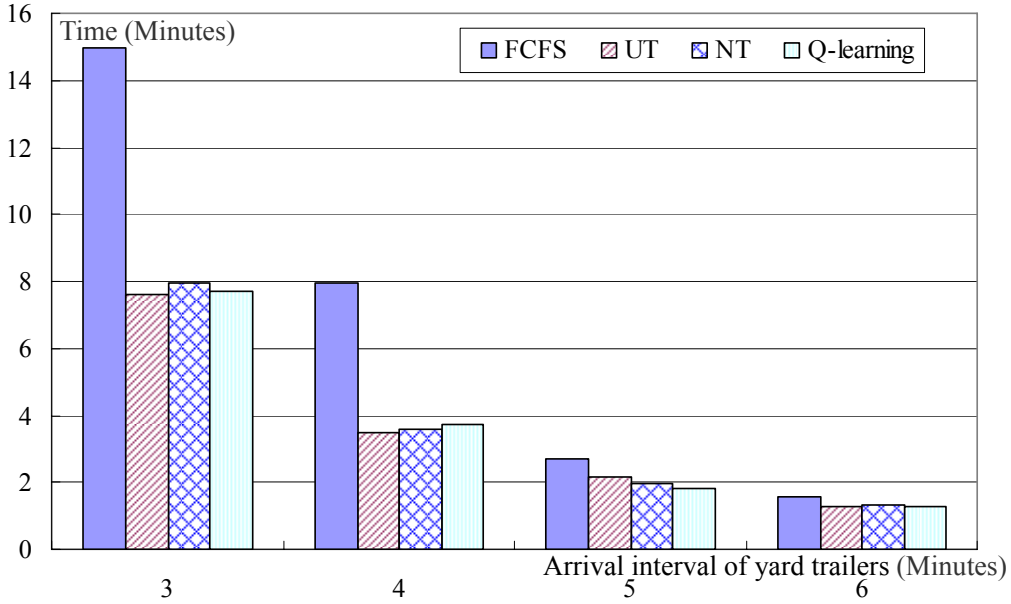

Fig.1 Average waiting time of yard trailers for different scheduling policies

3.3 Reinforcement learning for yard trailer scheduling

In container terminals, yard trailer transport container between quayside and storage yard. Usually, yard trailers are dispatched according to the operation order of quay cranes, the objective is to decrease the waiting time of quay cranes and improve the loading or unloading efficiency.

Taking unloading operation as an example, yard trailer transport a inbound container from quayside to storage yard first, and then return to quayside emptily to transport next inbound container. Let $J = \{1,2,\ldots i,\ldots,n-1,n\}$ denote the operation sequence of quay cranes, and i denote an inbound container; s denote the operation efficiency of quay cranes; λ_i denote the time for a yard trailer transport container i from quayside to storage yard; ST_i denote the starting time for container i, namely the time when a quay crane unloading container i from ship to a yard trailer; CT_i denote the completion time for container i, namely the time when yard trailer returning quayside after it has transport container i to storage yard; $V = \{v_1, v_2, \ldots, v_K\}$ denote the set of yard trailers.

If container i is transported by yard trailer k, $x_{ik} = 1$; and 0 otherwise. If the operation of container j is processed immediately after i by yard trailer k, $y_{ijk} = 1$; and 0 otherwise. Thus,

the model for YT scheduling can be formulated as:

$$\min(CT_n - ST_1) \tag{11}$$

$$\text{s.t. } ST_i \geq 0 \quad i = 2,3,\cdots,n \tag{12}$$

$$CT_i = ST_i + 2\lambda_i, \quad i = 1,2,\cdots,n \tag{13}$$

$$\sum_{k\in K} x_{ik} = 1, i = 1,2,\cdots n \tag{14}$$

$$\sum_{j\in N} y_{ijk} \leq 1, \forall k \in K, \forall i \in N \tag{15}$$

$$CT_i \leq ST_j + H(1 - y_{ijk}) \tag{16}$$

$$ST_i + s \leq ST_j + H(1 - y_{ijk}) \tag{17}$$

$$x_{ik}, y_{ijk} = 1 \text{ or } 0 \ \forall i, j \in N, k \in K \tag{18}$$

The objective function (11) is to minimize the total unloading time. Constraints (12) ensure that all the operation tasks begin after time zero. Constraints (13) denote the relation between starting and completion time of each operation task. Constraints (14) ensure that each operation task is assigned only one YT. Constraints (15) ensure that each operation task has at most one successor operation task. Constraints (16)-(17) denote the relation between two adjacent operation tasks. Constraints (18) are simple binary constraints.

The process of Q-learning algorithm for yard trailer scheduling is:

Step0: Initialize the value of Q. For all the state s and action a, let $Q(s,a) = 1$, $n = 1$.

Step1: Obtain the current state s, if $n > N$, the algorithm is end; and go to Step2, otherwise.

Step2: Select the action according to current state. The probability to execute certain action can be calculated by equation (3).

Step3: Execute the selected action, and obtain the immediate rewards r and the next state. The objective of our model is to minimize the waiting time of quay cranes, therefore, r denote the time penalty to execute certain action, namely the change of quay cranes waiting time, and r can be calculated by equation (19). Where, n', n are number of quay cranes waiting yard trailers at time t', t, D_i is the time that quay crane i can start next operation task.

$$r = \sum_{i=1}^{n'} \max(0, t' - D_i) - \sum_{i=1}^{n} \max(0, t - D_i) \tag{19}$$

Step4: Update Q according to equation (33).

Step5: Update the system state, let $n = n + 1$.

Step6: If the stop criterion is reached, stop the algorithm; and go to Step1 otherwise.

To apply Q-learning, the states and policies table should be designed. In this paper, the state is determined according to the number of waiting quay cranes (Table3), where, n denotes the

number of quay cranes. Three actions (dispatching rules) are used, namely assign yard trailers to the longest waiting (LW), assign yard trailers to the container with longest travel time (LT), and assign a yard trailers to a fixed quay crane (SCO). Thus the table for $Q(s,a)$ can be denoted as Table 2.

Table 2 State policy for Q-learning algorithm for yard trailer scheduling

State	State criteria	LW	LT	SCO
Dummy state	No waiting quay crane (QC)	0	0	0
Dummy state	One waiting QC	0	0	0
1	Number of waiting QCs=1	Q(1,1)	Q(1,2)	Q(1,3)
2	Number of waiting QCs =2	Q(2,1)	Q(2,2)	Q(2,3)
3	Number of waiting QCs =3	Q(3,1)	Q(3,2)	Q(3,3)
4	Number of waiting QCs =4	Q(4,1)	Q(4,2)	Q(4,3)
5	Number of waiting QCs =5	Q(5,1)	Q(5,2)	Q(5,3)
i	Number of waiting QCs =i	Q(i,1)	Q(i,2)	Q(i,3)
n	Number of waiting QCs =n	Q(*n*,1)	Q(*n*,2)	Q(*n*,3)

Suppose 6 quay cranes are assigned to process unloading operation; the processing time of quay cranes are generated from uniform distribution of $U(100,150)$ seconds; the number of yard trailers are 30; the number of unloading containers are 400; and the travel time of yard trailers from quayside to yard storage follows the uniform distribution of $U(8,11)$ seconds. The number of iterations is 10,000. Let $\alpha = 0.1$, and $\gamma = 0.1$. Results indicate that the algorithm can reach convergence within 10,000 iterations.

Furthermore, numerical tests are used to compare Q-learning algorithm and other three scheduling rules e.g. LW, LT, and SCO. Results are shown as Fig.2.

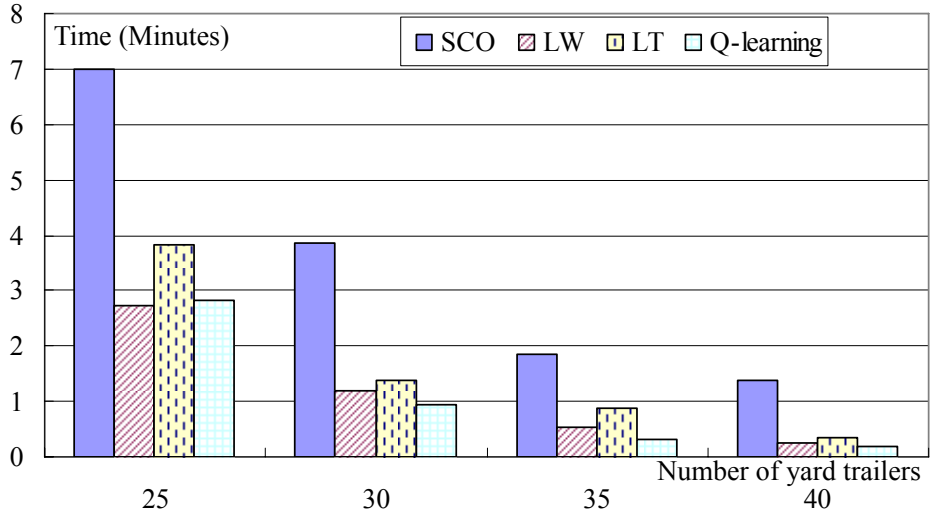

Fig.2 Average quay crane waiting time of different scheduling policies

Fig.2 indicates that the average quay crane waiting time of the four methods and the selected probability of LW, LT, SCO rules in Q-learning. From the results, we can find that SCO is the worst rule of three rules (LW, LT and SCO), and the LW is the best one. When the number of yard trailers is 25, Q-learning is not the best methods comparing to other three rules, but in other conditions (when the numbers of yard trailers are 30, 35, and 40 respectively), Q-learning is the best one. This indicates that with the increase of yard trailers, Q-learning

algorithm becomes more efficient comparing to other three rules.

4. METHOD INTEGRATING Q-LEARNING AND SIMULATION

4.1 Integrating framework

The framework for method integrating Q-learning and simulation (RLSO) is shown as Fig.3. There are two kinds of agents, namely autonomous agents and global agents. Based on the state s given by simulation module, autonomous agents select action a with a probability considering the "state-action" information. Then the action is executed, reward value r is fed back to autonomous agents by simulation module. And the "state-action" information is updated according to the reward value r. After a learning period, a solution can be obtained, and the information is fed back to global agents. The global agents use the information to update the "state-action" information, and thus supervise the learning process of autonomous agents. The "state-action" information includes state, action, Q value, the selected number of each "state-action" pair etc.

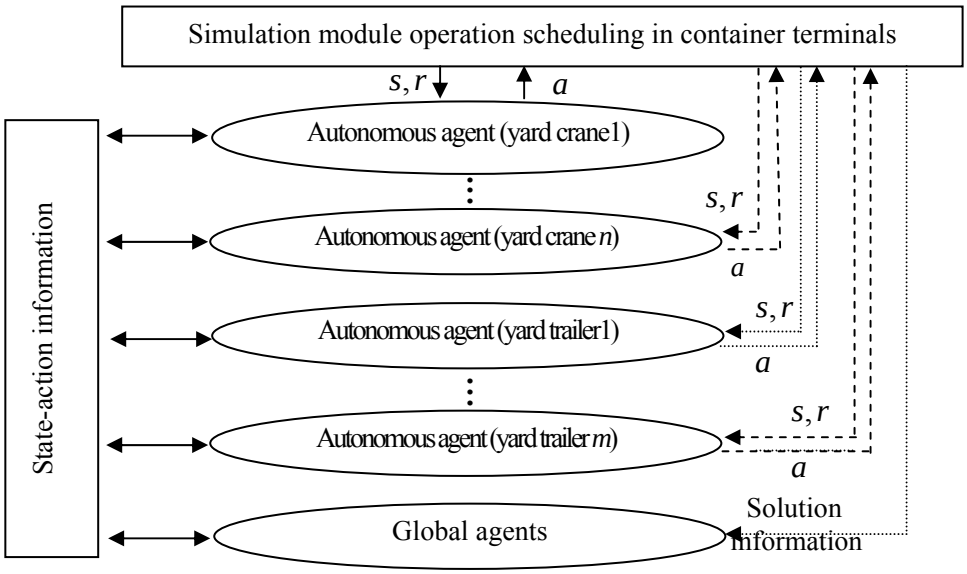

Fig.3 Method integrating Q-learning algorithm and simulation

4.2 Design of state and action

In the loading/unloading process of container terminals, yard cranes and yard cranes are dispatched according to the operation sequence of quay cranes. And the operation sequence of quay cranes is pre-determined by loading/unloading plan. Therefore, in real-time scheduling, we need not consider the dispatching rules for quay cranes, but the dispatching rules for yard trailers and yard cranes.

In our method, yard cranes and yard trailers are regarded as autonomous agents. These two kinds of agent are heterogeneous, which have different set of "state-action" pairs. The design of states and actions of agents are shown as Table1 and Table2.

4.3 Calculation of reward function

In the operation of container terminals, reduction of waiting time of quay cranes can improve the loading/unloading efficiency. Therefore, the waiting time of quay cranes is regarded as reward function. Let D denote the waiting time of quay cranes at time t in a learning period. Autonomous agent i executes action $a_t(i)$ under state $s_t(i)$, and then the operation task J_t is processed. After task J_t being finished, the waiting time of quay cranes and the system state will change to D' and $s_t'(i)$ respectively. Thus, the reward function can be denoted as equation (20):

$$r_t(i) = D' - D \tag{20}$$

This design of reward function helps to decrease the delay of quay cranes, and thus to improve the efficiency of quay cranes and decrease the whole operation time.

4.4 Action strategy of global agents

Let D_n denote the quay crane waiting time obtained by autonomous agents of current learning period; D_{n-1} denote quay crane waiting time of previous learning period. Then the supervisor information can be denoted as $\Delta = D_n - D_{n-1}$. Thus the action strategy of global agents is: when a "state-action" (s,a) is selected in such learning period, the Q function is updated according to equation (21).

$$Q(s,a) = Q(s,a) + \Delta \tag{21}$$

For a "state-action" pair, if $D_n < D_{n-1}$, then $\Delta < 0$, thus the Q value of this "state-action" pair will decrease, and this means that the probability to select this action will increase. On other hand, if $D_n > D_{n-1}$, then $\Delta > 0$, the Q value of this "state-action" pair will increase, and this means that the probability to select this action will decrease. By this means, the global agents can supervise the autonomous agents to select actions that cause the reduction of quay crane waiting time.

5. NUMERICAL TESTS

Firstly, numerical tests are used to indicate the validity of our method (RSLO). We compare RSLO with the method proposed by Lau (2008). Lau developed a scheduling model for automated container terminals, and designed a multi-level genetic algorithm (MLGA) to obtain the optimal operation sequence for quay cranes, AGVs, and yard cranes. We substitute AGVs of Lau's model for yard cranes, and the results are compared with those of RSLO.

Details of the test data are as follows:

- Unloading process is considered.
- The available quay cranes are 3; each quay crane is dispatched 4 yard trailers and 2 yard cranes.
- The processing time of quay cranes are generated from uniform distribution of $U(100,150)$ s, and the processing time of yard cranes follows the uniform distribution of $U(260,320)$ s
- Storage location in yard for each container is selected randomly. The storage location determines the processing time for yard trailers.

According to the number of quay cranes, yard cranes, yard trailers, and the handled containers, 10 scenarios are designed. The total operation time and computation time of the two methods are shown as Table 3. Also, the makespans obtained by RSLO and MLGA are compared with the global lower bound. The relative deviation (RD) is calculated by the following formula:

$$\mathrm{RD} = \frac{C_{\max} - \mathrm{LB}}{\mathrm{LB}} \times 100 \tag{22}$$

Where $C_{\max}$ is the makespan obtained by the developed algorithms and LB is its lower bound which can be calculated by the formula developed by Zeng and Yang (2007).

Table 3 Results of RLSO and MLGA

No	Scenarios				RSLO			MLGA		
	Number of containers	QC	YC	YT	Operation time (m)	Computation time (s)	RD (%)	Operation time (m)	Computation time (s)	RD (%)
1	10	1	2	5	23.1	0.6	0.52	23.0	1.0	0.02
2	20	1	2	5	48.6	2.0	1.15	48.0	3.2	0.03
3	40	2	4	8	51.6	6.4	1.26	51.1	24.3	0.12
4	50	2	4	10	61.2	10.5	1.93	60.2	67.6	0.36
5	80	2	4	8	108.6	21.7	2.47	107.3	150.1	1.24
6	100	2	4	8	130.0	39.2	3.14	128.5	474.2	1.98
7	200	3	6	15	164.8	80.9	4.28	162.1	2574.2	2.62
8	300	3	6	15	250.4	104.7	5.20	245.0	4185.9	2.94

From Table 3, we can find that RLSO method can decrease computation time comparing to MLGA. This is because that the objective of MLGA is to optimize the operation orders of all equipments. For a three-stage flow shop problem similar to container terminals, the solution space increases rapidly with the increase of task and the equipment quantity, thus the computation time of optimization algorithm increase greatly. Instead of search optimal operation order, the objective of RSLO method is to obtain proper dispatching rules or scheduling strategies for all equipments, thus it can decrease the computation time. Moreover, the solution quality of MLGA is better than that of RSLO, but the relative deviation can be controlled within the scope of 5%. This indicates that RSLO can improve the computation efficiency greatly at the cost of slight decrease of the solution quality.

In realistic scheduling, because of the uncertain factors, such as operation delay and mechanical breakdown, processing loading or unloading operations entirely by the pre-determined order obtained by optimization algorithm may face great difficulty. The optimal sequence can not adjust according to real-time state. The objective of Q-learning algorithm is to obtain the self-adaptability scheduling strategies, and the scheduling strategies can be adjusted according to system state. Therefore, Q-learning algorithm has more maneuverability comparing to optimization method.

Furthermore, we compare RSLO method with FCFS dispatching rule using the Scenarios in table 3. Presently, FCFS rule is widely used to dispatch and schedule yard trailers and yard cranes in container terminals. FCFS rule is easy to implement, but it can not ensure to obtain the optimal scheduling scheme. The results of RLSO and FCFS are shown as Fig.4. From

results we can find that RSLO can improve the solution quality comparing to FCFS, which indicate that the "state-action" strategy obtained by RSLO is better than simple FCFS rule.

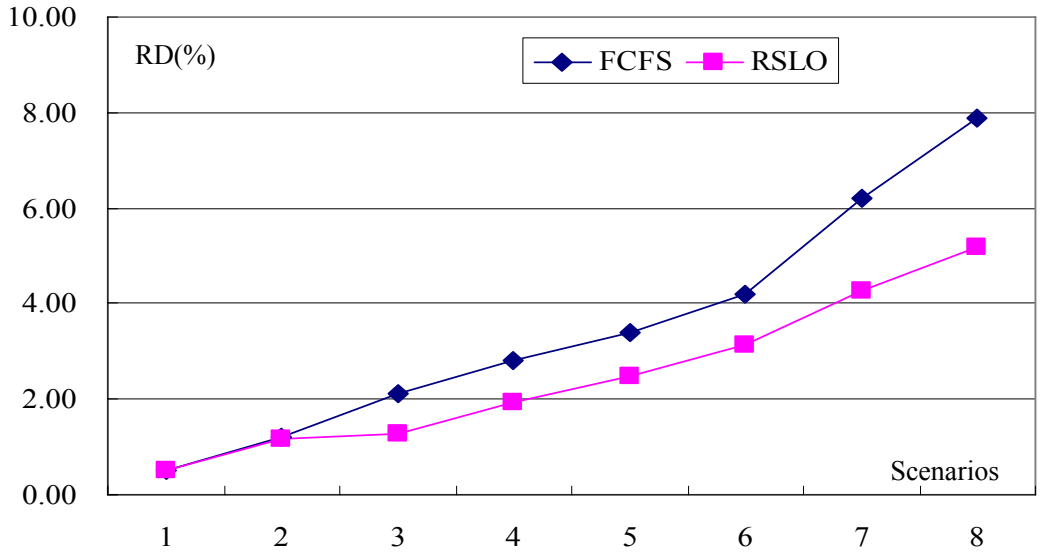

Fig. 4 Comparing RSLO with FCFS rules

6. CONCLUSIONS

Reinforcement learning is an important machine learning method, and has been widely used in control system and automaton. The studies of reinforcement in flow shop problem have gained more and more attention recently. In this paper, we integrate Q-learning algorithm with simulation to solve the scheduling problems in container terminals. Results indicate that the integrated method can not only improve computation efficiency, but improve the self-adaptability of scheduling scheme in container terminals.

The method proposed can be extended in several directions. The first is to design reward function, dispatching rules, and action strategies more efficiently; and thus improve the solution quality and adaptability of the proposed method. The second is to study the interaction among different types of agents, and thus improve the coordination of operation system of container terminals.

REFERENCES

Allaoui H., Artiba A. (2004) Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints, **Computers & Industrial Engineering**, 47(4), 431-450.

Aydin M. Emin, Öztemel Ercan. (2000) Dynamic job-shop scheduling using reinforcement learning agents, **Robotics and Autonomous Systems**, 33 (2-3) 169-178.

Bish Ebru K. (2003) A multiple-crane-constrained scheduling problem in a container terminal, **European Journal of Operational Research**, 144(1), 83-107.

Chen Lu, Bostel Nathalie, Dejax Pierre et al. (2007) A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal, **European Journal of Operational Research**, 181(1), 40-58.

Daganzo Carlos F. (1989) The crane scheduling problem, **Transportation Research Part B**, 23(3), 159-175.

Goodchild A.V, Daganzo C.F. (2007) Crane double cycling in container ports: Planning

methods and evaluation. **Transportation Research Part B**, 41(8), 875-891.

Jin Zhihong, Yang Zan, Ito Takahiro. (2006) Metaheuristic algorithms for the multistage hybrid flow shop scheduling problem, **International Journal of Production Economics**, 100(2), 322-334.

Kim Kap Hwan. (2004) A crane scheduling method for port container terminals, **European Journal of operational research**, 156(3,1), 752-768.

Kim Kap Hwan , Park Kang Tae. (2003) A note on a dynamic space-allocation method for outbound containers, **European Journal of Operational Research**, 148(1)92-101.

Kim Kap Hwan, Lee Keung Mo, Hwang Hark. (2003) Sequencing delivery and receiving operations for yard cranes in port container terminals, **International Journal of Production Economics**, 84(3), 283-292.

Kozan Erhan, Preston Peter. (1999) Genetic algorithms to schedule container transfers at multimodal terminals, **International Transactions in Operational Research**, 6(3), 311-329.

Lau Henry Y.K, Zhao Ying. (2008) Integrated scheduling of handling equipment at automated container terminals. **International Journal of Production Economics**, 112(2), 665-682.

Lee Der-Horng, Cao Zhi, Meng Qiang. (2007) Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm, **International Journal of Production Economics**, 107(1), 115-124.

Lee Der-Horng, Wang Hui Qiu, Miao Linxin. (2008) Quay crane scheduling with non-interference constraints in port container terminals, **Transportation Research Part E**, 44(1), 124-135.

Lin, H.-T., Liao, C.-J. (2003) A case study in a two-stage hybrid flow shop with setup time and dedicated machines. **International Journal of Production Economics** 86 (2), 133-143.

Linna Richard, Liu Ji-yin, Wan Yat-wah. (2003) Rubber tired gantry crane deployment for container yard operation, **Computers & Industrial Engineering**, 45(3), 429-442.

Liu Chin-I, Ioannou P.A. (2002) A comparison of different AGV dispatching rules in an automated container terminal, **The IEEE5th Conference on Intelligent Transportation Systems Singapore**, 880-885.

Maurizio Bielli, Azedine Boulmakoul, Mohamed Rida. (2006) Object oriented model for container terminal distributed simulation. **European Journal of Operational Research**, 175(3),1731-1751.

Ng W.C. (2005) Crane scheduling in container yards with inter-crane interference, **European Journal of Operational Research** 164(1), 64-78.

Nishimura Etsuko, Imai Akio, Stratos Papadimitriou. (2005) Yard trailer routing at a maritime container terminal, **Transportation Research Part E**, 41(1), 53-76.

Shabayek A.A., Yeung W.W. (2002) A simulation model for the Kwai Chung container terminals in Hong Kong, **European Journal of Operational Research**, 140(1), 1-11.

Vis, I. F.A. (2005) Minimum vehicle fleet size under time-window constraints at a container terminal, **Transportation science**, 39(2), 249-260.

Wang Yi-Chi, Usher John M. (2004) Learning policies for single machine job dispatching, **Robotics and Computer-Integrated Manufacturing** 20 (2-3), 553-562.

Won Young Yun, Yong Seok Choi. (1999) A simulation model for container-terminal operation analysis using an object-oriented approach. **International Journal of Production Economics**, 59(1-3), 221-230.

Zeng Qingcheng, Yang Zhongzhen.(2007) A Hybrid flow shop scheduling model for loading outbound container in container terminals, **Journal of Eastern Asia Society for Transportation Studies**,Vol. 7, 2927-2939.

Zhang Chuqian, Liu Jiyin, Liu Yat-wah Wan etc. (2003) Storage space allocation in container terminals, **Transportation Research Part B**, 37 (10) 883-903.

Zhang Chuqian, Wan Yat-waha, Liu Jiyin. (2002) Dynamic crane deployment in container storage yards, **Transportation Research Part B** 36(6), 537-555.