TELKOMNIKA, Vol. 11, No. 2, February 2013, pp. 653~658
ISSN: 2302-4046

Agent-Based Automatic Shore Operating Scheduling for a Container Terminal

Nannan Yan*, Yuan Zhou
Shanghai Maritime University 2660 Pu Dong Da Dao, 86-021-58711770
*Corresponding author, e-mail: nnyan@shmtu.edu.cn

Abstract

This paper established berth allocation and quay crane scheduling models of shore operating for a container terminal. Furthermore, the structure of the berth - quay crane scheduling agent is also given. Finally, an example using genetic algorithm to resolve the berth allocation model and agent technology is shown. The experiment result shows that the use of intelligent theory and technology can provide an effective way for container terminal operating scheduling.

Keywords*: berth allocation, quay crane scheduling, agent*

1. Introduction

The container terminal operating scheduling includes berth allocation and quay crane scheduling. Berth allocation means that berths are assigned for a ship before the arrival of the ship at terminals. Quay crane scheduling is defined as the allocation of a reasonable number of quay cranes for each berth and the suiTable loading and unloading order for quay cranes. Due to the randomness, real-time and complexity of container loading and unloading operations, the building of a complete analytical model for container terminal operating scheduling is very difficult. With the creation and development of many related theories and optimization techniques in the field of artificial intelligence, many new methods in operation scheduling field are presented. A new intelligent scheduling method based on multi-agent system (MAS) provides a new method for the resolving of scheduling problems.

In recent years, some scholars have studied the application of theories and technologies based on agent and MAS in the field of port. Rebollo established a container terminal production management system based on MAS.In this system, various resources, such as quay cranes, ship plans, are mapped to corresponding agents. The purpose of this system is to obtain the shortest ship staying time in port [1]. Thurston proposed a distributed agent technology to study thecoordination of container terminal handling operations, which mainly resolves the problem of the dispatching of quay cranes and trucks. In this study, a simulation system based on Java was developed [2]. Considering the yard plans and the transportation plans, Satoshi Hoshino established an automated guided vehicle (AGV) transportation system based on agent to the goal of efficiency [3]. Paper [4] proposed an optimization model based on (MAS) to resolve the problem of the automatic container terminal production dispatching.

2. A Dynamic Berth Assignment Model

Some referred parameters in the dynamic berth assignment model are: the arriving time for ships, the start job time of ships at berths, the container loading/unloading time, the departure time of ships.The followings are assumed: (1) the different arrival times of ships are considered as undetermined variables, (2) to determine the maximum accepTable waiting time for ships according to different ships, (3) the berths must meet the physical conditions of a ship (water depth and length), (4) each ship has one and only one berthing opportunity and (5) each berth is obtainable only for one ship.The objective function is as the following:

Received August 27, 2012; Revised December 23, 2012; Accepted January 11, 2013

$$Min\left(\sum_{i\in B}\sum_{j\in V} b_j - A_j + C_{ij}\right)X_{ijk}, \forall k \in U \quad (1)$$

The descriptions of variables in expression (1): B means the berth number set. V identifies the number set of ships at port. U stands for the job number set of ship j which is assigned to berth i.b_j means the time when ship j accepts service. A_j means the time when ship j arrives. C_{ij} means the period of loading/unloading time for ship j at berth i. X_{ijk} equals to 1 if ship j is assigned to berth i while X_{ijk} equals to 0 if ship j is not assigned to berth i.

The constraints are:

$$\sum_{j\in V} S_j = S \quad (2)$$

$$\sum_{i\in B}\sum_{k\in U} X_{ijk} = 1, \forall j \in V \quad (3)$$

The above expression (2) ensures the total amount of ships served in berths should be equal to the total amount of all arriving ships. The above expression (3) ensures each ship should be served at least one time.

$$0 \le b_j - A_j \le wt_{j,} \forall j \in V \quad (4)$$

Expression (4) ensures that each ship must be served after arriving according to time requirements and the importance of the ship. The waiting time of each ship must be less than the accepTable maximum waiting time wt_j.

$$(W_i - D_j)_{xijk} \ge 0, \forall_i \in B, j \in V, k \in U \quad (5)$$

Expression (5) ensures the depth of berth is not less than the ship's draft depth. W_i means the depth of berth i, D_j means the ship's draft depth.

$$(P_i - I_j)_{xijk} \ge 0, \forall_i \in B, j \in V, k \in U \quad (6)$$

Expression (6) ensures that the length of ship *j* is less than the length of berth *i*. L_j means the length of ship *j*. P_i means the length of berth *i*.

In this model, the ship's arriving time to port, the berthing location and the quay cranes assigned for each berth are undetermined dynamic factors. So we use the genetic algorithm to solve the problem in this paper.

3. Quay Crane Scheduling Model

Quay crane scheduling model can be expressed as a triple:

QCS=(Task, Qc, Rule) (7)

Task means the set of all loading and unloading tasks of berthing ships. Qc means the quay crane set which can be scheduled by all loading and unloading operations. Rule means the rule set which should followed by quay crane scheduling and quay crane operations.

$$Task = \bigcup_{i=1}^{n} Task\ i \quad (8)$$

$$Task_i=(tid_i, io_i, deck_i, sbay_i, block_i, ybay_i, cn_i, s_i, t_i, pri_i) \quad (9)$$

Tid_i is the serial number of $task_i$. io_i means whether $task_i$ is the loading task or the unloading task. $deck_i$ means that $task_i$ is on the deck or in the cabin. $sbay_i$ is the bay number of $task_i$. $block_i$ is the block number in the yard of $task_i$. $Ybay_i$ indicates the bay number in the block of $task_i$. cn_i indicates the number of containers waiting for loading/unloading. S_i indicates the beginning time of taski. t_i indicates the time needed by finishing $task_i$. pri_i is the priority of $task_i$.

$$Qc = \bigcup_{j=1}^{m} qc\ j \quad (10)$$

$$qc_j=(qcid_j, status_j, pos_j, n_j, s_j) \quad (11)$$

$qcid_j$ indicates the serial number of quay crane j. $status_j$ indicates the current status of quay crane j. pos_j indicates the current bay of quay crane j. n_j is the quantity of the tasks waiting for quay crane j. s_j indicates the earliest idle time for quay crane j.

Suppose there are n tasks waiting for loading/unloading and there are m available quay cranes. For $task_i = (tid_i, io_i, deck_i, bay_i, cn_i, s_i, t_i, pri_i)$ $(1≤i≤n)$, $pri_i = 0$, which means $task_i$ needs to be scheduled.

Suppose $task_i$ $(1≤i≤n)$ is waiting for the service of quay crane j（1≤j≤ m. Suppose $task_q$ (1≤q≤ n) is waiting for the service of quay crane p（1≤p≤m）.

$$\{qc_j =(qcid_j, status_j, pos_j, t_j) \quad task_i =(tid_i , io_i , deck_i , bay_i , cn_i, s_i , t_i , pri_i)\} \quad (12)$$

$$\{qc_p=(qcid_p, status_p, pos_p, t_p) \quad task_q=(tid_q, io_q, deck_q, bay_q, cn_q, s_q, t_q, pri_q\} \quad (13)$$

The basic scheduling rules can be expressed as:

if ((bay_i==bay_q) and (io_i==1) and (io_q==0)) then ((pri_i=1) and (pri_q=2)) /* if loading task and the unloading task are at the same bay, unloading task should be finished before loading task.*/

if ((bay_i==bay_q) and ((io_i==0) and (io_q==0)) and (($deck_i$==1) and ($deck_q$==0))) then ((pri_i=4) and (pri_q=3)) /* if unloading tasks are at the same bay, the unloading task on the deck should be finished before the unloading task in the cabin. */

if ((bay_i==bay_q) and ((io_i==1) and (io_q==1)) and (($deck_i$==1)and ($deck_q$==0))) then ((pri_i=1) and (pri_q=2)) /* if loading tasks are at the same bay, the loading task in the cabin should be finished before the loading task on the deck.*/

if ((bay_i<bay_q) and ($|bay_i-bay_q|$>=8) and (pos_j<pos_p) then ((pri_i=1) and (pri_q=1))/* Quay cranes can't work across each another. */

4. The Structure of the Berth-Quay Crane Scheduling Agent (BQSA)

After finishing building the berth-quay crane assignment and scheduling model, the structure of the BQSA should be designed.

When a ship arrives, the task request of berthing and loading/unloading is sent to the task agent by the agent platform. The BQSA receives the task request from the task agent. Therefore, the BQSA should have communication capabilities for information exchange with the outside world.In addition, the BQSA should be able to store the model of the berth-quay crane assignment and scheduling. Also, the solution algorithm for the berth-quay crane assignment and scheduling should be stored. At the same time, the BQSA should be able to manage the rules of the loading/unloading operation. In summary, we designed the structure of the BQSA as shown in Figure 1. The functions of the components of the BQSA are shown as the following: (1) communication control unit including the communication interface and the message interpreter. The communication interface is responsible for communicating with the outside world and the integral part of the message interpreter is responsible for the explanation of the received message. (2) resource management center. It is responsible for accepting the registration of agents and resource information query. (3) task buffer pool. When several tasks arrive, they can be put into the task buffer pool waiting for being assigned. (4) job control module. It is arranged to resolve the berth allocation and quay crane scheduling problem according to the solutions stored in the model library. (5) learning module. BQSA records and summarizes the allocation and scheduling results, and adds new knowledge to knowledge database. (6) berth / quay crane database. It is arranged to store the information of berths and quay cranes which are managed by BQSA. (7) knowledge/quay crane scheduling rule database. It is responsible for storing each allocation and scheduling result. (8) model database. Different solutions for berth and quay-crane allocation and scheduling problems are stored in model database.

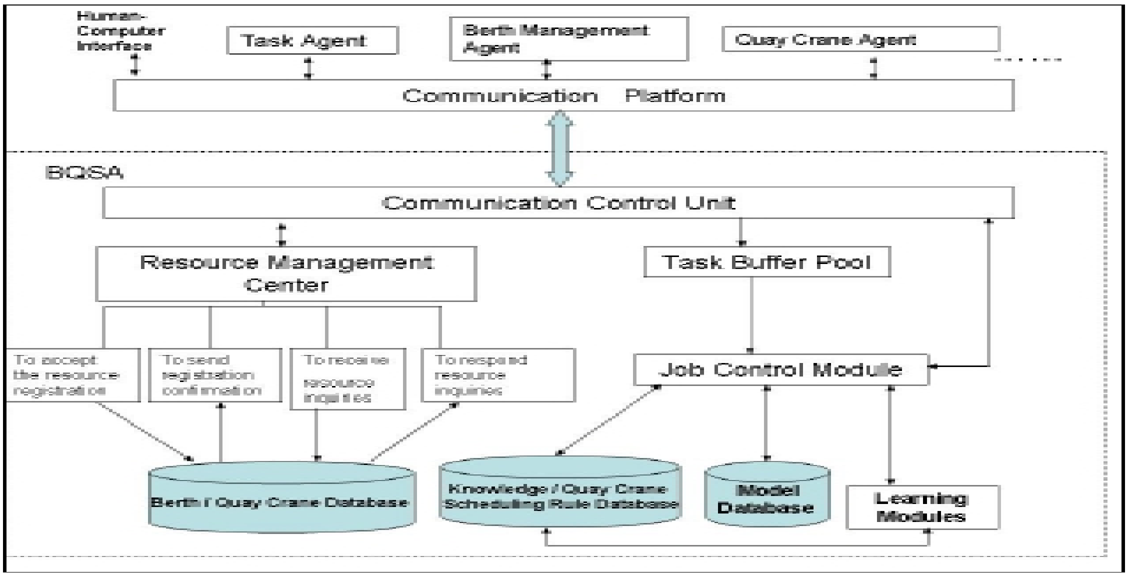

Figure 1. The Structure of the Berth - Quay Crane Scheduling Agent

5. Solution Examples

We take a container terminal which has 3 berths and is provided with 12 quay cranes for an example. The loading/unloading speed of the quay crane is 27 containers per hour. The work data of 12 ships is shown as Table 1. The maximum accepTable waiting time for ships is from 3-10 hours . The loading/unloading tasks of each ship is shown as Table 2.

Table 1. The Basic Work Data of Arriving Ships

Arriving Ships/Vessels	Arriving Time		The Number of Containers
	Day	Moment	
Ship 1	The 1st Day	01:40	960
Ship 2	The 1st Day	05:20	1790
Ship 3	The 1st Day	08:30	980
Ship 4	The 1st Day	13:00	1825
Ship 5	The 1st Day	22:00	1060
Ship 6	The 1st Day	23:00	980
Ship 7	The 2nd Day	05:00	1525
Ship 8	The 2nd Day	06:00	1035
Ship 9	The 2nd Day	20:00	1080
Ship 10	The 2nd Day	21:00	890
Ship 11	The 3rd Day	18:00	1670
Ship 12	The 3rd Day	20:00	1580

The tested basic parameters of genetic algorithm are that: (1) population size is 100. (2) the times of heredity iteration is 1000. (3) genetic crossover probability is 0.9. (4) genetic variation probability is 0.05.

Table 2. The Examples of Loading/Unloading Tasks

Task No.	Task Bay No.	Deck/Cabin	Loading/Unloading	The SequeceNumber in One Bay	The Number of Containers
I01D	01	D	I	1	1
I01H	01	H	I	2	1
I02H	02	H	I	2	31
E02H	02	H	E	4	12
I03D	03	D	I	1	2
I05D	05	D	I	1	1
...	...	...	...	...	...
I30D	30	D	I	1	7
I30H	30	H	I	2	1
E30H	30	H	E	4	15
E30D	30	D	E	4	17
I34D	34	D	I	1	14
I34H	34	H	I	2	12

Suppose all berths satisfy the arriving ships. According to the assumed experiment parameters, the best value is 39.22 after running the genetic algorithm program. The best chromosome is 1 4 8 12 0 2 6 9 13 0 3 5 7 10. So ship 1, ship 4, ship 8 and ship 12 are assigned for berth No. 1 which acquires 4 quay cranes. Ship 2, ship 6, ship 9 and ship 13 are assigned for berth No. 2 which acquires 4 quay cranes. Ship 3, Ship 5, ship 7 and ship 10 are assigned for berth No. 3 which also acquires 4 quay cranes. The dispatching arrangement is shown as Table 3.

Table 3.The Example of the Best Beth Allocation Using Genetic Algoritm

Arriving Ships/ Vessels	The Berthing Berth	Arriving Time		Time (hours)	Departure Time		Loading/ Unloading Speed Volumes/ Hour)
		Day	Moment		Day	Moment	
Ship 1	1	The 1st Day	01:40	9. 33	The 1st Day	10:50	105
Ship 2	2	The 1st Day	05:20	13. 66	The 1st Day	19:00	131
Ship 3	3	The 1st Day	08:30	9. 25	The 1st Day	17:45	106
Ship 4	1	The 1st Day	13:00	13. 33	The 2nd Day	2:10	130
Ship 5	3	The 1st Day	22:00	9. 83	The 2nd Day	7:50	108
Ship 6	2	The 1st Day	23:00	9. 5	The 2nd Day	8:30	103
Ship 7	3	The 2nd Day	05:00	11. 33	The 2nd Day	19:10	135
Ship 8	1	The 2nd Day	06:00	9. 66	The 2nd Day	15:40	107
Ship 9	2	The 2nd Day	20:00	8. 5	The 3rd Day	4:30	127
Ship 10	3	The 2nd Day	21:00	1. 25	The 3rd Day	7:15	87
Ship 11	1	The 3rd Day	18:00	12. 5	The 4th Day	6:30	134
Ship 12	2	The 3rd Day	20:00	13. 83	The 4th Day	9:50	114

5. Conclusion

This paper uses agent technology to build the structure of the Berth-Quay crane Scheduling Agent(BQSA). The berth allocation model and the quay crane scheduling rules of the BQSA are also shown in this paper. Experiment result shows that the use of agent technology and other related intelligent theory can provide an effective support for shore operating scheduling for a container terminal.

References

[1] Rebollo, Julian, Carrascosa, Botti. *A Multi-Agent System for the Automation of a Port Container Terminal.* Autonomous Agents 2000 workshop on Agents in Industry. Barcelona, Spain. 2000.

[2] Thurston HU. *Distributed Agent Architecture for Port Automation*. 26th International Computer Software and Applications Conference. Oxford. UK. 2002; 81-90.
[3] Satoshi Hoshino, Jun Ota, Akiko Shinozaki, Hideki Hashimoto. *Design of an AGV Transportation System by Considering Management Model in an ACT Intelligent Autonomous Systems*. Book Editors IOS Press. 2006.
[4] Miguel Rebollo, Vicente Julian, Carlos Carrascosa, Vicente Botti. *A Multi-Agent System for the Automation of a Port Container Terminal.* 26th International Computer Software and Applications Conference. Oxford. UK. 2002; 81-90.