Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II
WCECS 2009, October 20-22, 2009, San Francisco, USA

Space Allocation for Vessel Services with Irregular Calling Pattern

Mingkun Li and W.C. Ng

Abstract— **This paper studies storage space allocation in yard for export containers of vessel services which need a relatively stable storage plan for an optimal stacking and loading schedule. The planning will directly affect terminal operations and performance while the irregular calling pattern of those services largely increases the difficulty of planning. By introducing an integer programming method and two-phase solving algorithm, solutions can be generated optimally to minimize the total difference of occupied slots in yard blocks and avoid possible traffic jams during the loading process.**

***Index Terms*—Container terminals, Operations Research, Yard planning.**

I. INTRODUCTION

With the growth of world trade, cargo transportation in containers has been a major way in commercial logistics. In the market vessel services take a big share for providing a comparatively lower price than that by air or railway. The increasing cargo volume puts high pressure on ports to improve the operations in container terminals and keep the competitive advantage. Customers need efficient and effective services from quay side operations to storage yard handling while the operation cost should be under control as well. In Hong Kong, Shanghai and many other Asian cities land is a particularly limited resource and its costs undertake a large portion of the total cost. Storage space planning in yard area is considered a key issue for improving the terminal productivity and keeping low cost.

Yard is the temporary storage place for import and export containers which is normally divided into a number of blocks. A typical block contains 40 to 60 slots that each slot may have 6 to 8 rows. Containers could be stacked to several tires depending on the resource of land. Once containers have been in stack, they should not be moved to other slots until they are loaded to trucks or vessels to avoid high costs of re-handling. In yard, export containers normally occupy a large number of slots before they are loaded to ships. Big shipping lines often deploy vessels to run some regular vessel services, to call terminals for a loading job at schedule. Export containers for these ships arrive at the terminal several days before the vessels' calling date such that a large amount of storage space would be needed to satisfy the steadily increasing number of space requirement for storing these containers. When there are vessels calling the terminal, containers need to be located, transported and loaded to vessels efficiently. To facility the handling process and avoid traffic bottleneck for loading in any block, operators must make the space allocation plan for these containers to reserve the proper amount of slots and balanced distribute to each yard blocks before their arrival to the terminal according to the schedule. However, such a schedule, including the amount of containers arriving at the terminal during each time period and the vessel's arrival date could be changed timely due to many unexpected factors. Hence, a dynamic space allocation is needed for meeting the changeable schedules of vessel services.

Manuscript received June 5, 2009; revised July 8, 2009.

Mingkun Li is with the Department of Industrial and Manufacturing System Engineering, The University of Hong Kong, Hong Kong (phone number: 00852-28597967; email: lukeli@hkusua.hku.hk).

W. C. Ng is with the Department of Industrial and Manufacturing System Engineering, The University of Hong Kong, Hong Kong (email: ngwc@hku.hk).

II. LITERATURE REVIEW

Many research works have been put on yard planning and operations in yard. Zhang et al.[1] studied the workload imbalance problem in space allocation and developed a method to avoid the possible transportation bottlenecks. They provided a solution method to balance the number of space allocated to blocks first by formulating the problem as an integer programming model. One part of the objective function is to minimize the total difference of workload among blocks during the whole time period, i.e.

$$\text{Minimize} \sum_{t=1}^{T} \{\max_i (D_{i,t} + L_{i,t}) - \min_i (D_{i,t} + L_{i,t})\}$$

where $D_{i,t}$ was denoted as the number of vessel discharge containers stored in block *i* discharged from vessels of period *t* and $L_{i,t}$ was defined to be the number of containers stored in block *i* that are going to be loaded to vessels of period *t*. Thereafter they determined the exact block for export and import containers achieving a minimum sum of internal transportation costs. Kim and Kim[2] presented a time based scheduling model for the storage of inbound containers which can reduce the occupation of long term containers in yard place. Kim and Park[3] applied dynamic space allocation methods to outbound containers. By using duration-of-stay rule and sub-gradient optimization, some experiments were designed to compare allocation result of different heuristics. Yard planning has attracted many study works while only a few of them focus on such problems to determine exact position for export containers with an optimal workload balance in handling process.

Other research areas for operations in container terminals also attract many research works. Some works have been put on berth allocation and crane and vehicle arrangement. Imai et al.[4] studied the multi-user container terminals and developed a genetic algorithm to schedule the berth service. Priority of vessels is considered a factor to affect the model and the result. Another model[5] was built to solve MUT problem with continuous locations. Zhang et al.[6] addressed some solutions for dynamic crane deployment. By estimated the workload among different blocks, a mixed integer programming model was formulated to find the optimal time and route for cranes and minimize the total delayed workload. Imal et al.[7] regarded the stowage and load planning as a multi-objective problem and applied a weighting method to solve the multi-objective integer program.

This yard planning problem aims at giving the balance of the workload among occupied yard blocks and avoids possible jams during the loading process. Such an objective is widely considered in many other research fields. Kathryn [8] provided four options for the objective of workload balance in flexible manufacturing systems which include balancing the workload of each machine in a system of pooled machines with equal sizes. The formulation of this objective function provides ideas for achieving a similar objective of the yard planning problem.

III. Integer Programming Model for Solving the Problem

As stated in section 1, the yard planning problem is to decide the exact position of slots in each block assigned for storing export containers of vessel services from time to time so as to achieve an optimal balance of workload among blocks when there is a loading job. The workload is measured by the number of slots with loading job in each block when a vessel calls the terminal. Due to many uncertain circumstances, the vessel calling pattern in this problem is not in regular cycles, and schedules of vessel services could be changed from time to time. To solve this problem with dynamic calling patterns, a rolling horizon planning method is introduced in this section: Supposed that one planning horizon consists of *T* days, an allocation plan will be made for services with known arrival schedules at the first day of each planning horizon, the next planning will be done based on latest information when time period goes to the next day which is the beginning of next horizon so that the plan will be updated from time to time and at each planning epoch only space allocated to export containers at the first day is fixed and will be executed. Running the solution method on such a rolling horizon approach, we need to give the following feasible conditions and assumptions for the problem.

Assumptions:

1. The schedule of arrival of containers and the vessel will be submitted to terminal operators at the first day of container arrival to the terminal while the information in the schedule may be updated during the time period. Operators will not reserve space for those services whose containers have not arrived at the terminal on the first day of the planning horizon. Without losing generality, suppose that the storage period of containers for one vessel will not exceed *T* days. The number of containers arrived to the terminal increases with time until all containers are loaded to vessels at the loading day. Below is an example of space requirement at the first day of one planning horizon consisted of 7 days with schedules for 5 services.

Table I
Slots requirement of one planning horizon of 5services

Servic e	Slots requirement						
	Period						
	1	2	3	4	5	6	7
1	1	2	2	2	2	3	5
2	2	4	7	9	10	15	16
3	2	8	9	20	0	0	0
4	3	5	5	11	0	0	0
5	7	7	12	12	12	0	0

2. Once containers have been in stack to a certain place by the plan, they should not be moved to other slots until they are loaded to the vessel, i.e. the initial status of slots allocated to vessels must be checked first in each planning and the next allocation plan should be made accordingly.
3. For facilitating the loading job, normally there is an upper limit number of slots assigned to the same vessel in one block.

Therefore, according to the assumptions the data known at the beginning of a planning horizon are:

C_i the space capacity of slots in block *i*,

$R_{j,t}$ the number of slots required by service *j* at period *t*,

$\hat{S}$ the upper limit of slots belonging to one service in one block.

And the initial status of allocated slots to service *j* in block *i* at the beginning of a planning horizon is defined as

$$X^i_{k,j,0} = \begin{cases} 1, & \text{if slot } k \text{ of block } i \text{ is occupied by service } j \\ & \text{at the beginning of the planning horizon} \\ 0, & \text{otherwise} \end{cases}.$$

The following variables are defined to formulate the problem as an integer programming model.

$$X^i_{k,j,t} = \begin{cases} 1, & \text{if slot } k \text{ of block } i \text{ is assigned to service } j \\ & \text{at period } t \\ 0, & \text{otherwise} \end{cases},$$

α_t the smallest number of slots with loading jobs among all blocks at period *t*,

β_t the largest number of slots with loading jobs among all blocks at period *t*.

Denote the set $\{1,2,...,I\}$ by **I**, the set $\{1,2,...,K\}$ by **K**, the set $\{1,2,...,T\}$ by **T**, the set $\{1,2,...,J\}$ by **J** and the set of $X^i_{k,j,t}$ by **X**. Let Ω_j be the day which service *j* calls for loading operation.

Then, for each planning horizon, the yard planning problem can be stated as the following integer programming problem.

Problem P1

$$\text{Minimize } \sum_{t=1}^{T} (\beta_t - \alpha_t)$$

Subject to

$$X^{i}_{k,j,t-1} \le X^{i}_{k,j,t} \quad k \in \mathbf{K}, i \in \mathbf{I}, j \in \mathbf{J}, t = 1,\ldots,\Omega_j \tag{1}$$

$$\sum_{j=1}^{J} X^{i}_{k,j,t} \le 1 \quad k \in \mathbf{K}, i \in \mathbf{I}, t \in \mathbf{T} \tag{2}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K} X^{i}_{k,j,t} = R_{j,t} \quad j \in \mathbf{J}, t \in \mathbf{T} \tag{3}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K} X^{i}_{k,j,t} \le C_i \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{4}$$

$$\sum_{k=1}^{K} X^{i}_{k,j,\Omega_j} \le \hat{S} \quad i \in \mathbf{I}, j \in \mathbf{J} \tag{5}$$

$$\sum_{j \in \mathbf{J}, \Omega_j = t}\sum_{k=1}^{K} X^{i}_{k,j,t} \ge \alpha_t \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{6}$$

$$\sum_{j \in \mathbf{J}, \Omega_j = t}\sum_{k=1}^{K} X^{i}_{k,j,t} \le \beta_t \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{7}$$

In the objective function, $\beta_t - \alpha_t$ is the largest number of difference of slots with loading jobs at period *t* among all blocks. The objective of problem P1 is to minimize the total number of workload imbalances among all the blocks for the whole time period. Constraint (1) ensures that the size of occupied slots by each service grows when time approaches the loading day. Constraint (2) states that each slot in a block can be assigned to at most one service at any one time. Constraint (3) states that the total number of occupied slots for one service among all blocks should equal to the space requirement of this service at period *t*. Constraint (4) gives space capacity for each block which the number of total assigned slots in the block can not be greater than the capacity. Constraint (5) means that the number of slots assigned to service *j* in any block should not be greater than the upper bound $\hat{S}$. Constraints (6) and (7) contribute to establish a linear model for the objective function.

An experiment by ILOG CPLEX running on a desktop computer with 2.4 GHz CPU is carried out to evaluate the possibility of finding optimal solutions for the problem. In the experiment, the storage yard has 10 blocks serving for around 10 services in a planning horizon with 7 days, and the planning period is 90 days. The computational results reveal that normally these numerical examples can be solved optimally while some could take around one hour for generating a result if the number of services increases. An efficient algorithm is needed for solving this problem.

IV. ALGORITHM FOR SOLVING PROBLEM P1

Since solving problem P1 could be very time consuming, this section aims at developing an alternative model with simplified constraints and a two-phase solution idea for approaching solution for the problem. In phase 1, we will determine the number of slots assigned to each service in each block at every time period instead of the exact position of slots.

Phase 1:

Define $U^{i}_{j,t}$ the number of slots occupied by the cluster of service *j* in block *i* at period *t* and $U^{i}_{j,0}$ the number of containers that have been stacked in block *i* for service *j* at the beginning of the planning horizon. The problem of determining the number of slots assigned to services in each block at each time period can be stated as:

Problem P2

$$\text{Minimize } \sum_{t=1}^{T}(\beta_t - \alpha_t)$$

Subject to

$$U^{i}_{j,t-1} \le U^{i}_{j,t} \quad i \in \mathbf{I}, j \in \mathbf{J}, t = 1,\ldots,\Omega_j \tag{8}$$

$$\sum_{i=1}^{I} U^{i}_{j,t} = R_{j,t} \quad j \in \mathbf{J}, t \in \mathbf{T} \tag{9}$$

$$\sum_{j=1}^{J} U^{i}_{j,t} \le C_i \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{10}$$

$$U^{i}_{j,\Omega_j} \le \hat{S} \quad i \in \mathbf{I}, j \in \mathbf{J} \tag{11}$$

$$\sum_{j \in J \,\&\, \Omega_j = t} U^{i}_{j,t} \ge \alpha_t \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{12}$$

$$\sum_{j \in J \,\&\, \Omega_j = t} U^{i}_{j,t} \le \beta_t \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{13}$$

The objective of problem P2 is same as that of problem P1. Constraint (8) ensures that the occupied slots of one service in any block increase when time approaches the loading day. All the number $U^{i}_{j,t}$ should be greater than or equal to $U^{i}_{j,0}$ the number of containers that have been stacked in block *i* at the beginning of the planning horizon. Constraint (9) states that the sum of occupied slots of one service among all blocks should equal to the space requirement of this service. Constraint (10) gives space capacity for each block. Constraint (11) control the size of slots assigned to one service in one block. Constraints (12) and (13) contribute to establish a linear model for the objective function.

When the value of $U^{i}_{j,t}$ is determined, the position of slots in blocks allocated to services will be determined in phase 2.

Phase 2:

In this paper we apply the policy of scattered allocation to determine the position of containers that no specific area will be reserved for those containers before their arrival. As stated in assumptions, only the space allocation on the first day in the plan will be executed, thus the stacking idea will be introduced only for determining positions of slots on this day. Let $\overline{\mathbf{K}}_i$ be the set of available slots in block *i*. The allocation procedure can be stated as SAP:

Step 1: Set *i*=1. For all *j*, $j \in \mathbf{J}$, if $U^{i}_{j,1} \ne 0$, allocate slots in $\overline{\mathbf{K}}_i$ to services in sequence started from the smallest number of slots in the set where the number allocated to service *j* is $U^{i}_{j,1} - U^{i}_{j,0}$.

Step 2: For $i = 2,\ldots,I$, repeat the allocation.

Step 3: Do the allocation as in step 1 and 2 for the first day of each planning horizon.

The properties of problem P1 and P2 are analyzed and discussed in the following lemmas.

Lemma 1: The optimal solution of Problem P1 is a feasible solution of Problem P2.

Proof: This lemma will be proved by showing that the optimal solution of P1 satisfies all constraints of problem P2.

Suppose that $\{\hat{X}, \hat{\alpha}, \hat{\beta}\}$ is the optimal solution of Problem P1. Hence, the optimal objective function value of Problem P1 = $\sum_{t=1}^{T}(\hat{\beta}_t - \hat{\alpha}_t)$. It is clear from the definition of $U_{j,t}^{i}$ that for $i \in I, j \in J, t \in T$,

$$U_{j,t}^{i} = \sum_{k=1}^{K} X_{k,j,t}^{i} \; . \qquad (14)$$

It follows from Expression (14) and Constraint (3) that $\sum_{i=1}^{I}\sum_{k=1}^{K}\hat{X}_{k,j,t}^{i} = \sum_{i=1}^{I} U_{j,t}^{i} = R_{j,t}$. Thus, $\{\hat{X}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraint (9).

By replacing $\sum_{k=1}^{K}\hat{X}_{k,j,t}^{i}$ with $U_{j,t}^{i}$ in Constraints (4), (5), (6) and (7), it is clear that $\{\hat{X}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (10), (11), (12) and (13).

Since $\{\hat{X}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (1), it is obvious that $\hat{X}_{k,j,t-1}^{i} \le \hat{X}_{k,j,t}^{i}$ for $k \in \mathbf{K}, i \in \mathbf{I}, j \in \mathbf{J}, t = 1, \ldots, \Omega_j$.

Hence, it follows from Expression (14) that $U_{j,t-1}^{i} = \sum_{k=1}^{K}\hat{X}_{k,j,t-1}^{i} \le \sum_{k=1}^{K}\hat{X}_{k,j,t}^{i} = U_{j,t}^{i}$. Thus, $\{\hat{X}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (8).

Therefore, the optimal solution of Problem P1 satisfies all the constraints of P2. Hence, the lemma is true.

Since the objective functions of Problem P1 and P2 are the same, it is clear from Lemma 1 that the following corollary is true.

Corollary 1: The optimal objective function value of Problem P1 is greater than or equal to the optimal objective function value of Problem P2.

Lemma 2: Suppose that $\{\hat{U}, \hat{\alpha}, \hat{\beta}\}$ is the optimal solution of P2, the position of slots is determined by SAP with the value of $\hat{U}_{j,t}^{i}$, then an optimal solution can be constructed for Problem P1.

Proof: Corollary 1 shows that the optimal objective value of P2 is not larger than that of Problem P1.

Showing that vacant space will be allocated to $\hat{U}_{j,t}^{i}$ - $\hat{U}_{j,t-1}^{i}$ without reallocating the assigned slots for $\hat{U}_{j,t-1}^{i}$ in the procedure of Algorithm SAP, it definitely satisfies Constraints (1) and (2).

Obviously, the allocation of SAP is proceeded for satisfying the space requirement of $\hat{U}_{j,t}^{i}$ such that it satisfies Expression (14).

By replacing $\hat{U}_{j,t}^{i}$ with $\sum_{k=1}^{K} X_{k,j,t}^{i}$ in Constraints (9), (10), (11), (12) and (13), it is clear that the constructed solution $\{X, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (3), (4), (5), (6) and (7).

Thus, the lemma is proved showing that the constructed solution is a feasible solution of Problem P1 and achieves the optimal objective value.

V. Result discussion and conclusions

The above lemmas and corollary show that the two-phase solving idea is an effective way for solving problem P1. Noted that there are less variables in problem P2, optimal solutions can be generated by ILOG CPLEX in an acceptable time which is proved by our further experiments. The algorithm is efficient for solving the yard planning problem as well. Hence, this research introduces an effective and efficient method to make plans for storing containers with the optimal balanced workload for the loading process while it leaves space for further studies on containers stacking in clusters. Unlike the idea of scattered allocation of slots, continuous slots allocation is needed in a cluster since containers belonging to the same vessel may need to be arranged close to each other. Thus, space in clusters should be reserved for containers before the vessel arrives while the clusters' space planning could face challenges from unexpected changes of calling schedules.

References

[1] C. Zhang, J. Liu, Y.-w. Wan, K. G. Murty, and R. J. Linn, "Storage space allocation in container terminals," *Transportation Research Part B: Methodological*, vol. 37, pp. 883-903, 2003.

[2] K. H. Kim and K. Y. Kim, "Optimal price schedules for storage of inbound containers," *Transportation Research Part B: Methodological*, vol. 41, pp. 892-905, 2007.

[3] K. H. Kim and K. T. Park, "A note on a dynamic space-allocation method for outbound containers," *European Journal of Operational Research*, vol. 148, pp. 92-101, 2003.

[4] A. Imai, E. Nishimura, and S. Papadimitriou, "Berth allocation with service priority," *Transportation Research Part B: Methodological*, vol. 37, pp. 437-457, 2003.

[5] A. Imai, X. Sun, E. Nishimura, and S. Papadimitriou, "Berth allocation in a container port: using a continuous location space approach," *Transportation Research Part B: Methodological*, vol. 39, pp. 199-221, 2005.

[6] C. Zhang, Y.-w. Wan, J. Liu, and R. J. Linn, "Dynamic crane deployment in container storage yards," *Transportation Research Part B: Methodological*, vol. 36, pp. 537-555, 2002.

[7] A. Imai, K. Sasaki, E. Nishimura, and S. Papadimitriou, "Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks," *European Journal of Operational Research*, vol. 171, pp. 373-389, 2006.

[8] K. E. Stecke, "Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems," *Management Science*, vol. 29, pp. 273-288, 1983.