

Generating Scenarios for Simulation and Optimization of Container Terminal Logistics

Sönke Hartmann^{a,b}

^aOR Consulting, Bornstraße 6, D-20146 Hamburg, Germany.
E-mail: soenke.hartmann@gmx.de

^bChristian-Albrechts-Universität zu Kiel, Lehrstuhl für Produktion und Logistik, D-24098 Kiel, Germany. E-mail: hartmann@bwl.uni-kiel.de

Abstract. This paper introduces an approach for generating scenarios of sea port container terminals. The scenarios can be used as input data for simulation models. Furthermore, they can be employed as test data for algorithms to solve optimization problems in container terminal logistics such as berth planning and crane scheduling. A scenario consists of arrivals of deep sea vessels, feeder ships, trains, and trucks together with lists of containers to be loaded and unloaded. Moreover, container attributes such as size, empty, reefer, weight, and destination are included. The generator is based on a large number of parameters that allow the user to produce realistic scenarios of any size. The purpose of this paper is to outline the parameters that are important to produce realistic scenarios of high practical relevance and to propose an algorithm that computes scenarios on the basis of these parameters. The generator discussed here has been developed within the simulation project at the HHLA Container-Terminal Altenwerder in Hamburg, Germany. Nevertheless, its structure is general enough to be applied to any other terminal as well.

Keywords. Container logistics, container terminal, scenario generator, simulation.

1 Introduction

Since the 1960s, the container has gained an enormous importance in worldwide trade and transportation of goods. This is due to both increasing containerization (which means that the number of goods transported in containers has steadily grown) and increasing world trade. As a consequence, new container terminals are being built and existing ones are extended in order to cope with the growing number of containers. In addition, the container terminals face the challenge of turning around not only more but also larger ships in the shortest possible time. To meet these objectives, container terminals employ innovative (and often automated) equipment and optimize their logistic processes. Given the increasing importance of efficient container terminal logistics, it is no surprise that practitioners as well as researchers continuously develop new optimization approaches as well as simulation models.

Optimization problems include the allocation of berths to arriving vessels (see Guan and Cheung [11], Imai et al. [14, 15], Lim [21]) as well as scheduling the loading and unloading operations of quay cranes (see Daganzo [5], Peterkofsky and Daganzo [24]). A berth assignment approach which simultaneously considers quay crane capacities has been developed by Park and Kim [23]. Internal transportation of containers, particularly between the quay and the stack, has been studied for

straddle carriers (see Böse et al. [4], Kim and Kim [18], Steenken et al. [26]) and for automated guided vehicles (see Bae and Kim [2], Evers and Koppers [7], Grunow et al. [10]). The problem of allocating and scheduling stacking cranes has been addressed by Zhang et al. [30]. In addition, a general approach for scheduling equipment such as straddle carriers, automated guided vehicles, and stacking cranes as well as manpower has been proposed by Hartmann [13]. Finally, strategies for locating containers in the yard have been discussed for two different cases, namely import containers (see de Castilho and Daganzo [6], Kim and Kim [16]) and export containers (see Kim et al. [17], Taleb-Ibrahimi et al. [27]). An approach covering both cases has been proposed by Zhang et al. [29]. A more comprehensive overview of literature on optimization issues is given by Meersmans and Dekker [22]. While these papers consider sea port container terminals, an optimization approach for an inland container terminal (i.e., without vessels or ships) has been presented by Aliche [1].

Simulation models are developed to evaluate the dynamic processes on container terminals. This allows to generate and analyze statistics such as average productivity, average waiting time (e.g., of a quai crane waiting for a straddle carrier), and average number of shuffle moves in the stack. This way, potential bottlenecks can be identified. Depending on the application, detailed simulation models usually cover both the physical resources (particularly the equipment such as cranes and vehicles) and the components for control and strategies (hence, simulation models provide a testing environment for optimization algorithms). Simulation projects can be carried out when building a new terminal (see Schütt and Hartmann [25]) or when analyzing or modifying an existing one (see Legato and Mazza [20], Yun and Choi [28]). In either case, plans concerning the logistic processes (such as capacity extensions, alternative stacking strategies or new scheduling algorithms) can be tested by means of simulation before they are actually implemented. Moreover, simulation models can be employed as a decision support system for the terminal management (see Gambardella et al. [8, 9]). For example, one could use the actual data of the next hours to simulate the next shift in order to evaluate impact of decisions (e.g., on the assignment of resources) in advance.

This paper introduces an approach to generate scenarios for container terminals. The goal of the generator is to produce realistic scenarios that provide all required input data for detailed simulation models of container terminals. Such data is usually necessary for simulations of container terminals to be built or extended. Of course, for such projects, appropriate real data is not available when the simulation is carried out, thus artificially generated data is needed. On the basis of adjustable parameters, the generator computes arrivals for deep-sea vessels, feeder ships, trains, and trucks including lists of containers to be unloaded and loaded. By specifying the parameters appropriately, one can generate realistic scenarios for terminals of different sizes. It is possible to adjust the distributions among the modes of transport, the ship sizes, the arrival distributions over time, and the distributions of various container properties. The generator has been developed within the simulation project of the new HHLA Container-Terminal Altenwerder in Hamburg, Germany (for details on this terminal see Baker [3]). Nevertheless, it is general enough to cover almost any seaport container terminal.

The paper is organized as follows. After a description of the general concept, it outlines the parameters that have been identified as being important to produce realistic scenarios for container terminal logistics. Next, we define a procedure to construct scenarios on the basis of these parameters. The procedure is then evaluated by comparing the generated scenarios with real-world statistics. Finally, we give a detailed discussion of applications of the proposed generator and close the paper with a few concluding remarks.

2 General Concept

In this paper, a scenario for a container terminal is defined as the data concerning all ship, train, and truck arrivals within a specified horizon including the information related to the containers being delivered or picked up. That is, for each day within the horizon, the scenario includes the

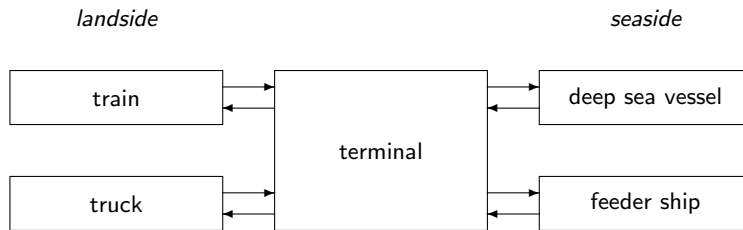


Figure 1: Modes of transport

arrival time of any ship, train, and truck together with the number of containers delivered and picked up. For each of these containers, detailed information on size (20' oder 40'), type (reefer, oversized etc.), and destination are given. Before we describe the parameters and algorithms of the generator in detail, this section summarizes some aspects of the overall concept.

2.1 Modes of Transportation

The generator distinguishes four different types of transport modes which are depicted in Figure 1. On the seaside we have large vessels for world-wide service as well as (typically smaller) feeder ships for regional service. The landside is made up by trains and trucks. With these four predefined transport modes, it is possible to adjust realistic parameters of the individual transport modes (usually, a vessel carries substantially more containers than a train). Equally important, the distribution of containers among the four transport modes can be specified (for example, one can determine how many of the containers delivered by truck will be picked up by vessel and by feeder ship). Hence, the generator allows for container flows between any of the four modes. Considering the port of Hamburg (and hence also the Container-Terminal Altenwerder), the container flow between large vessels and feeder ships is of particular importance. Therefore, we distinguish these two categories instead of having just a single seaside mode “ship.” Nevertheless, it is possible to adjust the parameters in a way that only one type of ship is considered.

Note that our definition of transport modes is more general than approaches in the literature (see, e.g., Zhang et al. [30]) which often consider only one seaside and one landside mode and container flows either from landside to seaside (often referred to as “export”) or from seaside to landside (“import”).

2.2 Interdependencies

The interdependencies within a real-world container terminal are considerably complex. This includes the distribution of containers among the transport modes on delivery and pick-up as mentioned above. At the same time, the distribution of arrival times and sizes of the means of transportation must be considered, as must be the dwell time distribution of containers (the dwell time is the time a container spends on the terminal until it is picked up). In other words, all containers must be assigned to a transport mode for delivery and to a transport mode for pick-up such that the distributions on transport modes, transport mode sizes, transport mode arrival times, and container dwell times are matched simultaneously. For example, this means that there has to be a concentration of container deliveries for a specific vessel in the days before the vessel arrives as well as a concentration of container pick-up in the days after the vessel’s arrival.

With regard to these interdependencies, the goal of the generator concept is twofold: First, it should provide parameters that allow to control the distributions mentioned above. Second, it should provide an algorithm that observes these distributions simultaneously in order to reflect the interdependencies.

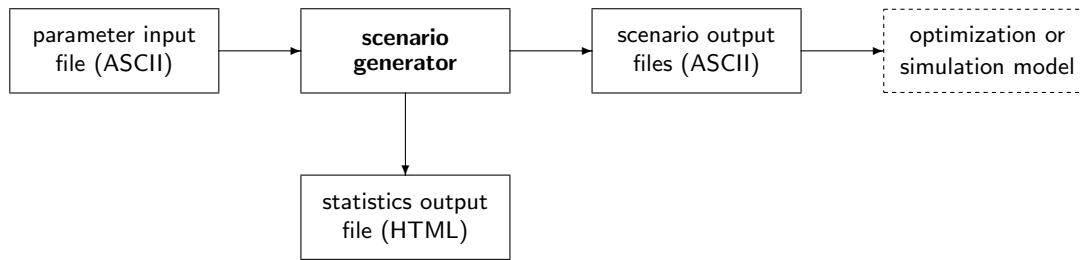


Figure 2: File concept of the scenario generator

2.3 File Concept

The generator works with several files. An overview of the file concept is sketched out in Figure 2. The user specifies the settings of the input parameters in a text file. A detailed description of the parameters is given in Section 3. After reading the parameters and computing a scenario, the generator writes two types of output files which are summarized in the following paragraphs.

The scenario itself is written into text files which can then serve as input for simulation models or optimization algorithms (for each day in the horizon and each of the four transport modes, a separate file is produced in order to avoid huge output files). The scenario data contains all information that is needed to perform a detailed simulation. The structure of the output files can be summarized as follows. The means of transportation are listed in the order of arrival. Each vessel, feeder ship, train, and truck is described by a line that contains the transport mode, a unique ID, an arrival day and time, and the numbers of containers to be loaded and unloaded. For vessels and feeder ships, also the length and the number of bays are given. Then the attributes of all containers to be loaded and unloaded follow, that is, a unique container ID, size, destination, weight, information on the type (empty, reefer, dangerous content, oversized), as well as pick-up information (which transport mode will pick it up on what day).

For illustration, we consider an example output file for deep sea vessels. Table 1 displays a few lines of such a file. Two types of lines are distinguished. Lines describing a transport mode arrival start with “T:” while lines describing a container start with “C:”. After the line for the transport mode arrival, there is a line for each container to be unloaded and, subsequently, a line for each container to be loaded. In what follows, the numbers of Table 1 are underlined in order to ease the explanation. The first line of Table 1 contains the information on the arrival of a vessel. The transport mode is indicated by first number; 1 stands for vessel (2 would stand for feeder, 3 for train, and 4 for truck). The vessel with ID 18 arrives on day 8 of the horizon at 7:40 in the morning. It will unload 824 containers and load 788. The length of the vessel is 280 m, it has 20 bays. The next line specifies the first container to be unloaded. The container with ID 8127 will be picked up by transport mode 2, i.e. a feeder, with ID 147 on day 14. The destination of the container (port, city) is 4. Next, 1 indicates a 40’ container (0 would imply a 20’ container) with a weight of 28 tons. It is neither a reefer container (0) nor an IMO container (0) nor an OOG container (0). A reefer container would be indicated by 1, and IMO and OOG containers would be indicated by a number > 0 referring to the specific type (Subsection 3.4 provides an explanation of reefer, IMO, and OOG containers). The last number in the line, 0, indicates that it is not an empty container. The following lines describe containers 8128 and 8129, the remaining containers to be unloaded are skipped in Table 1. Then the containers to be loaded are given, starting with container 32892. The format is the same as for the containers to be unloaded. Note that the pick-up information (transport mode, ID, and day of pick-up) is set to -1 since it makes only sense for containers to be unloaded (of course, all these containers are picked up by vessel 18 arriving on day 8).

The generator produces a separate statistics file (the format of this file is HTML so it can

T:	1	18	8	7	40	824	788	280	20	
C:	8127	2	147	14	4	1	28	0	0	0
C:	8128	2	85	9	3	1	23	0	3	0
C:	8129	3	109	12	8	1	36	1	0	0
[...]										
C:	32892	-1	-1	-1	2	1	41	0	0	0
C:	32979	-1	-1	-1	1	0	17	0	0	1
C:	33031	-1	-1	-1	1	1	25	0	0	0
[...]										

Table 1: Example output file of the scenario generator

T	horizon (number of days)
T_1	first day, $T_1 \in \{\text{Monday}, \dots, \text{Sunday}\}$
C_i	number of containers delivered by mode i within the horizon T
c_{ij}	fraction of containers picked up by mode j among all containers arriving by mode i , $i, j \in \{\text{vessel}, \text{feeder}, \text{train}, \text{truck}\}$
τ	maximum dwell time (days)
δ_t	fraction of containers with a dwell time of t days, $t \in \{0, \dots, \tau\}$

Table 2: General parameters

easily be viewed with a browser). This file illustrates the generated scenario by providing an overview, and it offers a possibility to check the scenario with respect to plausibility. The input parameter settings are displayed together with the corresponding aggregated information from the generated scenario (e.g., fraction of 20' containers), which allows for a convenient comparison. For more complex parameters which describe distributions, the correlation between parameters and generated figures is given (e.g., distribution of container dwell times). Some additional information is also provided, including vessel schedules as well as the overall numbers of containers arriving and departing each day, together with the resulting number of containers in the stack.

3 Scenario Parameters

The parameters allow the user to specify the scenario to be generated. The goal was to determine parameters for which planning data and/or statistics are typically available in practice because we want to produce scenarios that are as realistic as possible.

The following subsections describe the parameters which are classified into four groups. These are general parameters like the horizon and the number of containers arriving within that horizon, parameters concerning the sizes of the modes of transport, parameters dealing with arrival frequencies, and parameters reflecting container properties like size and weight distribution.

3.1 General Parameters

A summary of the general parameters is given in Table 2. In what follows, they are discussed in more detail. The first of the general parameters are related to the horizon of the scenario to be generated. The length T of the horizon (in days) has to be specified along with the first day T_1 of the horizon which can be a Monday, Tuesday and so on.

Next, we consider the number of containers which arrive at the terminal within the horizon. We define parameters for the numbers C_{vessel} , C_{feeder} , C_{train} , and C_{truck} of containers that arrive

at the terminal by deep sea vessel, feeder ship, train, and truck, respectively, during the horizon of T days. Clearly, this parameter set allows to produce scenarios for container terminals of any size.

For example, a medium sized terminal handles about one million containers per year. Typically, more than 50% of the containers arrive via the seaside. For the port of Hamburg, typical figures are 45% arriving by vessel, 15% by feeder ship, 15% by train, and 25% by truck.

The following parameters take care of the distribution of the containers among the four modes of transport when picked up at the terminal. Let us consider two modes of transport $i, j \in \{\text{vessel, feeder, train, truck}\}$ (for the sake of readability, we refer to the transport modes by symbolic constants). We denote as c_{ij} the fraction of those containers that are picked up by mode j among those that arrive by mode i . In other words, of the C_i containers arriving by mode i , $C_i \cdot c_{ij}$ containers are picked up by mode j . Note that this definition leads to the condition $\sum_j c_{ij} = 1$ which must be fulfilled for each mode i . The number of containers picked up by train, for example, can be computed as

$$C_{\text{vessel}} \cdot c_{\text{vessel,train}} + C_{\text{feeder}} \cdot c_{\text{feeder,train}} + C_{\text{train}} \cdot c_{\text{train,train}} + C_{\text{truck}} \cdot c_{\text{truck,train}}.$$

If none of the containers arriving by train will be picked up by train, one simply has to set $c_{\text{train,train}} = 0$. By specifying all C_i and c_{ij} , the user has completely defined the number of containers and their distribution among the four modes of transport on arrival and pick-up.

To give the reader an idea of realistic figures (again for the port of Hamburg), 30% of the containers arriving by vessel are picked up by feeders, 30% by train, and 40% by truck. Between 75% and 95% of the containers arriving by the other three transport modes are picked up by vessel. It is interesting to note that some (although only a very few) containers arriving by a landside mode are also picked up by a landside mode (which can be captured by the general parameter structure).

The last set of the general parameters determines the dwell time, that is, the number of days a container spends on the terminal before it is picked up. First, the maximum dwell time τ (in days) has to be specified. Then for each possible dwell time $t = 0, \dots, \tau$ the fraction d_t of containers with that dwell time has to be given. For example, $d_1 = 0.25$ means that 25% of the containers are picked up after one day in the stack. Additionally setting $d_2 = 0.15$ and $d_3 = 0.1$, one has specified that approximately 50% of the containers are picked up in the first three days after arrival (which is a realistic figure). Also note that the average dwell time (which can be derived from the distribution) has a significant impact on the level to which the terminal stack capacity is filled. The average dwell time varies between terminals, but realistic values range between 3 and 6 days.

3.2 Parameters for Means of Transportation

The second group of parameters determines the modes of transport. An overview is given in Table 3. Let us consider the deep sea vessels first. The user can define several classes of vessels, that is, different vessel sizes. After specifying the number $n_{\text{vessel}}^{\text{class}}$ of classes, each class $k \in \{1, \dots, n_{\text{vessel}}^{\text{class}}\}$ is described by the following parameters. Each class is associated with a range of the number of containers that a vessel of this class will unload. (Note that a range is given only for containers to be unloaded. It will become clear below that the number of containers to be loaded is determined by the generator without a parameter range.) Next, ranges for the length and the number of bays of the vessel class have to be given. Finally, the fraction (i.e., relative occurrence) of vessels of each class has to be specified. In reality, the number of containers unloaded by a deep sea vessel varies within a broad range (often between 50 and 1500). This makes the definition of different vessel classes indispensable.

So far, we have described the parameters for deep sea vessels. The parameters for feeder ships are analogous. For trains, the same parameters apply with the exception that length and, of course, number of bays are not considered. Finally, for trucks, different classes are not considered. For the sake of simplicity, we assume that one truck either delivers a single container or picks up a single container (in the real world application this generator was originally designed for, no data on the distribution of trucks carrying different numbers of containers were available).

n_i^{class}	number of classes of mode of transport $i \in \{\text{vessel, feeder, train}\}$
s_{ik}	fraction of class k for mode $i \in \{\text{vessel, feeder, train}\}$
\underline{S}_{ik}	minimum number of delivered containers in class k of mode $i \in \{\text{vessel, feeder, train}\}$
\overline{S}_{ik}	maximum number of delivered containers in class k of mode $i \in \{\text{vessel, feeder, train}\}$
\underline{L}_{ik}	minimum ship length in class k of mode $i \in \{\text{vessel, feeder}\}$
\overline{L}_{ik}	maximum ship length in class k of mode $i \in \{\text{vessel, feeder}\}$
\underline{B}_{ik}	minimum number of bays in class k of mode $i \in \{\text{vessel, feeder}\}$
\overline{B}_{ik}	maximum number of bays in class k of mode $i \in \{\text{vessel, feeder}\}$
F	factor for the maximal relation of containers unloaded and loaded by a vessel, feeder, or train

Table 3: Parameters for means of transportation

D_{id}	fraction of mode $i \in \{\text{vessel, feeder, train, truck}\}$ arriving on day $d \in \{\text{Monday}, \dots, \text{Sunday}\}$
n_i^{timeslot}	number of time slots of arrivals of mode $i \in \{\text{vessel, feeder, train, truck}\}$
h_{ik}	fraction of arrivals in timeslot k for mode $i \in \{\text{vessel, feeder, train, truck}\}$
\underline{H}_{ik}	first hour of time slot k for mode $i \in \{\text{vessel, feeder, train, truck}\}$
\overline{H}_{ik}	last hour of time slot k for mode $i \in \{\text{vessel, feeder, train, truck}\}$

Table 4: Parameters for arrival frequencies

The last parameter F limits the relation of the number of containers unloaded and loaded by the same vessel, feeder, or train. If the number of containers to be unloaded and loaded are denoted as U and L , respectively, this parameter implies that the condition $L \leq F \cdot U$ must be fulfilled. The idea behind this is that the classes defined above only determine the number of containers to be unloaded. In order to keep the number of containers to be loaded realistic, it can be limited by this parameter. Note that the parameter F is the same for vessels, feeder ships, and trains (since trucks carry only a single container by definition, it does not apply to trucks). The application of parameter F will become more clear in Subsection 4.4.

3.3 Parameters for Arrival Frequencies

The third group of parameters determines the distribution of arrivals over time (for a summary see Table 4). For each of the four transport modes, the arrival frequencies can be influenced separately. This is necessary because the four modes usually have different arrival distributions over time.

First, for each day $d \in \{\text{Monday}, \dots, \text{Sunday}\}$ and each transport mode i , the fraction D_{id} arriving on that day has to be given. For example, $D_{\text{feeder}, \text{Monday}} = 0.23$ means that 23% of the feeders arrive on Mondays. Over the seven days of the week, these fractions must sum up to 1, that is, we require $\sum_{d=\text{Monday}}^{\text{Sunday}} D_{id} = 1$ for each mode i . Next, one day can be partitioned into time slots which allow to capture the arrival rates within a day. Note that the arrival characteristics of a transport mode within a day are assumed to be the same on each day. A time slot is given by the start and end hour (ranging between 0 and 24) and by the fraction arriving within that time slot. Hence, a time slot can reflect a single hour or blocks of several hours.

Considering the notation of Table 4, we show how to reflect the arrival behavior of trucks as an example. Due to German law, trucks are not allowed to travel on Sundays which leads to a fraction of $D_{\text{truck}, \text{Sunday}} = 0$ for container terminals in Germany. Within a day, trucks arrive rather in the daytime than during the night, and their arrivals peak at early afternoon. In a rather rough approach, we could define $n_{\text{truck}}^{\text{timeslot}} = 4$ time slots of 6 hours each (e.g., the second time slot would cover the hours between $\underline{H}_{\text{truck}, 2} = 6$ and $\overline{H}_{\text{truck}, 2} = 12$). The fractions of the four time slots could

$p^{20'}$	fraction of 20' containers ($1 - p^{20'}$ is the fraction of 40' containers)
p^{empty}	fraction of empty containers
p^{reefer}	fraction of reefer containers
p^{IMO}	fraction of IMO containers, i.e., containers with dangerous goods
n^{IMO}	number of different IMO container types
p_k^{IMO}	fraction of containers of IMO type k among all IMO containers
p^{OOG}	fraction of OOG containers, i.e., oversized containers
n^{OOG}	number of different OOG container types
p_k^{OOG}	fraction of containers of OOG type k among all OOG containers
$n^{\text{weight},20'}$	number of weight groups for 20' containers
$p_k^{\text{weight},20'}$	fraction of 20' containers with weight group k
$\underline{w}_k^{20'}$	minimum weight of weight group k for 20' containers
$\overline{w}_k^{20'}$	maximum weight of weight group k for 20' containers
$w_{\text{empty}}^{20'}$	weight of empty 20' container (weight parameters for 40' containers are analogous)
n^{dest}	number of container destinations
p_{ik}^{dest}	fraction of containers with destination k among all containers picked up by mode $i \in \{\text{vessel, feeder, train, truck}\}$

Table 5: Parameters for container properties

be $h_{\text{truck},1} = 0.1$, $h_{\text{truck},2} = 0.3$, $h_{\text{truck},3} = 0.4$, and $h_{\text{truck},4} = 0.2$.

3.4 Parameters for Container Properties

The properties of a container (such as size, weight, and destination port) are important for vessel stowage as well as for determining a location for the container in the yard (cf. Kim et al. [17]). Therefore, they are relevant for detailed simulations. In what follows, we summarize the parameters that allow to specify the distribution of various container attributes. An overview is given in Table 5.

The first parameter defines the container size. We distinguish 20' and 40' containers. The user has to enter the fraction $p^{20'}$ of 20' containers; $1 - p^{20'}$ is the fraction of 40' containers. A realistic value for $p^{20'}$ is between 0.4 and 0.5. Next, the fraction p^{empty} of empty containers has to be given.

The following parameters are related to special container types for which specific constraints for yard locations must be observed. Reefer containers require a connection (usually to electricity) to keep their content cool or frozen. These connections are available only in designated areas of the yard. Other containers may contain dangerous goods such as specific chemicals. These containers are often called IMO containers after the International Maritime Association (IMO). Here, complex restrictions have to be considered. For example, a minimum distance between the yard locations of two containers with certain different chemicals must be observed in order to avoid the danger of a chemical reaction. Finally, OOG (out of gauge) containers are oversized containers which require more space than normal containers. For each of these three categories, the fraction has to be given (for example, p^{OOG} denotes the fraction of OOG containers among all containers).

The IMO and OOG containers may be further classified into different groups. In both cases, the number of groups has to be specified, along with the fraction of each group. For illustration, consider the following example of $n^{\text{OOG}} = 2$ different OOG container groups. Let us assume that 30% of the OOG containers are open top containers for which the load exceeds the container height such that no other container may be put on top of them. Moreover, 70% of the OOG containers are open on one side such that the width of the container is exceeded by the load. These two OOG groups can then be captured by defining the fractions $p_1^{\text{OOG}} = 0.3$ and $p_2^{\text{OOG}} = 0.7$. If, however, separate groups of OOG containers are not needed, one would simply define a single group by

setting $n^{\text{OOG}} = 1$ and $p_1^{\text{OOG}} = 1$. Considering the IMO containers, the parameters allow to apply the standard categorization defined by the International Maritime Association.

The next set of parameters specifies the container weight. Starting with 20' containers, the number of weight groups has to be specified. Then for each weight group the fraction of that group together with a lower and an upper weight limit are entered (the latter two can be interpreted as, e.g., metric tons). Subsequently, weight groups for 40' containers have to be defined. Finally, the weights of empty 20' and empty 40' containers are required.

The last set of parameters for container attributes is concerned with the destination of a container. First, the user specifies the number n^{dest} of destinations. Considering container terminals in Hamburg as an example, the destinations would subsume destination ports for large vessels such as Hong Kong and Singapore, destination ports for feeder ships such as Gothenborg, and destinations for trains such as Munich. As the destination of a container may be unknown (especially if a container will be picked up by a truck), it is useful to define an additional destination representing an unknown destination. Obviously, the destination is related to the means of transportation that picks up the container. Therefore, for each of the four transport modes, a separate distribution among the destinations can be given. For mode of transport $i \in \{\text{vessel, feeder, train, truck}\}$ and destination $k \in \{1, \dots, n^{\text{dest}}\}$, p_{ik}^{dest} represents the fraction of containers with destination k among all containers picked up by transport mode i . Clearly, this implies $\sum_{k=1}^{n^{\text{dest}}} p_{ik}^{\text{dest}} = 1$ for each mode $i \in \{\text{vessel, feeder, train, truck}\}$. For example, let us assume that destination $k = 3$ refers to an Asian port like Hong Kong. If 10% of the container picked up by large vessels go to Hong Kong but none of the containers picked up by feeder ships, one will set $p_{\text{vessel},3}^{\text{dest}} = 0.1$ and $p_{\text{feeder},3}^{\text{dest}} = 0$. It should be noted that, even for detailed simulations, it might not be necessary to model "real" destinations and ports exactly. Often, a number of abstract destinations will be sufficient.

4 Algorithm for Generating a Scenario

In this section, we summarize the algorithm that constructs a scenario. The goal of the algorithm is to compute a scenario in a way that the parameter settings are observed as closely as possible. Before we describe the algorithm which consists of four successive stages, we have a brief look at procedures that will be used in several parts of the algorithm.

4.1 Preliminaries

In several parts of the generator, we have to find an assignment that should match a distribution given as parameters as closely as possible. For example, the user can define classes of vessels (as well as feeders and trains). These classes determine the size of the vessels. Of course, when we compute vessels, the distribution of the generated vessel sizes should match the parameter settings. Similarly, the distributions of the arrival days and times should match the distributions selected by the user.

Let us give an abstract formulation of this problem setting. We assume that we have objects (e.g., vessels or containers) each of which must be assigned one out of m properties (e.g., arrival time slots for vessels or sizes for containers). The goal is to construct an assignment such that property $j \in \{1, \dots, m\}$ occurs with a fraction f_j (these fractions correspond to the distributions given by the parameters). In the generator, we employ the following two alternative procedures to assign properties to objects.

Randomized assignment. For each object i , the property is assigned randomly where the probability to select property $j \in \{1, \dots, m\}$ is given by fraction f_j . This is, of course, the straightforward approach. However, if the number of objects is relatively small, this method might generate distributions that do not match the fractions. Therefore, in those cases, we employ the following modified method.

Optimized assignment. In order to match the given fractions more accurately, we define the following simple greedy procedure. The first object $i = 1$ is assigned a property $j \in \{1, \dots, m\}$

randomly. Then we successively consider objects $i > 1$. The i -th object, $i > 1$, is assigned a property based on the previously generated assignments of objects $1, \dots, i - 1$. Let z_j denote the number of times property j has been selected so far. Object i is assigned property j^* with $f_{j^*} - \frac{z_{j^*}}{i} = \max\{f_j - \frac{z_j}{i} \mid j = 1, \dots, m\}$ (ties are broken arbitrarily). Subsequently, we update $z_{j^*} := z_{j^*} + 1$. This way, we always select the property with the largest difference between the selected fraction and the currently generated one. The assignment step is repeated until a stopping criterion is fulfilled, that is, until each object has been assigned a property.

4.2 Generating Transport Modes and Containers to be Unloaded

The first stage of the generator produces arriving vessels, feeders, trains, and trucks along with the related containers to be unloaded. Let us begin the description of the related procedure for vessels.

We have C_{vessel} containers that arrive by vessel. This number will now be split into individual vessels. Here, the distribution of the vessel classes as given by the parameters has to be observed. We apply the optimized assignment procedure described in Subsection 4.1. In each step of that procedure, a new vessel class $k \in \{1, \dots, n_{\text{vessel}}^{\text{class}}\}$ is selected. Based on the information related to class k , a specific vessel is determined. The number S of containers to be unloaded by that vessel is drawn randomly from $\{\underline{S}_{ik}, \dots, \overline{S}_{ik}\}$. Similarly, the length and the number of bays are selected randomly from the respective parameter intervals related to class k . This way, we obtain a new vessel in each step of the greedy procedure. In order to reflect the remaining number of containers, we set $C_{\text{vessel}} := C_{\text{vessel}} - S$ whenever we have determined a new vessel. The above step is repeated until no containers are left ($C_{\text{vessel}} = 0$) or until the number of remaining containers is smaller than the number that would be required for the smallest vessel type (in the latter case, the remaining containers are randomly added to the vessels that have already been defined).

The generation of feeders and trains is analogous with the exception that length and number of bays are not considered for trains. The construction of trucks, however, is different. Since we assume that a truck carries only a single container, we simply define a truck for each of the C_{truck} containers arriving by truck.

At the end of this stage, we have vessels, feeders, trains, and trucks together with the respective numbers of containers to be unloaded.

4.3 Generating Arrival Day and Time

In the second stage of the generator, an arrival day and time is computed for each vessel, feeder ship, train, and truck that was generated in the previous stage. We describe the computation of the deep sea vessels in more detail. The procedure for the remaining three transport modes is analogous.

The first step deals with the day of arrival. So far, we know the fraction of vessels that should arrive on each of the seven days of the week. This information is given by the parameters $D_{\text{vessel},d}$ for day $d \in \{\text{Monday}, \dots, \text{Sunday}\}$. We now have to transform these parameters such that they reflect the fractions of the days of the horizon. For each day $t = 1, \dots, T$ of the horizon, we set $D'_{\text{vessel},t} := D_{\text{vessel},d}$ if the t -th day of the horizon corresponds to day $d \in \{\text{Monday}, \dots, \text{Sunday}\}$ with respect to the first day T_1 . Then we calculate the fraction of vessels arriving on day $t = 1, \dots, T$ as

$$D''_{\text{vessel},t} = \frac{D'_{\text{vessel},t}}{\sum_{q=1}^T D'_{\text{vessel},q}}.$$

With this distribution, we are now ready to compute an arrival day for each vessel. Again, we apply the optimized assignment procedure of Subsection 4.1.

Having fixed the arrival day for each vessel, the second step computes a time on that day. Recall that the parameter settings include a specification of time slots $1, \dots, k, \dots, n_{\text{vessel}}^{\text{timeslot}}$ along with a distribution $h_{\text{vessel},k}$. Using the optimized assignment procedure of Subsection 4.1, we determine a

time slot for each vessel based on the given distribution. Subsequently, we consider the exact time of arrival. If a vessel is assigned time slot k , we draw the hour of arrival from $\{\underline{H}_{\text{vessel},k}, \dots, \overline{H}_{\text{vessel},k} - 1\}$. Finally, we draw the minutes past the hour from the interval $\{0, \dots, 59\}$.

At the end of this stage, we have vessels, feeders, trains, and trucks together with the respective number of containers to be unloaded as well as the arrival day and time.

4.4 Generating Container Pick-up

So far, we have generated lists of arriving means of transportation together with arrival date and time as well as the numbers of containers delivered. While we already know which container is delivered by which vessel, feeder ship, train, or truck, we now determine for each container which vessel, feeder ship, train, or truck will pick it up. For all containers, the following three steps are executed. Let us consider a container for which we want to compute the pick-up information. The assignment of pick-up properties to the container under consideration is done using the randomized assignment procedure of Subsection 4.1.

In the first step, we determine the transport mode that will pick up the container. If the container is delivered by transport mode i , then we use c_{ij} as the probability that it will be picked up by mode j ($i, j \in \{\text{vessel, feeder, train, truck}\}$), see Table 2. Let $j^* \in \{\text{vessel, feeder, train, truck}\}$ denote the transport mode selected to pick up the container.

The second step is to determine the day on which the container will leave the terminal. With t_1 we denote the day on which the container arrives at the terminal. Now we determine the dwell time of the container using the dwell time distribution. Let Δ^* be the selected dwell time. The day on which the container is picked up is then determined as $t_2 = t_1 + \Delta^*$.

Finally, the third step is to find a means of transportation of mode j^* that arrives on day t_2 . Among the means of transportation that fulfill these requirements, some might not be eligible for pick-up because they have already reached their maximal load. Given a means of transportation with U containers to be unloaded that has already been assigned L containers to load, it can only receive more containers to pick up if we have $L + 1 \leq F \cdot U$ (for the definition of parameter F see again Table 3). Let M be the set of the eligible means of transportation of mode j^* arriving on day t_2 . When we select one of them to pick up the container under consideration, we want larger means of transportation (e.g., larger vessels) to have a higher probability to be selected. To do so, we reflect the size of the means of transportation $m \in M$ by the number U_m of containers to be unloaded (recall that the number of containers to be unloaded has already been fixed). More precisely, the probability to chose $\mu \in M$ is defined as $p(\mu) = \frac{U_\mu}{\sum_{m \in M} U_m}$. Three special cases that may occur in the third step should briefly be mentioned:

- If no eligible means of transportation exist on that day (either because none of mode j^* arrive or because those that arrive have already reached their maximal load), then we start again with the first step.
- If day t_2 is not in the horizon (i.e., $t_2 > T$), then a means of transportation is not determined.
- If the transport mode j^* is truck, we define a new truck to pick up the container given that truck arrival on that day is possible (recall that we assumed trucks to deliver or pick up a single container, thus the selection mechanism of existing means of transportation as described above cannot be applied).

Observe that at the end of this stage we know for each vessel, feeder ship, train, and truck which containers it unloads and which containers it loads. Of course, we also know for each container which vessel, feeder ship, train, or truck delivers it and which one picks it up. Hence, we have completed the relationship between means of transportation and individual containers. We have taken into account the parameters concerning the classes (or sizes) of the means of transportation, the arrival time distributions over the days and within a day, and the dwell time distribution. In particular, the generation of container pick-up as described above ensures that

there is a concentration of container deliveries for a specific vessel in the days before the vessel arrives (and, analogously, a concentration of container pick-up in the days after the vessel's arrival).

4.5 Generating Container Properties

At this point, we have individual containers, each of which is associated with individual means of transportation for arrival and pick-up. The last stage of the generator consists of the assignment of properties to the containers. The procedure is quite straightforward. It makes use of the parameters of Section 3.4 and the randomized assignment of Subsection 4.1.

Using the fractions $p^{20'}$, p^{empty} , p^{reefer} , p^{IMO} , and p^{OOG} as probabilities, the related properties are fixed for each container. Next, for the containers that have been assigned the IMO attribute, a specific IMO type is selected. This is done using the fractions of the different IMO types as probabilities. Analogously, OOG containers are assigned an OOG type.

The destination of a container is determined in a similar way. However, the choice of the destination depends on the transport mode that picks up the container. That is, if a container is picked up by mode $i \in \{\text{vessel, feeder, train, truck}\}$, then the probability to select destination $k \in \{1, \dots, n^{\text{dest}}\}$ is given by p_{ik}^{dest} .

Finally, a container is assigned a weight. Here, the weight parameters related to the container size (20' or 40') are applied. If the container has been defined as empty, the related parameter for the weight of empty containers is used. Otherwise, a weight group k is drawn according to the related fractions which are employed as probabilities. The actual weight w is randomly chosen from the interval determined by the minimal and maximal weight of selected weight group k , that is, we select $w \in \{\underline{w}_k^{20'}, \dots, \bar{w}_k^{20'}\}$ or $w \in \{\underline{w}_k^{40'}, \dots, \bar{w}_k^{40'}\}$, respectively.

5 Experimental Validation

5.1 Evaluation of Generated Scenarios

In this section, we show that the proposed algorithm is appropriate for constructing scenarios that match the selected parameters. We report on experiments in which we generated scenarios based on the following setting. We have a horizon of 4 weeks and a total of 76,600 containers arriving within the horizon (this corresponds to a medium sized terminal with one million containers per year). The remaining parameter selections were done using statistics of the HHLA container terminal Burchardkai in Hamburg, Germany. For this setting, ten scenarios were generated.

The size of the output on disk is approximately 8.6 MB for such a scenario (i.e., all transport mode and container property information). On a Pentium 4 based computer with 1.6 GHz, the computation took on the average 9.9 seconds for one scenario (we coded the generator in C and used the lcc compiler under Windows XP). Hence, the algorithm is reasonably fast.

In order to validate the computed scenarios, we compared the statistics computed from the generated scenarios with the real-world statistics that were used as parameters. For the sake of brevity, we restrict the presentation of the results to the most important statistics concerning transport modes, arrival frequencies, and container dwell time. Table 6 summarizes the respective correlations between scenario characteristics and real-world statistics. With the exception of the dwell time distribution, the correlations are measured separately for the transport modes (recall that the respective parameters are given separately for the transport modes as well). The correlations are given as average values over the ten scenarios generated. As we can see, all correlations are clearly above 0.8, and most are even above 0.95. Only some of the correlations related to vessels are below 0.95 which is due to the fact that the number of vessels arriving within the horizon is relatively small.

The results indicate that the algorithm of the generator is capable of constructing scenarios in which the selected parameters are observed considerably well. Hence, if real-world statistics are

Correlation type	overall	vessel	feeder	train	truck
distribution of unloaded containers among transport modes	–	1.000	1.000	1.000	1.000
distribution of transport mode size classes	–	0.994	1.000	0.986	–
distribution of arrivals among weekdays	–	0.842	0.996	0.995	0.958
distribution of arrivals among time slots within a day	–	0.886	0.994	1.000	1.000
distribution of container dwell times	0.996	–	–	–	–

Table 6: Average correlations of generated data and control parameters

Correlation type	assignment	vessel	feeder	train	truck
distribution of arrivals among time slots within a day	randomized	0.350	0.650	0.998	1.000
distribution of arrivals among time slots within a day	optimized	0.886	0.994	1.000	1.000

Table 7: Average correlations of generated data and control parameters w.r.t. assignment method

used as input parameters, the generator produces realistic scenarios. These findings were confirmed in several other experiments based on different parameter settings.

5.2 Effect of Assignment Method

This subsection briefly analyzes the impact of the two different procedures for assigning properties to objects described in Subsection 4.1. Recall that we have defined a simple randomized method and an optimized method for assignment. Considering the assignment of time slots, Table 7 compares the resulting correlations if these two methods are used. For vessels and feeders, the optimized approach leads to time slot distributions that match the parameters by far better. The difference for trains is much smaller, but this is due to our parameter selection (due to the real-world statistics, we have only three time slots for trains whereas we have 24 time slots for the other three modes). Finally, since we have a huge number of trucks, the randomized method would be sufficient there.

Summarizing, the results confirm our decision to employ the optimized assignment approach for time slot selection. Similar results were obtained for the other steps of the generator.

6 Applications

Parameter-based generation of data is an important field in operations research (see, e.g., Hall and Posner [12], Kolisch et al. [19]). In scientific research, generated data is needed to carry out experiments with algorithms or modeling approaches. Parameters are employed to generate data systematically according to a specific experimental design. In practice, decisions on algorithms or strategies in control systems are made only if they have been tested before. While real-world data is often preferred for such tests, artificially generated data is used if real-world data is unavailable. This is the case for new container terminals to be built, for existing ones that will be expanded, and for analyzing future developments such as more ship arrivals or larger vessels. For applications in practice, it is of particular importance that the artificially generated data is realistic (otherwise, the results would not be an appropriate basis for decision support). The following subsections outline several applications of the generator proposed in this paper.

6.1 Stack Simulation at the Container-Terminal Altenwerder

The generator described in this paper has been developed within the simulation project of the HHLA Container-Terminal Altenwerder in Hamburg, Germany. It was applied within a simulation

study of the strategies for selecting yard blocks and slots for arriving containers. The study was carried out in the planning phase of the terminal, thus real-world data was not available. The purpose of the study was to test, improve, and parameterize the stacking strategies. A simulation-based approach was necessary to analyze the dynamic container arrivals and departures in a realistic online environment.

When a container arrives at the terminal, a position in the stack has to be determined. This is done using strategies which are associated with the following goals. First and most important, the number of shuffle moves has to be minimized. A shuffle move occurs if two containers stand on top of each other and the lower one has to be picked up first. In such a case, the upper one has to be moved to another position in the stack, which reduces the stacking crane (or straddle carrier) capacities for productive moves. While shuffle moves cannot be totally avoided, stacking strategies select positions such that the number of shuffle moves is kept to a minimum. Second, the distance over which a stacking crane (or straddle carrier) transports a container through a yard block should be minimized in order to use the equipment capacities efficiently.

In order to achieve the goals mentioned above, the stacking strategies make the selection on the basis of the information associated with a container. Considering the shuffle move minimization, two containers with the same properties (in particular, same vessel for pick-up and same weight class) can be stacked on top of each other (in such a case, the upper one can be picked first). It can also be a good idea to put a container on top of another if the expected departure time of the upper container is earlier than that of the lower one. The second goal implies that, e.g., a position on the seaside should be preferred for a container which arrives with a feeder and will be picked up by a vessel. Such considerations are usually employed within a priority based evaluation method for all available positions in the stack, and the best stack is selected (for reasons of confidence, details of the stacking strategies at the Container-Terminal Altenwerder cannot be given here).

The generator proposed in this paper was employed to produce the input data for the simulation of the stacking strategies. The parameters of the generator were adjusted such that they captured the specific planning assumptions of the Container-Terminal Altenwerder (e.g., the expected number of containers turned over per year). The remaining parameter settings were done on the basis of statistics from the HHLA Container-Terminal Burchardkai. In addition to the transport mode arrivals over time, the container properties were of particular importance because they are required by the stacking strategies. Each scenario (and thus also each simulation run) covered a horizon of several weeks (note that the impact of stacking strategies could not be observed within a shorter horizon).

The simulation model was developed using the emPlant software package. It was designed as follows. In a first step, a list of events is constructed from the scenario under consideration. For example, for a truck arrival with a request of a specific container, a related pick-up event is created for the truck's arrival time. Considering the arrival of a vessel, the arrival times of the containers to be unloaded are calculated on the basis of an assumed quai crane productivity. That is, for the containers arriving by a vessel, one event related to a single container is created every n seconds starting with the vessel's arrival time (the loading process as well as feeder and train arrivals are treated analogously). In the second step, the simulation moves through the event list, that is, a container related to an arrival event is put in the stack, and a container associated with a pick-up event is removed from the stack. The equipment (quai cranes, automated guided vehicles, stacking cranes) is not modeled explicitly since the transportation process was not in the focus of this study (in fact, modeling the transportation process would have slowed down the simulation runs). The stacking strategies and the stack itself are modeled on a detailed level. An arrival event triggers the stacking strategy computation and, subsequently, an update of the stack data with respect to the selected position. A pick-up event triggers again an update of the stack data with respect to the position from which the requested container is removed. In reality, an arriving container may be associated with missing or incorrect information on its pick-up time and transport mode. Since this is important when analyzing stacking strategies, the simulation model contains parameters to distort the input scenarios. Several statistics were incorporated into the simulation model in

order to evaluate different strategies, in particular number of shuffle moves (overall and for different container types), number of free groundslots, stacking crane distance per container, stacking height for different container types, and level of stack utilization over time. The model also contains a visualization with a view of individual containers as well as container types in the yard blocks.

For each generated scenario, a large number of simulation runs were carried out in order to test alternative stacking strategies and parameter settings for strategies (again, for reasons of confidence, the simulation results cannot be given here). The originally considered strategies did not perform well, thus they were modified on the basis of the simulation. Hence, the simulation led to improved stacking strategies before the terminal started operation.

6.2 Further Applications

The main purpose of the generator presented in this paper is to produce scenarios that can be used as input data for simulation models. Simulation approaches are used by researchers and practitioners to study the dynamic processes in container terminal logistics. Typically, a simulation model covers the terminal resources (particularly equipment such as quai cranes, straddle carriers, and stacking cranes), the stack, and the control strategies (e.g., equipment scheduling and stack reservation). The subject of a simulation study determines the level of detail to which a component is modeled. Generally speaking, arrivals of transport modes together with arriving and requested containers are the events that cause the execution of processes on the terminal (e.g., transportation tasks, stack reservation). Therefore, an event-based simulation model requires input data concerning transport mode arrivals (and hence container arrivals and requests) over time. This is the data that is produced by the generator proposed here. Given that realistic input data is provided, simulation models can be employed to support decisions on required resource capacities, to detect possible bottlenecks, and to select and adjust strategies, objective functions, and algorithms (e.g., for equipment scheduling).

In addition to simulation studies, the proposed generator can also be applied in experiments with optimization approaches that are not embedded into an online environment. The following well-known optimization problems require input data that can be produced by our generator.

In the berth allocation problem (see Guan and Cheung [11], Imai et al. [14, 15], Lim [21]), a list of ship arrivals is given. These ships have to be assigned a berth at the quai (which, of course, has a limited length). The objective is to minimize the waiting time for a free berth and hence the total time in port for the ships. Our generator produces ship arrivals over time which are needed as input data for this optimization problem. In addition to the arrival date and time of the ships, it also generates the number of containers to be loaded and unloaded. On the basis of a productivity rate, this number of containers can easily be converted into the time needed at the quai. Adding the quai length, one has obtained all problem data for berth allocation.

The quai crane scheduling problem (see Daganzo [5], Peterkofsky and Daganzo [24]) deals with the assignment of a limited number of quai cranes to vessels which arrive over time. A quai crane is allowed to move from ship to ship. Hence, a quai crane schedule determines for each quai crane at what time it is working on which vessel. Again, the objective is to minimize the time in port. Similarly to the berth planning problem, the generator produces the input data for the quai crane scheduling problem.

Both problems have been combined to a more realistic problem in which the berthing time of vessels depends on quai crane capacities (see Park and Kim [23]). Also for this integrated problem, the generator can be employed to produce test data. Note that, whereas the full scenario data was required for the simulation study described in Subsection 6.1, only parts of the scenarios are necessary for these optimization problems. Both the landside information (i.e., arrivals of trains and trucks) and the container properties can be skipped here.

7 Conclusions

In this paper, we presented an approach to generate scenarios for container terminals. A scenario contains data on arrivals of vessels, feeder ships, trains, and trucks together with lists of the containers to be delivered and picked up and the container attributes (size, weight etc.). The generation of a scenario is controlled by means of various parameters. The goal was to develop an easy-to-use generator that produces realistic scenarios. Therefore, we selected parameters that allow to use statistics and planning information that are typically available in practice. By employing real-world information for the parameters, the artificially generated scenarios become realistic.

After a description of the parameters and the algorithms of the generator, we have demonstrated by experiments based on real-world statistics that the algorithm produces scenarios that match the parameter settings appropriately well. Subsequently, we have sketched out various applications in research and practice. While the main purpose of the generator is to produce input data for simulation models of container terminals, it can also be used to generate test data for well-known optimization problems such as berth planning and quay crane scheduling. In particular, we have described a practical application of the generator within the simulation project at the HHLA Container-Terminal Altenwerder in Hamburg, Germany. In this project, the strategies for determining a position in the stack for arriving containers were examined. This application has shown that the generator is well suited for projects in practice.

Although the generator is already considerably general, extensions of the generator might be promising tasks in future research. A possible extension of the generator could be to compute the positions of the individual containers on the vessels and feeder ships. These positions are relevant when stowage plans are part of the problem considered. This is of particular importance if the process of loading vessels is examined in full detail. Often, each position on the vessel is associated with a specific container. This imposes a partial order on the containers to be loaded. Alternatively, clusters of cells may be associated with container attributes rather than individual containers. In the latter case, there are more degrees of freedom concerning the loading order, and specific strategies are needed to exploit this setting. Hence, the design of an extended generator (and particularly that of the additional parameters) would depend on the application of the stowage plans.

Acknowledgement. This research project has been carried out for the HHLA Container-Terminal Altenwerder in Hamburg, Germany, when the author was with LOGAS Gesellschaft für logistische Anwendungssysteme, Hamburg, Germany.

References

- [1] K. Alicke. Modeling and optimization of the intermodal terminal Mega Hub. *OR Spectrum*, 24:1–17, 2002.
- [2] J. W. Bae and K. H. Kim. A pooled dispatching strategy for automated guided vehicles in port container terminals. *International Journal of Management Science*, 6:47–70, 2000.
- [3] C. Baker. Altenwerder – the details. *Port Development International*, (07/08):24–25, 1999.
- [4] J. Böse, T. Reiners, D. Steenken, and S. Voß. Vehicle dispatching at seaport container terminals using evolutionary algorithms. In R. H. Sprague, editor, *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 377–388. IEEE, Piscataway, 2000.
- [5] C. F. Daganzo. The crane scheduling problem. *Transportation Research B*, 23:159–175, 1989.
- [6] B. de Castilho and C. F. Daganzo. Handling strategies for import containers at marine terminals. *Transportation Research B*, 27:151–166, 1993.
- [7] J. J. M. Evers and S. A. J. Koppers. Automated guided vehicle traffic control at a container terminal. *Transportation Research A*, 30:21–34, 1996.
- [8] L. M. Gambardella, G. Bontempi, E. Taillard, D. Romanengo, G. Raso, and P. Piermari. Simulation and forecasting of an intermodal container terminal. In A. G. Bruzzone and E. J. H. Kerckhoffs, editors, *Simulation in Industry – Proceedings of the 8th European Simulation Symposium*, pages 626–630. SCS, Ghent, Belgium, 1996.
- [9] L. M. Gambardella, A. E. Rizzoli, and M. Zaffalon. Simulation and planning of an intermodal container terminal. *Simulation*, 21:107–116, 1998.

- [10] M. Grunow, H.-O. Günther, and M. Lehmann. Dispatching multi-load AGVs in highly automated seaport container terminals. *OR Spectrum*, 26:211–235, 2004.
- [11] Y. Guan and R. K. Cheung. The berth allocation problem: Models and solution methods. *OR Spectrum*, 26:75–92, 2004.
- [12] N. G. Hall and M. E. Posner. Generating experimental data for computational testing with machine scheduling applications. *Operations Research*, 49:854–865, 2001.
- [13] S. Hartmann. A general framework for scheduling equipment and manpower at container terminals. *OR Spectrum*, 26:51–74, 2004.
- [14] A. Imai, K. Nagaiwa, and C. W. Tat. Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, 31:75–94, 1997.
- [15] A. Imai, E. Nishimura, and S. Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research B*, 35:401–417, 2001.
- [16] K. H. Kim and H. B. Kim. Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, 59:415–423, 1999.
- [17] K. H. Kim, Y. M. Park, and K.-R. Ryu. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124:89–101, 2000.
- [18] K. Y. Kim and K. H. Kim. A routing algorithm for a single straddle carrier to load export containers onto a container ship. *International Journal of Production Economics*, 59:425–433, 1999.
- [19] R. Kolisch, A. Sprecher, and A. Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41:1693–1703, 1995.
- [20] P. Legato and R. M. Mazza. Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research*, 133:537–547, 2001.
- [21] A. Lim. The berth planning problem. *Operations Research Letters*, 22:105–110, 1998.
- [22] P. J. M. Meersmans and R. Dekker. Operations research supports container handling. Technical Report EI 2001-22, Econometric Institute, Erasmus University Rotterdam, 2001.
- [23] Y.-M. Park and K. H. Kim. A scheduling method for berth and quai cranes. *OR Spectrum*, 25:1–23, 2003.
- [24] R. I. Peterkofsky and C. F. Daganzo. A branch and bound solution method for the crane scheduling problem. *Transportation Research B*, 24:159–172, 1990.
- [25] H. Schütt and S. Hartmann. Simulation in Planung, Realisierung und Betrieb am Beispiel des Container-Terminals Altenwerder. In D. P. F. Möller, editor, *Frontiers in Simulation — Simulationstechnik, 14. Symposium in Hamburg*, pages 425–430. SCS, Ghent, Belgium, 2000.
- [26] D. Steenken, A. Henning, S. Freigang, and S. Voß. Routing of straddle carriers at a container terminal with the special aspect of internal moves. *OR Spectrum*, 15:167–172, 1993.
- [27] M. Taleb-Ibrahimi, B. de Castilho, and C. F. Daganzo. Storage space vs. handling work in container terminals. *Transportation Research B*, 27:13–32, 1993.
- [28] W. Y. Yun and Y. S. Choi. A simulation model for container-terminal operation analysis using an object-oriented approach. *International Journal of Production Economics*, 59:221–230, 1999.
- [29] C. Zhang, J. Liu, Y.-W. Wan, K. G. Murty, and R. J. Linn. Storage space allocation in container terminals. *Transportation Research B*, 37:883–903, 2003.
- [30] C. Zhang, Y.-W. Wan, J. Liu, and R. J. Linn. Dynamic crane deployment in container storage yards. *Transportation Research B*, 36:537–555, 2002.