Chapter 13
Container Rehandling at Maritime Container Terminals

Marco Caserta, Silvia Schwarze, and Stefan Voß

Abstract
In this paper, we review recent contributions dealing with the rehandling of containers at maritime container terminals. The problems studied in the paper refer to a post-stacking situation, *i.e.* problems arising after the stacking area has already been arranged. In order to increase efficiency of loading/unloading operations, once updated information about the state of the containers as well as of the vessels becomes available, it is possible to reshuffle the container yard, or a portion of it, in such a way that future loading operations are carried out with maximal efficiency. The increase in efficiency of loading/unloading operations has a bearing on the berthing time of the vessels, which, in turn, is a widely accepted indicator of port efficiency. Three types of post-stacking problems have been identified, namely (i) the remarshalling problem, (ii) the premarshalling problem, and (iii) the relocation problem. With respect to each of these problems, a thorough explanation of the problem itself, its relevance and its connections with other container handling issues are offered. In addition, algorithmic approaches to tackle such problems are summarized.

13.1 Introduction

Container terminals can be seen as buffers within larger logistic chains encompassing worldwide distribution systems. The major purpose of using container terminals is to serve as transshipment points. Container terminals are used as temporary storage points for containers, so that, *e.g.* unloading operations from a vessel and loading operations onto a train or a truck need not be synchronized.

Marco Caserta · Silvia Schwarze · Stefan Voß
Institute of Information Systems - University of Hamburg
Von-Melle-Park 5, 20146 Hamburg, Germany
email: marco.caserta@uni-hamburg.de, schwarze@econ.uni-hamburg.de, stefan.voss@uni-hamburg.de

J. W. Böse (ed.), *Handbook of Terminal Planning,*
Operations Research/Computer Science Interfaces Series 49,
DOI 10.1007/978-1-4419-8408-1_13, © Springer Science+Business Media, LLC 2011

Broadly speaking, a container terminal can be divided into three major areas: The quayside, *i.e.* the side in which vessels are berthed, the landside, *i.e.* the side in which other means of transportation operate (trucks, trains), and the container yard, *i.e.* the area in which containers are stored for future operations. The way in which the container yard is managed is of paramount importance in determining the efficiency of a port. Due to the fierce competition on the global market, container terminal operators are forced to increase the efficiency of their operations in order to capture and retain their customers.

As pointed out by a number of authors, *e.g.* Choe et al (2009), Park et al (2009), Stahlbock and Voß (2008), and Zhang et al (2003), some performance indicators of container terminal efficiency are: (i) the vessel berthing time, and (ii) the throughput of the quay cranes, *i.e.* the efficiency in unloading/loading containers from/to vessels. While such key performance indicators can be improved through the use of new technology, such as new equipment, terminal layout re-design, etc. the efficiency of container terminal operation can also be enhanced by *optimizing* the way in which such operations are carried out. More specifically, a great deal of attention should be devoted to the definition of efficient container stacking policies.

Moreover, in the stages of design, construction and operation of a container terminal, simulation tools play a crucial role, examples are given in, *e.g.* Gambardella et al (1998) and Yun and Choi (1999). Optimization methods, like those addressing rehandling and stacking operations at ports are suited to extend and enhance classical simulation approaches. Simulation recreates dynamical processes of real life within a (computerized) model. Experimental studies based on those models aim to gain insights on system behavior and efficient layouts of, *e.g.* manufacturing or transportation systems. Simulation has turned out to be a powerful tool for container terminal planning and its complex processes. Questions of interest are, among others, the layout of the terminal itself, including location and size of facilities (container yards, mooring, maintenance areas, etc.), design and operation of transport systems (automated guided vehicles, cranes, etc.), and modeling of container flows. Consequently, the development of simulation methods for container terminal planning has attracted research interest and respective models have been applied successfully to ports all over the world.

In terms of methodologies, several approaches like discrete-event simulation, multi-agent systems, petri-nets, or integrated simulation-optimization exist. The latter combines simulation with optimization methods by establishing the simulation tool on a higher level, having the permission to call optimization methods on a sublevel. The role of the optimization tool might differ. For instance, the optimization algorithm can take over a tactical position and be used to define and control general system parameters on an aggregate level (see Saccone and Siri (2009)). This approach is not to be mixed up with *simulation optimization* (see, *e.g.* Swisher et al (2000)) where optimization techniques are employed to fit parameters of the simulation itself *before* starting the simulation. In an alternative setting, optimization tools could be used to take decisions on a detailed, operational level. For instance, while analyzing transport systems at a container terminal using simulation, it could become necessary to call optimization tools that solve particular rehandling and stack-

ing problems to obtain information about capacity utilization of cranes and vehicles. Along the same line, while designing a terminal layout through simulation, analysis of detailed stacking operations at container yards could become necessary to determine transport and handling times. The availability of fast optimization techniques is a crucial issue of integrated simulation-optimization tools, as typically optimization methods will be called quite often. Thus, the development of efficient optimization techniques is an important matter of terminal planning.

As highlighted in Dekker et al (2006), stacking can be seen as a three-level problem. *Strategic* stacking decisions must be made with respect to the layout of the container yard, the type of equipment, and the design itself of the container terminal. *Tactical* stacking decisions are concerned with decisions that affect capacity in the medium term, *e.g.* whether a pre-stacking area should be used, whether prearrangement policies should be implemented (remarshalling, premarshalling, etc.). Finally, *operational* stacking decisions deal with the identification of slots to be assigned to containers, the rehandling of containers within the yard, the berth allocation problem, the assignment of equipment to tasks, the definition of a loading/unloading (stowage) plan, etc. In this paper, we deal with operational stacking decisions, with a special focus on collecting previous work dealing with those operations that are carried out upon an existing stack of containers.

Let us therefore suppose that a block of the container yard has already been filled with a number of stacked containers. One of the key goals of any major port is to reduce the berthing time of vessels. Therefore, if possible, it is worthwhile to use some time before a vessel reaches the port to rearrange, or prearrange containers in such a way that the subsequent loading operations are carried out in the fastest and most efficient way possible. However, some relevant issues with respect to how to carry out these types of preliminary operations arise, *e.g.*:

- How can the loading area be rearranged with the minimum amount of crane movements?
- How can it be ensured that, during the preliminary operations of rearrangement, interference among cranes is prevented?

The focus of this paper is to provide an overview of current research on post-stacking policies, *i.e.* what could be done to prepare a stack of containers to increase the effectiveness of the future loading operations. For reasons that will become clearer later on, these policies play a vital role especially in outbound operations, *i.e.* loading of containers to a vessel. However, most of the proposed approaches could also be used to deal with inbound operations, *i.e.* loading of containers to trains or trucks.

In this paper, we use the term *stacking* to indicate the policy of container handling in which containers are piled up vertically. Typically, once containers are stacked in a pile, they can only be accessed from above, in a *L*ast *I*n *F*irst *O*ut (LIFO) fashion. Consequently, a trade-off between the effective use of the container yard surface and the minimization of container handling operations arise. On the one hand, the limited stacking area of a container yard pushes in the direction of increasing the height of the stacking area to maximize the total number of containers that can be

accommodated within the yard; on the other hand, the minimization of the number of unproductive moves within the yard leads to the creation of stacks with a maximum height of one container. Therefore, the definition of a stacking policy should take into account conflicting objectives. According to Dekker et al (2006), a stacking strategy should take into account at least three objectives: (i) efficient use of storage space, (ii) efficient inbound and outbound transportation and (iii) avoidance of unproductive moves.

As pointed out by Dekker et al (2006) as well as by Taleb-Ibrahimi et al (1993), at least two different types of stacking operations can be identified: On one side of the container terminal yard, *i.e.* the quayside, *export containers* are dealt with. These are containers that arrive in the terminal in some way (*e.g.* via land) and must be loaded onto the vessel. Typically, these containers arrive somehow randomly and moreover, relevant data, like weight information, is not always given at that time. Usually, export containers arrive up to two weeks before being loaded to the vessel. However, the detailed loading sequence becomes available only shortly before the loading process. Since the stacking strategy to be used to pile up export containers can only exploit partial information, re-handling operations might be introduced to make use of information becoming available over time. On the other side of the yard, *i.e.*, the landside, containers that are unloaded from the vessel and are finally shipped via hinterland transportation are treated. These are called *import containers*. While the arrival of such containers is somehow predictable, the departure of import containers is related to the time of arrival of trucks and therefore, subject to higher variability. Consequently, using high stacking areas can lead to inefficiencies. It is worth noting though, that some authors identify a third category of containers, *i.e. transhipment containers*. These are containers that are unloaded from a vessel, temporarily stored in the yard, and loaded back to a different vessel. However, as pointed out by Kim and Park (2003), many times transshipment containers can be dealt with as if they were export containers.

A basic ingredient in the container terminal management is the *stowage plan*. A stowage plan for each ship to be loaded is defined. Such plan specifies which containers should be loaded onto the vessel and which exact position within the vessel should be occupied by each container. Factors such as containers' weight, destination, type of goods transported (*e.g.* hazardous material) are taken into account when the stowage plan is computed and released.

Whenever the stowage plan is known some time in advance, *remarshalling* operations can be performed to reduce the need for further relocation of containers. The goal of the remarshalling phase is to create stacking areas that fully take into account precedences among containers. This implies that whenever containers are to be loaded from the yard onto the vessel, the number of unproductive moves is minimum. It is worth noting though, that during the remarshalling phase containers are not retrieved and the number of containers in the stacking area is kept unchanged. The remarshalling phase is aimed at *reshuffling* containers so that in the subsequent loading or unloading phases containers can be retrieved with a minimum amount of unproductive moves.

Finally, if the configuration of a stacking area is given, regardless of whether the area is made up by import or export containers, and if containers must be retrieved following a pre-defined sequence, a *relocation* problem might arise. Whenever a container that must be retrieved first is found below containers that will be retrieved at a later time, there arises a need to relocate such containers within the stacking area in order to make the high priority container accessible. In this context, the relocation of a lower priority container is an unproductive move required to free up and retrieve a higher priority container underneath it. Therefore, while in the remarshalling phase the total number of containers in the stacking area is kept unchanged, in the relocation problem containers are retrieved one at a time, following a prespecified order.

In this paper we assume that a *priority* is associated with each container in the stacking area, where this priority could account for a number of different factors. Some of these factors defining the priority of a container are, (i) category: *e.g.* containers with the same priority might belong to the same category and could be piled up on top of each other; (ii) departure time: *e.g.* containers with earlier departure time will have higher priority than containers with later departure time; (iii) size and weight: *e.g.* typically containers with higher weight are not stored on top of containers with lower weight, in order to respect overall ship balancing constraints.

We follow the typical terminology adopted in the context of container terminal operation. We indicate with the term *bay* a two-dimensional portion of the container yard, made up by a number of *stacks*, *i.e.* the width, and *tiers*, *i.e.* the height, as illustrated in Figure 13.1. A *block* is a set of consecutive bays, as presented in Figure 13.2. Finally, a *container yard* is made up by a set of blocks.

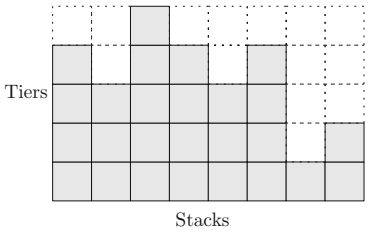

Fig. 13.1 A bay

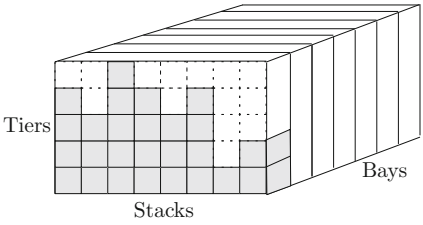

Fig. 13.2 A block

The two terms *retrieving* and *rehandling* are used to describe movement of containers. More specifically, the term *retrieving* is used to indicate a movement of a container from the bay to the vessel. Conversely, we use the term *rehandling* to indicate a move of a container within the container yard, both in the case of intra-bay or intra-block movements.

Finally, the focus of our paper is on three problems in the context of container terminal operation, namely the *B*locks *R*elocation *P*roblem (BRP), the *R*e*M*arshalling *P*roblem (RMP), *i.e.* intra-block marshalling and the *P*re*M*arshalling *P*roblem (PMP), *i.e.* intra-bay marshalling.

In all cases we consider the layout of the stacking area as given, *i.e.* the position and priority of each container in the stacking area is known. Therefore, our interest is not centered on finding effective stacking policies but rather, given a stacking area, we wish to determine how containers should be *rehandled* in order to minimize the total number of unproductive movements.

According to Choe et al (2009), Park et al (2009), and Kang et al (2006b), the unproductive movement of containers in the different phases of the container management process, *i.e.* rehandling, is perceived as the major source of inefficiency in most container terminals. Therefore, special emphasis has been put on finding approaches aimed at minimizing the total amount of rehandling needed during a vessel unloading/loading cycle. Consequently, a number of references dealing with this type of problem can be found under the captions "minimizing the number of rehandling operations," *e.g.* Caserta et al (2009a), Caserta et al (2009c), and Kim and Hong (2006) or, alternatively, "minimizing rehandling time," *e.g.* Choe et al (2009), Kang et al (2006b), Park et al (2009).

The structure of the paper is as follows: In Section 13.2 an overview of stacking approaches, *i.e.* approaches aimed at defining how containers should be piled up in the stacking area is presented. The section concludes illustrating why, no matter how well-thought the stacking policy can be, container rehandling is going to be needed during the loading phase. Therefore, Section 13.3 is devoted to the presentation of marshalling problems, aimed at reshuffling the storage area in order to eliminate or reduce the total number of future rehandling. Section 13.4 deals with a different type of problem, the blocks relocation problem. Exact and heuristic approaches for this special type of rehandling problem are presented. Section 13.5 constitutes a bridge between rehandling problems at maritime container terminals and similar problems arising in different realms. Some references to related work in other application domains are provided in this section. Finally, Section 13.6 concludes offering a brief overview of the current status in the container handling discipline along with a glimpse of future challenges and opportunities.

13.2 Container Stacking

In this section, we present a brief overview of approaches aimed at managing container terminals with respect to defining an effective stacking policy. The papers presented below study, often via *simulation*, different strategies for the container stacking problem. An overview of storage and stacking logistics at container terminals is provided in Steenken et al (2004) and Stahlbock and Voß (2008).

The stacking problem arising at container terminals is the following: Export containers of various destinations, weights, and due dates arrive at a container yard in a random fashion. Very often, not all relevant data is given at the arrival time, *e.g.* the precise weight is usually given only shortly before loading the container to the vessel. In addition, some containers need special treatment, such as containers carrying hazardous material or reefer containers. The task is to find good storage slots for

incoming containers, where several objectives might be addressed, *e.g.* minimizing the number of future relocations, minimizing the overall crane utilization, or storing containers of equal destination within the same bays. The stacking problem requires optimization techniques that work with uncertain information and allow to deal with data arising in an online fashion.

Dekker et al (2006) present an interesting overview of some of the most relevant optimization problems at container terminals and propose a number of alternative policies for stacking containers in a yard. They test the validity of the proposed approaches through simulation. The simulation study concerns a time period of 15 weeks with a movement of around 175,000 containers and generates arrival and departure times for each container. Based on the stochastic set of data, a stacking algorithm made up by different stacking policies has been tested and compared against the naive random stacking policy. Total number of reshuffles, cranes workload, and level of occupation are used as performance measures of each stacking policy. The focus of the paper is on evaluating stacking strategies, *i.e.* how stacks should be filled up to achieve efficiency with respect to these performance measures. However, no details about how containers are reshuffled, *i.e.* which rules are employed, are given.

Kang et al (2006b) present a simulated annealing algorithm for the identification of stacking policies when the information about the weight of containers is not certain. The authors first explain that one of the principal reasons why rehandling occurs is due to the lack of precise information about the weight of containers. Ship balancing constraints require heavy containers to be placed at the bottom of the ship. Therefore, in order to minimize rehandling during loading operations heavy containers should always be placed on top of lighter containers when they are piled up at the container yard, so that they can easily be reversed when loaded onto the vessel. However, when the containers are brought into the terminal by trucks, only an estimate, and not a precise value of container weights is available. As a result, rehandling during the loading phase cannot be entirely avoided. The authors present a simulated annealing algorithm that allows to define an appropriate stacking policy, *i.e.* the "best" stack to be used by an incoming container is identified. A simulation approach is used: first, a random sequence of incoming containers is generated following predefined probability distribution functions that account for the likeliness of receiving containers of a given weight class. These probability distribution functions are derived from historical data. Next, a strategy to be evaluated is applied to stack the incoming containers. Once all the incoming containers have been stacked, the expected number of rehandling operations required to load these containers to a vessel is estimated. The amount of rehandling is taken as a measure of "goodness" of the stacking strategy. In order to have a more thorough evaluation, the simulation cycle is repeated over multiple runs. It is also worth pointing out that, in order to increase the accuracy of prediction of the actual container weights based upon the estimates, a machine learning approach (decision trees) to derive a classifier is employed. The authors report an improvement in accuracy when the learning mechanism is used.

Yang and Kim (2006) consider the problem of finding the stacking policy of incoming containers that minimizes the total number of rehandling operations. Each container is characterized by arrival and departure dates. When a container with early departure date is placed below another container with late departure date, a rehandling is deemed necessary. The focus of the paper is on the definition of "groups" of containers, defined as a collection of containers that can be stored together on the same stack and subsequently retrieved in any order without incurring extra rehandling. An example is given by containers with the same destination port, size and weight class. A mathematical model is proposed with the aim of creating such groups of homogeneous containers and identifying which portion of the bay should be used to stack each group. The goal of the model is, therefore, to identify the group stacking policy with minimum rehandling. Two versions of the problem are tackled by the authors: a *static* version, in which the precise information about departure date of each container is known in advance, and a *dynamic* version of the same problem, in which the retrieval date of a container becomes known only at its arrival. While in the static version of the problem the whole planned schedule can be used to find the optimal stacking policy, in the dynamic version of the stacking problem a position is assigned to a container whenever the container arrives at the yard, using updated information about the container as well as considering the current state of the bay.

Kim et al (2000) tackle the problem of determining the storage location of export containers with the aim of minimizing the expected number of future rehandling. Containers are considered one at a time, in a dynamic fashion. Whenever a new container reaches the container yard information about its weight, the current state of the bay and the distribution of the weight groups within the bay are used to assign a slot to the incoming container. An optimization model based upon dynamic programming is employed, under the strong assumption that a container is rehandled at most once. Two basic terms of the dynamic programming recursion are: (i) the probability of arrival of a container of a given weight group (estimated from historical data), and (ii) the expected number of extra rehandling generated when a container of a given group is placed at a specific slot within the current bay. Given these two terms, the dynamic programming scheme iterates over the number of available empty slots within the bay and minimizes the expected number of rehandling operations for a given set of incoming containers.

Taleb-Ibrahimi et al (1993) study the stacking problem from two different perspectives. First, they propose a set of rules aimed at determining the amount of space required to stack a set of containers without moving each container more than a prespecified number of times. Next, they study the dual problem, where the amount of space available for container stacking is kept fixed and a method to organize the storage area while minimizing the number of rehandling operations is given.

A number of authors propose "richer" approaches, in which the container handling problem is tackled from a broader perspective. These "integrated" approaches typically attempt to take into account a pool of factors affecting the total berthing time, *e.g.* traveling time of containers from the quayside to the container yard, stacking policy within the container yard, and rehandling policy.

Kozan and Preston (1999) present an integrated model that considers both the unloading and stacking problems, together. The problem is modeled as a job-shop machine scheduling problem, where the set of containers to be moved corresponds to the set of jobs to be processed and the set of available cranes corresponds to the set of machines to be used. They propose an interesting characterization of setup times. Two subproblems are identified: (i) which containers should be assigned to which crane, *i.e.* in the context of scheduling problems, this corresponds to finding which jobs should be processed on which machine; and (ii) the scheduling sequence for each crane, *i.e.* how jobs should be processed on each machine. The authors point out that, while the proposed model resembles the classical scheduling problem, a major difference consists in the way in which setup times are computed. While in the classical scheduling problem setup times are exclusively dependent on the job immediately preceding the current job, in their model the setup time, *i.e.* the time it takes to access a specific container within a stack, depends on the order of scheduling of containers initially stored on top of the target container. In this paper they propose a model aimed at minimizing the total berthing time, computed as the sum of the setup and traveling times of all containers handled. Minimizing the traveling time corresponds to finding the proper assignment of containers to cranes and minimizing the setup time corresponds to finding the optimal rehandling policy for stacked containers. The model is solved by using a genetic algorithm, where the chromosome representation captures both the container-crane assignment and the scheduling problem per crane.

In a similar fashion, Kozan and Preston (2006) present a mathematical model for the integrated problem of determining the optimal storage strategy and container handling scheduling. The main goal of the authors is to define an approach that minimizes the total throughput time, which is seen as the sum of two terms: The transferring time of the containers to the storage area and the handling time of all containers from a ship at berth. Consequently, a *container transfer model* and a *container location model* are proposed, respectively. The two separated models are finally integrated into a single model and iteratively solved using a hybrid tabu search/genetic algorithm.

Other integrated approaches are presented in *e.g.* Froyland et al (2008), Lee et al (2006), and Kozan (2000).

A different and yet somehow related problem is the one studied by Kim and Park (2003), where the problem of how to pre-allocate storage space for export containers, *i.e.* containers that are going to be loaded onto a vessel, is investigated. Such space should be allocated with the goal of maximizing efficiency of future loading operations. As pointed out by the authors, the process of determining storage locations of export containers can be decomposed into two major steps: (i) space allocation problem, and (ii) container location problem. While problem (ii) deals with the identification of the specific storage location of a container within a bay, *i.e.* stack and tier number, problem (i) only defines the amount of space and the specific area, *i.e.* how many and which bays need to be assigned to store export containers for a specific vessel. Some basic criteria are enforced in determining such area, *e.g.* bays assigned to a vessel should be located near the vessel berthing position, containers

of different groups should not be mixed in the same bay etc. A mathematical model and a solution approach based upon heuristic rules are presented in the paper.

Further works dealing with the storage location assignment problem are, *e.g.* Bazzazi et al (2009), Park and Seo (2009), Han et al (2008), Zhang et al (2003), and Preston and Kozan (2001).

As pointed out by many authors, *e.g.* Park et al (2009), Kang et al (2006b), and Kim and Bae (1998), rehandling during the loading phase cannot be avoided altogether, even after a well-planned strategy aimed at identifying a stacking policy that minimizes the total number of expected rehandling is employed. Some of the reasons why this occurs are that containers to be shipped to different vessels are stored together due to the limited space capacity of the yard, the precise information about the weight of containers is not available until just before the loading operations begin and the loading plan is not yet determined when a container arrives at the yard.

13.3 Remarshalling and Premarshalling

As mentioned in the introduction, the goal of the remarshalling problem is to reshuffle containers so that no further relocations, *i.e.* unproductive moves are required when the loading/unloading phase is performed. As formalized in Choe et al (2009), "remarshalling refers to the task of relocating export containers into a proper arrangement for the purpose of increasing the efficiency of the loading operations."

Two types of marshalling activities are carried out at a typical container terminal yard: (i) intra-block *remarshalling*, in which containers that are scattered around within a block are rearranged into designated bays within the same block; (ii) intra-bay *premarshalling*, in which containers within the same bay are reshuffled. In both cases, the goal is to minimize the number of future unproductive moves.

Typically, intra-block remarshalling refers to the problem of moving a set of containers to pre-specified bays within the same block. As indicated in Kang et al (2006a), the bays in which the target containers are located before remarshalling are called *source* bays and the empty bays to which these containers should be moved are called *target* bays. Containers within a block are characterized by two types of information:

- a *group* or *category*, accounting for, *e.g.* the port of destination. In order to minimize the distance traveled by the cranes during the loading phase, containers belonging to the same group are placed in adjacent slots within the same block;
- a *priority*, accounting for, *e.g.* weight information, order of retrieval, etc. Within the same group, containers should be stacked by ensuring that no container with lower priority is found on top of a container with higher priority.

Therefore, the two-objective problem of intra-block remarshalling is aimed at grouping together containers belonging to the same category and, for each set of

containers of the same category, at piling up such containers taking into account priorities.

On the other hand, intra-bay premarshalling is motivated by the use of a specific technology. As pointed out by Lee and Chao (2009) and Lee and Hsu (2007), yards that use rail mounted gantry cranes as major container handling equipment typically solve the marshalling problem at bay level. For safety reasons, in some terminals where access of containers to and from the block is usually from the side, a gantry crane is not moved from one bay to another, even within the same block, while carrying a container. Therefore, in those terminals, to move a container from one bay to another, it would be necessary to temporarily unload the container from the crane, put it on a truck, move the truck and, possibly the empty crane to the target bay, pick up the container from the truck with the crane and store the container within the target bay. This operation is time consuming and, therefore, it is avoided whenever possible. This consideration motivates the study, from a practical perspective, of the intra-bay premarshalling problem. The goal of the premarshalling problem is, therefore, to rehandle containers within the same bay in order to eliminate (or minimize) future rehandling while minimizing the total number of rehandling operations during the premarshalling process itself.

Intra-block remarshalling could be seen as more than a simple extension of intra-bay premarshalling, since more than one crane could be used to handle the containers. Therefore, typically the remarshalling problem also encompasses some considerations with respect to avoiding or minimizing interference among cranes within the same block.

Conversely, as illustrated in Figure 13.3, the premarshalling problem could be seen as a special case of the remarshalling problem, in which the following characteristics arise:

- only one source bay is given;
- only one target bay is given;
- the target bay coincides with the source bay;
- only one crane is used and, therefore, there is no need to take into account crane scheduling interference issues.

To the best of the authors knowledge, the computational complexity of the RMP has not been addressed yet. We close this gap in Section 13.4, after introducing the BRP in detail. It turns out that the RMP is NP-hard which can be proven by transforming the NP-hard BRP to the RMP.

Choe et al (2009) study the intra-block remarshalling problem where more than one crane is used to handle containers. Therefore, interference among cranes is taken into account. The problem takes as input a current configuration, in which containers are placed on source stacks and is aimed at relocating such containers into target stacks in such a way that two constraints are satisfied: (i) after remarshalling, all containers can be loaded without further rehandling; and (ii) during remarshalling, each container should be moved from its source bay to its target bay without re-

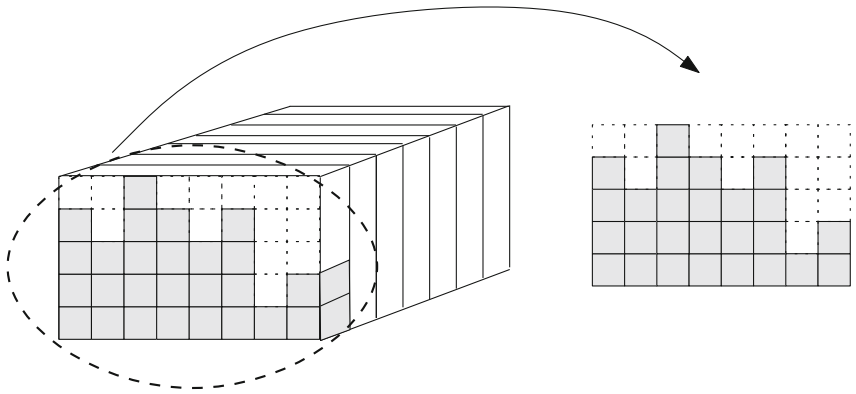

Fig. 13.3 PMP as special case of RMP.

handling. The authors propose a two-phase algorithm: The first phase is devoted to identifying the target slots to which handled containers should be moved, and the second phase is aimed at finding an optimal schedule of the cranes to actually perform the relocation of containers. The proposed algorithm, based upon simulated annealing, is aimed at finding a rehandling-free configuration of the block that can be achieved in the minimum amount of time. Based upon a partial order graph that captures all the feasible moves leading from the current block configuration to a target configuration, at each step of the search phase the algorithm evaluates the goodness of a candidate solution configuration by heuristically creating a crane schedule and estimating the time needed to complete remarshalling to achieve that particular configuration.

Park et al (2009) analyze the remarshalling problem with respect to export containers at the intra-block level, *i.e.* the reshuffling of containers is done within the same block. Typical dimensions of the considered problem are 41 bays per block, where each bay is made up by 10 stacks and 6 tiers. A block is managed by the use of two cranes, a first crane dealing with export containers and a second one dealing with import containers. Due to the large size of the considered blocks, the authors identify two sources of inefficiencies in the handling of containers. The first one is related to the horizontal movement of the cranes used to load containers to the vessel. Typically, export containers are unloaded from tracks and, therefore, are piled up near the landside of the block. This means that during the loading operations the crane operating on the waterside is forced to travel long distances toward the landside of the block to pick up export containers, hence affecting the overall time of the loading operations. A second source of inefficiency can be ascribed to the stacking of high priority containers below low priority containers, forcing a rehandling of the uppermost containers. The authors present a two-stage heuristic algorithm. The first stage uses heuristic rules to identify where, *i.e.* in which stacks containers must be relocated. Stacks are selected with the aim of avoiding future rehandling of containers during the loading operations. In the second stage of the algorithm, a co-

operative co-evolutionary algorithm is used to identify the precise slot within which containers should be relocated (stack and tier), along with the order of movement of the containers to be reshuffled. Two populations are created to identify the slots and to define the order of movement. Information is exchanged in the following way: initially, a solution for the target slots identification is found; such solution is then used as input to the subproblem dealing with the movement sequence. In turn, the movement sequence defined by this last subproblem is used to find a better set of target slots and the cooperative approach is repeated in cycles.

Similarly, Kang et al (2006a) deal with the intra-block remarshalling problem, where containers are reshuffled at a block level, moving them into designated slots within the same block. As for the previous works, they deal with export containers and the objective is to find a rearrangement that avoids future rehandling during the loading operations. As in Choe et al (2009), multiple cranes are used within a block and, therefore, interference among cranes is also minimized. The problem takes as input a current configuration in which containers are placed on source stacks and is aimed at relocating such containers into target stacks in such a way that two constraints are satisfied: (i) after remarshalling, all containers can be loaded without further rehandling; and (ii) during remarshalling, each container should be moved from its source bay to its target bay without rehandling. The proposed approach is similar to the one of Choe et al (2009), since a two-phase algorithm is designed. First, a set of target locations is defined ensuring that the two main constraints are enforced. Next, a partial order graph is created with the goal of finding a set of feasible moves leading from the source configuration to the target configuration. The partial order graph captures all the possible moves leading from source to target configuration, while respecting the two aforementioned constraints. Next, simulated annealing is used to find the solution that minimizes the overall time required to carry out the remarshalling operations. Finally, a heuristic is employed to find a crane's feasible schedule. An interesting point brought out by the authors is related to the notion of neighbor solutions. Given a partial order graph, a neighbor of such graph is obtained by appropriately modifying the current one via the application of swapping among containers stored on different stacks of the same bay.

In a seminal work, Kim and Bae (1998) deal with the problem of how to efficiently move a set of containers from source bays to target bays. Containers in the target bays should be accommodated according to a pre-specified layout, called target layout. The intra-block remarshalling problem is decomposed into two subproblems: (i) the bay matching and move planning problem, in which each source bay within the block is matched with the target bay in the target layout. Decisions with respect to how many containers should be moved between any two bays are made at this stage. This part of the problem is solved using dynamic programming (to define the bay matching needs) and the transportation algorithm (to plan the movement of containers among bays and assignment to cranes). Whenever crane interferences arise throughout the container movements, the bay matching is called again under additional constraints that prohibit the conflicting bay matching; (ii) the movement sequencing problem, in which the actual movements required to reach the target layout are scheduled. Kim and Bae (1998) adopt a "macroscopic" perspective of the

problem, *i.e.* only the number of containers per group type and bay are considered, whereas the actual positions and rehandling within a bay are neglected.

More recently, Lee and Hsu (2007) proposed an optimization model for the intra-bay premarshalling problem. As previously mentioned, the study of the intra-bay problem is motivated by the specific technology used at the container yard. Rail mounted gantry cranes are moved from one bay to the other while being empty, *i.e.* the trolley of the crane does not carry any container. Therefore, in order to move a container from one bay to another, even within the same block, the use of an auxiliary truck to temporarily store the container to transport it from the source bay to the destination bay is required. Such an inter-bay operation is obviously time consuming and consequently should be avoided whenever possible. For this reason, the authors study the problem of intra-bay premarshalling. The premarshalling process reshuffles the containers within a bay in order to reach a final bay layout that does not require further rehandling during the loading phase. The authors work under the following four basic assumptions:

- reshuffling takes place only within the same bay;
- containers are assumed to be of the same length;
- each crane is involved in the loading of one ship at a time;
- the loading order of containers is known.

These authors propose an integer programming model based upon a multi-commodity network flow problem. The network accounts for two dimensions, time and space. Each level of the network describes a specific point in time and captures the state of the bay at that instant. Connections between different levels of the network account for moves of containers over time and space, *i.e.* edges within the network are used to model the movement of a container from one stack to another in a given time period. The basic mathematical model, along with some extensions, is presented in the paper. Finally, in order to reduce the number of variables and to make the model tractable, some simplifications are introduced. One drawback of the model worth mentioning concerns the need to pre-define a parameter T, *i.e.* the total number of time periods required to completely reshuffle the bay. While the goal of the problem is to find the rehandling pattern that sorts out the whole bay in the minimum amount of moves, *i.e.* in the minimum amount of time periods T (which is unknown), the appropriate choice of the value of T has a strong bearing on the computational time required to solve the model. If T is chosen too large, then a very large number of variables is created and, therefore, the MIP solver might not be able to reach the optimal solution in a reasonable amount of computational time. On the other hand, if T is chosen too small, a feasible solution might not even exist. Some analysis about this trade off is presented by the authors.

Lee and Chao (2009) propose a different algorithm for the same premarshalling problem. In order to overcome the limitations imposed by the size of the integer programming model of Lee and Hsu (2007), the authors propose a heuristic approach aimed at minimizing the number of movements required to complete the premarshalling process. More specifically, a bi-objective problem is proposed: On the one hand, the authors attempt to create a reshuffled bay that requires the mini-

mum amount of rehandling during the loading phase; on the other hand, such desired configuration should be reached in the minimum amount of steps, *i.e.* the final configuration should be reached minimizing the total number of rehandling operations. The approach is hybrid in the sense that heuristic techniques, such as neighborhood search and mathematical programming techniques, such as integer programming are intertwined to deal with different subproblems. First, the neighborhood search heuristic is used to find a chain of movements to sort out the bay in such a way that the number of further rehandling required during the loading phase is minimum. Next, a binary integer programming model is solved to reduce the number of movements required to reach that final configuration. A number of minor heuristic rules are used to foster the effectiveness of the proposed algorithm. Some comments on Lee and Chao (2009) together with a simple lower bound calculation can be found in Voß (2008).

Caserta and Voß (2009) present a metaheuristic algorithm for the premarshalling problem. The central idea of the approach relies on iteratively solving to optimality smaller portions of the original problem. The usual assumptions, *i.e.* premarshalling is carried out within the same bay, containers are assumed to be of the same size and loading preferences are known, are made. The algorithm consists of four different phases, in which ideas from the Corridor Method, roulette-wheel selection and local search techniques are intertwined to foster intensification around an incumbent solution. The algorithm is stochastic in nature and is based upon the use of a set of greedy rules that bias the behavior of the scheme toward the selection of the most appealing moves.

13.4 Relocation and Retrieval

The BRP is closely related to the previously discussed pre- and remarshalling problems but with one major difference: While pre- and remarshalling problems only consider rehandling operations, BRP also allows for retrieving operations, *e.g.* moving a container from a bay to a destination vessel. Retrieving and rehandling operations might be carried out in parallel for the BRP. Consequently, the number of containers in the bay decreases for the BRP, whereas the number of containers in the bay (block) remains constant for premarshalling (remarshalling) problems.

More precisely, the BRP is described by the following properties:

- A single bay is considered and consequently operations are carried out by a single crane.
- Containers are piled up vertically in stacks, *i.e.* only the uppermost container of each stack is accessible for rehandling or retrieving and each container is either placed on the ground or on top of another container.
- The number of stacks describes the width of the bay.
- The height of stacks is bounded by the number of tiers.
- The total number of containers in the bay is denoted by N.

- Each container in the bay is associated with a priority number, where more than one container might belong to the same priority group (indicated by the priority number). Moreover, the location of each container is given in advance.
- Containers have to be retrieved from the bay according to their priority number, *i.e.* a container with a certain priority can only be retrieved if all containers with higher priorities have already been removed.
- Containers that are to be removed next are called *target containers*. Rehandling operations become necessary, if no target container is accessible.
- A majority of models given in literature add the following condition: *(A1)* Only containers located in the same stack as and above the current target container are allowed to be rehandled (see, *e.g.* Kim and Hong (2006), Caserta et al (2009a), Caserta et al (2009c), and Caserta et al (2009b)).

The objective of the BRP is to retrieve all the containers from the bay in the prescribed order while minimizing the number of rehandling operations. As pointed out at the beginning of this section, the BRP is closely related to remarshalling. In particular, the BRP can be considered as a specific case of remarshalling as each BRP can be transformed to a remarshalling problem by the following procedure:

Transformation BRP to RMP: Given a BRP with a certain bay defined by a number of stacks and a number of tiers. Generate a remarshalling problem with a single source bay of the same size by carrying over the layout from the BRP bay to the remarshalling source bay. Moreover, generate a single and empty target bay with one stack of height equal to the number of containers to be retrieved (N) and impose the use of a single crane for carrying out the rehandling operations. A solution to the remarshalling problem with a minimum number of rehandling operations is then an optimal solution to the BRP. The transformation process is illustrated in Figure 13.4. The transformation "BRP to RMP" implies the following result:

Lemma 1 *The remarshalling problem is* NP*-hard.*

Proof 1 *Assume there exists a polynomial algorithm for RMP. Then each BRP instance could be solved in polynomial time as a polynomial transformation from BRP to RMP exists. However, BRP is proven to be* NP*-hard (Caserta et al (2009b)) and, unless* P = NP *holds true, there is no such polynomial algorithm.*

So far, only a few publications have discussed the BRP. Kim and Hong (2006) describe the BRP together with assumption *(A1)* and suggest two solution procedures. First, in an exact approach a branch-and-bound method is described that branches over all possible bay layouts resulting from the retrieval of a target container. For determining lower bounds, the *confirmed rehandlings* are counted and added to the already realized rehandling, where each container located above a container with higher priority causes a confirmed rehandling. Second, Kim and Hong (2006) proposes a heuristic method based on the *E*xpected *N*umber of *A*dditional *R*ehandling (ENAR). Each time, if there is more than one rehandling operation possible, choose

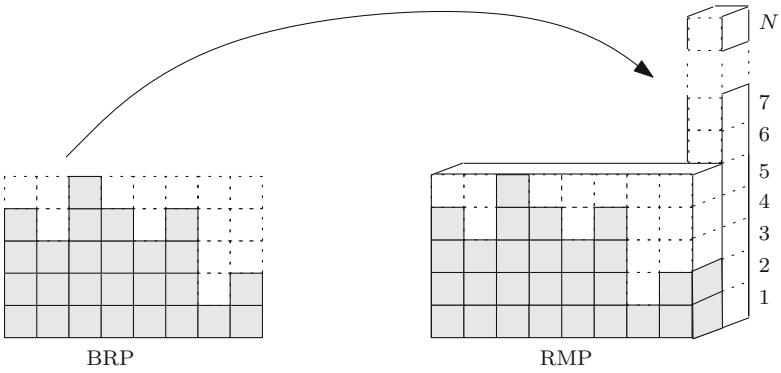

Fig. 13.4 Transformation BRP to RMP

the one that minimizes the ENAR for the resulting bay configuration. Experimental studies are carried out for scenarios where precedence relations are given among individual containers, *i.e.* each priority group has exactly one member and for scenarios where precedence relations are given for container groups. The exact branch-and-bound approach is compared with the heuristic method and an average increase of the number of rehandling operations up to 7.3 % is reported for using the heuristic instead of the exact method.

Caserta et al (2009a) describe a novel encoding of the bay to a binary matrix and describe the benefit of this encoding in terms of computational matters. Fast access to information about the current layout of the bay is enabled and fast transformation of bays when rehandling takes place is possible which, in turn allows a lean and fast implementation of the solution method. The BRP as well as the premarshalling problem are stated in the notion of the binary encoding. Nevertheless, the focus of the article is on investigating the BRP under assumption *(A1)*. In particular, a random-guided look-ahead procedure is implemented. This metaheuristic approach is based on a set of simple rules that are used to compute heuristic solutions based on any initial bay configuration. The objective function value of these heuristic solutions serves as a score for the quality of a found partial solution and as an upper bound for the search procedure. The experimental study compares the results of the presented procedure against those of Kim and Hong (2006) and Caserta et al (2009c) and proves the quality of the proposed method by showing a decreased average number of rehandling operations in the solution.

Caserta et al (2009c) present a corridor method algorithm for the blocks relocation problem, in which a dynamic programming scheme is presented and used in a metaheuristic fashion. The approach belongs to the realm of hybrid algorithms, since mathematical programming techniques are used within a metaheuristic framework, iteratively solving to optimality "constrained" versions of the original BRP.

To the best of our knowledge, Caserta et al (2009b) is up to now the only work presenting mathematical model descriptions for the BRP. Two mathematical formulations are proposed, where the first one (BRP-I) is not taking into account condition

(A1), whereas the second one (BRP-II) does. Thus, BRP-I is exploring a larger solution space and an example showing that BRP-I is indeed able to find better solutions than BRP-II is presented. On the other hand, assumption *(A1)* allows a leaner formulation which results in shorter computational times for solving the problem using a commercial MIP-solver. Consequently, using BRP-II, larger instances can be addressed compared with BRP-I, as reported in the computational study. In addition, in the same work, upper bounds on the number of rehandling operations are presented. Furthermore, the complexity of the BRP is stated as NP-hard for the BRP, as well as for the BRP under assumption *(A1)*. Finally, a simple heuristic rule is proposed and measured against the exact solution and the heuristic solution of Kim and Hong (2006).

13.5 Related Work in Different Fields

Stacking, sorting and rehandling problems are discussed not only in the context of containers and ports, but also in different areas like warehousing, production planning, and artificial intelligence. This section is not meant to give a comprehensive overview of the work in different fields, but to refer the interested reader to related notions and concepts.

Some warehouses are organized following the stacking principle by storing uniform items piled up on top of each other, where access is only granted for the uppermost item. Stacking operations in those warehouses follow similar rules as in container yards. However, a major difference between container yards and warehouses is given by the item flow, as warehouses have to offer retrieving and receiving operations in parallel (see, *e.g.* Nishi and Konishi (2009)), whereas in container yards the receiving operations are usually completed before the retrieval operations take place. Moreover, in general warehouses handle a much larger number of items than container yards. In addition, the physical properties of the items in a warehouse might differ from those of a box-shaped container. For instance, in the steel industry coils are stored by stacking them on top of each other. The resulting storage setting is not forming "stand-alone" stacks, as each coil is placed on top of two consecutive coils from the row below (see, *e.g.* Zäpfel and Wasner (2006)).

Also the handling of trains involves stacking operations; see, *e.g.* Felsner and Pergel (2008). A train can be seen as a sequence of wagons. It might happen that the wagon sequence of a single train needs to be changed or that the wagons of several trains have to be *reshuffled* to new collections of trains. These operations are physically carried out on dead end sidings, where trains or part of trains can be stored intermediately and taken away later on. Thus on dead end sidings trains can be "stacked" together and moreover, rehandling of wagons is possible. Each of those dead end sidings relates to a stack in the container yard, where only the uppermost container/wagon is accessible.

A well-known concept in artificial intelligence is that of blocks-world (see, *e.g.* Romero and Alquézar (2004), Gupta and Nau (1992)). The blocks-world is carried

out on a "table" where blocks are stacked on top of each other. A typical blocks-world instance consists of a given initial table state and a desired goal state. The task is to transform the initial state to the goal state with a minimum number of moves. Variants of blocks-world incorporate limitations on the table size and different levels of given conditions for the goal state. Gupta and Nau (1992) prove the NP-hardness of blocks-world and Caserta et al (2009b) show that the BRP is a particular case of blocks-world.

13.6 Conclusion and Future Challenges

Ever since the first containers were introduced in the early 1960s, container handling techniques and strategies have always been key factors in measuring the efficiency of major ports. However, due to the growth of container vessels in recent years, whenever one of such ships berths at a port, a number of containers that would have been unthinkable some time ago must be handled in just a few hours. This poses a serious challenge for container terminal operators, since the volume of traffic has grown exponentially while the available surface for managing such traffic has remained virtually unchanged. Therefore, optimization techniques for handling and rehandling containers acquire a prominent role in fostering efficiency of container terminal operation.

In this paper we have presented a survey of techniques for post-stacking situations, *i.e.* once the stacking area has already been filled with containers. Exploiting the fact that updated information becomes available while the vessel approaches the port, export containers could be rearranged prior to the arrival of the vessel with the objective of minimizing the time required for future loading operations. We have identified three classes of post-stacking problems:

- the *remarshalling* problem, *i.e.* the problem in which one wants to rehandle containers from a set of source bays to a set of target bays within the same block. Such containers are rehandled in such a way that future loading operations will be carried out with the maximum efficiency;
- the *premarshalling* problem, *i.e.* the problem in which one wants to reshuffle containers within the same bay. The crane used to carry out these operations is kept fixed over the specific bay and, therefore, no horizontal movements of the crane are allowed. As in the previous problem, the final layout of the bay is such that the number of future relocations required to load the containers is minimized, or eliminated altogether;
- the *relocation* problem, *i.e.* the problem in which one wants to retrieve containers following a prescribed list of precedences among containers. Such retrieval operations are carried out minimizing the total number of rehandling operations.

Some of the challenges concerning the aforementioned problems from the operations research point of view are, *e.g.* the design of efficient algorithms for online

optimization, the use of recent findings in the metaheuristic field and the development of broader, integrated approaches for container terminal logistics.

With respect to the design of algorithms for online optimization, as presented in this paper, a number of heuristic rules have been proposed for any of these problems. However, a more thorough effort with respect to the design of learning mechanisms inspired by metaheuristics could be made, in a fashion similar to what is proposed by Bazzazi et al (2009), Lee and Chao (2009) and Kozan and Preston (1999). Perhaps one of the major challenges facing operations research experts in designing metaheuristic algorithms consists in proposing alternative encodings for such problems. The dynamic nature of the problem itself seems to impose a "constraint" in the type of encoding used. However, ideas about alternative encoding schemes for any of these problems could lead to the design of radically different approaches. As it often happens with metaheuristic algorithms, such approaches should be fast enough to be used in dealing with online optimization.

When it comes to exploring recent findings in the metaheuristic field, one of the new trends concerns connecting metaheuristic paradigms with classical mathematical programming techniques. In general, such mathematical programming techniques are used in a metaheuristic "philosophy," in the sense that the powerful exact technique is exploited only on a limited portion of the solution space, or perhaps over different portions of the solution space, as opposed to attempting to solving the original problem to optimality. This could lead to more powerful algorithms that could produce higher quality solutions. Some ideas about how to use mathematical programming techniques in the spirit of metaheuristics are provided in Maniezzo et al (2009). Similarly, a first application of such approach in the context of the blocks relocation problem has been proposed in Caserta et al (2009c).

Finally, a third avenue of opportunity could be offered by recent advances in computer technology and parallel computing. Due to the astonishing increase in computational power, problems that in the past were seen as intractable, today can be dealt with using, *e.g.* parallel or grid computing. The interesting aspect is that broader models, perhaps taking into account more than one single optimization problem at a time, can now be considered and solved. The current trend so far has been the one of "divide and conquer," *i.e.* to take the original logistic problem and to divide it into smaller problems. Each one of these smaller problems is then solved, either heuristically or exactly. However, due to this spur in computing power, perhaps it is now possible to tackle larger problems taking into account a bigger portion of the whole picture. This idea is in line with what is attempted by, *e.g.* Kozan and Preston (1999), Kozan and Preston (2006), and Lee et al (2006).

References

Bazzazi M, Safaei N, Javadian N (2009) A genetic algorithm to solve the storage space allocation problem in a container terminal. Computers and Industrial Engineering 56(1):44–52

Caserta M, Voß S (2009) A Corridor Method-based Algorithm for the Premarshalling Problem. In: Giacobini M, Brabazon A, Cagnoni S, Di Caro G, Ekart A, Espacia-Alcázar A, Farooq M, Fink A, Machado P, McCormack J, O'Neill M, Neri F, Preuss M, Rothlauf F, Tarantino E, Yang S (eds) Applications of Evolutionary Computing, Lecture Notes in Computer Science, vol 5484, Springer, Berlin, pp 788–797

Caserta M, Schwarze S, Voß S (2009a) A New Binary Description of the Blocks Relocation Problem and Benefits in a Look Ahead Heuristic. In: Cotta C, Cowling P (eds) Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, vol 5482, Springer, Berlin, pp 37–48

Caserta M, Schwarze S, Voß S (2009b) A mathematical formulation and complexity considerations for the blocks relocation problem, Working Paper, Institute of Information Systems, University of Hamburg

Caserta M, Voß S, Sniedovich M (2009c) Applying the corridor method to a blocks relocation problem. OR Spectrum (DOI: 10.1007/s00291-009-0176-5)

Choe R, Park T, Oh MS, Kang J, Ryu KR (2009) Generating a rehandling-free intra-block remarshaling plan. Journal of Intelligent Manufacturing (DOI: 10.1007/s10845-009-0273-y)

Dekker R, Voogd P, van Asperen E (2006) Advanced methods for container stacking. OR Spectrum 28(4):563–586

Felsner S, Pergel M (2008) The Complexity of Sorting with Networks of Stacks and Queues. In: Halperin D, Mehlhorn K (eds) Algorithms – ESA 2008: 16th Annual European Symposium, Lecture Notes in Computer Science, vol 5193, Springer, Berlin, pp 417–429

Froyland G, Koch T, Megow N, Duane E, Wren H (2008) Optimizing the landside operation of a container terminal. OR Spectrum 30(1):53–75

Gambardella LM, Rizzoli AE, Zaffalon M (1998) Simulation and planning of an intermodal container terminal. Simulation 71(2):107–116

Gupta N, Nau DS (1992) On the complexity of blocks-world planning. Artificial Intelligence 56(2–3):223–254

Han Y, Lee LH, Chew EP, Tan KC (2008) A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub. OR Spectrum 30(4):697–720

Kang J, Oh MS, Ahn EY, Ryu KR, Kim KH (2006a) Planning for Intra-block Remarshalling in a Container Terminal. In: Ali M, Dapoigny R (eds) Advances in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence, Springer, Berlin, pp 1211–1220

Kang J, Ryu KR, Kim KH (2006b) Deriving stacking strategies for export containers with uncertain weight information. Journal of Intelligent Manufacturing 17(4):399–410

Kim KH, Bae JW (1998) Re-marshalling export containers in port container terminals. Computers and Industrial Engineering 35(3–4):655–658

Kim KH, Hong GP (2006) A heuristic rule for relocating blocks. Computers & Operations Research 33(4):940–954

Kim KH, Park KT (2003) A note on a dynamic space-allocation method for outbound containers. European Journal of Operational Research 148(1):92–101

Kim KH, Park YM, Ryu KR (2000) Deriving decision rules to locate export containers in container yards. European Journal of Operational Research 124(1):89–101

Kozan E (2000) Optimising container transfers at multimodal terminals. Mathematical and Computer Modelling 31(10–12):235–243

Kozan E, Preston P (1999) Genetic algorithms to schedule container transfers at multimodal terminals. International Transactions in Operational Research 6(3):311–329

Kozan E, Preston P (2006) Mathematical modeling of container transfers and storage locations at seaport terminals. OR Spectrum 28(4):519–537

Lee LH, Chew EP, Tan KC, Han Y (2006) An optimization model for storage yard management in transshipment hubs. OR Spectrum 28(4):539–561

Lee Y, Chao SL (2009) A neighborhood search heuristic for pre-marshalling export containers. European Journal of Operational Research 196(2):468–475

Lee Y, Hsu NY (2007) An optimization model for the container pre-marshalling problem. Computers & Operations Research 34(11):3295–3313

Maniezzo V, Voß S, Stützle T (eds) (2009) Matheuristics: Hybridizing Metaheuristics and Mathematical Programming. Springer, Berlin

Nishi T, Konishi M (2009) An optimisation model and its effective beam search heuristics for floor-storage warehousing systems. International Journal of Production Research (DOI: 10.1080/00207540802603767)

Park C, Seo J (2009) Assembly block storage location assignment problem: revisited. Production Planning and Control 20(3):216–226

Park K, Park T, Ryu KR (2009) Planning for Remarshalling in an Automated Container Terminal using Cooperative Coevolutionary Algorithms. In: SAC '09: Proceedings of the 2009 ACM Symposium on Applied Computing, ACM, New York, pp 1098–1105

Preston P, Kozan E (2001) An approach to determine storage locations of containers at seaport terminals. Computers & Operations Research 28(10):983–995

Romero AG, Alquézar R (2004) To Block or Not to Block? In: Lemaitre C, Reyes C, Gonzalez J (eds) Advances in Artificial Intelligence – IBERAMIA 2004, Lecture Notes in Computer Science, vol 3315, Springer, Berlin, pp 134–143

Saccone S, Siri S (2009) An integrated simulation-optimization framework for the operational planning of seaport container terminals. Mathematical and Computer Modelling of Dynamical Systems 15(3):275–293

Stahlbock R, Voß S (2008) Operations research at container terminals: a literature update. OR Spectrum 30(1):1–52

Steenken D, Voß S, Stahlbock R (2004) Container terminal operations and operations research – a classification and literature review. OR Spectrum 26(1):3–49

Swisher JR, Hyden PD, Jacobson SH, Schruben LW (2000) Simulation Optimization: A Survey of Simulation Optimization Techniques and Procedures. In: WSC '00: Proceedings of the 32nd Conference on Winter Simulation, Society for Computer Simulation International, San Diego (California), pp 119–128

Taleb-Ibrahimi M, De Castilho B, Daganzo CF (1993) Storage space vs handling work in container terminals. Transportation Research B 27B(1):13–32

Voß S (2008) Extended Mis-Overlay Calculation for Pre-Marshalling Containers. Technical Report, Institute of Information Systems, University of Hamburg, Hamburg

Yang JH, Kim KH (2006) A grouped storage method for minimizing relocations in block stacking systems. Journal of Intelligent Manufacturing 17(4):453–463

Yun WY, Choi YS (1999) Simulation model for container-terminal operation analysis using an object-oriented approach. International Journal of Production Economics 59(1):221–230

Zäpfel G, Wasner M (2006) Warehouse sequencing in the steel supply chain as a generalized job shop model. International Journal of Production Economics 104(2):482–501

Zhang C, Liu J, Wan Y, Murty KG, Linn RJ (2003) Storage space allocation in container terminals. Transportation Research B 37(10):883–903