

STORAGE ALLOCATION IN AUTOMATED CONTAINER TERMINALS: THE UPPER LEVEL

Xia Mengjue ¹

Zhao Ning ²

Mi Weijian ²

¹ Scientific Research Academy

² Logistics Engineering College

Shanghai Maritime University, 1550 Haigang Avenue, Shanghai, China. 201306

ABSTRACT

Nowadays automation is a trend of container terminals all over the world. Although not applied in current automated container terminals, storage allocation is indispensable in conventional container terminals, and promising for automated container terminals in future. This paper seeks into the storage allocation problem in automated container terminals and proposed a two level structure for the problem. A mixed integer programming model is built for the upper level, and a modified Particle Swarm Optimization (PSO) algorithm is applied to solve the model. The applicable conditions of the model is investigated by numerical experiments, so as the performance of the algorithm in different problem scales. It is left to future research the lower level of the problem and the potential benefit of storage allocation to automated container terminals.

Keywords: storage allocation, automated container terminal, upper level model, PSO algorithm

INTRODUCTION

Container terminals are important interfaces of global container transportation, which connect trucks on land and container vessels over sea. Different in arrival and departure directions, containers passing through a container terminal could be divided into three classes. One is inbound containers from vessels to trucks; the second is outbound containers from trucks to vessels; the third is transshipment containers from vessels to vessels. No matter the class it belongs to, one container is unloaded to the storage yard on its arrival, and stored there for some time till it is loaded to some other vessel or truck and leave the terminal.

Storage space allocation is a decision problem in determining the stacking location of containers in the storage yard,

especially in Asian container terminals. Before the arrival of one container vessel, some of the storage space in the yard are designated for the outbound and transshipment containers to the vessel, and some other are designated for the inbound and transshipment containers from the vessel. Although the specific stacking locations are not determined in this allocation, it is confined the storage spaces available for the containers from and to the vessel. It is widely agreed by terminal operators that, proper allocation of storage spaces has a far-reaching impact on the loading and unloading efficiencies of the vessels, and at the same time contributes a lot to a high utilization of the storage space.

Following the trend of large vessels, automation is nowadays a trend for container terminals in major ports all over the

world. In recent years, the capacity of the largest container vessels have reached up to 18000 TEUs, with their lengths kept at not much more than 400 meters. To serve these huge vessels collecting and discharging probably thousands of containers at a time, terminals built in the past few years are much larger than before, with yard areas extending, and stacking cranes equipped in the yard increasing at the same time. In case that the yard cranes are operated at site following the traditional way, in which one worker is needed for every yard crane in work, a large number of operators are needed for the terminal and the manual cost will be very high. As a result, yard cranes in large terminals are usually automated reformed, hence multiple cranes could be remote controlled simultaneously by a single operator, and the manual cost of the terminal could be kept in a reasonable range.

Storage allocation is an important decision in conventional container terminals, while things are a bit different in the current automated ones. In these automated terminals located mainly in Europe, the storage spaces of containers to be loaded onto or unloaded from some vessel are decided by rule, i.e. to the nearest crane or to the crane with the shortest queue, while no space allocation is applied. Rule-based allocation is a practical strategy in European automated container terminals, where the information of some containers is missing or incorrect now and again. However, in other automated container terminals where the container information is always complete and correct, pre-determined storage allocation plans may lead to better performance.

This paper deals with the storage allocation problem in automated container terminals where container information is complete and correct, especially for those terminals under construction in Asian countries which will be put into use in the near future. This problem is divided into two levels, and only the upper level is discussed in this paper. The rest of the paper is organized as follows. Section 2 gives a literature review on storage allocation problems in conventional terminals and storage allocation strategies in automated ones. A detailed description to the automated container terminal and storage space allocation is given in section 3. Section 4 presents a mixed integer programming model with multiple objectives for the storage space allocation problem in automated container terminals. In section 5, a modified Particle Swarm Optimization (PSO) Algorithm is proposed to solve the model. In section 6 numerical experiments are conducted to show the performance of this algorithm in various problem scales. A conclusion is driven in the last section.

LITERATURE REVIEW

Storage space allocation is a fully studied problem in conventional container terminals. Some researchers focus on allocations within a given time length, while others are interested in the designation of yard templates which could be used repeatedly every week.

Mi Chao et al. [1-5] have done a bunch of researches about information acquisition of inbound and out bound containers, which supports an information-complete storage space

allocation decision.

Kap Hwan Kim and Kang Tae Park [6] discussed the storage space allocation problem for outbound containers only. With no grouping strategy applied, containers are distinguished only by vessels. Blocks were treated as simple container holders, without consideration on structure. The objective of their problem is to minimize the total travelling distance.

Chuqian Zhang et al. [7] studied the container storage allocation problem in Hong Kong. Inbound and outbound containers are both considered in their problem, and a rolling horizontal approach is applied, following which storage plan is made daily for the coming three days. Balance of workloads was set the objective of their problem, while special attention is paid to workloads related to vessel loading and unloading. Still, no grouping strategy or block structure is mentioned in their work.

Thanks to the team led by Loo Hay Lee and Ek Peng Chew, many researches has been done to the storage allocation problem in transshipment hubs, as for terminals in Singapore. Loo Hay Lee et al. [9] studied the storage yard allocation problem on a known yard template, considering only the outbound and transshipment containers that are to be loaded. Blocks were divided into sub-blocks with fixed capacity, and the set of sub-blocks which could be reserved for vessels was known. They considered both 20ft and 40ft containers, while grouped only to their destination vessel respectively. A work balancing protocol was proposed to reduce traffic congestion, and a model was built for a weekly template in which the number of yard cranes deployed to the allocated storage locations were minimized. Yongbin Han et al. [11] presented an extension to the former study. In their problem, a sub-blocks could be reserved for any vessel in the planning horizon. The considered the workloads from both loading and unloading. Liang Ping Ku et al. [17] proposed a generic yard plan template problem specification, which was believed to contribute to the design of an IT-based search engine for an optimum planning strategy. In a later research [19], they proposed a robust approach in finding a nimble yard template for a storage spaces allocation problem under uncertainty, in which the arrival schedule of vessels might change. Xinjia Jiang [23, 30] researched further into dynamic yard planning problems. In a former study [18], a space sharing strategy was proposed for the yard template planning problem. Some bays in a block are treated as sharing spaces, which could be shared by the sub-blocks on both sides. In a latter study [30], a flexible space sharing strategy was further proposed, in which storage locations could be reserved alternately for containers of different vessels. Both strategies led to higher utilization of the storage spaces.

Weijian Mi et al. [12] sought into storage allocation problem for outbound containers only. The objective function of their model was a combination of travel distance, work balance among blocks and work balance among vessels.

Mohammad Bazzazi [13] proposed an extended storage space allocation problem for inbound containers. Different types of containers were considered in the problem, and the objective of their model was the balance of workloads among blocks in every time period.

Der-Horng Lee et al. [14] proposed an approach which integrates the truck scheduling problem and the storage allocation problem. Containers were allocated to storage locations which led to a minimized total travel distance of trucks.

Etsuko Nishimura [15] addressed the storage allocation problem of transshipment containers. Transshipment containers were grouped to different feeder vessels, and each group was allocated a block hence the container handling time of the feeder ships could be minimized.

Yanwei Zhang et al. [16] studied the storage planning problem for outbound containers in container terminals. Only 20 ft containers were considered in the problem, and a three level model was built for this problem. In the first level, containers of the same vessel were allocated to blocks which lead to a minimized total transfer distance.

Mingkun Li and Shiyang Li [20] studied the yard template designing problem for outbound containers in terminals with limited storage space and heavy workload. Spaces are allocated in bays, which could be reserved for various vessels in different periods.

Youn Ju Woo and Kap Hwan Kim [21] proposed a method for allocating storage spaces to outbound containers grouped to destination ports. Spaces were reserved on arrivals of containers, and a rule-based method was addressed for determining the sizes of spaces to be reserved.

Yeon-Ho Jeong et al. [24] proposed a block assignment method based on QC schedule. Aiming at the balance of workloads, a block was selected for a coming container group according to YC workload and space availability.

Der-Horng Lee et al. [25] studied the management of transshipment flows in a transshipment container terminal. Containers to different feeder vessels are located to different blocks hence both the total travel distance and the workload imbalance could be minimized. In a later study [28], he extended the problem to a large container transshipment hub with multiple terminals. Containers were grouped to inbound and outbound vessels. An integral programming model was developed, for the berthing terminal of every vessel, and the storage block for every container group, aiming at the minimization of total inter-terminal and intra-terminal handling costs.

Pei-yu Li and Xiao-ming Sun [26] addressed the storage space allocation problem in a novel container terminal layout, in which containers are carried between the quayside and the storage yard by vehicles running on crossing rails. Some rails were laid parallel to the quay line below the QCs, while others were perpendicular to the quay line, connecting the rails laid at the quayside, and YCs in the blocks. Given that the number of transport channels limited, a non-linear model was proposed, aiming at the balance of YC workloads in every planning horizon.

Lu Chen and Zhiqiang Lu [28] addressed the storage location assignment problem for outbound containers, which was a two stage problem consisting of a storage space allocation problem in the upper level and a storage location determination problem in the lower level. In the upper level, containers were

assumed to be the same size, and grouped by the destination ports. Container groups are allocated to bays, aiming at the minimization of total travel distances, and balance of workloads among blocks.

Seung Hwan Won et al. [29] proposed an integrated decision-making framework for yard-planning problem in container terminals, in which various resources were considered, including spaces, YCs, yard trucks and driving lanes. A minimum-cost network-flow model was built, with an objective to minimize the total costs of container flows.

Deniz Türsel Eliiyi et al. [31] considered an outbound container storage problem at some container terminal in Turkey. They proposed a two-stage approach, similar to Lu Chen and Zhiqiang Lu [23], in which containers of the same vessel are assigned to several bays in the storage yard in the first step, while their objective was that containers of the same vessel should be located as concentrated as possible.

Omor Sharif and Nathan Huynh [32] modeled a container terminal as a network of berths, gates and blocks. The storage space allocation problem of containers among blocks was solved in real-time, with the objectives of workload balance and travel distance minimization.

L Zhen [34,36] made an exploratory study in yard template designing problem of transshipment hubs under uncertainty. This research was a further research of Liang Ping Ku et al. [20], for reason that the berthing location was treated uncertain, aside from the arrival time of vessels.

Dissimilar to the conventional container terminals, in automated ones the storage allocation of containers are decided in real-time (online) by rules. To the extent of our knowledge, no research could be found on the storage allocation problem in automated container terminals. In this regard, simulation is a usual approach to study different allocation strategies.

Rommert Dekker et al. [10] investigated into online container stacking policies at an automated container terminal. Two stacking strategies, namely random stacking and category stacking, are compared by simulation. The results showed that, the former strategy stacking containers of the same category in the same stack outperformed the latter one stacking containers in random stacks for fewer reshuffles.

Bram Borgman et al. [18] evaluated the performance of random strategy and leveling strategy on stacking a container by simulation. The difference of the two online strategies was, whether the stacking position of a container was decided in random, or an unfilled stack got the priority in case that the top container was to be retrieved later. Trade-off could be observed in simulation experiments, between travel time of the cranes and the probability of reshuffles.

Taejin Park et al. [22] proposed an online search algorithm for inbound containers which dynamically adjusts and optimizes a stacking policy. For every incoming container to the storage yard, the stacking block was determined depending on the short-term workload of cranes in each block, while the stacking position was determined in comprehensive considerations.

Mingzhu Yu and Xiangtong Qi [33] focused on the retrieval efficiency of inbound containers in automated

container terminals. The stacking positions of containers were determined following a segregation strategy, in which containers were always stacked upon another one that is to be retrieved later. It is verified in this paper that, the approach allocating multiple container batches together outperformed the approach allocating one batch at a time.

The ultimate goal of our research is, to discuss the potential benefit of storage allocation in automated container terminals, in case that it could be applied instead of online strategies in future automated container terminals. However, only part of the storage allocation problem is studied in this paper; the remaining part is left to further research, so as the benefit evaluation of storage allocation plans when applied instead of online strategy. The works in this paper could be characterized as follows.

- » It is put forward in this paper the storage allocation problem in automated container terminals, a novel problem to the existing ones in conventional terminals. Moreover, it is proposed that this problem could be decomposed into two levels.
- » A mixed integral programming model with rolling horizontal approach is proposed for the upper level of the problem.
- » A modified particle swarm optimization (PSO) algorithm is proposed to solve the simplified model. The modified algorithm makes a multi-stage search process with multiple swarms, hence better solution could be found than using the original PSO algorithm. Moreover, for large scale problems in particular, a simplified coding method is proposed, which leads to shorter particle length and faster convergence.

PROBLEM DESCRIPTION

Storage allocations in conventional and automated container terminals are similar problems while not the same. The priorities of the two allocations varies, according to differences in block layouts and operation modes between the two terminals. Therefore, storage allocation in automated container terminals could be treated a novel problem from that in conventional ones.

STORAGE ALLOCATION IN CONTAINER TERMINALS

To launch the container handling works of vessels, a series of decisions must be made before the handling process whether in a conventional terminal or an automated one. According to Steenken D et al. [8], these decisions include but are not limited to:

Berth allocation: to allocate a berth for every vessel before its arrival, and to predict the departure time of the vessel.

Storage allocation: to reserve spaces in the yard for the temporal storage of outbound, inbound and transshipment containers.

Stowage planning: to assign the containers stacked in the yard and to be loaded one by one to the respective cells of their

destination vessel.

Quay crane scheduling: to allocate quay cranes to vessels or sections of vessels, and to determine the handling sequence of each quay crane.

For every vessel to visit the terminal, the four decisions above are executed in sequence, as shown in Fig. 1 below. Berth allocation is the first decision to be made, after which the storage allocation, in which the space for containers to be loaded onto and unloaded from the vessel is reserved on the basis of berth allocation. Generally, the spaces reserved for containers will not be quite far from the berth allocated to the vessel, and the storage allocation must be done before the terminal starts to collect outbound containers for this voyage. Stowage planning is carried out after the cut-off time of a vessel, when the container collection of this voyage expires. The quay crane scheduling is made later before the vessel handling.

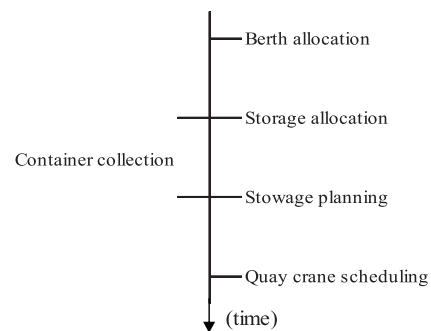


Fig. 1 Sequence of main decisions before vessel handling

The storage spaces in the storage yard of a container terminal is separated into blocks, which are continuous storage spaces separated by roads or some other spaces not for storage. Blocks in a container terminal are lined up, while the layouts of blocks differs in conventional terminals from automated ones. The storage space in a block is a three-dimension bay-row-tier space, as is shown in Fig. 2 below. Generally, a 20ft container is stored in a single cell, and a 40ft container is stored in two neighboring cells in a same row and a same tier. Two neighboring bays in the same block is often called a double bay, which is designated for containers larger than 20ft. Containers even larger than 40ft are put in double bays in the two double

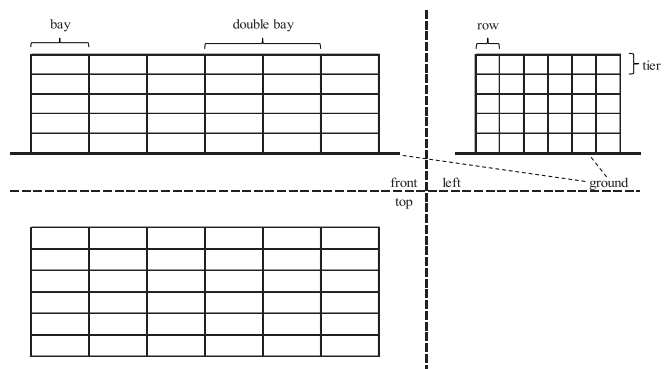


Fig. 2 Orthographic view of a block

bays at the ends of a block. There are no brackets between neighboring cells in a block, hence containers located in the same bay (or double bay) and the same row are piled up as a stack. It is the practice of container terminals that, containers located in the same stack should be of the same size.

To the convenience of the following stowage planning, spaces in a container terminal are allocated to containers in units of container groups when making a storage plan. Generally, containers from or to a vessel is divided into container groups, and some blocks or some bays in some blocks are allocated to such groups but not to single containers. Within a container group, specific stowage space is not a major concern when making the plan. Outbound containers are usually grouped according to sizes, types, destination ports, weight levels, and etc., hence containers collected in a same stack could be interchangeable when making a stowage plan, which is an effective method to reduce reshuffles when loading them onto the vessel. Similarly, inbound containers are also grouped according to sizes, types and owners. Because inbound container has less groups, it is a common scene in container terminals that, the scale of inbound container groups are often much larger than the scale of outbound groups.

The fundamental objective of storage allocation is, to optimize vessel handling efficiency while ensuring that storage capacity of the terminal is not exceeded. Handling of every container from or to a vessel is a multi-stage process using Quay Cranes (QC), Yard Cranes (YC) and horizontal movers (HM), which is the handling system of containers. To load a container from some block to a vessel, an YC picks it up and puts it onto a HM, which sends this container to a QC according to some work plan, after which the container is loaded by that QC onto the destination vessel. A reversed process is carried out when unloading a container from a vessel to some block. As the scarcest and most efficient equipment in the handling system, QCs should be taken full advantage of to maximize the handling efficiency of the whole system. In other words, QCs should be kept handling as long as possible. For this purpose, attentions must be paid to following two key features.

- Numerical relationship between QC and YCs.

Since the technical handling efficiency of QCs are much higher than that of YCs, it is a major key point in ensuring the handling efficiency of QCs to keep a “one-to-many” relationship between QCs and YCs. Thus, for each QC, it is preferred that multiple YCs could be designated to a QC handling thread, otherwise the operation of the QC would be interrupted because of lack of YC handling capacity. However, it is hard to determine the exact best number of YCs that is needed to serve a QC, for reason that YCs could handle some extraneous containers when serving the QC, i.e. containers handled by some other QC, or containers collected from / delivered to somewhere out of the terminal.

- Turnaround times of HMs.

Turnaround times of HMs is another key point in keeping a high QC handling efficiency. In the visit interval of a QC,

a HM spends time at some YC and on the road, Hence the turnaround time of a HM could be lengthened by far travel distance, blocks on the road, and long waiting time at some YC. Repeated long turnaround times may interrupt the handling of QC, no matter how many HMs are designated to it.

A major difficulty in making a storage plan is that, the handling launch time and duration of the container groups could not be estimated accurately when making a storage plan. Since the stowage plan and QC schedules are made afterwards, handling sequences of the container groups and containers within each group are unknown for storage space allocation. If two groups are located in spaces where only one YC is allowed to work at a time, an YC will have to take account of them simultaneously when they are handled at the same time, which causes inefficiency.

Due to differences in layouts and operations, as are listed below, storage allocation problems in conventional container terminals and in automated ones are a bit different. Except for the priorities on travel distance, work balance and space utilization, the two problems pay attentions to different targets, which are explained in the rest of this section.

- Direction of blocks and locations of Input / Output (I/O) points
- Numerical relationship between YCs and blocks
- Operation modes of YCs

COMPARISON OF STORAGE ALLOCATIONS IN TWO KINDS OF TERMINALS

Storage layouts in conventional container terminals and automated ones are different [30]. Fig. 3 is a typical storage layout of a conventional container terminal. Blocks are positioned in parallel to the quay line, and the I/O points are scattered on a path lane for HMs alongside the block. At every block, storage and retrieval of containers to or from one bay are executed at a same I/O point. The YCs could be either Rail Mounted Gantry Cranes (RMG) that are confined to blocks on the same rail track, or Rubber Tired Gantry Cranes (RTG) that could move freely between all the blocks. As a result, the number of YCs that work at the same block is variable at different times. The presence of an YC and a HM at the I/O point is required when storing or retrieving a container, and no gantry move is needed in the handling process.

Travel distance between allocated spaces and the berth position of vessels is a major objective to be optimized in storage allocation problems in conventional container terminals. This distance should be minimized to reduce turnaround time. Work balance among blocks is also paid much attention to, since it contributes a lot to the “one-to-many” relationship between QCs and YCs. However, due to the fact that multiple YCs are allowed to work simultaneously at one block, workload balance is often carried out among sections of some adjacent bays within the blocks, called “sub-block” as in [4, 12, 18, 20], in which only one YC is permitted at most. Space utilization is the least important priority to be considered. In most cases,

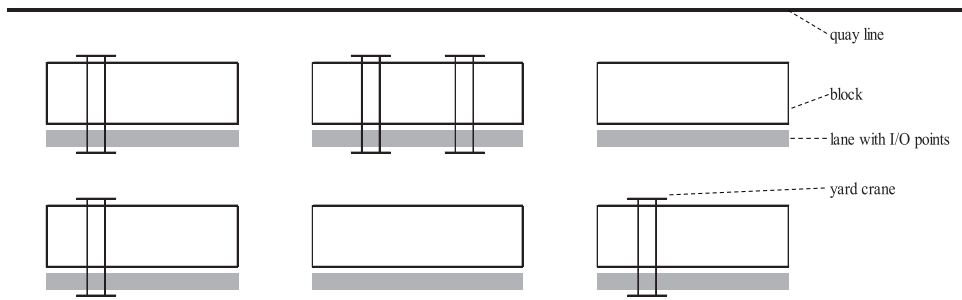


Fig. 3 Storage layout of a conventional container terminal

it is only treated as a constraint of the allocation problem.

Besides the three priorities above, there are other optimization factors, and some principles should be followed when making a storage plan, which are listed below.

- Interference of HMs between sub-blocks in a same block should be considered. Since the I/O points of a same block are located on the same path, hence a HM may be blocked by some other HM ahead, either at its destination I/O point or still on the way, which adds to the HM's turnaround time. It is recommended that inbound containers and outbound containers are allocated with spaces in different blocks, hence interferences between HMs for loading and unloading in the same block could be prevented.
- Distance of the nearby bays allocated for the same container group should also be minimized, which is related to gantry movements of YCs when handling these containers in sequence. Spaces allocated in a same sub-block to the same container group should attract neighboring bays to store similar containers, hence the gantry movement of YC will not be very long, which leads to shorter waiting time of HMs at the YC, and shorter turnaround time of HMs as a result. Likewise, containers allocated to the same bay should be of same size.
- Large container groups should be located in multiple blocks, or multiple sub-blocks distant to each other in some same block. Owing to the fact that containers in a same group are always handled in succession by one QC, locating them in such a scattered way enables simultaneous handling of multiple YCs, as to keep the "one-to-many" relationship between QC and YCs.

Fig. 4 is a storage layout of an automated container terminal. Distinct from the conventional ones, terminal blocks are positioned perpendicular to the quay line, and I/O points are located at both ends of the blocks. Seaside I/O points are designated for vessel loading and unloading, while landside points are for container collection and delivery. YCs in this layout are almost all RMGs characterized by high gantry moving speed, known as Automated Stacking Cranes (ASC). The Number of YCs in every block is fixed, usually 2 YCs in one block, in which the seaside YC focuses on vessel loading and unloading, while the landside one is for container collection and delivery in the daylight, and storage reshuffling at night. Differ from those in the conventional terminals, YCs in this layout following a double-cycling operation mode. An YC could pick up a container from some bay in the block, carry it to the I/O point and put it down, then pick up another container from the I/O point, carry it to some bay in the block and put it down. In this way, an YC could handle stacking and retrieving containers alternatively, known as the double-cycling operation mode. And YC gantry movement is inevitable in stacking or retrieving containers, and the operation time of different bays varies. The further one bay is to the I/O point, the longer the YC operation time will be.

In an automated container terminal, travel distance between allocated spaces and the berth position of vessels is no longer a major optimization objective of the storage allocation problem. HMs for vessel handling are confined between the quay line and the blocks, hence the travel distances are much shorter than those in the conventional terminals. Owing to the fact that there is only one YC in each block for vessel handling, work balance among blocks should be seriously considered. Still, space utilization is the least important objective to be considered. Consequently, there are some differences between

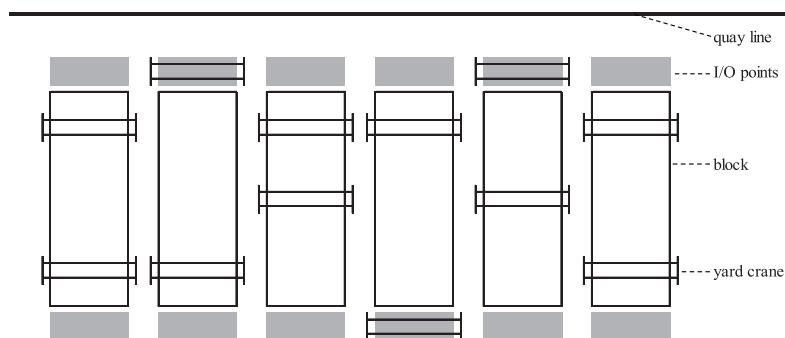


Fig. 4 Storage Layout of an automated container terminal

storage allocations in conventional terminals and in automated ones, which are listed below.

- Influences between I/O points in automated container terminals are much less than those in the conventional ones, to the extent that it could even be omitted when making a storage plan. Consequently, there is no need to distinguish inbound and outbound containers, and they could be located in the same block.
- It is no longer recommended that containers of the same group should be located in neighboring bays, and containers of different sizes could be allocated in the same bay. However, it will be appreciated that container groups which is likely to be handled simultaneously are allocated with bays closer to the I/O point.
- Storage allocation in automated container terminals could be treated as a two level problem. The upper level is to determine only the number of available stacks allocated to each container group; the lower level decides the specific locations of these stacks in each block. Solutions of the upper level are responsible to the “one-to-many” relationship between QCs and YCs, the travel distance and the space utilization. Solutions of the lower level are only related to YC handling times of the containers, which have nothing to do with the upper level model. As a result, the storage allocation problem in automated container terminals could be treated in a two-level manner, in which the upper level and lower level problems are solved separately to improve computing efficiency.

MATHEMATICAL MODEL

Storage planning in automated container terminals could be abstracted into planning models executed following a rolling-horizon approach, a common way in making storage plans in the conventional container terminals. As mentioned before, only the upper level model is proposed in this section.

PLANNING HORIZON AND ASSUMPTIONS

Automated container terminals runs in two modes by day and at night. Although vessel handling is carried out 24 hours a day in the workdays, container collections and deliveries are handled only in the daylight. During the night hours when no HM is allowed into the terminal, containers stored in the blocks are reshuffled if necessary. As described in [30], mainly two kinds of reshuffles are executed overnight. One is called re-marshalling, in which some containers to be loaded in the coming day are moved to a new location near the seaside end. The other is called pre-marshaling, in which some containers unloaded from vessels are carried in the opposite direction. Thanks to the reshuffles during the night, outer HM and vessel handlings in the next day could be fastened for shorter YC gantry movements.

Due to the operation mode alternations every day, in automated container terminals, a daily storage plan is needed. As shown in Fig. 5 below, the horizon of a storage plan covers a

whole day, while the deadline of a storage plan for the coming day is laid to the beginning of the current night, before the reshuffles are executed. A reshuffle plan should be made just before the storage plan, hence the storage spaces available for the inbound containers in the next day could be determined. This pattern goes on continuously, howbeit the reshuffle plans are not content of this paper.

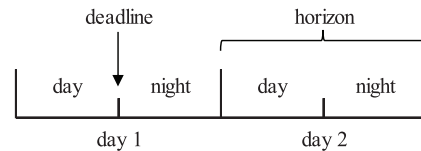


Fig. 5 Deadline and horizon of a storage plan

A mixed integral programming model is proposed here for every storage plan, based on several assumptions as are listed below:

- (1) The total volume capacity of the blocks are always enough for storage allocation.
- (2) Container information is all known when making a new storage plan, so as the available storage spaces in each block.
- (3) The berth plan is known, including the berthing positions of the corresponding vessels, so as the berthing start times and berthing end times. On this basis, travel distances between container groups and blocks could be calculated, and the handling time windows of container groups are determined as the berthing start time and berthing end time of the corresponding vessel.
- (4) Only 20ft and 40ft general containers are considered in this model. Containers of other sizes or types are excluded, for instance 45ft containers, refrigerated containers, and etc.
- (5) Only inbound and outbound containers are considered in this model. Transshipment containers are not concerned in this paper.
- (6) Reshuffles at night could be always well executed, hence outbound containers to be loaded in the coming day could be all positioned in the storage spaces as defined in the storage plan, and inbound containers unloaded in the current day could be relocated to some positions near the landside I/O points.
- (7) Containers are divided into groups before storage planning, according to rules as described in section 3.1. Storage spaces are allocated to container groups, but not to individual containers.
- (8) Spaces are allocated in stacks. One stack could be allocated to one container group at most. Moreover, number of stacks needed for every container group is pre-determined.
- (9) Container groups are preferred to be divided and scattered in a recommended number of blocks. The value of these recommended numbers are related to the scale and direction of the container groups, and could be determined when making a storage plan. Based on this division, a new concept, namely allocation unit, is used in the model instead of container group. The number of allocation units for every container group equals to its recommended division

number, and the total number of stacks required for the allocation units equals to that of the container group.

For better understanding of the assumptions, some explanations of assumption (3), (8) and (9) are given here as a further step. In assumption (3), it is believed that the handling time of container groups cannot be accurately predicted since the handling sequence among them are not decided when making a storage plan. As a result, the handling time windows of container groups are determined according to the berth start time and end time of the corresponding vessels in this paper. Shorter time window for container groups will be better in case that the handling sequences of container groups could be pre-determined. In assumption (8), the number of free stacks needed for a container group depends on the number of containers in the group, and the volume of a free stack. In this paper, this number is set as the minimum of free stacks that could hold all the containers in the group. Larger numbers will also be feasible, while leading to less space utilization. In assumption (9), division numbers of the container groups are fixed to recommended values. This assumption causes a smaller solution space excluding storage plans in which container groups could be scattered in a number of blocks that differs to the recommended value. Adoption of this assumption leads to a great reduction of decision variables needed in this model. However, in case that the spaces needed for allocation is close to the total available spaces in the terminal, the solution of this model may be probably not a satisfactory storage plan, or even no feasible solution could be found at all.

THE MODEL FORMULATION

Notations used in the upper level model are listed as follows.

- I total number of allocation units, indexed by integer i and j . $1 \leq i, j \leq I$.
- G total number of groups, indexed by integer g . $1 \leq g \leq G$.
- B total number of blocks, indexed by integer b . $1 \leq b \leq B$.
- S_i size of allocation unit i . If unit i consists of 20ft containers, $S_i = 1$; else $S_i = 2$.
- L_i direction of allocation units. If unit i consists of outbound containers, $L_i = 1$; else $L_i = 2$.
- WS_i start of time window of allocation unit i .
- WE_i end of time window of allocation unit i .
- D_{ib} travel distance between block b and the handling point of allocation unit i .
- DU_i maximal travel distance between unit i and the blocks.
- DL_i minimal travel distance between unit i and the blocks.
- N_g number of stacks needed for group g .
- R_{ig} relationship between unit i and group g . If unit i belongs to group g , $R_{ig} = 1$; else $R_{ig} = 0$.
- P_g division number of group g .
- F_b maximal number of 40ft free stacks in block b .
- T_b maximal number of 20ft free stacks in block b .

- δ_{ij} time window overlap of allocation units i and j .
- μ_{ij} total time length of time windows of allocation units i and j .
- η_{ij} the time span of time windows of allocation units i and j .

Decision variable are shown as follows.

- x_{ib} 0-1 variable. If allocation unit i is allocated to block b then $x_{ib} = 1$, else $x_{ib} = 0$.
- n_i nonnegative integer variable, for the stacks needed for allocation unit i .

The constraints in this stage are listed as follows.

$$\sum_i x_{ib} n_i (S_i - 1) \leq F_b, \forall b \quad (1)$$

$$\sum_i x_{ib} n_i S_i \leq T_b, \forall b \quad (2)$$

Equation (1) and (2) are for the volume capacities of blocks. Besides the volume capacity constraint on TEU as in Equation (2), special volume capacity constraint for 40ft stacks should also fulfilled as well, hence the containers allocated to the block could be indeed placed in the block. An example is shown in Fig. 6, which is a top view of a block section. In this block section of 4 bays and 3 rows, 6 stacks are occupied by some other containers and could not be used in the current storage plan, as the gray cells in the figure. There seems to be enough spaces for 3 40ft stacks by checking the TEU capacity, however only 1 40ft stack could be actually placed in this block section.

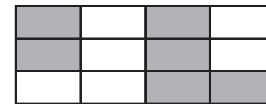


Fig. 6 Illustration for constraints on block capacity

$$\sum_i n_i R_{ig} = N_g, \forall g \quad (3)$$

Equation (3) is a constraint on the total stack number of allocation units in a same group.

$$\sum_b x_{ib} = 1, \forall i \quad (4)$$

Equation (4) means that every allocation unit should be allocated to one block only.

Some objectives are listed as following.

$$\text{Min } f_1 = \text{Min} \frac{\sum_{i,b} x_{ib} n_i (D_{ib} - DL_i)}{\sum_{i,b} x_{ib} n_i (DU_i - DL_i)} \quad (5)$$

Objective (5) is for the minimization of travel distance. Since travel distance is not a major priority in making a

storage plan, this objective is presented in a normalized form.

$$\text{Min } f_2 = \text{Min} \sum_i \left| n_i - \sum_g (B_{ig} N_g / P_g) \right|^2 \quad (6)$$

Objective (6) means that it is preferred that the number of stacks in need for every container groups are divided evenly.

$$\text{Min } f_3 = \text{Min} \sum_{b,i,j} x_{ib} x_{jb} \delta_{ij} \text{Min}(n_i, n_j) \quad (7)$$

$$\delta_{ij} = \begin{cases} 0, i = j \text{ or } L_i \neq L_j \\ \frac{\mu_{ij} - \eta_{ij} + |\mu_{ij} - \eta_{ij}|}{2(WE_i - WS_i)} + \sum_{b,g} x_{ib} x_{jb} B_{ig} B_{jg}, \text{ otherwise} \end{cases} \quad (8)$$

$$\mu_{ij} = WE_i - WS_i + WE_j - WS_j \quad (9)$$

$$\eta_{ij} = \text{Max}(WE_i, WE_j) - \text{Min}(WS_i, WS_j) \quad (10)$$

Objective (7) means that allocation units of the same direction with overlapping handling time windows are better not allocated in the same block. Equation (8-10) are definitions of parameters δ_{ij} , μ_{ij} and η_{ij} respectively. These equations are concluded from all the possible overlap schemes of two units' time windows, as is shown in Fig. 7 below. It is supposed in the figure that, the time window length of unit i is longer than that of unit j . The time window of unit i starts earlier than that of unit j in scheme (a), (b) and (c), while it is opposed in scheme (d) and (e). In scheme (a) handling of unit j starts later than the end of unit i 's time window; in scheme (b) handling of unit j starts before the end of unit i 's time window; in scheme (c) handling of unit j ends before the end of unit i 's time window. In scheme (d), handling of unit j ends before unit i 's time window, while in scheme (e) handling of unit j ends after the beginning of unit i 's time window. It could be concluded from all the schemes that, overlap occurs when and only when sum of time window duration of the two units is longer than their time span, which is the time range between the earlier start time and the later end time of the two windows. For reason that allocation units of the same container group are preferred to be allocated in different

blocks, equation (8) is designed that δ_{ij} will be larger than 1 when and only when unit i and j belongs to a same group and allocated to a same block.

The global objective is shown as equation (11) below, in which α_1 , α_2 and α_3 are coefficients for objectives (5), (6) and (7) respectively.

$$\text{Min } f = \text{Min} \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 \quad (11)$$

SOLUTION METHOD

Owing to the fact that the objectives formulas f_2 and f_3 in the model are not linear, the model cannot be solved using those well-developed linear solvers, including CPLEX as an instance. Intelligence algorithms are feasible methods in solving these none linear optimization model. Considering that the scale of storage allocation problems in practical terminals may be quite large, Particle Swarm Optimization (PSO) seems a promising optimizing mechanism for fast convergence. Since single PSO searching process will probably end in a local optimum, a PSO-based progressive (PSOP for short) algorithm is proposed in this paper as the solution method of the model raised in section 3. In this progressive algorithm, the best solution found in the current searching process is treated as knowledge in a next searching process, which attempts to get a better solution than the current stage.

ENCODING REPRESENTATION AND GROUP INITIALIZATION

In this algorithm, a particle is encoded in an array, which has a length of twice the number of allocation units. The former half of this array are real numbers, the minimal integer larger than which indicates the number of block that the units are allocated to. The latter half of the array are integer numbers, indicating the number of free stacks needed for the allocation units. Fig. 8 gives an example of a particle for a solution including two allocation units. In this particle, allocation unit 1 needs 7 free stacks and is allocated to block 1, while allocation unit 2 needs 6 free stacks and is allocated to block 6. As a distinction of the two halves of a particle, notation pf is used for the former half of the array, while pl is

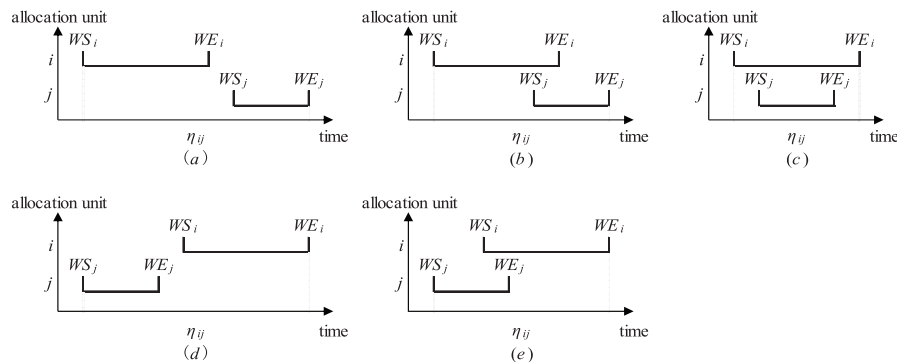


Fig. 7 Possible time window overlap schemes of two allocation units

for the latter half. The speed of a particle is coded as an array of real numbers. The length of the speed array equals to the number of allocation units.

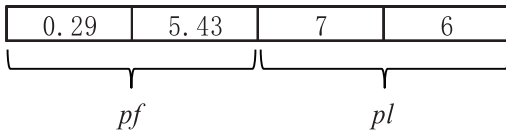


Fig. 8 An example of particle encoding

The particles, speeds, the swarm's best array and the particles' best arrays, are initialized in the following manners when a searching stage starts. Elements in the former half of a particles are randomly valued in range $(0, B)$, and those in the latter half are given an integer number, valued according to the container group which the allocation units belong to. In order to minimize the value of f_2 in the initialization, number of free stacks for the allocation units in a same group g are valued as integers near N_g/P_g , and the sum of these numbers are kept as N_g at the same time. The speeds are valued randomly in range $(-B, B)$. The particles' best arrays are set the particles themselves, and the swarm's best array is set the current best particle in the swarm.

FITNESS FUNCTION AND PARTICLE REPAIR TECHNIQUE

With the encoding representation and group initialization method described above, constraint (3) and (4) are already fulfilled. However constraints (1) and (2) are not satisfied yet: the capacity of some block may be exceeded so that too many allocation units are allocated to it at the same time, which leads to an infeasible solution. Due to the random initialization and iteration, dissatisfactions of particles with constraint (1) and (2) are inevitable in the searching process. Hence, a punishment is added to the fitness function for the volume exceeding of blocks, and a repair technique is proposed for infeasible particles.

The fitness function F of particles is defined as equation (12) below. In this equation, f is defined as in equation (11), E^F is the total number of 40ft stacks exceeding the block capacities, E^T is the total number of stacks in TEU that exceed the block capacities. Notation α_p is the punishment coefficient, valued much larger than α_1 , α_2 and α_3 as in equation (11).

$$F = f + \alpha_p (E^F + E^T) \quad (12)$$

The particle repair technique is designed with greedy strategy. The repair procedure starts with the array of current particle. The technique changes the array step by step. A new array is generated from the current array by moving one container from an exceeded block to another, in which an allocation unit of the same container group is allocated. Every possible move makes a new array, and the fitness value is calculated. Only the best array is kept as the new particle array, and other arrays are deleted. In case that the current array is

feasible or no better than the last one, it is used to replace the array of the particle and the repairing ends. Otherwise, the repairing goes on to a next step.

Although the repairing technique is effective in changing an infeasible array into a feasible one, the burden of repairing process on time consumption is quite heavy. Usage of repairing technique in the searching processes should be limited, otherwise the computation time of the algorithm will be unacceptably long. Consequently, this technique is applied only to some infeasible particles, the fitness value of which are no larger than a pre-defined value. Moreover, repairing is not executed in every iteration of the searching process.

ITERATION AND PROGRESSIVE SEARCHING STRATEGY

The total number of iterations is fixed in a searching process. In the first searching process, the former halves of particles are iterated following the standard iteration mechanism, as is presented in equation (13) and (14), and the latter halves are set as the initial arrays in every iteration. In the equations below, pf is for the former half of a particle array and v is for the speed array. pbf is for the former half of the particle's best array and sbf is that of the swarm's. w_v , w_p and w_s are coefficients, which decrease linearly as the iterations go on. Fitness value of particles are calculated in each iteration, and the best arrays of particles and the swarm are updated once better arrays are found in the searching process.

$$v = w_v \cdot v + w_p (pbf - pf) + w_s (sbf - pf) \quad (13)$$

$$pf = pf + v \quad (14)$$

The best solution found in the first searching process is probably not the best solution of the model. In search of a better solution, progressive searching are executed repeatedly, till the fitness value of the best solution found in the current searching process is not much better, i.e. to a predefined extent, than that of the last one. Mechanisms of a progressive searching and the first searching are almost the same. The only difference of progressive searching to the first one is the determination of the swarm's best particle. In the first searching process, this particle is a result of the current searching, while in a progressive searching process, the swarm's best particle is inherited from a last searching process.

Similar to the repair technique, progressive searching is as well a potential burden to the time consumption of the searching process, if not properly treated. The computation time of the algorithm may be quite long when progressive searching are executed repeatedly, while the decreases of best fitness values between two successive searching processes are very small. To avoid this, requirements are attached to the decreasing margin of best fitness values between successive searching processes. At the beginning of a searching process, a required best value F^* is calculated according to the best

fitness value of a last searching process F_{last} and an attenuation coefficient d , as is shown in Equation (15) below. In case that the best fitness value of this searching process is no more than F' , F_{last} is updated and another progressive searching starts. Otherwise, the algorithm terminates, returning the best particle array found in the last two searching processes as the solution.

$$F' = (1 - d)F_{last} \quad (15)$$

The outline of the PSOP algorithm is shown by flowchart as in Figure 9.

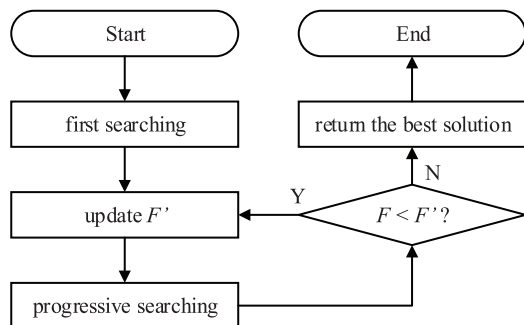


Fig. 9 Outline of the PSOP algorithm

NUMERICAL EXPERIMENTS

In order to verify the effectiveness of the PSOP algorithm, and to test the performance of the PSOP algorithm for storage allocation problems in different scales, experiments are carried out in solving examples generated in three terminals, different in berth lengths and total number of blocks, as shown in Table 1. In each terminal, a berth plan of vessels is first generated, so as the free spaces available for storage allocation in every block. On this basis, examples are generated in random according to various capacity rates (noted as r_s), which equals to the value of spaces needed for allocation divided by the total available spaces in the terminal. 10 examples are generated for every r_s , ranging from 0.05 to 0.8 with an interval of 0.05, and each example is solved 20 times with PSOP algorithm. All the experiments are executed in a desktop work station with 28 GB memory and 2 Inter Xeon E5520 @ 2.26 GHz processors.

The main purpose of the experiments in this section is to simulate applicable conditions of the model and algorithm proposed in this paper. As described in assumption (9), this model is a simplification of the practical storage allocation problem, hence it is not assured that a feasible solution could always be found, especially when value of r_s is rather large. Since the fitness value of the solution found is with little practical significance, a secondary attention is paid to the solving times only. Moreover, the benefit and burden of the progressive searching processes in solving the examples is investigated, so as the burden of repair processes. The parameters of the algorithm are set as follows. There are 30

particles in a swarm, and the iteration time of every searching process is fixed to 100. The values of α_1 , α_2 , α_3 and α_p are set 1, 1, 2 and 10 respectively. The value of w_v is fixed to 0.7, while values of w_p and w_s decrease from 1 down to 0 in every searching process. Repairs are executed every 10 iterations, while applied only to particle arrays of which the value $E^F + E^T$ is no more than 15. The value of d is set to 0.05.

Table 1 Scales of terminals

terminal scale	berth length (m)	total number of blocks
small	400	9
medium	700	18
large	1000	27

EXPERIMENTS OF SMALL TERMINAL

A berth plan of 4 vessels for a small terminal is shown in Fig. 10 below.

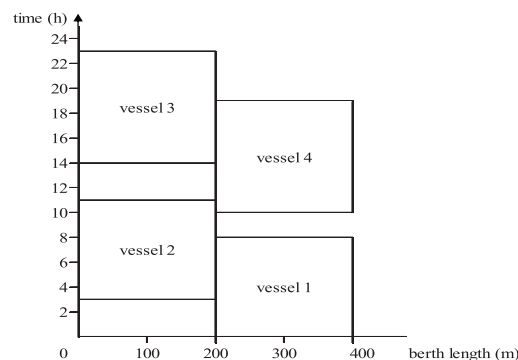


Fig. 10 Berth plan of the small terminal

The performance of the PSOP algorithm in the small terminal is shown in Fig. 11.

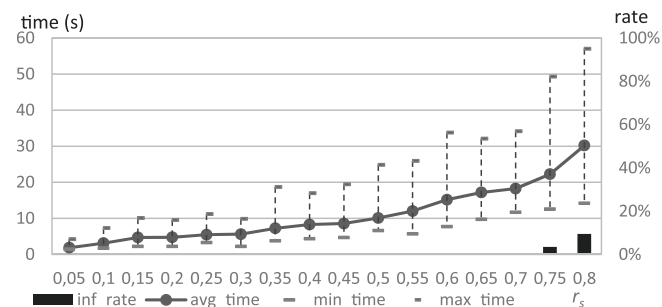


Fig. 11 Performance of PSOP in the small terminal

Attentions are paid to the solving times of the algorithm, and the chances of infeasible solutions in different capacity rates. Notations avg_time, min_time and max_time are the average, maximal and minimal solving times respectively, of which the values are presented by the left vertical axis.

Notation inf_rate is the chance of infeasible solutions among the examples, of which the value is presented by the right vertical axis. It could be observed from the figure that, the solving times increase with the capacity rate, and a solution could always be obtained in one minute. For examples with a capacity rate no more than 0.7, a feasible solution could be assured.

The effect and burden of progressive searchings are shown in Fig. 12. Notations avg_t_first and avg_t_prog are the average time of first searching process and progressive searching processes of the examples respectively. They are prepared in percentage by histograms. Notation avg_fit_dec is the percentage of fitness decrease after first searching processes. It could be observed from the figure that, progressive searching processes lead to a 8% decrease of fitness value on average, while taking roughly 1.4 times as long as the time consumed by first searching processes.

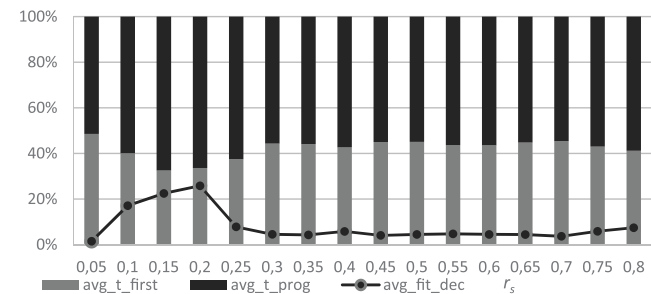


Fig. 12 Effect and burden of progressive searching in the small terminal

The burden of repair process is shown in Fig. 13 below. Notation avg_t_repair and avg_t_comp are the average time spent by repair processes and other computation processes, which are presented in percentage by histograms, and valued by the left axis. Notation avg_n_repair is the average times that repairing processes is executed in solving an example. Except for an inflection point near capacity rate 0.6, the average number of repairs executed in solving an example increase as the capacity rate raises. The percentage of time spent in repairing goes up with the capacity rate, up to roughly 20% when the capacity rate is 0.8.

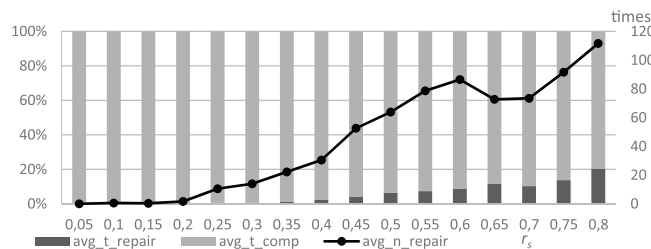


Fig. 13 Burden of repair technique in the small terminal

EXPERIMENTS OF MEDIUM SIZED TERMINAL

A berth plan of 6 vessels for a small terminal is shown in Fig. 14 below.

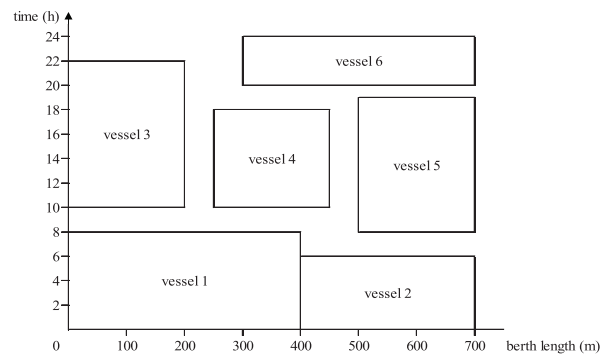


Fig. 14 Berth plan of the medium sized terminal

The performance of the PSOP algorithm in the medium sized terminal is shown in Fig. 15. According to this figure, the solving times increase with the capacity rate, and a solution could always be obtained in 100 seconds. Examples with a capacity rate no more than 0.65 will be assured with a feasible solution.

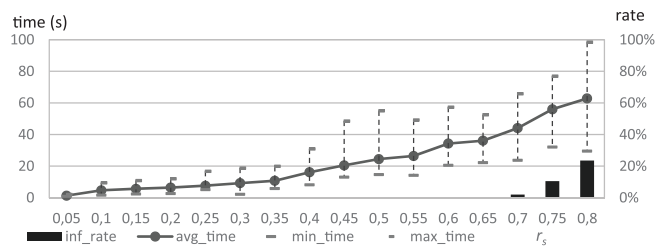


Fig. 15 Performance of PSOP in the medium sized terminal

The effect and burden of progressive searching are shown in Fig. 16 below. Here progressive searching processes lead to a 16% decrease of fitness value on average, while taking roughly 1.7 times as long as the time consumed by first searching processes.

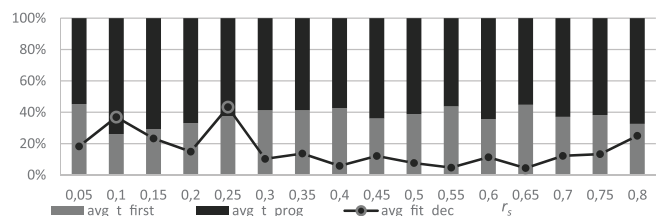


Fig. 16 Effect and burden of progressive searching in the medium sized terminal

The burden of repair process is shown in Fig. 17 below. Similarly, Except for an inflection point near capacity rate

0.45, the average number of repairs executed in solving an example increase as the capacity rate raises. The percentage of time spent in repairing goes up with the capacity rate, up to roughly 25% when the capacity rate reaches 0.8.

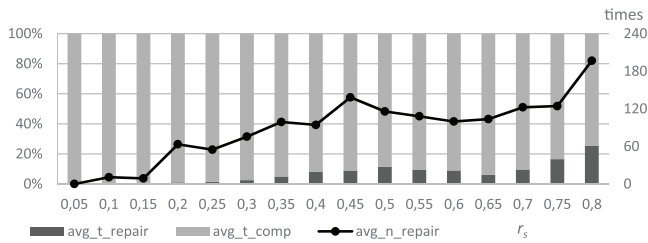


Fig. 17 Burden of repair technique in the medium sized terminal

EXPERIMENTS OF LARGE TERMINAL

A berth plan of 8 vessels for a large terminal is shown in Fig. 18 below.

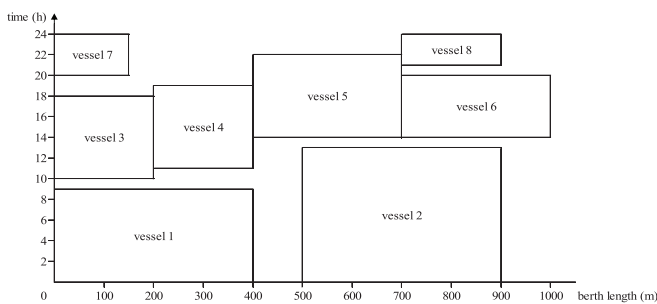


Fig. 18 Berth plan of the large terminal

The performance of the PSOP algorithm in the large terminal is shown in Fig. 19. In this figure, the solving times increase with the capacity rate, and a solution could always be obtained in 200 seconds. For examples with a capacity rate no more than 0.6, a feasible solution could be ensured.

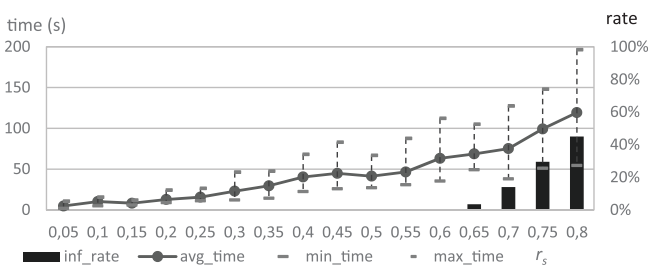


Fig. 19 Performance of PSOP in the large terminal

The effect and burden of progressive searching are shown in Fig. 20 below. Here progressive searching processes lead

to a 23.5% decrease of fitness value on average, while taking roughly 2.1 times as long as the time consumed by first searching processes.

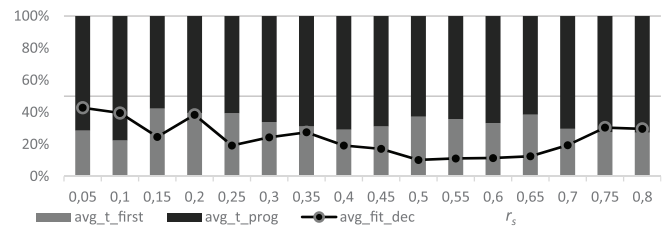


Fig. 20 Effect and burden of progressive searching in the large terminal

The burden of repair process is shown in Fig. 21. Obvious inflection sections could be seen near capacity rates 0.4 and 0.6. Generally, the average number of repairs executed in solving an example increase with the capacity rate. The percentage of time spent in repairing goes up with the capacity rate, up to roughly 28% when the capacity rate reaches 0.8.

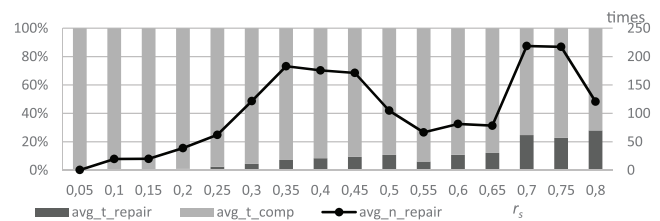


Fig. 21 Burden of repair technique in the large terminal

SUMMARY

According to the experiments above, some conclusions could be drawn as listed below.

- The model and algorithm raised in this paper is effective in solving storage allocation problems in container terminals in case that the capacity rate is not very large, and the solving time could be limited within several minutes. However, for large capacity rates, roughly more than 0.7, the solution will not be feasible every time.
- Progressive searching is an effective approach in further optimizing the result of a single PSO searching process, while the searching time of the whole algorithm will be lengthened, possibly doubled or tripled in terminal of different scales.
- Computation burden caused by repairing processes is roughly in proportion. The percentage of time taken by repairing processes could be limited to 30%, in all terminal scales and capacity rates.

CONCLUSION

This paper focus on the storage allocation problem in

automated container terminals. The problem is divided into two levels, and only the upper level problem is discussed in this paper. With some simplifications of the actual problem, a mixed integer programming model is proposed, and a progressive PSO algorithm is applied for solution. It is verified by numerical experiments that, the algorithm could converge to a feasible best solution in several minutes, in case that the ratio of space needed for allocation to the available space in the terminal is not extremely high.

This paper is a research to the storage allocation problem in automated container terminals, which is expected of great value to the automated container terminals to be built in near future. Given that almost all container information could be pre-determined by state of the art techniques as in some conventional terminals in China, pre-determined storage plans may lead to higher handling efficiency than real-time allocation strategies by rules. As a future work, the lower level of the problem and the potential efficiency increase will be discussed.

ACKNOWLEDGEMENT

This research is supported by the Young Scholar of Distinction Project for Postgraduate Students at Shanghai Maritime University (YXR2015109).

REFERENCE

1. Mi Chao, Zhang Zhiwei, Huang Youfang, Shen Yang, A FAST AUTOMATED VISION SYSTEM FOR CONTAINER CORNER CASTING RECOGNITION, *Journal of Marine Science and Technology-Taiwan*, 2016, 24(1): 54-60.
2. Mi Chao, Shen Yang, Mi Weijian, Huang Youfang, Ship Identification Algorithm Based on 3D Point Cloud for Automated Ship Loaders, *Journal of Coastal Research*, 2015, 2015(SI.73):28-34.
3. Mi Chao, Zhang Zhiwei, He Xin, Huang Youfang, Mi Weijian, Two-stage classification approach for human detection in camera video in bulk ports, *Polish Maritime Research*, 2015, 22(SI.1):163-170.
4. Mi Chao, He Xin, Liu Haiwei, Huang Youfang, Mi Weijian, Research on a Fast Human-Detection Algorithm for Unmanned Surveillance Area in Bulk Ports, *Mathematical Problems in Engineering*, 2014, Article No.386764.
5. Mi Chao, Liu Haiwei, Huang Youfang, Mi Weijian, Shen Yang, Fatigue alarm systems for port machine operators, *ASIA LIFE SCIENCES*, 2016, 25(1):31-41.
6. Kim K H, Park K T. A note on a dynamic space-allocation method for outbound containers [J]. *European Journal of Operational Research*, 2003, 148(1): 92-101.
7. Zhang C, Liu J, Wan Y, et al. Storage space allocation in container terminals [J]. *Transportation Research Part B: Methodological*, 2003, 37(10): 883-903.
8. steenken D, Voß S, Stahlbock R. Container terminal operation and operations research-a classification and literature review [J]. *OR spectrum*, 2004, 26(1): 3-49.
9. [9] Lee L H, Chew E P, Tan K C, et al. An optimization model for storage yard management in transshipment hubs [J]. *Or Spectrum*, 2006, 28(4): 539-561.
10. Dekker R, Voogd P, van Asperen E. Advanced methods for container stacking [M] // *Container terminals and cargo systems*. Springer Berlin Heidelberg, 2007: 131-154.
11. Han Y, Lee L H, Chew E P, et al. A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub [J]. *OR Spectrum*, 2008, 30(4): 697-720.
12. Weijian Mi, Wei Yan, Junliang He and Daofang Chang. An investigation into yard allocation for outbound containers [J]. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 2009, 28(6): 1442-1457.
13. Bazzazi M, Safaei N, Javadian N. A genetic algorithm to solve the storage space allocation problem in a container terminal [J]. *Computers & Industrial Engineering*, 2009, 56(1): 44-52.
14. Lee D H, Cao J X, Shi Q, et al. A heuristic algorithm for yard truck scheduling and storage allocation problems [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2009, 45(5): 810-820.
15. Nishimura E, Imai A, Janssens G K, et al. Container storage and transshipment marine terminals [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2009, 45(5): 771-786.
16. Zhang Y, Zhou Q, Zhu Z, et al. Storage planning for outbound container on maritime container terminals [C] // *Automation and Logistics, 2009. ICAL'09. IEEE International Conference on. IEEE*, 2009: 320-325.
17. Ku L P, Lee L H, Chew E P, et al. An optimisation framework for yard planning in a container terminal: case with automated rail-mounted gantry cranes [J]. *OR spectrum*, 2010, 32(3): 519-541.
18. Borgman B, van Asperen E, Dekker R. Online rules for container stacking [J]. *OR spectrum*, 2010, 32(3): 687-716.
19. Ku L P, Chew E P, Lee L H, et al. A novel approach to yard planning under vessel arrival uncertainty [J]. *Flexible Services and Manufacturing Journal*, 2012, 24(3): 274-293.
20. Li M, Li S. An effective heuristic for yard template design

- in land-scarce container terminals [C] // Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on. IEEE, 2011: 908-912.
21. Woo Y J, Kim K H. Estimating the space requirement for outbound container inventories in port container terminals [J]. *International Journal of Production Economics*, 2011, 133(1): 293-301.
 22. Park T, Choe R, Hun Kim Y, et al. Dynamic adjustment of container stacking policy in an automated container terminal [J]. *International Journal of Production Economics*, 2011, 133(1): 385-392.
 23. Jiang X, Lee L H, Chew E P, et al. A container yard storage strategy for improving land utilization and operation efficiency in a transshipment hub port [J]. *European Journal of Operational Research*, 2012, 221(1): 64-73.
 24. Jeong Y H, Kim K H, Woo Y J, et al. A Simulation Study on a Workload-based Operation Planning Method in Container Terminals [J]. *Industrial Engineering & Management Systems*, 2012, 11(1): 103-113.
 25. Lee D H, Jin J G, Chen J H. Schedule Template Design and Storage Allocation for Cyclically Visiting Feeders in Container Transshipment Hubs [J]. *Transportation Research Record: Journal of the Transportation Research Board*, 2012, 2273(1): 87-95.
 26. Li P, Sun X. Storage Space Allocation Planning in the New Container Terminal [C] // *Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012)*. Springer Berlin Heidelberg, 2013: 429-437.
 27. Lee D H, Jin J G, Chen J H. Terminal and yard allocation problem for a container transshipment hub with multiple terminals [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2012, 48(2): 516-528.
 28. Chen L, Lu Z. The storage location assignment problem for outbound containers in a maritime terminal [J]. *International Journal of Production Economics*, 2012, 135(1): 73-80.
 29. Won S H, Zhang X, Kim K H. Workload-based yard-planning system in container terminals [J]. *Journal of intelligent manufacturing*, 2012, 23(6): 2193-2206.
 30. Jiang X, Chew E P, Lee L H, et al. Flexible space-sharing strategy for storage yard management in a transshipment hub port [J]. *OR spectrum*, 2013, 35(2): 417-439.
 31. Türsel Eliiyi D, Mat G, Ozmen B. Storage Optimization for Export Containers in the Port of Izmir [J]. *PROMET-Traffic&Transportation*, 2013, 25(4): 359-367.
 32. Sharif O, Huynh N. Storage space allocation at marine container terminals using ant-based control [J]. *Expert Systems with Applications*, 2013, 40(6): 2323-2330.
 33. Yu M, Qi X. Storage space allocation models for inbound containers in an automatic container terminal [J]. *European Journal of Operational Research*, 2013, 226(1): 32-45.
 34. Zhen L. Yard template planning in transshipment hubs under uncertain berthing time and position [J]. *Journal of the Operational Research Society*, 2012, 64(9): 1418-1428.
 35. Carlo H J, Vis I F A, Roodbergen K J. Storage yard operations in container terminals: Literature overview, trends, and research directions [J]. *European Journal of Operational Research*, 2014, 235(2): 412-430.

CONTACT WITH THE AUTHOR

Mi Weijian

mwj@shmtu.edu.cn

Logistics Engineering College
Shanghai Maritime University
1550 Haigang Avenue
Shanghai
China. 201306
CHINA