Volume 36 Issue 5 May 2009 ISSN 0305-0548

computers & operations research

*Part Special Issue:*
**Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X)**
*Guest Editors:*
**Juan A. Mesa, Blas Pelegrin and Justo Puerto**

**Editor: Stefan Nickel**

Available online at
ScienceDirect
www.sciencedirect.com

Computers & Operations Research 36 (2009) 1711 – 1725

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

# Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal

Matthew E.H. Petering[a,*], Katta G. Murty[b]

[a]Department of Industrial and Manufacturing Engineering, University of Wisconsin—Milwaukee, Milwaukee, WI 53201, USA
[b]Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117, USA

ARTICLE INFO

Available online 29 April 2008

*Keywords:*
Container terminal
Maritime shipping
Block length
Yard crane deployment
Simulation
Terminal design
Gross crane rate

ABSTRACT

As more and more container terminals open up all over the world, terminal operators are discovering that they must increase quay crane work rates in order to remain competitive. In this paper we present a simulation study that shows how a terminal's long-run average quay crane rate depends on (1) the length of the storage blocks in the terminal's container yard and (2) the system that deploys yard cranes among blocks in the same zone. Several different block lengths and yard crane deployment systems are evaluated by a fully dynamic, discrete event simulation model that considers the detailed movement of individual containers passing through a vessel-to-vessel transshipment terminal over a several week period. Experiments consider four container terminal scenarios that are designed to reproduce the multi-objective, stochastic, real-time environment at a multiple-berth facility. Results indicate that a block length between 56 and 72 (20-ft) slots yields the highest quay crane work rate, and that a yard crane deployment system that restricts crane movement yields a higher quay crane work rate than a system that allows greater yard crane mobility. Interestingly, a block length of 56–72 slots is somewhat longer than the average block in use today. The experiments provide the first direct connection in the literature between block length and long-run performance at a seaport container terminal. The simulator can be suitably customized to real, pure-transshipment ports and adequately tuned to get an appreciable prescriptive power.

## 1. Introduction

The recent increase in international trade of finished consumer goods has placed the maritime container shipping industry at the center of our global economy. Today, almost all overseas shipping of furniture, toys, footwear, clothing, auto parts, bananas, electronics components, and computers is done via standardized 20, 40, and 45 ft long steel containers aboard deep-sea container vessels. In addition, the amount of fruit, vegetables, fish, meat, and general foodstuffs shipped in refrigerated containers is increasing.

As of January 2008, the world cellular fleet consisted of 4282 vessels of various sizes solely devoted to transporting containerized cargo. The total capacity of these vessels was the equivalent of some 10.7 million 20 ft containers [1]. Four years earlier, in February 2004, the figure was only 6.54 million 20 ft containers [2]. This represents a growth rate exceeding 1% *per month*. Such a rapid expansion of the container sector, combined with a heightened concern over customer service and security, has made the container shipping industry a major focus of operational research in the past decade. Indeed, with today's just-in-time global supply chain, improving the efficiency of container shipping processes is more important than ever.

This paper focuses on design and operations control problems at seaport container terminals. Container terminals are the places in seaports where container vessels are loaded and unloaded, and where containerized cargo is temporarily stored while awaiting a future journey. In the world's largest terminals, there are hundreds of trucks and cranes operating around the clock—24 h a day, 365 days a year—to transfer more than 100,000 containers between ships and shore each week. A brief summary of container terminal equipment and operations now follows.

Containers (boxes) come in three standard lengths—20, 40, and 45 ft—and are 8 ft wide and either 8.5 or 9.5 ft high. When a container vessel arrives at the port, the terminal provides a berth where it docks. Then the QCs (quay cranes, shore cranes) at the berth begin unloading and loading it, with each QC handling cargo in a different section along the length of the vessel known as a hatch. Larger vessels may have up to two dozen hatches, so typically 3–4 QCs work on a vessel at a time.

Cargo passes through a container terminal in three ways: it may be imported, exported, or transshipped. Fig. 1 shows each of these

---

[*] Corresponding author. Tel.: +1 414 229 3448; fax: +1 414 229 6958.
*E-mail address:* mattpete@uwm.edu (M.E.H. Petering).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.04.007

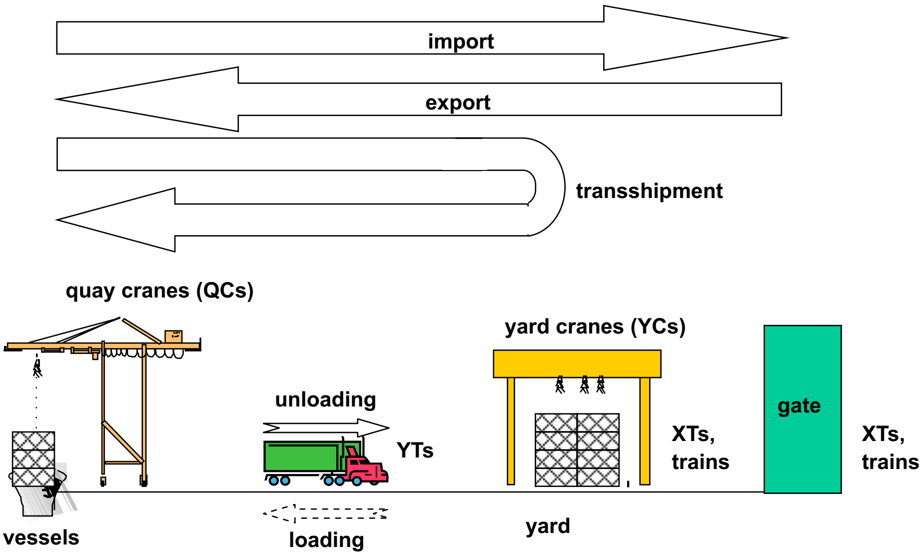

**Fig. 1.** Cargo may pass through a container terminal in three different ways.

processes and the equipment involved. Import containers arrive by vessel and leave the terminal via truck or train. These containers are first unloaded by a QC which puts them onto a YT (yard truck, internal tractor, prime mover, hustler, UTR) waiting under it on the ground. The YT then takes the container to a storage yard (container yard, yard) where a YC (yard crane, rubber tired gantry crane, RTGC, transtainer, TT) picks it off the YT's trailer and places it in a stack in the yard. At some later time, a YC retrieves the container from the yard and places it onto an XT (external truck) or train, which then takes the container to its final mainland destination.

Export containers arrive by truck or train and leave the terminal by vessel. They are packed off-site by exporters and delivered by XTs or trains to the container terminal. At the terminal these containers are removed from XTs or trains, and put into temporary storage in the yard by the YCs. When the vessel into which they are to be loaded docks at the terminal, these containers are retrieved from storage by YCs and carried by YTs to the berth; these YTs park under the appropriate QC which lifts the containers and loads them into the vessel.

Containers that are transshipped both arrive and depart by vessel. Upon arrival, each such container is unloaded by a QC, transported via YT from the quay to the yard, and then placed in a stack in the yard by a YC. When the vessel into which it will be loaded arrives at the terminal, the container is then retrieved by a YC, transported via YT from the yard to the quay, and then loaded by a QC into the vessel.

Fig. 2 shows the general layout of a container terminal from a bird's eye view. Vessels, QCs, YCs, and YTs are labeled for easy identification. In the layouts in common use today, the yard is divided into rectangular regions called blocks. The width of a block is typically divided into seven rows—six for stacks of containers and the seventh for trucks that interact with the YCs. Traffic lanes for trucks occupy the spaces between blocks. Blocks are divided along their length into 20 ft sections called slots. A typical block is about 40 slots long. The region occupied by a stack of 20 ft containers is called a groundslot (20 ft stack) and that occupied by a stack of 40 ft containers is a 40 ft stack. Forty foot stacks occupy two adjacent groundslots in the same row. In each stack, containers are stored one on top of the other 3–6 tiers high depending on the height of the YC serving the block.

YCs transfer containers between trucks (YTs or XTs) and the stacks in the yard. They straddle the entire width of the block beneath them and move along the length of the block. A zone is a sequence of blocks that together form a single lane for YC movement. In Fig. 2, blocks 1–3 are in zone 1; blocks 4–6 are in zone 2; and so on. YCs move easily within each block and from block to block within a zone; such movement is called linear gantrying. But to move from one zone to another, YCs have to spend at least 15 min making vertical turns in a maneuver called cross-gantrying, which is very time consuming. A cross-gantrying YC blocks the road to other vehicles for an extended period of time, thus disrupting traffic.

A QCs top speed is typically 40 lifts/h (container moves between vessel and shore) if it does not have to wait for YTs under it to take away the import containers it is unloading, or to bring export containers for it to load while it is loading. However, QCs at most terminals average only 25 or so lifts/hour. Each lift typically involves one 40 ft container, one 20 ft container, or two 20 ft containers. The average number of lifts achieved at a terminal per QC working hour is known as the GCR (gross crane rate, QC rate). GCR is perhaps the most important performance measure of a terminal. Another important measure by which terminals are judged is the average vessel turnaround time, which is highly correlated (with negative correlation) to the GCR. In this paper, we use GCR as the measure of performance to maximize. To attain a high GCR, the flow of containers back and forth between the shore and the yard has to proceed smoothly like clockwork, so that QCs do not incur idle time waiting for YTs.

Before we continue, it is important to mention that several alternative cargo handling systems are currently being used in container terminals around the world. In this paper, we assume a YT- and RTGC-based operation in which all equipment is manually controlled by human operators. Such a system is usually found in the land-scarce terminals in Asia and other crowded regions of the world. Other container handling systems may utilize one or more of the following types of equipment: rail-mounted gantry cranes (RMGCs); automated stacking cranes (ASCs); straddle carriers; top-handlers; reach-stackers; side-picks; automated guided vehicles (AGVs); and automated lifting vehicles (ALVs).

RMGCs are similar to, but bulkier than, RTGCs. Unlike RTGCs, they are mounted on rails and cannot make cross-gantry moves. Unlike RTGCs, they can gantry while carrying a container. Thus, RMGC-YT handover points may be located at the front and rear ends of a block, whereas RTGC-YT handover points are always located in a traffic lane along the side of a block. RMGCs may interfere with each other

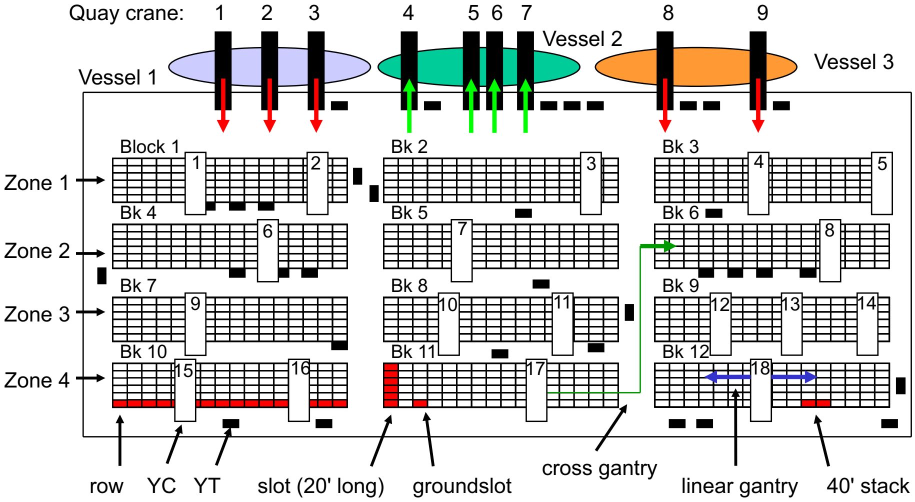

**Fig. 2.** Bird's eye view of a container terminal.

since they are usually of a different size. ASCs are automated versions of RMGCs. Straddle carriers are vehicles that combine the lifting and stacking capabilities of RTGCs and the traveling capability of YTs. They straddle blocks that are one row across and can carry single containers directly above stacks that are up to three containers high when moving through a block. Top-handlers are essentially huge forklifts that stack containers up to 4 high in the yard. They may only access the exterior rows of a block. Reach-stackers are bulkier than top-handlers and can access some of the interior rows of a block. Side-picks are lighter than top-handlers and are used for handling empty containers. AGVs are automated versions of YTs. They are typically used with ASCs. ALVs combine the traveling capability of AGVs with the lifting, but not stacking, capability of straddle carriers. Unlike AGVs, ALVs can pick up containers directly from the ground and transport them within the terminal without interacting with a crane.

Here we described the equipment and operations inside container terminals only briefly to help the reader understand our strategy described later. The remaining sections of the paper are organized as follows. Section 2 discusses the importance of maximizing GCR. In Section 3, we describe the problem in greater detail. Section 4 summarizes the relevant literature. Section 5 describes the simulation model used in the experiments. Section 6 describes the experiments, presents the results, and discusses their significance. Concluding remarks are made in Section 7.

## 2. What is the importance of maximizing GCR in container terminals?

For the terminal operator, two powerful incentives for maximizing GCR are (1) the economic incentive of higher business turnover using the same equipment and labor force and (2) the prestige and reputation for the terminal that comes with it. For the vessel operator (i.e. shipping line), a higher GCR means that vessels spend less time at port and more time at sea. With annual revenues for the largest container terminal operating companies at around $2 billion, and revenues for the largest liner shipping companies as much as 10 times this amount, the potential financial gains that could be achieved by improving GCR are enormous.

With the maritime container shipping industry playing an increasingly important role in the global economy, maximizing GCR is also important from a public policy perspective. Indeed, container terminals are notorious for being bottlenecks in the global supply chain, so even a 1% improvement in productivity at a major terminal could generate significant benefits for businesses around the world. Finally, maximizing GCR is smart from an environmental perspective. If existing terminals can operate at higher efficiencies, the need for building new terminals will be diminished. A higher vessel utilization will also diminish the need for constructing additional vessels. Thus, environmental concerns provide yet another powerful motivation for improving the GCR at existing terminals.

## 3. Problem description

In this paper, we are investigating how two parameters—(1) the system that deploys YCs among blocks in the same zone and (2) the length of the storage blocks in a terminal's container yard—affect the overall, long-run performance of a container terminal as measured in terms of GCR. To understand these relationships, we must discuss container yard operations in more detail.

### *3.1. YC deployment*

We first discuss the YC deployment system. To attain the highest GCR, it is vitally important that the activities of the YCs be properly coordinated so the YCs serve the QCs effectively. However, this is easier said than done. Firstly, the maximum handling capacity of a YC is roughly 25 lifts/h, much slower than a QCs 40 lifts/h. Secondly, YCs, unlike QCs, must multi-task. In particular, when more than one vessel is present, YCs have to store import containers being unloaded by QCs while also retrieving stored export containers to feed to other QCs that are loading vessels. For example, YC 6 in Fig. 2 has to serve not only YTs that are bringing containers from QCs 1–3 but also YTs that are retrieving containers to be loaded by QCs 4–7 into vessel 2. Thirdly, YCs move great distances while QCs are virtually immobile. The above factors mean that at least 2–3 YCs are typically needed per QC to keep the QCs smoothly working at a high speed. Fourthly, YCs have highly variable gantrying and container handling times. For example, according to a major container terminal operator, the time taken by a YC to handle a single container is a triangularly distributed random variable with parameters (1.2, 2.0, 3.4) min. Fig. 3 shows the probability density function of this distribution.

Finally, YCs that come too close to each other are subject to slowdowns. Fig. 4 illustrates how such a slowdown can develop. In the figure, Trucks 1 and 2 are attempting to transfer containers to/from stacks in the yard that lie beneath YCs 1 and 2, respectively. The trucks, each 55 ft long, are assumed to be mobile and the YCs are assumed to be stationary. If YCs 1 and 2 are less than 170 ft apart, Truck 2 does not have enough space to pull into the handling lane and become parallel to the storage block before reaching YC 2. Even if Truck 2 somehow manages to reach YC 2, Truck 1 does not have enough space to pull out of the handling lane and into the bypass lane before hitting the backside of Truck 2. In other words, the trucks have difficulty (A) gaining access to the downstream YC and (B) departing from the upstream YC. The overall result is a YT traffic jam that cuts YC productivity in half. Thus, two YCs in the same block must be separated by at least 170 ft (i.e. eight slots when accounting for spaces between stacks of containers) in order for two streams of trucks to be served independently by the YCs without delay.

The above paragraph highlights the challenging nature of real-time YC operations control problems at a container terminal. With these ideas in mind, we have considered in this paper two systems which govern the deployment of YCs among blocks in the same zone. Both systems are tested and shown to be viable in a real-time environment. Experimental results indicate that one of these systems is superior to the other.

### 3.2. Block length

Another factor affecting GCR is the design and layout of the storage yard. In this paper, we show how the sizes of the storage blocks (in particular the lengths of the blocks) can affect a terminal's long-run GCR, assuming that the total yard storage capacity and the number of YCs and YTs deployed remains unchanged. Block length affects GCR indirectly through the YTs and YCs. Block length affects the YTs as follows. When the blocks are longer (i.e. larger), there are fewer blocks and hence fewer spaces between the blocks for roadways, which means the likelihood of a YT finding a straight-line path from a random origin to a random destination within the terminal diminishes. This undesirable effect, however, may be compensated by the fact that the terminal as a whole is smaller; thus, the average distance as the crow flies between origin and destination diminishes.

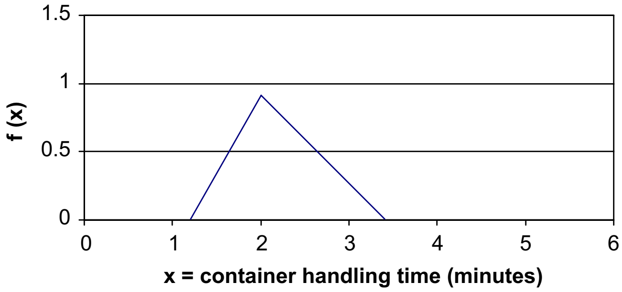

**Fig. 3.** The p.d.f. of YC handling time.

Block length presents a similar trade-off for the YCs. When the blocks get larger, there are fewer blocks in each zone, so the stacks in each zone are closer together on average. Thus, the average distance gantried by a YC between consecutive container handling operations decreases. This desirable effect, however, may be outweighed by an overall increase in YC interference (i.e. YC clashing), in which the work rates of adjacent YCs are reduced because they are working in close proximity in the same block (see Fig. 4). The purpose of the current paper is to show that by means of simulation experiments it is possible to identify the block length that strikes the optimal balance between these situations.

## 4. Literature review

The literature relevant to the current investigation includes all papers on container terminals that discuss (1) inter-block YC deployment, (2) terminal design, (3) simulation modeling, or (4) the literature itself. A total of 51 such papers were found by the authors.

Excellent surveys of recent research on container terminal operations have been done by Meersmans and Dekker [3], Stahlbock and Voss [4], Steenken et al. [5], and Vis and de Koster [6]. A good overview of container terminal operations is given by Günther and Kim [7]. A concise summary of the various operational decisions made in container terminals is given in Murty et al. [8]. Han et al. [9] is a very recent article that studies a storage yard management problem at a vessel-to-vessel transshipment terminal similar to that considered here.

Inter-block YC deployment involves two related problems: (1) determining how many YCs should be deployed in each block at each point in time and (2) deciding when and how YCs should linear- or cross-gantry from one block to another. Only four articles consider the problem of inter-block YC deployment. Cheung et al. [10], Linn et al. [11], Linn and Zhang [12], and Zhang et al. [13] develop methods for allocating YCs among yard blocks and for scheduling inter-block YC moves. The first, third and fourth articles present integer programming models of long-range YC deployment problems, but do not embed these models within a simulation model to test their performance in a real-time environment. The second article uses a simulation model to test the proposed algorithm on a set of 21 days' worth of work. However, YTs are virtually ignored by the assumption that their travel times are always 0.

Unlike the models presented in the literature, the YC deployment systems considered in this paper allow inter-block YC moves to be initiated at any time, not just at regular time intervals (e.g. every 4 h). In addition, although mathematically less sophisticated than the models in the literature, the two YC deployment systems presented here are directly tied into a detailed simulation model so that their

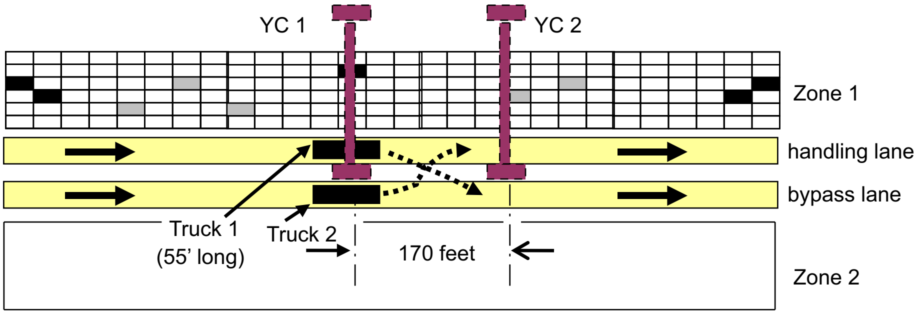

**Fig. 4.** Two YCs in the same block must be separated by at least 170 ft in order for two streams of (55-ft long) trucks to be independently served by the two YCs without delay.

performance in a real-time environment can be measured. Thus, this is the first article to show how different YC deployment systems and/or models affect the overall efficiency of a container terminal as measured by GCR or average vessel turnaround time.

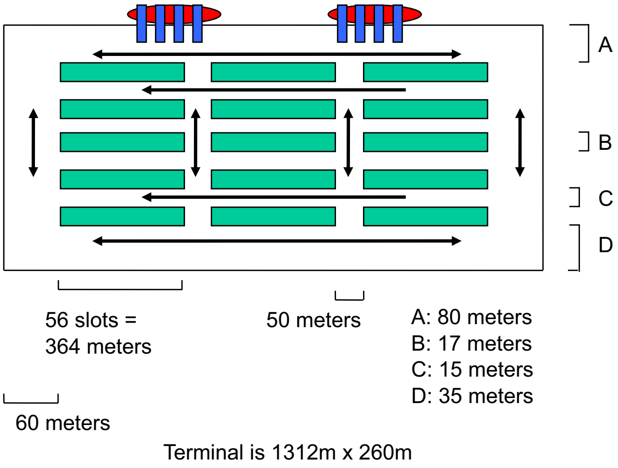

**Fig. 5.** Layout of the small container terminal when the block length is 56 slots.

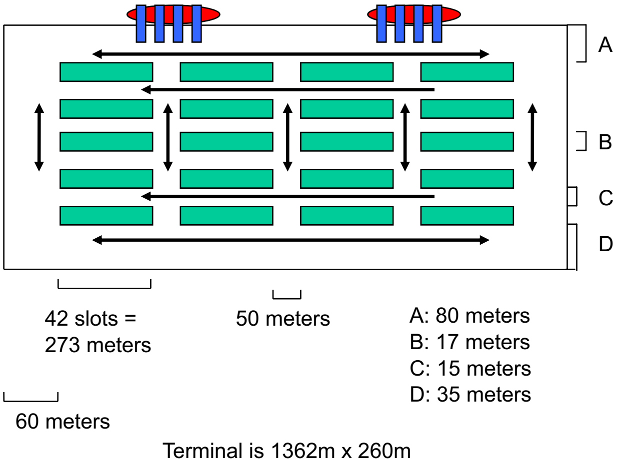

**Fig. 6.** Layout of the small container terminal when the block length is 42 slots.

Only seven articles that present numerical results on container terminal design could be found in the literature. The following five articles compare the performance of different types of handling equipment: Duinkerken et al. [14], Liu et al. [15], Nam and Ha [16], Vis [17], and Yang et al. [18]. Kim et al. [19] and Liu et al. [20] compare the performance of storage yard layouts in which blocks are parallel versus perpendicular to the berthed vessels. These papers use a static, equation-based approach and a simulation methodology, respectively. The former article finds the parallel layout to be superior, but the latter arrives at the opposite conclusion. The current paper assumes a parallel layout (see Figs. 2, 5–7). To the authors' knowledge, this is the first paper to specifically explore how block length affects the operations at a container terminal.

Thirty-eight container terminal simulation models were found in the literature: Alessandri et al. [21], Bielli et al. [22], Borovits and Ein-Dor [23], Bruzzone and Signorile [24], Canonaco et al. [25], Chung et al. [26], Dekker et al. [27], Demirci [28], Dragovic et al. [29], Duinkerken et al. [14], El Sheikh et al. [30], Froyland et al. [31], Gambardella et al. [32,33], Grunow et al. [34], Hartmann [35], Hayuth et al. [36], Kia et al. [37], Kozan [38], Lee et al. [39], Legato and Mazza [40], Linn et al. [11], Liu et al. [15,20], Merkuryev et al. [41], Nam et al. [42], Nevins et al. [43], Ottjes et al. [44], Parola and Sciomachen [45], Petering and Murty [46], Petering et al. [47], Pope et al. [48], Sgouridis et al. [49], Shabayek and Yeung [50], Silberholz et al. [51], Thiers and Janssens [52], Yang et al. [18], and Yun and Choi [53]. The emphasis of these studies ranges from strategic to operational aspects of container terminal management. The articles by Liu et al. [15,20] offer perhaps the most comprehensive models among those listed above. Their models are unique in that they (A) measure performance using a global indicator such as GCR or average vessel turnaround time; (B) track several other performance measures such as yard utilization, truck productivity, and YC productivity; and (C) show how different values for an input parameter (e.g. the type of handling equipment used, the vehicle fleet size, or the layout of the terminal) affect the overall performance of the terminal as measured by GCR or average vessel turnaround time. One limitation common to their models, and most other simulation models in the literature, is that they only consider a single-berth facility. Thus, all yard equipment is devoted to serving a single vessel at a given time. In addition, the duration of time that is simulated is only one day (i.e. the time taken for processing a single vessel). The simulation model presented below, on the other hand, considers the detailed operations at a multiple-berth facility over an extended time period that is user-defined (e.g. three weeks, six months).

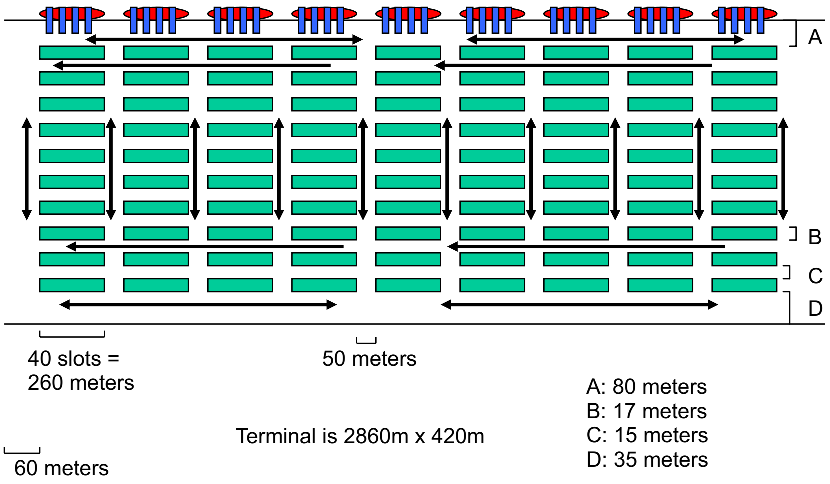

**Fig. 7.** Layout of the large container terminal when the block length is 40 slots.

Overall, a review of the literature has yielded many outstanding articles but no models that test how proposed terminal designs and/or *real-time* equipment control systems affect the overall performance (i.e. GCR) of a *multiple-berth* container terminal over *an extended period of time*. In the following section, we present a simulation model capable of performing such tests, and we show how it is being used to obtain new numerical results comparing alternate yard layouts and YC deployment systems for seaport container terminals.

## 5. Simulation model of a seaport container transshipment terminal

### 5.1. Introduction

We now present a discrete event simulation model of operations inside a seaport container transshipment terminal. The model, based on the authors' combined experience at container terminals on several continents and on extended discussions with managers and staff members at these terminals, is designed to show the dependence of GCR on the lengths of the blocks in the terminal's container yard and on the system that deploys YCs among the blocks in the same zone in the container yard in real time. The model is noteworthy in that (A) it allows direct comparisons to be made between alternate terminal designs and real-time yard control systems at a multiple-berth facility; (B) it measures performance according to the industry standard metric, GCR; (C) it considers all major entity types—vessels, QCs, YCs, and YTs; (D) all cranes and trucks have stochastic handling and traveling times; (E) delays are propagated realistically from one part of the system to another (e.g. from yard to quay); and (F) the arrival, stay, and departure of every individual container is explicitly modeled. The model simulates the activities associated with individual containers, vessels, QCs, YCs, YTs, and groundslots in the storage yard over an arbitrarily long, user-defined time period to a level of detail indicated by the list of events in Table 1. In short, it has been designed to reproduce the multi-objective, stochastic, real-time environment at a multiple-berth facility.

### 5.2. Events

Table 1 shows the different events that occur within the simulation model. The event times are continuous quantities, taking values in minutes along the positive real number line. Overall, there are two kinds of events: primary and secondary. Only primary events may appear in the future event list (FEL). A primary event generally (1) causes an immediate change in the state of the container terminal, (2) triggers other events, some primary and some secondary, that occur immediately along with the primary event, and/or (3) causes other primary events to be placed at some future time in the FEL. The event scheduling/time advance algorithm is used to ensure that all events take place in proper chronological order. Note that several events listed in Table 1 are calls to algorithms that make container storage, YC deployment, YC dispatching, and YT dispatching decisions in real time.

### 5.3. Main features and limitations of simulation model

A complete description of the simulation model is provided in Petering [54]. Less detailed descriptions are available in Petering and Murty [55] and Petering et al. [56]. The model has several important features. First of all, it has built-in systems for selecting storage locations for arriving containers, for routing and dispatching YCs in real time, and for dispatching YTs in real time. These systems are discussed in Petering and Murty [55], Petering et al. [56], and Petering [54], respectively.

The model accommodates two container sizes (20 and 40 ft). YTs are allowed to haul two 20 ft containers at the same time. In other words, YTs have dual-load capability. Although the total number of YTs is fixed throughout each simulation run, the number of active YTs varies depending on the number of busy berths. In particular, the number of active YTs is a fixed multiple of the number of busy berths at all times. Inactive YTs are only reactivated when a new vessel starts to dock at the terminal.

QCs handle one container at a time. Each QC handles containers according to a sequence with limited flexibility for changing the order in which lifts are performed. This means that during vessel loading, YTs arriving beneath a QC may not be served until the YTs assigned to the QC's earlier jobs have already passed the appropriate containers to the QC. Thus, a poorly sequenced arrival of YTs beneath a QC that is loading a vessel can significantly harm GCR as measured by the simulation model. This is particularly relevant for modeling the operations at large terminals where it is more likely for YTs to overtake one another along the way to the quay.

Another feature of the model is that the YC control system automatically prevents YC–YC interference. In particular, YCs are not allowed to pass each other within the same zone. Moreover, YCs are dispatched so they never are closer than 8 slots = 170 ft at any time (see Fig. 4). The inclusion of a YC–YC interference prevention system means that performance, as measured by the simulation model, actually deteriorates as additional YCs are added to a terminal where YCs are already plentiful. The model also assumes that there are bypass lanes, such as that depicted in Fig. 4, for trucks along the horizontal roadways between storage blocks. These bypass lanes allow

**Table 1**
Discrete events in the simulation model.

| Primary events | Secondary events |
|---|---|
| Vessel arrives | Vessel starts berthing |
| Vessel finishes berthing | Vessel starts un-berthing |
| Vessel finishes un-berthing | QC job sequences generated |
| All QC job sequences scanned | QC job sequences erased |
| QC finishes handling | QC starts handling |
| Multiple YC deployment algorithm called | General storage and retrieval algorithm called |
| YC finishes cross-gantry | Real-time container storage algorithm called |
| YC finishes linear gantry | YC cross-gantry scoring re-computed |
| YC finishes handling | Individual YC dispatching algorithm called |
| Workcenter YT assignment algorithm called | Individual on-the-fly YC dispatching algorithm called |
| YT finishes journey | YC starts cross-gantry |
| Check terminal status consistency | YC starts linear gantry |
| Start data collection | YC starts handling |
|  | Individual YT dispatching algorithm called |
|  | YT starts journey |

YTs to reach more than one YC in a block simultaneously without creating a traffic jam.

The traveling and handling times for all machines (QCs, YTs, YCs) follow probability distributions that are user-defined. The particular distributions used in the experiments are given in Section 6. Containers may only be passed from one machine to another if the machines satisfy a "handshake'' criterion. In other words, the transfer of individual containers between cranes and trucks is realistically and explicitly modeled. During vessel loading, for example, the event "QC starts handling'' in Table 1 is only triggered if one of the following conditions is satisfied: (1) the QC has just finished loading a container and the next container is ready to be loaded or (2) the YT bringing the next container in the sequence has just arrived beneath the QC. In other words, the event "QC starts handling'' can only be triggered by an occurrence of the event "QC finishes handling'' or "YT finishes journey,'' subject to many restrictions (see Table 1).

The model also tracks the groundslots where individual containers are stored when they are sitting in the storage yard. Containers are assigned storage locations immediately after being placed onto YTs during unloading. Tracking individual groundslots and storage locations is important for several reasons. First of all, these locations affect the journey times of YTs that are transporting cargo between the yard and quay. They also affect the times taken by YCs to gantry between jobs. Most importantly, they affect the performance of YCs that work in close proximity through the 170-ft separation requirement (see Fig. 4). Each container that is stored in the yard is retrieved from the same location before being loaded onto a vessel. Vessels may not leave the terminal unless all of their containers have been unloaded and loaded.

The model's main limitations are as follows. Firstly, only 20 and 40 ft standard dry containers are considered; 45-ft long, refrigerated, dangerous goods (DG), and other kinds of containers that make up about 15% of overall cargo volume are ignored. Secondly, YT congestion on the roads and at handling points is not explicitly modeled. Thus, the effect of YT–YT interference on operational performance is not measured. This means that, barring a probabilistic miracle, adding more YTs to a scenario *never* erodes the observed performance of the system. To the authors' knowledge, this shortcoming is found within all other container terminal simulation models in the literature. However, the modeling of YTs is still fairly detailed. For example, the expected YT travel time between two locations in the terminal depends on (A) the length of the shortest route along terminal roadways that adheres to traffic rules (e.g. one-way restrictions along the narrow avenues between storage blocks) that connects the origin and destination, (B) the number of turns involved in the route selected in part A, and (C) whether the YT is empty or laden.

The most important limitation is that the model only considers a pure-transshipment terminal. That is, all containers enter and leave the terminal by vessel. At least two of the world's busiest 20 container ports in 2006—Singapore and Tanjung Pelepas (Malaysia)—were primarily transshipment ports, with roughly 80% of cargo throughput being transferred from one vessel to another. In addition, at least two other ports in the top 10—Kaohsiung and Dubai—had a significant proportion ($>40\%$) of transshipment cargo.

The pure-transshipment assumption leads to several other assumptions. Firstly, there are no gates, no XTs, no rail yards, and no trains in the simulation model. Thus, every container sitting in the yard is destined to be loaded onto a vessel. In other words, the entire storage yard is an export yard. Containers in an export yard are typically stored according to a few important categories such as length, height, weight class, loading vessel, and destination port. Containers having identical status in all five categories are considered to be in the same group. Such containers are essentially substitutable with one another for vessel loading purposes. Indeed, if $C$ containers in the same group are scheduled for loading onto a particular vessel, the order in which these containers are loaded is irrelevant; any of the $C!$ permutations is equally desirable. Thus, containers in the same group are essentially substitutable once they are sitting in the yard, and the QC job sequences are not really sequences of individual containers, but sequences of containers from specified groups. In such an environment, it makes sense to require that all containers in a stack belong to the same group. In other words, the contents of each stack in the yard should be homogenous. Such a policy improves YC performance because it eliminates the need for YCs to shift containers directly from one stack to another stack in order to dig out containers at lower levels. Indeed, whenever a container belonging to a particular group needs to be retrieved from the yard, the homogenous stacking policy guarantees that there is always such a container on the top of a stack. Such a homogenous stacking policy is commonly adopted in export yards at terminals throughout the world, and we enforce such a policy in the simulation model. Thus, we also assume that YCs do not shift containers directly between stacks. These assumptions reflect actual practice in most export yards around the world, but they clearly do not account for the limited number of shifting moves that are inevitably required in a real setting due to special circumstances such as a last-minute request to change a container's loading vessel.

Although it considers a pure-transshipment facility, the model captures many important features of non-transshipment (i.e. import/export) terminals. Indeed, the vessel-to-vessel transshipment operations considered by the model are very similar to the export operations at non-transshipment terminals. This is because transshipment and export cargo both depart a terminal by vessel. Thus, the strategies used for storing such cargo in the yard, and moving such cargo from the yard to the quay for loading, are essentially the same.

### *5.4. Inputs and outputs*

As indicated earlier, the model has the flexibility to investigate a variety of terminal designs and algorithms for real-time yard control. The model has over 100 user-defined input parameters which remain constant for an entire simulation run. These parameters define the overall size and shape of the container terminal (including the number of berths, the number of stacks in the yard, and the layout of the yard); the number of QCs, YCs, and YTs deployed; the operating speed and variability of the QCs, YCs, and YTs; the general cargo profiles for the vessels that visit the terminal; the number of vessels visiting the terminal each week; the number of weeks to be simulated; and the settings for the algorithms that control equipment in real time.

The outputs from the model include the average GCR observed during the course of the simulation; the total number of QC lifts made; the average berth occupancy; the total number of YC lifts made; the average YC productivity (number YC lifts/hour of YC time); the total number of containers placed on YT trailers; the average YT productivity (number of containers placed on YT trailers per hour of active YT time); the total number of dual loads on YTs (i.e. the number of 20 ft containers that are placed on already half-full YT trailers); the average time that QCs, YCs, and YTs spend waiting for YTs, YTs, and (QCs or YCs), respectively; and the CPU runtime in minutes.

### *5.5. Vessel schedules and initialization*

In the simulation program, the terminal operates according to a regular weekly schedule. In other words, the terminal sees one vessel from a particular liner service at more-or-less the same time each week. The vessels visiting the terminal each week are assumed to be following the same routes as their respective counterparts from the previous week. Most major container terminals operate in this manner.

The terminal's weekly vessel schedule is generated at the beginning of each simulation run. This schedule consists of a home berth assignment and scheduled weekly arrival time for each liner service. Home berths are assigned so the liner services are evenly distributed among the berths. After assigning home berths, the model randomly generates each liner service's scheduled weekly arrival time, under the constraint that the vessels visiting each berth should be present during non-overlapping time intervals.

The actual arrival time of a vessel during the course of the simulation may deviate from the scheduled arrival time. If a vessel's home berth is available when it arrives, the vessel immediately begins berthing at its home berth. Otherwise, if some berth is available, the vessel immediately begins berthing at the berth closest to its home berth. If no berth is available, the vessel drops anchor in the harbor and joins the end of the queue of vessels that are waiting for a berth.

The container yard is empty at the beginning of each simulation run. During week 1, vessels bring in containers and the majority of container traffic flows from the quay to the yard. By the end of week 1, the yard becomes "filled up'' and equilibrium between the quay-to-yard and yard-to-quay flows is established; during weeks 2 and later, these two flows balance out. Data collection starts at the beginning of week 2.

## 6. Experiments, results, and discussion

### 6.1. Experimental setup: general

The simulation program was written and compiled using the Professional Edition of Microsoft Visual C++ 6.0. Experiments were run in the Windows XP environment using a 2.0 GHz Pentium 4 desktop computer with 512 MB of RAM.

In all simulation runs, data collection started at the beginning of week 2. The simulation terminated after three weeks worth of vessels were fully processed at the terminal. This instant may fall within week 4 or later if significant backlogging has occurred.

The experiments considered two different container terminals—a small terminal and a large terminal—and two different yard fleet size scenarios for each terminal—"less equipment'' and "more equipment''. Thus, a total of four different container terminal scenarios were considered. Table 2 gives the major specifications of these scenarios. Comprehensive experiments considered every possible combination of block length and YC deployment system for each of these four scenarios.

Seven different yard layouts were considered for the small terminal. These layouts correspond to block lengths of 168, 84, 56, 42, 28, 24, and 14 slots, respectively. Figs. 5 and 6 show the physical layout of the small terminal when the block length is 56 and 42 slots, respectively. The latter layout has one additional vertical roadway and therefore gives rise to a larger overall terminal area. All layouts for the small terminal consist of 5 yard zones and 168 slots of yard capacity per zone, where the blocks in each zone are 6 container rows wide. Note in Table 2 that 20 and 25 YCs are deployed at the small terminal with less and more equipment, respectively. Twenty and 25 are both divisible by 5 (the number of zones), so there are an equal number of YCs (4 and 5) operating in each zone in the less and more equipment scenarios, respectively.

Twelve different yard layouts were considered for the large terminal. These layouts correspond to block lengths of 360, 180, 120, 90, 72, 60, 40, 36, 30, 24, 20, and 18 slots, respectively. The layouts with longer blocks have fewer blocks and hence fewer vertical roadways. Thus, these layouts give rise to a smaller overall terminal area. The physical layout of the large terminal when the block length is 40 slots is depicted in Fig. 7. All layouts for the large terminal consist of 10 yard zones and 360 slots of yard capacity per zone, where each block has six rows of containers. Note in Table 2 that 90 and 110 YCs are deployed at the large terminal with less and more equipment, respectively. Ninety and 110 are both divisible by 10 (the number of zones), so there are an equal number of YCs (9 and 11) operating in each zone in the less and more equipment scenarios, respectively.

As Figs. 5–7 indicate, YTs may travel in both directions on all vertical lanes and on the two horizontal lanes at the top and bottom of the terminal, but may only travel one way down the horizontal lanes in the middle of the terminal.

In all experiments, GCR is measured as follows:

$$\text{GCR} = (\text{total number of QC lifts})/(\text{total number of QC hours beside a busy berth}).$$

A busy berth is defined as a berth with a vessel alongside such that at least one QC is transferring containers between the vessel and the shore. Since the denominator is directly proportional to the average vessel turnaround time, GCR is inversely proportional to the average vessel turnaround time.

### 6.2. Experimental setup: QC, YT, and container storage algorithm settings

The default values of the model's input parameters were chosen based on information received from a major container terminal operator. In all experiments, the time taken by a QC to handle a single container is a triangularly distributed random variable with parameters (1.0, 1.5, 2.0) min. This distribution has a mean of 1.5 min. That is, we assume that the long-run technical work rate of each QC, if it

**Table 2**
Fully dynamic container terminal scenarios considered in the experiments.

| | Small terminal less equipment | Small terminal more equipment | Large terminal less equipment | Large terminal more equipment |
|---|---|---|---|---|
| Vessel calls per week | 10 | 10 | 90 | 90 |
| Expected QC lifts per vessel | 3600 | 3600 | 1920 | 1920 |
| Expected QC lifts per week | 36,000 | 36,000 | 172,800 | 172,800 |
| Berths | 2 | 2 | 9 | 9 |
| QCs | 8 | 8 | 36 | 36 |
| Groundslots in yard | 5040 | 5040 | 21,600 | 21,600 |
| Yard zones | 5 | 5 | 10 | 10 |
| Max. cont. stacking height in yard | 6 | 6 | 5 | 5 |
| Ratio of 20 ft to 40 ft conts. | 2:1 | 2:1 | 3:2 | 3:2 |
| YCs | 20 | 25 | 90 | 110 |
| YCs per zone | 4 | 5 | 9 | 11 |
| YTs | 40 | 72 | 180 | 324 |
| Active YTs per busy berth | 20 | 36 | 20 | 36 |

is never starved of YTs, is 40 lifts/h. This means that the maximum GCR possible for any experiment is 40 lifts/h. The variance in QC handling time is due to (1) the trial-and-error method by which QC operators first latch onto containers that are to be hoisted, (2) the varying skill levels of the YT operators with whom the QCs interact, (3) the clashing or conformity of the individual styles of the QC and YT operators involved in an individual transaction, (4) the different vessel cell locations to/from which containers are transferred (some are close to the shore/ground, others far), and (5) the possibility that a container may have to be hoisted higher than normal in order to avoid hitting other containers already stacked on deck.

In all experiments, YTs travel 40 km/h on average when empty and 25 km/h on average when carrying one or more containers. They spend on average 10 s making a turn. Their actual travel time for a given journey may range anywhere from 30% below to 30% above the expected travel time. Preliminary experiments, many of which are described in Petering [54], led us to identify one YT dispatching system, out of several alternatives, for use in the experiments in this paper. The YT dispatching system that we adopted performed well in a variety of situations. It combines all active YTs into a single pool and allows YTs to carry two, 20-ft containers simultaneously only if they have the same pick-up location (QC or stack in the yard). Whenever a YT becomes free, the system identifies the "most starved QC" and assigns to the YT the earliest container(s) in that QC's job sequence that have not already been assigned to another YT. In other words, the YT serves the "most idle QC" first. Two containers are assigned to the YT if the two earliest container(s) identified above are both 20 ft long and have the same pick-up location.

Containers are assigned storage locations in real time immediately after being placed onto YTs during unloading. The experiments described in Petering and Murty [55] allowed us to identify one container storage location assignment system, out of several alternatives, for use in the experiments in this paper. The system chosen for this paper performed well in a variety of situations. Overall, the system tries to disperse both the containers unloaded by the same vessel and the containers loaded onto the same vessel throughout the storage yard to make sure that several YCs are supporting each working QC during vessel unloading and loading. This helps to minimize the QC idle time during both these operations. The dispersion of containers also helps to equalize each YC's workload at any given point in time.

The container storage location assignment system works as follows. Immediately after a QC finishes placing a container onto a YT, the system scans the entire storage yard to see if there is a stack in the yard with available capacity that already stores containers belonging to the same group to which the container belongs. If one or more such stacks are found, the system assigns the container to one of these stacks. Otherwise, the system assigns the container to an empty stack in the yard that is in the block with the lowest combined (1) number of YTs heading to or already waiting at the block and (2) number of retrievals of containers in the block that are expected to coincide with (i.e. clash with) the retrieval of the current container if the container were placed in the block. The overall decision is made by weighting these two measures. Put another way, the container storage location assignment system tries to strike the ideal balance between (1) minimizing the delay/congestion associated with placing the container into the yard and (2) minimizing the delay/congestion associated with retrieving the container from the yard when the time comes to retrieve it.

### 6.3. *Experimental setup: two YC deployment systems*

In all experiments, YCs gantry at an average rate of one slot every 4 s. Their actual gantry time between two slots may range anywhere from 30% below to 30% above the expected gantry time. The time taken by a YC to handle a single container is a triangularly distributed random variable with parameters $(1.2, 2.0, 3.4)$ min. YCs are not allowed to make inter-zone cross-gantry moves like that indicated by the arrow to the right of YC 17 in Fig. 2.

Two systems for deploying YCs among blocks in the same zone were considered in the experiments. The first system restricts the movement of YCs by setting a minimum and maximum number of YCs that must be present in each block at all times. These values are equal to (number of YCs per zone/number of blocks per zone) rounded down and up to the nearest integer, respectively. For example, for the small terminal with more equipment with two blocks per zone, the minimum and maximum number of YCs allowed in a block at the same time are $\lfloor \frac{5}{2} \rfloor = 2$ and $\lceil \frac{5}{2} \rceil = 3$, respectively. (Five is the number of YCs per zone in this scenario.) This system is called a "restrictive" system because it restricts the freedom of YCs to move between blocks in the same zone. The second YC deployment system allows YCs to move freely about the various blocks within a zone with no limit on the minimum or maximum number of YCs that can be present in a block at the same time. This system is referred to as the "free" YC deployment system. Note that the restrictive and free deployment systems are identical when there is only one block per zone.

Both YC deployment systems use the same real-time YC dispatching algorithm. The experiments described in Petering et al. [56] allowed us to identify one such YC dispatching algorithm, out of several alternatives, for use in the experiments in this paper. This algorithm, which performed well in a variety of situations, works as follows. Whenever a YC becomes free, the dispatching algorithm first identifies the set of all jobs (i.e. container moves) that have been assigned to YTs that are already waiting near the YC or are soon expected to appear near the YC. Among these jobs, the algorithm assigns to the YC the *retrieval* job that is most urgent from the point of view of the QC loading sequences. If there is no retrieval job to be done, the algorithm assigns to the YC the *storage* job nearest to the YC.

### 6.4. *Small terminal results*

Our experiments for the small terminal considered two yard fleet size scenarios, seven yard layouts (block lengths), and two different YC deployment systems. Every possible combination of these three parameters was considered. Since there is only one YC deployment system for one of the yard layouts, a total of $2 \times (7 \times 2 - 1) = 26$ different setups were investigated.

The two different yard fleet sizes are as follows. The scenario with "less equipment" has only 20 YCs and 40 YTs, while the scenario with "more equipment" has 25 YCs and 72 YTs (Table 2).

Each of the seven yard layouts gives rise to an overall yard capacity of 5040 groundslots. These layouts divide each zone into (1) one 168-slot block, (2) two 84-slot blocks, (3) three 56-slot blocks, (4) four 42-slot blocks, (5) six 28-slot blocks, (6) seven 24-slot blocks, and (7) twelve 14-slot blocks, respectively. The number of interior vertical traffic lanes for YTs in these layouts is 0, 1, 2, 3, 5, 6, and 11, respectively. Thus, the overall terminal area is enlarged as we proceed from the beginning to the end of the list.

Table 3 shows the experimental results for the small terminal with less equipment. Each row in the table corresponds to a different combination of block length and YC deployment system. Each row of data was obtained by averaging the results from six independent simulation runs (i.e. replications) of 3 weeks each (with data collection beginning at the start of week 2), and rounding the average to the nearest integer or hundredth. Thus, the table includes data from a total of $13 \times 6 = 78$ independent experiments. Each experiment was completed in less than 2 min of CPU time.

Sixteen different performance measures were tracked in each experiment. These measures include the GCR, average YC productivity,

**Table 3**
Experimental results for the small terminal with less equipment.

| Blk length | YC deploy | GCR | bthOcc | YCProd | YTProd | %vol | %area | QCLifts | waitU | waitL | YCLifts | waitS | waitR | gant | YTHauls | dualLd | trv/cnt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 168 slots | | 28.14 | 0.95 | 10.37 | 5.46 | 46 | 88 | 77,688 | 456 | 5357 | 77,735 | 2659 | 101 | 3137 | 77,713 | 23,810 | 4.80 |
| 84 slots | Restrictive | 31.07 | 0.87 | 10.47 | 5.99 | 45 | 87 | 75,847 | 245 | 3532 | 75,852 | 1435 | 18 | 2818 | 75,845 | 23,203 | 3.35 |
| | Free | 30.23 | 0.90 | 10.54 | 5.84 | 45 | 87 | 77,676 | 367 | 4007 | 77,725 | 1387 | 18 | 3405 | 77,699 | 23,843 | 3.35 |
| 56 slots | Restrictive | 31.20 | 0.87 | 10.47 | 6.01 | 44 | 86 | 74,504 | 291 | 3325 | 74,487 | 1051 | 7 | 2904 | 74,487 | 22,840 | 2.95 |
| | Free | 30.32 | 0.89 | 10.50 | 5.86 | 45 | 88 | 76,713 | 150 | 4121 | 76,735 | 1077 | 9 | 3709 | 76,725 | 23,499 | 2.96 |
| 42 slots | Restrictive | 31.02 | 0.88 | 10.50 | 5.98 | 44 | 86 | 76,793 | 193 | 3684 | 76,797 | 1126 | 6 | 2420 | 76,793 | 23,536 | 2.80 |
| | Free | 29.94 | 0.91 | 10.56 | 5.79 | 44 | 86 | 77,418 | 496 | 4055 | 77,430 | 887 | 6 | 3864 | 77,421 | 23,701 | 2.80 |
| 28 slots | Restrictive | 30.22 | 0.90 | 10.51 | 5.84 | 44 | 86 | 76,249 | 413 | 3902 | 76,273 | 879 | 5 | 3457 | 76,261 | 23,329 | 2.69 |
| | Free | 29.34 | 0.93 | 10.53 | 5.68 | 45 | 87 | 77,884 | 395 | 4575 | 77,912 | 752 | 5 | 4450 | 77,895 | 23,893 | 2.69 |
| 24 slots | Restrictive | 29.58 | 0.92 | 10.50 | 5.72 | 45 | 87 | 75,798 | 394 | 4263 | 75,796 | 803 | 6 | 3866 | 75,800 | 23,255 | 2.68 |
| | Free | 28.91 | 0.94 | 10.54 | 5.60 | 45 | 87 | 77,582 | 693 | 4546 | 77,616 | 694 | 5 | 4632 | 77,599 | 23,800 | 2.67 |
| 14 slots | Restrictive | 26.92 | 0.97 | 10.10 | 5.23 | 46 | 89 | 77,462 | 488 | 6198 | 77,503 | 645 | 7 | 5733 | 77,480 | 23,691 | 2.78 |
| | Free | 26.55 | 0.98 | 10.07 | 5.16 | 46 | 89 | 77,049 | 333 | 6622 | 77,091 | 622 | 7 | 6035 | 77,073 | 23,603 | 2.79 |

GCR, gross crane rate (total QC lifts/total hours of QC time beside a busy berth). bthOcc, berth occupancy (average fraction of berths occupied at any instant). YCProd, average yard crane productivity (total YC lifts/total hours of YC time). YTProd, average yard truck productivity (total number of containers placed on YT trailers/total hours of active YT time). %vol, average percentage of storage volume occupied at any instant. %area, average percentage of storage area occupied at any instant. QCLifts, total number of QC lifts made. waitU, average amount of time each QC spends waiting for YTs to appear during unloading (in minutes, summed up over an entire simulation run). waitL, average amount of time each QC spends waiting for YTs to appear during loading (in minutes, summed up over an entire simulation run). YCLifts, total number of YC lifts made. waitS, average amount of time each YC spends waiting for YTs to bring conts, to be stored (in minutes, summed up over an entire simulation run). waitR, average amount of time each YC spends waiting for YTs with retrieval requests to appear (in minutes, summed up over an entire simulation run). gant, average amount of time each YC spends linear gantrying (in minutes, summed up over an entire simulation run). YTHauls, total number of containers placed on YT trailers. dualLd, total number of 20 ft containers placed on YT trailers that are already half full. trv/cnt, average amount of time each YT spends traveling per container hauled (i.e. per single container drop-off + pick-up operation) (in minutes).

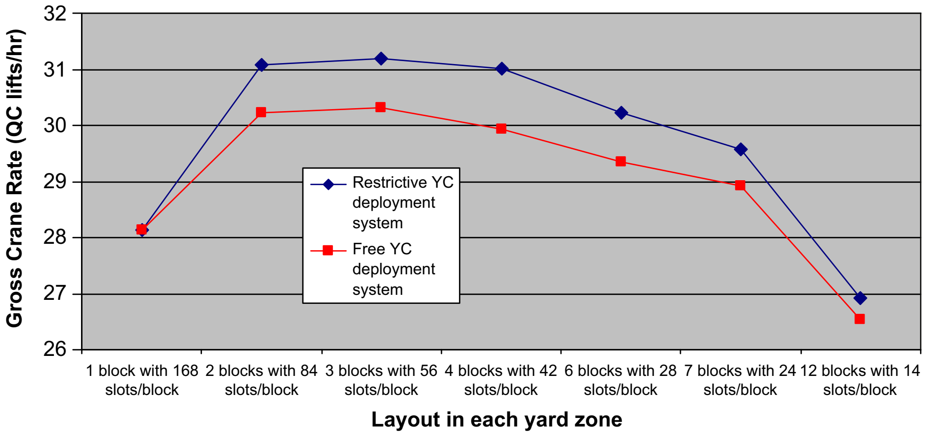

**Fig. 8.** Performance of seven different block lengths and two different YC deployment systems at the small terminal with less equipment.

average YT productivity, average percentage of the yard storage volume occupied at any instant, average percentage of the yard storage area occupied at any instant, and several other key performance indicators. A key to these performance measures is provided beneath Table 3.

Fig. 8 graphically displays how the GCR, the most important performance measure, is affected by the yard layout and YC deployment system. This figure shows that the restrictive YC deployment system outperforms the free YC deployment system for all possible block lengths, and by about 3% (i.e. 1 QC lift/hour) on average. Also, GCR is essentially concave with respect to the block length, attaining its maximum value when the block length is 56 slots for both YC deployment algorithms.

Table 3 offers some explanations for these observations. First of all, note that the numbers in the "waitL" column are much larger than those in the "waitU" column. This indicates that most QC delays occur during loading, not unloading. In addition, the "waitL" column reveals that the QCs are spending significantly less (more) time waiting for YTs to appear beneath them during loading when the restrictive (free) YC deployment system is used. These observations directly account for the superiority of the restrictive YC deployment system. Indeed, note the high negative correlation between GCR and the terms in the "waitL" column, especially for the experiments involving the restrictive YC deployment system. In particular, GCR is the highest (at 31.2 lifts/h) precisely when the QC waiting time during loading is the lowest (3325 min over a 2-week data collection period). The superiority of the restrictive YC deployment system probably originates from the reduced amount of YC gantrying associated with this system, as shown in the "gant" column, compared with the free YC deployment system. Indeed, by restricting the mobility of the YCs, the restrictive system is actually improving performance by forcing the YCs to spend their time handling nearby containers and not going on "wild goose chases." Finally, note in the "trv/cnt" column that the average YT travel time per container hauled

**Table 4**
Experimental results for the small terminal with more equipment.

| Blk length | YC deploy | GCR | bthOcc | YCProd | YTProd | %vol | %area | QCLifts | waitU | waitL | YCLifts | waitS | waitR | gant | YTHauls | dualLd | trv/cnt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 168 slots | | 36.48 | 0.76 | 8.48 | 3.88 | 43 | 85 | 75,943 | 21 | 1060 | 75,960 | 2375 | 61 | 1976 | 75,948 | 23,251 | 4.84 |
| 84 slots | Restrictive | 37.59 | 0.74 | 8.49 | 3.99 | 43 | 85 | 76,250 | 0 | 630 | 76,228 | 1469 | 12 | 2168 | 76,228 | 23,363 | 3.36 |
| | Free | 37.22 | 0.74 | 8.46 | 3.95 | 43 | 85 | 76,197 | 0 | 796 | 76,228 | 1442 | 11 | 2321 | 76,219 | 23,346 | 3.37 |
| 56 slots | Restrictive | 37.72 | 0.72 | 8.37 | 4.00 | 43 | 84 | 74,443 | 0 | 577 | 74,397 | 1177 | 6 | 2125 | 74,398 | 22,785 | 2.97 |
| | Free | 37.17 | 0.75 | 8.57 | 3.96 | 43 | 84 | 77,499 | 0 | 833 | 77,553 | 1111 | 6 | 2487 | 77,531 | 23,749 | 2.96 |
| 42 slots | Restrictive | 37.23 | 0.75 | 8.54 | 3.95 | 43 | 85 | 76,084 | 0 | 785 | 76,067 | 1100 | 4 | 2025 | 76,064 | 23,334 | 2.80 |
| | Free | 37.05 | 0.74 | 8.47 | 3.94 | 43 | 84 | 74,421 | 0 | 814 | 74,426 | 922 | 3 | 2557 | 74,429 | 22,815 | 2.81 |
| 28 slots | Restrictive | 37.20 | 0.74 | 8.47 | 3.95 | 43 | 85 | 75,168 | 0 | 797 | 75,159 | 1018 | 4 | 2112 | 75,157 | 23,044 | 2.70 |
| | Free | 36.87 | 0.75 | 8.52 | 3.92 | 43 | 85 | 75,381 | 0 | 917 | 75,420 | 827 | 3 | 2986 | 75,413 | 23,111 | 2.70 |
| 24 slots | Restrictive | 37.11 | 0.75 | 8.57 | 3.95 | 43 | 85 | 76,554 | 0 | 860 | 76,566 | 941 | 4 | 2443 | 76,555 | 23,473 | 2.69 |
| | Free | 36.81 | 0.75 | 8.52 | 3.91 | 43 | 85 | 76,792 | 0 | 964 | 76,815 | 797 | 3 | 3179 | 76,811 | 23,595 | 2.69 |
| 14 slots | Restrictive | 36.10 | 0.76 | 8.47 | 3.84 | 43 | 85 | 76,573 | 0 | 1282 | 76,566 | 709 | 4 | 3665 | 76,561 | 23,466 | 2.79 |
| | Free | 35.71 | 0.77 | 8.49 | 3.80 | 43 | 84 | 74,371 | 2 | 1384 | 74,357 | 622 | 4 | 3756 | 74,360 | 22,767 | 2.78 |

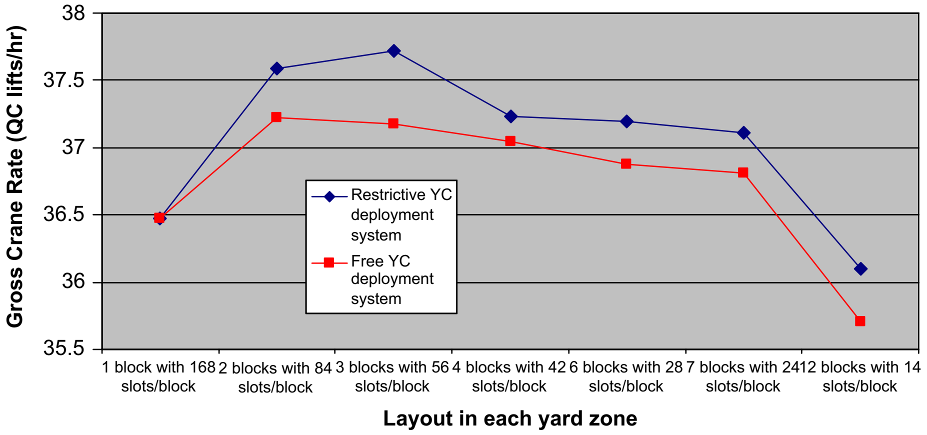

**Fig. 9.** Performance of seven different block lengths and two different YC deployment systems at the small terminal with more equipment.

is essentially convex with respect to the block length. This convexity confirms our earlier speculations in Section 3 that a changing block length presents competing trade-offs for the YT fleet.

Table 4 shows the experimental results for the small terminal with more equipment. Again, each row of data was obtained by averaging the results from six independent simulation runs and rounding the average to the nearest integer or hundredth. Thus, the table includes data from a total of $13 \times 6 = 78$ experiments. Each individual experiment was completed in less than 1 min of CPU time.

Sixteen different performance measures were tracked in each experiment. Fig. 9 graphically displays how the GCR, the most important performance measure, is affected by the yard layout and YC deployment system. This figure shows once again that the restrictive YC deployment system outperforms the free YC deployment system for all possible block lengths, this time by about 1% on average. Also, GCR is almost concave with respect to the block length, attaining its maximum value when the block length is 56 slots for the restrictive YC deployment system and when the block length is 84 slots for the free YC deployment system. Overall, the GCR with more equipment is roughly 25% higher than the GCR with less equipment.

Table 4 contains some of the same trends that were in Table 3. In particular, most QC delays occur during loading, not unloading. This is not surprising given the fact that YTs bringing containers to the quay during loading are typically not substitutable, whereas YTs bringing their empty trailers to the quay during unloading are substitutable. Once again, the "waitL" column reveals that the QCs are spending much less time waiting for YTs to appear beneath them during loading when the restrictive YC deployment system is used. This directly accounts for the superiority of the restrictive YC deployment system. Indeed, note once again the high negative correlation between GCR and the terms in the "waitL" column, especially for the experiments involving the restrictive YC deployment system. Once again, the superiority of the restrictive YC deployment system apparently originates from the reduced amount of YC gantrying associated with this system as shown in the "gant" column. Finally, note again in the "trv/cnt" column that the average YT travel time per container hauled is essentially convex with respect to the block length. Not surprisingly, the values in this column are virtually identical to their counterparts in Table 3.

### *6.5. Large terminal results*

Our experiments for the large terminal considered two yard fleet size scenarios, 12 yard layouts, and the same two YC deployment systems investigated for the small terminal. Every possible combination of these three parameters was considered. Since there is only one YC deployment system for one of the yard layouts, a total of $2 \times (12 \times 2 - 1) = 46$ different setups were investigated.

The two different yard fleet sizes are as follows. The scenario with "less equipment" has only 90 YCs and 180 YTs, while the scenario with "more equipment" has 110 YCs and 324 YTs (Table 2).

**Table 5**
Experimental results for the large terminal with less equipment.

| Blk length | YC deploy | GCR | bthOcc | YCProd | YTProd | %vol | %area | QCLifts | waitU | waitL | YCLifts | waitS | waitR | gant | YTHauls | dualLd | trv/cnt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 360 slots | | 20.64 | 1.00 | 8.03 | 4.01 | 60 | 89 | 376,966 | 1080 | 12,343 | 376,987 | 7508 | 1289 | 4025 | 376,976 | 90,407 | 9.35 |
| 180 slots | Restrictive | 26.06 | 1.00 | 10.02 | 5.03 | 60 | 89 | 364,248 | 228 | 6991 | 364,285 | 3941 | 438 | 3781 | 364,261 | 87,233 | 6.22 |
| | Free | 25.36 | 1.00 | 9.78 | 4.90 | 60 | 89 | 366,401 | 241 | 7643 | 366,489 | 3976 | 442 | 4030 | 366,436 | 87,773 | 6.21 |
| 120 slots | Restrictive | 27.87 | 0.99 | 10.60 | 5.37 | 59 | 88 | 363,044 | 110 | 5662 | 363,106 | 3201 | 247 | 3452 | 363,070 | 86,990 | 5.30 |
| | Free | 26.72 | 0.99 | 10.20 | 5.15 | 59 | 88 | 362,944 | 147 | 6506 | 362,908 | 3095 | 247 | 4116 | 362,929 | 86,980 | 5.30 |
| 90 slots | Restrictive | 28.71 | 0.98 | 10.83 | 5.52 | 59 | 88 | 362,535 | 59 | 5077 | 362,507 | 2910 | 177 | 3492 | 362,526 | 86,839 | 4.88 |
| | Free | 27.09 | 0.99 | 10.34 | 5.22 | 60 | 89 | 363,023 | 96 | 6242 | 363,031 | 2742 | 177 | 4295 | 363,025 | 87,000 | 4.88 |
| 72 slots | Restrictive | 29.06 | 0.98 | 10.95 | 5.59 | 59 | 88 | 363,373 | 53 | 4916 | 363,369 | 2760 | 141 | 3469 | 363,367 | 86,960 | 4.67 |
| | Free | 27.22 | 0.99 | 10.37 | 5.25 | 59 | 88 | 365,454 | 89 | 6220 | 365,628 | 2538 | 141 | 4455 | 365,521 | 87,468 | 4.66 |
| 60 slots | Restrictive | 28.97 | 0.98 | 10.91 | 5.57 | 59 | 88 | 362,215 | 33 | 4950 | 362,245 | 2677 | 125 | 3521 | 362,224 | 86,772 | 4.53 |
| | Free | 27.18 | 0.99 | 10.35 | 5.24 | 59 | 88 | 363,267 | 88 | 6161 | 363,273 | 2393 | 124 | 4571 | 363,264 | 86,967 | 4.55 |
| 40 slots | Restrictive | 28.63 | 0.98 | 10.83 | 5.51 | 59 | 88 | 362,606 | 20 | 5200 | 362,604 | 2871 | 109 | 2717 | 362,605 | 86,880 | 4.40 |
| | Free | 26.65 | 0.99 | 10.21 | 5.14 | 59 | 88 | 364,682 | 121 | 6593 | 364,688 | 2190 | 102 | 5048 | 364,685 | 87,361 | 4.41 |
| 36 slots | Restrictive | 28.49 | 0.98 | 10.76 | 5.48 | 59 | 88 | 361,104 | 35 | 5256 | 361,089 | 2709 | 103 | 3116 | 361,097 | 86,472 | 4.39 |
| | Free | 26.48 | 0.99 | 10.13 | 5.11 | 60 | 89 | 363,764 | 91 | 6749 | 363,812 | 2158 | 97 | 5235 | 363,783 | 87,141 | 4.38 |
| 30 slots | Restrictive | 27.79 | 0.99 | 10.55 | 5.35 | 59 | 88 | 363,371 | 79 | 5759 | 363,444 | 2495 | 105 | 3810 | 363,395 | 87,050 | 4.43 |
| | Free | 25.92 | 0.99 | 9.95 | 5.00 | 60 | 89 | 364,423 | 96 | 7198 | 364,433 | 2103 | 97 | 5593 | 364,419 | 87,233 | 4.42 |
| 24 slots | Restrictive | 26.71 | 0.99 | 10.20 | 5.15 | 60 | 89 | 363,908 | 74 | 6597 | 363,915 | 2342 | 103 | 4733 | 363,905 | 87,147 | 4.45 |
| | Free | 25.25 | 1.00 | 9.74 | 4.88 | 60 | 89 | 365,937 | 151 | 7770 | 366,033 | 2050 | 96 | 6038 | 365,976 | 87,578 | 4.46 |
| 20 slots | Restrictive | 25.59 | 1.00 | 9.85 | 4.94 | 60 | 89 | 365,043 | 135 | 7491 | 365,059 | 2241 | 106 | 5533 | 365,041 | 87,579 | 4.55 |
| | Free | 24.52 | 1.00 | 9.47 | 4.74 | 60 | 89 | 366,551 | 161 | 8452 | 366,570 | 2045 | 101 | 6595 | 366,551 | 87,838 | 4.56 |
| 18slots | Restrictive | 24.95 | 1.00 | 9.63 | 4.82 | 60 | 89 | 366,137 | 206 | 8019 | 366,135 | 2189 | 107 | 6015 | 366,132 | 87,683 | 4.63 |
| | Free | 24.07 | 1.00 | 9.31 | 4.66 | 60 | 89 | 364,996 | 232 | 8786 | 364,985 | 2020 | 106 | 6849 | 364,990 | 87,391 | 4.63 |

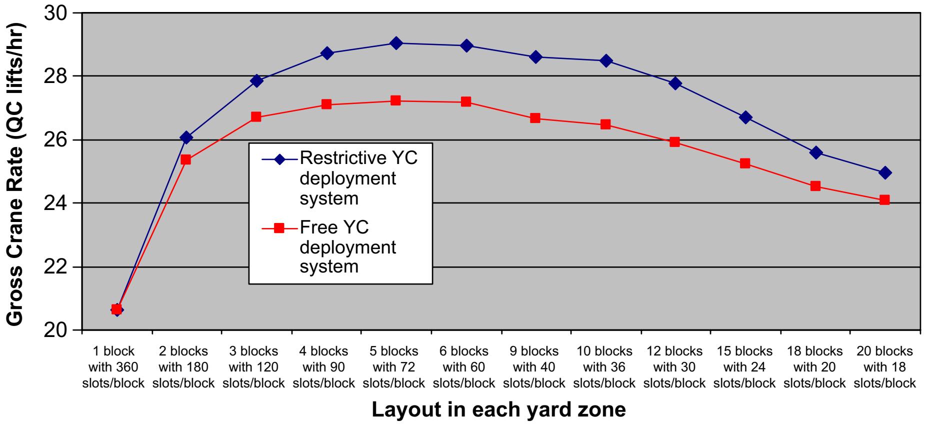

**Fig. 10.** Performance of 12 different block lengths and two different YC deployment systems at the large terminal with less equipment.

Each of the 12 different yard layouts gives rise to an overall yard capacity of 21,600 groundslots. These layouts divide each zone into (1) one 360-slot block, (2) two 180-slot blocks, (3) three 120-slot blocks, (4) four 90-slot blocks, (5) five 72-slot blocks, (6) six 60-slot blocks, (7) nine 40-slot blocks, (8) ten 36-slot blocks, (9) twelve 30-slot blocks, (10) fifteen 24-slot blocks, (11) eighteen 20-slot blocks, and (12) twenty 18-slot blocks, respectively. The number of interior vertical roadways in these layouts is 0, 1, 2, 3, 4, 5, 8, 9, 11, 14, 17, and 19, respectively. Thus, the overall terminal area is enlarged as we proceed from the beginning to the end of the list.

Table 5 shows the experimental results for the large terminal with less equipment. Each row in the table corresponds to a different combination of block length and YC deployment system. Each row of data was obtained by averaging the results from six independent simulation runs of 3 weeks each (with data collection beginning at the start of week 2), and rounding the average to the nearest integer or hundredth. Thus, the table is based on the results from a total of $23 \times 6 = 138$ experiments. Each experiment was completed in less than 30 min of CPU time.

Sixteen different performance measures were tracked in each experiment. Fig. 10 graphically displays how the most important performance measure—GCR—is affected by the yard layout and YC deployment system. This figure shows that the restrictive YC deployment system once again outperforms the free YC deployment system for all possible block lengths, this time by about 5% on average. Also, GCR is essentially concave with respect to the block length, attaining its maximum value when the block length is 72 slots for both YC deployment algorithms. Statistical tests indicate that the performance difference between this block length (GCR = 29.06) and the second-best block length (60 slots with GCR = 28.97) when the

**Table 6**
Experimental results for the large terminal with more equipment.

| Blk length | YC deploy | GCR | bthOcc | YCProd | YTProd | %vol | %area | QCLifts | waitU | waitL | YCLifts | waitS | waitR | gant | YTHauls | dualLd | trv/cnt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 360 slots | | 29.96 | 0.97 | 9.11 | 3.19 | 59 | 88 | 361,464 | 73 | 4189 | 361,487 | 5499 | 859 | 2169 | 361,482 | 86,509 | 9.39 |
| 180 slots | Restrictive | 35.41 | 0.84 | 9.23 | 3.74 | 58 | 87 | 359,940 | 1 | 1340 | 359,847 | 3603 | 364 | 2313 | 359,894 | 86,059 | 6.27 |
| | Free | 34.93 | 0.85 | 9.22 | 3.70 | 58 | 87 | 361,499 | 0 | 1527 | 361,522 | 3573 | 362 | 2427 | 361,516 | 86,635 | 6.25 |
| 120 slots | Restrictive | 36.55 | 0.81 | 9.23 | 3.86 | 58 | 87 | 359,804 | 0 | 833 | 359,834 | 3083 | 222 | 2324 | 359,824 | 86,079 | 5.33 |
| | Free | 35.88 | 0.82 | 9.22 | 3.79 | 58 | 87 | 358,235 | 0 | 1086 | 358,191 | 2928 | 216 | 2547 | 358,209 | 85,817 | 5.33 |
| 90 slots | Restrictive | 36.98 | 0.80 | 9.24 | 3.90 | 58 | 87 | 360,039 | 0 | 690 | 360,141 | 2858 | 165 | 2294 | 360,102 | 86,237 | 4.92 |
| | Free | 36.10 | 0.82 | 9.22 | 3.81 | 58 | 87 | 359,205 | 0 | 984 | 359,198 | 2667 | 158 | 2670 | 359,198 | 86,003 | 4.91 |
| 72 slots | Restrictive | 37.18 | 0.80 | 9.23 | 3.92 | 58 | 87 | 358,994 | 0 | 613 | 358,990 | 2767 | 136 | 2185 | 358,986 | 85,811 | 4.69 |
| | Free | 36.29 | 0.82 | 9.22 | 3.83 | 58 | 87 | 357,787 | 0 | 916 | 357,742 | 2502 | 128 | 2765 | 357,743 | 85,626 | 4.68 |
| 60 slots | Restrictive | 37.32 | 0.80 | 9.25 | 3.94 | 58 | 87 | 359,292 | 0 | 542 | 359,336 | 2692 | 122 | 2200 | 359,329 | 85,998 | 4.56 |
| | Free | 36.34 | 0.82 | 9.23 | 3.84 | 58 | 87 | 360,504 | 0 | 907 | 360,528 | 2409 | 114 | 2881 | 360,532 | 86,211 | 4.57 |
| 40 slots | Restrictive | 36.86 | 0.80 | 9.20 | 3.89 | 58 | 87 | 358,723 | 0 | 735 | 358,690 | 2631 | 103 | 2050 | 358,702 | 85,654 | 4.43 |
| | Free | 36.10 | 0.82 | 9.22 | 3.81 | 58 | 87 | 359,153 | 0 | 988 | 359,181 | 2209 | 92 | 3150 | 359,164 | 85,909 | 4.43 |
| 36 slots | Restrictive | 36.82 | 0.81 | 9.26 | 3.89 | 58 | 87 | 360,041 | 0 | 762 | 360,112 | 2706 | 103 | 1897 | 360,102 | 86,262 | 4.42 |
| | Free | 36.07 | 0.82 | 9.23 | 3.81 | 58 | 87 | 357,920 | 0 | 1023 | 357,835 | 2145 | 89 | 3227 | 357,871 | 85,779 | 4.42 |
| 30 slots | Restrictive | 36.94 | 0.80 | 9.23 | 3.90 | 58 | 87 | 358,694 | 0 | 712 | 358,634 | 2660 | 102 | 2007 | 358,664 | 85,904 | 4.44 |
| | Free | 35.94 | 0.82 | 9.22 | 3.80 | 58 | 87 | 360,205 | 0 | 1094 | 360,210 | 2114 | 90 | 3435 | 360,217 | 86,270 | 4.44 |
| 24 slots | Restrictive | 36.63 | 0.81 | 9.23 | 3.87 | 58 | 87 | 359,950 | 0 | 845 | 360,025 | 2436 | 96 | 2663 | 359,993 | 86,097 | 4.50 |
| | Free | 35.55 | 0.83 | 9.22 | 3.76 | 58 | 87 | 358,568 | 0 | 1254 | 358,586 | 2038 | 86 | 3700 | 358,570 | 85,627 | 4.49 |
| 20 slots | Restrictive | 36.18 | 0.82 | 9.25 | 3.82 | 58 | 87 | 359,420 | 0 | 1042 | 359,411 | 2301 | 96 | 3196 | 359,439 | 86,144 | 4.58 |
| | Free | 35.01 | 0.85 | 9.25 | 3.71 | 58 | 87 | 361,795 | 0 | 1525 | 361,943 | 2016 | 84 | 4000 | 361,877 | 86,679 | 4.56 |
| 18 slots | Restrictive | 35.76 | 0.83 | 9.22 | 3.78 | 58 | 87 | 361,334 | 0 | 1234 | 361,348 | 2216 | 98 | 3535 | 361,362 | 86,581 | 4.64 |
| | Free | 34.71 | 0.85 | 9.22 | 3.67 | 58 | 87 | 358,361 | 0 | 1674 | 358,300 | 1973 | 88 | 4127 | 358,326 | 85,731 | 4.64 |

restrictive YC deployment system is used is statistically significant at the 95% confidence level.

Table 5 contains the same trends that were visible in Tables 3 and 4. Once again, the numbers in the "waitL" column are much larger than those in the "waitU" column. That is, most QC delays occur during loading, not unloading. In addition, the "waitL" column reveals that the QCs are spending significantly less (more) time waiting for YTs to appear beneath them during loading when the restrictive (free) YC deployment system is used. This directly accounts for the superiority of the restrictive YC deployment system. Indeed, there is once again a high negative correlation between GCR and the items in the "waitL" column, especially for the experiments involving the restrictive YC deployment system. In particular, GCR is the highest (at 29.06 lifts/h) precisely when the QC waiting time during loading is the lowest (4916 min over a 2-week data collection period). Yet again, the superiority of the restrictive YC deployment system apparently originates from the reduced amount of YC gantrying associated with this system as shown in the "gant" column. Finally, note yet again in the "trv/cnt" column that the average YT travel time per container hauled is essentially convex with respect to the block length.

Table 6 shows the experimental results for the large terminal with more equipment. Again, each row of data was obtained by averaging the results from six independent simulation runs and rounding the average to the nearest integer or hundredth. Thus, the table is based on the results from a total of 23×6=138 experiments. Each individual experiment was completed in less than 20 min of CPU time.

Sixteen different performance measures were tracked in each experiment. Fig. 11 graphically displays how the most important performance measure—GCR—is affected by the yard layout and YC deployment system. This figure shows for the fourth time that the restrictive YC deployment system outperforms the free YC deployment system for all possible block lengths, this time by about 2% on average. Also, GCR is essentially concave with respect to the block length, attaining its maximum value when the block length is 60 slots for both YC deployment algorithms. Statistical tests indicate that the performance difference between this block length (GCR = 37.32) and the second-best block length (72 slots with GCR = 37.18) when the restrictive YC deployment system is used is statistically significant at the 95% confidence level. Overall, the GCR with more equipment is roughly 26% higher than the GCR with less equipment.

Table 6 contains the same basic trends that we observed in Tables 3–5. In particular, yet again, the "waitU" and "waitL" columns reveal that most QC delays occur during loading, not unloading. As mentioned earlier, this is primarily due to the reduced substitutability of YTs during loading. It may also be due to the fact that loading operations begin with slow-handling, slow-moving YCs retrieving cargo from a large storage area, whereas unloading operations begin with fast-handling QCs pulling containers from densely packed vessels. Once again, the "waitL" column reveals that the QCs are spending significantly less time waiting for YTs to appear beneath them during loading when the restrictive YC deployment system is used. This directly accounts for the superiority of the restrictive YC deployment system. Indeed, note for the fourth time the high negative correlation between GCR and the items in the "waitL" column, especially for the experiments involving the restrictive YC deployment system. Yet again, the superiority of the restrictive YC deployment system seems to be tied to the reduced amount of YC gantrying associated with this system as shown in the "gant" column. Finally, note again in the "trv/cnt" column that the average YT travel time per container hauled is essentially convex with respect to the block length. Not surprisingly, the values in this column are very close to their counterparts in Table 5.

### 6.6. Discussion

As Figs. 8–11 demonstrate, two major phenomena hold up across all experimental scenarios. First of all, the restrictive YC deployment

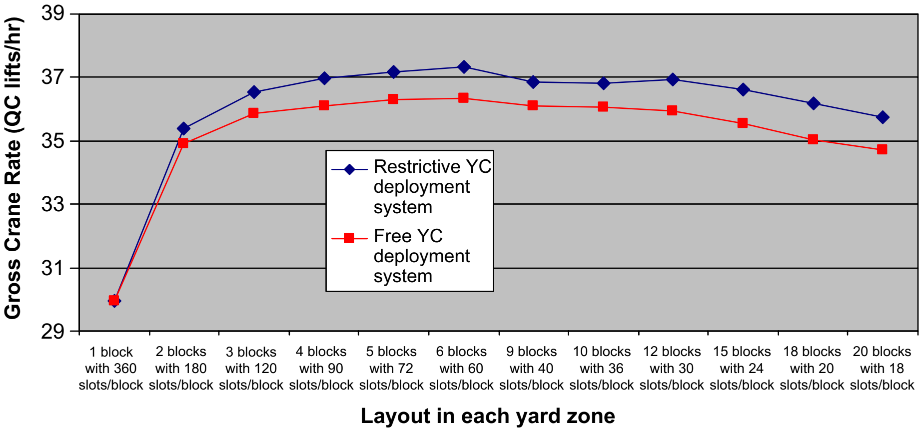

**Fig. 11.** Performance of 12 different block lengths and two different YC deployment systems at the large terminal with more equipment.

system is superior to the free YC deployment system. This superiority is manifested in all 12 combinations of block length and fleet size at the small terminal, and all 22 combinations of block length and fleet size at the large terminal. Managers of container terminal facilities would not be too surprised by these results. Indeed, to the authors' knowledge, most container terminals already restrict the inter-block movements of YCs to some degree. Nevertheless, these findings are useful in that they (1) validate current practice at container terminals and (2) are the first results in the literature to directly evaluate alternative real-time YC deployment systems for container terminals in terms of a long-run, global performance measure such as GCR or average vessel turnaround time. Interestingly, the superiority of the restrictive YC deployment system is more pronounced when the terminal is larger and when there is less equipment. In other words, the proper coordination of inter-block YC movements becomes more important as a terminal gets larger and/or YCs become scarcer.

The second phenomenon that holds up across all experimental scenarios is the concavity of GCR with respect to block length. Indeed, in each of the eight combinations of container terminal size (small or large), amount of equipment deployed (more or less), and YC deployment system (restricted or free), GCR is essentially concave with respect to block length.

In addition, the block length that yields the highest GCR is fairly robust across all experiments. Depending on the scenario, a block length of 56, 60, or 72 slots was found to yield the highest GCR, assuming the better YC deployment system is used. At the small terminal, the two block lengths yielding the highest GCR are 56 and 84 slots, regardless of the amount of equipment deployed or the YC deployment system used. Note that these are neighboring block lengths. Moreover, the two block lengths yielding the highest GCR at the large terminal, regardless of the amount of equipment deployed or the YC deployment system used, are 60 and 72 slots—precisely the only two block lengths we considered at the large terminal that belong to the interval [56, 84].

The above results are quite robust given the high stochasticity built into the simulation model and considering that block lengths as high as 360 slots and as low as 14 slots were considered. Indeed, they allow us to speculate that a block length of 56–72 slots would perform well for transshipment container terminals of varying sizes in the real world, assuming that such a block length is feasible. Fifty-six to 72 slots is somewhat longer than the average block in use at most terminals today, so our results beg the following question: Should the designers of real container terminal facilities be considering longer blocks? The authors hope the current paper will ignite a debate on this matter in the near future.

Despite some mild success, several questions remain on the issue of block length. Why does a block length of 72 slots perform the best at the large terminal with less equipment, while a block length of 60 slots performs the best at the large terminal with more equipment? More importantly, how does the optimal block length depend on the various characteristics of a container terminal? More studies, based on simulation, stochastic, and/or deterministic modeling methodologies, will probably be needed before we can provide any concrete answers to these questions.

## 7. Conclusion

In this paper, the authors have investigated how two parameters—(1) the length of the storage blocks in a terminal's container yard and (2) the system that deploys yard cranes among blocks in the same zone—affect the overall, long-run performance of a seaport container terminal as measured in terms of GCR (i.e. average quay crane work rate). Toward this end, the authors constructed a discrete event simulation model of terminal operations that was designed to reproduce the multi-objective, stochastic, real-time environment at a multiple-berth facility. The experiments considered four fully dynamic container terminal scenarios and 13, 13, 23, and 23 different block length and yard crane deployment system combinations for these scenarios, respectively. Six independent simulation replications were performed for each of the above setups, yielding 432 experiments in all.

Results from the simulation experiments are two-fold. Firstly, they indicate that GCR is concave with respect to block length and that a block length between 56 and 72 (20-ft) slots yields the highest GCR. Secondly, they resoundingly show that a yard crane deployment system that restricts yard crane movement yields a higher GCR than a system that allows greater yard crane mobility. It is worth noting that a block length of 56–72 slots is somewhat longer than the average block at most terminals today. While the experiments are not exhaustive enough to give our findings prescriptive power, they nevertheless do provide the first direct connection in the literature between yard layout (in this case block length) and long-run performance at a multiple-berth seaport container terminal.

Further research on the issue of block length could proceed in several directions. For example, additional simulation experiments which consider non-transshipment container terminals and various other scenarios could be performed. Future effort could also be devoted to identifying and developing a formal, queuing-based model that underlies the practical logistic processes at hand.

## Acknowledgment

The authors wish to thank three anonymous referees whose comments have greatly improved the quality of this paper.

## References

[1] Containerisation International. February 2008: 19.
[2] Containerisation International. March 2004: 5.
[3] Meersmans PJM, Dekker R. Operations research supports container handling. Econometric Institute Report 2001; EI 2001-22.
[4] Stahlbock R, Voss S. Operations research at container terminals: a literature update. OR Spectrum 2008;30:1–52.
[5] Steenken D, Voss S, Stahlbock R. Container terminal operation and operations research—a classification and literature review. OR Spectrum 2004;V26(1): 3–49.
[6] Vis IFA, de Koster R. Transshipment of containers at a container terminal: an overview. European Journal of Operational Research 2003;47(1):1–16.
[7] Günther H-O, Kim KH. Container terminals and terminal operations. OR Spectrum 2006;V28(4):437–45.
[8] Murty KG, Liu J, Wan Y-W, Linn R. A decision support system for operations in a container terminal. Decision Support Systems 2005;39(3):309–32.
[9] Han Y, Lee LH, Chew EP, Tan KC. A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub. OR Spectrum 2008, doi:10.1007/s00291-008-0127-6.
[10] Cheung RK, Li C-L, Lin W. Interblock crane deployment at container terminals. Transportation Science 2002;36(1):79–93.
[11] Linn R, Liu J-Y, Wan Y-W, Zhang C, Murty KG. Rubber tired gantry crane deployment for container yard operation. Computers & Industrial Engineering 2003;45(3):429–42.
[12] Linn RJ, Zhang C-Q. A heuristic for dynamic yard crane deployment in a container terminal. IIE Transactions 2003;35(2):161–74.
[13] Zhang C, Wan Y-W, Liu J, Linn RJ. Dynamic crane deployment in container storage yards. Transportation Research Part B: Methodological 2002;36(6): 537–55.
[14] Duinkerken MB, Dekker R, Kurstjens STGL, Ottjes JA, Dellaert NP. Comparing transportation systems for inter-terminal transport at the Maasvlakte container terminals. OR Spectrum 2006;V28(4):469–93.
[15] Liu C-I, Jula H, Ioannou PA. Design, simulation, and evaluation of automated container terminals. IEEE Transactions on Intelligent Transportation Systems 2002;3(1):12–26.
[16] Nam K-C, Ha W-I. Evaluation of handling systems for container terminals. Journal of Waterway, Port, Coastal, and Ocean Engineering 2001;127(3): 171–5.
[17] Vis IFA. A comparative analysis of storage and retrieval equipment at a container terminal. International Journal of Production Economics 2006;103(2):680–93.
[18] Yang C, Choi Y, Ha T. Simulation-based performance evaluation of transport vehicles at automated container terminals. OR Spectrum 2004;V26(2):149–70.
[19] Kim KH, Park YM, Jin MJ. An optimal layout of container yards. OR Spectrum 2008, doi:10.1007/s00291-007-0111-6.
[20] Liu C-I, Jula H, Vukadinovic K, Ioannou P. Automated guided vehicle system for two container yard layouts. Transportation Research Part C: Emerging Technologies 2004;12:349–68.
[21] Alessandri A, Sacone S, Siri S. Modelling and optimal receding-horizon control of maritime container terminals. Journal of Mathematical Modelling and Algorithms 2007;6:109–33.
[22] Bielli M, Boulmakoul A, Rida M. Object oriented model for container terminal distributed simulation. European Journal of Operational Research 2006;175(3):1731–51.
[23] Borovits I, Ein-Dor P. Computer simulation of a seaport container terminal. Simulation 1975;25:141–4.
[24] Bruzzone A, Signorile R. Simulation and genetic algorithms for ship planning and shipyard layout. Simulation 1998;71(2):74–83.
[25] Canonaco P, Legato P, Mazza R, Musmanno R. A queuing network model for the management of berth crane operations. Computers & Operations Research 2008;35:2432–46.
[26] Chung Y-G, Randhawa SU, McDowell ED. A simulation analysis for a transtainer-based container handling facility. Computers & Industrial Engineering 1988;14(2):113–25.
[27] Dekker R, Voogd P, van Asperen E. Advanced methods for container stacking. OR Spectrum 2006;V28(4):563–86.
[28] Demirci E. Simulation modeling and analysis of a port investment. Simulation 2003;79(2):94–105.
[29] Dragovic B, Park NK, Radmilovic Z, Maras V. Simulation modelling of ship-berth link with priority service. Maritime Economics & Logistics 2005;7(4):316–35.
[30] El Sheikh AAR, Paul RJ, Harding AS, Balmer DW. A microcomputer-based simulation study of a port. The Journal of the Operational Research Society 1987;38(8):673–81.
[31] Froyland G, Koch T, Megow N, Duane E, Wren H. Optimizing the landside operation of a container terminal. OR Spectrum 2008;30:53–75.
[32] Gambardella LM, Mastrolilli M, Rizzoli AE, Zaffalon M. An optimization methodology for intermodal terminal management. Journal of Intelligent Manufacturing 2001;V12(5):521–34.
[33] Gambardella LM, Rizzoli AE, Zaffalon M. Simulation and planning of an intermodal container terminal. Simulation 1998;71(2):107–16.
[34] Grunow M, Günther H-O, Lehmann M. Dispatching multi-load AGVs in highly automated seaport container terminals. OR Spectrum 2004;V26(2):211–35.
[35] Hartmann S. Generating scenarios for simulation and optimization of container terminal logistics. OR Spectrum 2004;V26(2):171–92.
[36] Hayuth Y, Pollatschek MA, Roll Y. Building a port simulator. Simulation 1994;63:179–89.
[37] Kia M, Shayan E, Ghotb F. Investigation of port capacity under a new approach by computer simulation. Computers & Industrial Engineering 2002;42(2–4): 533–40.
[38] Kozan E. Comparison of analytical and simulation planning models of seaport container terminals. Transportation Planning and Technology 1997;20:235–48.
[39] Lee TW, Park NK, Lee DW. A simulation study for the logistics planning of a container terminal in view of SCM. Maritime Policy & Management 2003;30: 243–54.
[40] Legato P, Mazza RM. Berth planning and resources optimisation at a container terminal via discrete event simulation. European Journal of Operational Research 2001;133(3):537–47.
[41] Merkuryev Y, Tolujew J, Blumel E, Novitsky L, Ginters E, Viktorova E. et al. A modeling and simulation methodology for managing the Riga harbour container terminal. Simulation 1998;71(2):84–95.
[42] Nam K-C, Kwak K-S, Yu M-S. Simulation study of container terminal performance. Journal of Waterway, Port, Coastal, and Ocean Engineering 2002;128(3):126–32.
[43] Nevins MR, Macal CM, Joines JC. A discrete-event simulation model for seaport operations. Simulation 1998;70(4):213–23.
[44] Ottjes JA, Veeke HPM, Duinkerken MB, Rijsenbrij JC, Lodewijks G. Simulation of a multiterminal system for container handling. OR Spectrum 2006;V28(4): 447–68.
[45] Parola F, Sciomachen A. Intermodal container flows in a port system network: analysis of possible growths via simulation models. International Journal of Production Economics 2005;97(1):75–88.
[46] Petering MEH, Murty KG. Simulation analysis of algorithms for container storage and yard crane scheduling at a container terminal. In: Proceedings of the second international intelligent logistics systems conference, Brisbane, Australia, 2006.
[47] Petering MEH, Wu Y, Li W, Goh M, Murty KG, de Souza R. Simulation analysis of yard crane routing systems at a marine container transshipment terminal. In: Proceedings of the international congress on logistics and supply chain management systems, Kaohsiung, Taiwan, 2006.
[48] Pope JA, Rakes TR, Rees LP, Crouch IWM. A network simulation of high-congestion road-traffic flows in cities with marine container terminals. The Journal of the Operational Research Society 1995;46(9):1090–101.
[49] Sgouridis SP, Makris D, Angelides DC. Simulation analysis for midterm yard planning in container terminal. Journal of Waterway, Port, Coastal, and Ocean Engineering 2003;129(4):178–87.
[50] Shabayek AA, Yeung WW. A simulation model for the Kwai Chung container terminals in Hong Kong. European Journal of Operational Research 2002;140(1):1–11.
[51] Silberholz MB, Golden BL, Baker EK. Using simulation to study the impact of work rules on productivity at marine container terminals. Computers & Operations Research 1991;18(5):433–52.
[52] Thiers GF, Janssens GK. A port simulation model as a permanent decision instrument. Simulation 1998;71(2):117–25.
[53] Yun WY, Choi YS. A simulation model for container-terminal operation analysis using an object-oriented approach. International Journal of Production Economics 1999;59(1–3):221–30.
[54] Petering MEH. Design, analysis, and real-time control of seaport container transshipment terminals. PhD dissertation, University of Michigan, USA; 2007.
[55] Petering MEH, Murty KG. Real-time container storage location assignment at a seaport container transshipment terminal, 2007, submitted.
[56] Petering MEH, Wu Y, Li W, Goh M, de Souza R. Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals, 2008, under revision.