Double-Cycling Strategies for Container Ships and Their Effect on Ship Loading and Unloading Operations

Article *in* Transportation Science · November 2006

DOI: 10.1287/trsc.1060.0148 · Source: DBLP

2 authors, including:

Anne Goodchild
University of Washington Seattle

TRANSPORTATION SCIENCE
Vol. 40, No. 4, November 2006, pp. 473–483
ISSN 0041-1655 | EISSN 1526-5447 | 06 | 4004 | 0473

DOI 10.1287/trsc.1060.0148

Double-Cycling Strategies for Container Ships and Their Effect on Ship Loading and Unloading Operations

Anne V. Goodchild
Department of Civil and Environmental Engineering, 121E More Hall, Box 352700, University of Washington, Seattle, Washington 98195-2700, annegood@u.washington.edu

Carlos F. Daganzo
Department of Civil and Environmental Engineering, 416A McLaughlin Hall, University of California, Berkeley, Berkeley, California 94720, daganzo@ce.berkeley.edu

Loading ships as they are unloaded (double cycling) can improve the efficiency of a quay crane and container port. This paper describes the double-cycling problem, and presents solution algorithms and simple formulae to determine reductions in the number of operations and operating time using the technique. We focus on reducing the number of operations necessary to turn around a row of a ship. The problem is first formulated as a scheduling problem, which can be solved optimally. A simple lower bound for all strategies is then developed. We also present a greedy algorithm that yields a simple and tight upper bound. The gap between the upper and lower bounds is so small that the formula for either bound is an accurate predictor of crane performance. The analysis is then extended to double cycling when ships have deck hatches. Results are presented for many simulated vessels, and compared to empirical data from a real-world trial. The research demonstrates that double cycling can create significant efficiency gains in crane productivity, typically reducing the number of cycles by about 20% and the operational time by about 10% when double cycling only below deck.

Key words: container port terminal; freight transportation; terminal operations
History: Received: September 2004; revisions received: April 2005, December 2005; accepted: January 2006.

Introduction

The volume of goods moved by container through the U.S. transportation system has grown dramatically over the past 15 years, but infrastructure has not been developed at a similar rate. In 2004, peak levels of container traffic through major U.S. West Coast ports jumped approximately 15% from the previous year. This caused significant port congestion; for example, containers required an additional week just to be moved from vessels through the marine terminals (Mongelluzzo 2005a). There is no reason to believe that this growth will not continue, except that our inland transportation infrastructure will not have sufficient capacity to carry it. This growth in container volumes will require additional capacity on the freight transportation network and through ports, in particular. Strategies are required that speed the movement of freight through the system, and specifically through terminals. In this research we consider such a strategy. Quay cranes are the most expensive single unit of handling equipment in port container terminals; because of this, one of the key operational bottlenecks at ports is quay crane availability (Crainic and Kim 2005). By improving quay-crane efficiency, ports can reduce ship turn-around time, improve port productivity, and improve throughput in the freight transportation system. The research presented in this paper addresses the key bottleneck to port productivity: quay-crane efficiency. In contrast to other measures to increase capacity such as terminal expansion and information technology deployments, double cycling, the method considered here, is a low-cost method to increase capacity; it does not require new technology or infrastructure. Although double cycling will not solve the capacity problem in the long term, it can be more quickly implemented than other solutions, and can be used to complement other strategies.

Double cycling is a technique that can be used to improve the efficiency of quay cranes by eliminating some empty crane moves. Instead of using the current method, where often all relevant containers are unloaded from the vessel before any are loaded (*single cycling*), containers are loaded and unloaded simultaneously (see Figure 1). This allows the crane to carry a container while moving from the apron to the ship (one move), as well as from the ship to the apron, thus doubling the number of containers transported in

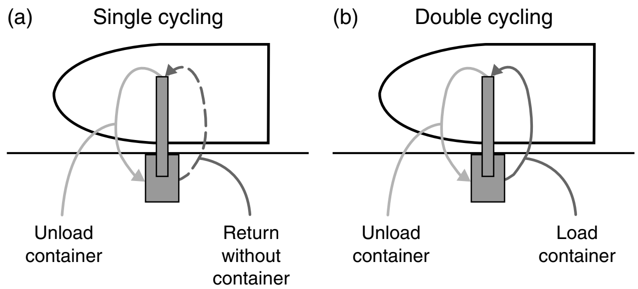

Figure 1 (a) Unloading Using Single Cycling; (b) Unloading and Loading with Double Cycling

a cycle (or two moves). This crane efficiency improvement can be used to reduce ship turn-around time and therefore improve port throughput, and address the capacity problem.

In their efforts to increase productivity, ports have undertaken various projects such as renovating and adding terminals, constructing and expanding intermodal facilities, and implementing new IT infrastructure (Mongelluzzo 2005b). Because crane productivity is so important, ports have also invested in various crane utilization improvement strategies. For example, dual hoist cranes have been developed that separate the crane's cycle into two subcycles that can be operated independently. Although double cycling is used to a limited extent in practice, and small-scale trials have been undertaken (TranSystems Corporation 2003), a broad implementation of double cycling has not occurred. One of the reasons for this is that small-scale trials have understated the benefits of double cycling which, as we will show, increase with ship size. The absence of a rigorous analysis of the efficiencies of double cycling leaves open the question of its impact on crane operations. This paper attempts to fill this void.

Because the necessary operational changes with double cycling are not well understood, some operators doubt that its benefits can overcome its operational costs. To alleviate this concern, we will assume the ship's loading plans are given, and that they are the same with and without double cycling. This is desirable, because shipping lines use software tools to create loading plans that accommodate, amongst other concerns, (a) vessel stability requirements, (b) priority of delivery, (c) placement constraints on hazardous materials, (d) refrigerated containers, (e) above- and below-deck storage, and (f) strategies to minimize the number of cycles necessary to unload containers at subsequently visited ports. From these tools, a sequence of operations is generated for the crane operator, foreman (who directs landside operations), and terminal management system. When considering the benefits of double cycling, we assumed that existing planning tools had been used to create a loading plan, as is current practice, and that this loading plan has made no accommodations for double cycling. We therefore consider changes only to the crane's sequence of operations. In practice, this would be determined at a planning stage, and the crane operator would be given a sequence of operations to carry out, in the same way that a sequence of operations is given when performing single-cycle operations. This way we demonstrate that double cycling is feasible and beneficial without changing the quayside operations. Double cycling does require some changes, but as described in Goodchild (2005), they are minor and beneficial, on the whole.

Problems of port design and operation are the subject of much academic research (see §1), and are currently the subject of much political attention (California State Assembly Bill 2650, 2002–2003, and 2042, 2003–2004); nevertheless, to date no study on double cycling has appeared in a scholarly journal. The ideas presented here are not meant to substitute for detailed terminal and vessel planning programs, which are well suited to managing a specific vessel and terminal configuration, but to provide portable insights into double cycling at a more general level.

1. Literature Review

A significant amount of operational research has addressed port problems. These works typically focus on strategic design planning issues such as the number of berths (Schonfeld and Sharafeldien 1985), the size of storage space (Kim and Kim 2002), the number of various pieces of equipment to install (Vis, de Koster, and Roodbergen 2001), and the trade-offs inherent in these choices (Taleb-Ibrahimi, Castilho, and Daganzo 1993). Also addressed are operational planning and control problems, including berth scheduling (Park and Kim 2003), berth assignment (Imai, Nishimura, and Papadimitriou 2001), quay-crane scheduling (Daganzo 1989; Peterofsky and Daganzo 1990), stowage planning and sequencing (Christiansen et al. 2005; Kim, Kang, and Ryu 2004), storage space planning (Castilho and Daganzo 1993), and dispatching of yard cranes and prime movers (Kim and Bae 1999). To date, most of this work utilizes queueing theory and stochastic models (Daganzo 1989), simulation (Lai and Leung 2000), and classical operations research techniques such as routing, network, and scheduling problems (Kim and Kim 2002). The operations research literature, however, has not yet addressed the double-cycling problem.

The only scientific work appears to be in the maritime economics literature, where researchers have explained the productivity gains from hatchless ships when double cycling is used (Bendall and Stent 1996). The work illustrates the impact that double cycling can have not just on port operation, but also on ship design. This work assumes that all containers except those above hatches can be double cycled, but does not address operational concerns.

Our paper examines these operational aspects. In the next section a framework is set for analysis of the problem. Section 2 also includes an example that is used to illustrate the problem's basic properties. In §3 we discuss a scheduling formulation to the problem. A lower bound to the problem is developed in §4. Section 5 presents the greedy algorithm and bounds its results from above. In §6 the problem is extended to accommodate ships with deck hatches. In §7 we develop a formula to convert benefits from number of cycles to an amount of time, compare results to empirical data, present the results of a computer program, and consider the economic impact of double cycling.

2. Modeling Framework

The layout of containers on a ship can be modelled as a three-dimensional matrix. Containers are stacked on top of one another, and arranged in rows (see Figure 2). One row stretches across the width of the ship. Large container vessels today typically hold 20 stacks of containers across the width of the ship, and up to 20 stacks along the length of the ship (40-foot equivalent units). Of course, we expect these figures to increase with the market penetration of Malacca-max carriers. Figure 2 gives a top and side view of a typical vessel (although the number of container stacks is not representative).

The complete operating cycle of the crane can be broken down into the following components:

(1) Locking to or unlocking from a container;

(2) Horizontal motion of the trolley, or trolley and container, across the ship;

(3) Vertical motion of the trolley, or trolley and container.

Between some crane cycles the crane may also move lengthwise along the ship. It is important to point out that the number of locking and unlocking operations is not affected by double cycling. In this research we will assume that dockside containers are ready for loading when required, and containers being unloaded can be quickly removed from the immediate area. In the initial analysis it will be assumed that ships lack hatch coverings, or doors on the deck that separate above-deck and below-deck storage. This assumption will be relaxed in §6.

Consider the case where a ship arrives in port with a set of containers on board to be unloaded and a loading plan for containers to be loaded. The loading plan indicates the placement of containers on the ship. Given are u_c and l_c, the number of containers to be unloaded and loaded, respectively, in each stack labelled c. Figure 3 is an example problem that will be used for illustrative purposes. Notice that in Figure 3, $u_A = 3$ and $l_A = 2$. A rehandle is a container that must be moved to access containers below it, but will be stowed again before the ship departs. Note that if any rehandles are necessary, we include these in the total number of loads and unloads. For example, if, during unloading, a container must be moved to access a container beneath it, the container moved will be counted as an unload. This container would then also be counted as a load when it is placed back in the original stack. We always assume that rehandles are replaced in the stack from which they were

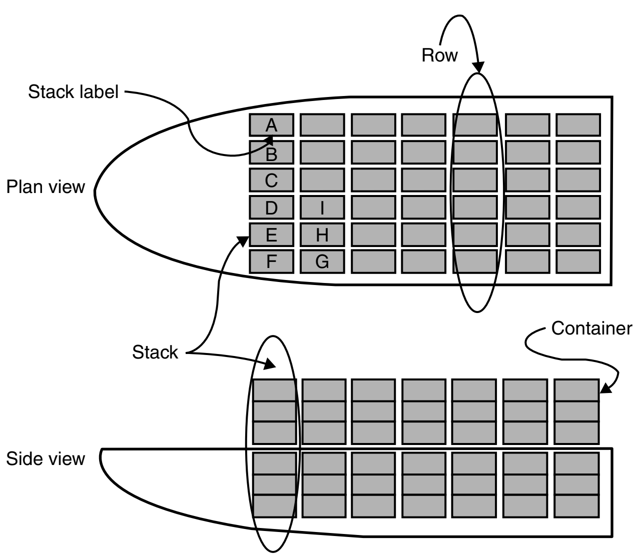

Figure 2 **Plan and Side Views of a Simplified Ship**
Note. Number of containers shown not representative of typical ship size.

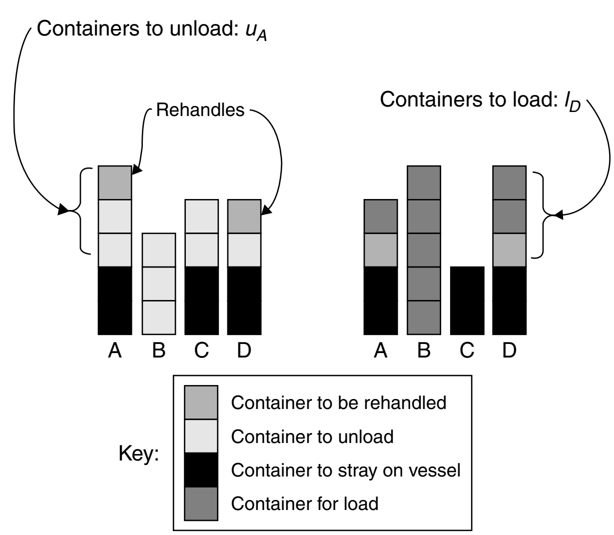

Figure 3 **Detailed Plan for Containers to be Unloaded and Loaded**

removed. Note that this will overestimate the amount of work necessary to unload and load a set of containers because the container is considered moved from the vessel to the shore and back to a location on the vessel. In this research we consider a move to be between the vessel and the apron, but in reality some rehandles may only be moved between locations on the vessel, typically a shorter distance.

If we consider that the time it takes to unload and load a ship is a measure of crane efficiency, then the goal of double cycling is to reduce the total turn-around time. A proxy for this is the number of cycles required to unload and load the ship. The number of cycles necessary to complete loading and unloading will be represented by the variable w. We will consider double cycling within one row of the ship. Due to the difficulty with which the crane moves laterally along the ship, it is not practical to consider double cycling across two rows. We will complete unloading and loading of one row before moving the crane lengthwise along the ship to the next row. We will consider time savings in §7, including the time required for the crane to move laterally along the vessel. A key feature of double cycling is the order in which the stacks within each row are handled; this is explained below.

Let S denote the set of stack labels in a row, $|S| = N$ the number of stacks in the set, and Π a permutation of S indicating an ordering of the stacks. A permutation is a one-to-one correspondence between the set of $n \in \{1, \ldots, N\}$ and $c \in S$, such that $\Pi(n) = c$, or $n = \Pi^{-1}(c)$. For example, in Figure 3, the set of stack labels is $S = \{A, B, C, D\}$. A permutation of these is $\{B, A, C, D\}$ given by the function Π_e where $\Pi_e(1) = B$, $\Pi_e(2) = A$, $\Pi_e(3) = C$, and $\Pi_e(4) = D$. We will restrict our attention to special cases of the generic double-cycling method described below.

- Choose an unloading permutation, Π'. Unload all containers in the first stack of the permutation, then all containers in the second stack of the permutation, proceed in this fashion until all stacks have been unloaded.
- Choose a loading permutation, Π, and load the stacks in that order. Load all containers in the first stack, then in the second, and so on. Loading can start in any stack as soon as it is empty or it contains just containers that should not be unloaded at this port. Once loading has begun in a stack, continue loading until that stack is complete.

In all cases of single or double cycling, we assume that the crane starts and finishes on the dock.

Figure 4(a) is a diagram for a single-cycling operation where the stacks of Figure 3 are handled in the order $\{A, B, C, D\}$ both for loading and unloading. Time is expressed in cycles. Note that loading operations must wait until cycle $w = 10$, when unloading

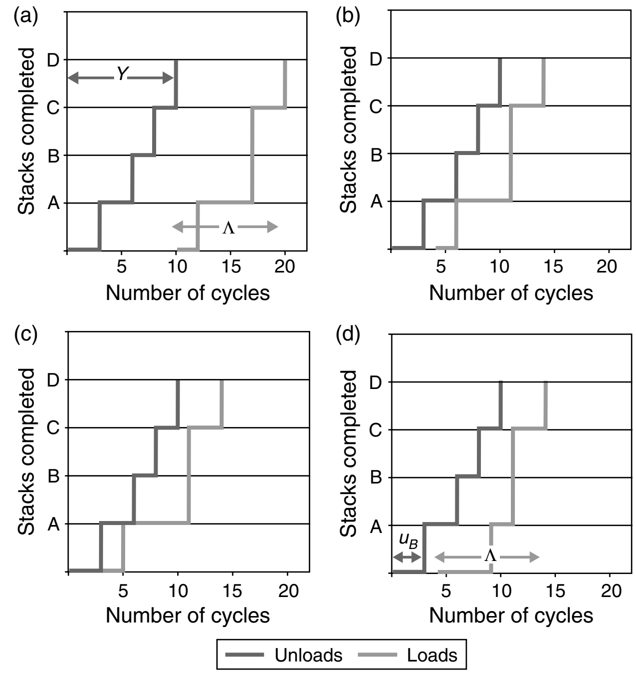

Figure 4 Turn-Around Time with Different Methods

Notes. (a) Single cycling with ordering A, B, C, D unloading starts at $w =$ 10, 20 cycles. (b) Double cycling with ordering A, B, C, D unloading starts at $w = 4$, 14 cycles. (c) Double cycling with ordering A, B, C, D unloading starts at $w = 3$, 14 cycles. (d) Double-cycling ordering B, A, C, D unloading starts at $w = 3$, 13 cycles.

is finished. The process requires $w = 20$ cycles. With single cycling, we assume the crane unloads each row of the vessel before loading any containers.

If we double cycle, we can still plot the unloading curve on the same diagram. Now, using the same sequence for unloading and loading, $\Pi' = \Pi = A, B, C, D$, we can shift the loading curve to the left as far as possible without overlapping the unloading curve. Figure 4(b) shows the maximum shift. Loading can start as early as $w = 4$ and the process would require only 14 cycles. The same number of cycles is obviously obtained if we start loading each stack as early as possible, as in Figure 4(c). This introduces some delay as the loading operations must wait one cycle for the unloading operations to be completed in stack B, but does not change the completion time.

With single cycling one cycle is required for every container. With double cycling, however, the number of cycles will depend on the sequence. Figure 4(d) shows that if the loading and unloading sequence is B, A, C, D, then the completion time is $w = 13$.

This framework considers the work of one crane, working on individual rows of a vessel. This does not limit our analysis to operations where only one crane works each vessel, because it can be reproduced for each crane, assuming the working areas of the vessel can be segmented by crane. A scheduling formulation for this problem is discussed in §3.

3. Scheduling Approach

Although double cycling involves only one physical machine (the quay crane), the problem can be formulated as a two-machine flow shop scheduling problem where one job corresponds to one stack and each job has two operations: an unloading operation that must be completed first, and a subsequent loading operation. The crane performs as the loading machine when its trolley is moving from the apron to the vessel and performs as the unloading machine when its trolley is moving from the vessel to the apron. We can assume that there is a separate machine for loading and another for unloading because in every cycle the crane can perform exactly one task of each type. As a result, the number of crane cycles available to do a task of either type is unaffected by what the crane does with tasks of the other type. Thus, the crane can operate on both tasks in the same way that two independent machines could handle one container per cycle.

Our problem is to determine the best unloading and loading sequences to minimize the maximum completion time (makespan). The formulation (shown in the appendix) includes a technological constraint: Stacks must be unloaded before they are loaded, but no precedence constraints. We assume all rehandles (containers that must be moved to access another container, but are to stay on the vessel) are loaded back into the stack from which they are unloaded, and that there are no constraints on the order in which a set of stacks is operated on by an individual machine (loading or unloading).

This problem can be solved optimally with Johnson's rule (1954). It has three key features. First, the assumption of uninterrupted loading and unloading of stacks is not restrictive; preemption cannot improve the solution. Second, it is sufficient to consider schedules in which the processing orders on the two machines are identical. Third, if the processing times are interchanged, then an equivalent inverse problem results.

Although Johnson's rule can be used to determine the optimal sequence for a specific vessel, it does not yield a simple formula for the number of cycles that could be used for port planning. In the next sections we develop such formulae. We start with a lower bound to the optimum.

4. A Lower Bound

Define

$$Y = \sum_{n=1}^{N} u_{\Pi'(n)} = \sum_{c \in S} u_c, \quad \text{and} \quad \Lambda = \sum_{n=1}^{N} l_{\Pi(n)} = \sum_{c \in S} l_c \tag{1}$$

as the total number of containers to unload and load. Recall from Figure 4(a) that, using single cycling, the number of cycles necessary to complete a row equals the number of containers to be moved:

$$Y + \Lambda. \tag{2}$$

For double cycling, with a specific loading permutation Π and unloading permutation Π', the number of cycles, w, must be at least $\Lambda + u_{\Pi'(1)}$. Obviously, then,

$$w \geq \Lambda + u_{\Pi'(1)} \geq \Lambda + \min_c(u_c). \tag{3}$$

Similarly, w must also be at least $Y + l_{\Pi(1)}$, and therefore

$$w \geq Y + l_{\Pi(1)} \geq Y + \min_c(l_c). \tag{4}$$

It follows that the number of cycles must satisfy for any Π and Π' (including the optimum):

$$w \geq \max\left\{\Lambda + \min_c(u_c),\, Y + \min_c(l_c)\right\}. \tag{5}$$

This is the proposed lower bound. We will now discuss an algorithm that provides an equally simple upper bound.

5. A Greedy Strategy and an Upper Bound

We propose to unload and load each stack as soon as possible, assuming that the loading and unloading sequences are given by the same greedy permutation, $\Pi' = \Pi = G$. The greedy permutation is obtained by ordering the stacks in descending order of the variable d_c where

$$d_c = l_c - u_c \quad \text{when } \Lambda \geq Y \tag{6}$$

$$d_c = u_c - l_c \quad \text{when } Y > \Lambda. \tag{7}$$

The rationale for Equations (6) and (7) is that we want the unloading operations to run ahead of the loading operations as much as possible.

We will assume in this section that stacks have been labelled by position in the handling sequence with the greedy strategy. So now u_j and l_j are the numbers of containers to be unloaded and loaded in the jth stack, $j = 1, \ldots, J$, where the sequence is given by the greedy strategy. We also define U_j as the cumulative time (in number of cycles) at which the jth stack is finished unloading: $U_j = u_1 + u_2 + \cdots + u_j$, and L_j as the combined operational time (in number of cycles) to load j stacks: $L_j = l_1 + l_2 + \cdots + l_j$.

Assume now that there are more loads than unloads, $\Lambda \geq Y$. In this case, $d_j = l_j - u_j$ and $D_j = L_j - U_j = d_1 + d_2 + \cdots + d_j$. Notice that $d_j \leq d_{j-1}$ because our strategy is greedy. Notice also that $D_J = \sum_{c \in S} d_c = \Lambda - Y \geq 0$ because there are more (or equal) loads than unloads.

Lemma 1. *If there are more* (*or equal*) *loads than unloads, then* $D_j \geq 0\ \forall j$.

Proof. Because there are more (or equal) loads than unloads, $D_J \geq 0$. Assume now that $D_j < 0$ for some j. For D_j to be negative, one of its components, d_k, must be negative for some $k \leq j$. This, however, would mean that $d_r < 0\ \forall r \geq k$, and also for $r \geq j$ (because the d_j sequence is decreasing for the greedy strategy). Hence, $D_J = D_j + d_{j+1} + \cdots + d_J$ is a sum of negative terms and would be negative, but this is a contradiction. Thus, there cannot be a $D_j < 0$ for some j. □

Lemma 2. *If there are more (or equal) loads than unloads, the number of cycles to complete operations with the greedy strategy, w_G, satisfies $w_G \leq \Lambda + \max_c\{u_c\}$.*

Proof. We assume that the loading operation is artificially postponed to a time s (number of cycles after the beginning of the loading operation) that avoids loading delays. The completion time with postponed loading is therefore $\Lambda + s$. Because the start of loading is postponed, this completion time must exceed or equal w_G, and can be used as an upper bound. If there are no intermediate delays, then the time to begin loading the jth stack, B_j, is $B_j = s + L_{j-1}$, i.e., the shift plus the time to load $j-1$ stacks (define $L_0 = 0$). We now look for the smallest s that guarantees that there are no intermediate delays to loading, i.e., that $B_j - U_j \geq 0$, $\forall j$. Note that $\{B_j - U_j\} = [s - u_j] + [L_{j-1} - U_{j-1}]$ and that the first term on the right side is nonnegative if we choose $s = \max_{c\in S}\{u_c\}$. We also see that the second term is nonnegative by Lemma 1. Thus, $B_j - U_j \geq 0$, $\forall j$, if $s = \max_{c\in S}\{u_c\}$. $\Lambda + s = \Lambda + \max_{c\in S}\{u_c\}$ is an upper bound to w_G, and so the lemma is proven. □

Lemma 3. *If $\Lambda \leq Y$, then $w_G \leq Y + \max_c\{l_c\}$.*

Proof. G in this case is defined by Equation (7). That Lemma 3 is true should be obvious by symmetry, because time reversals map any problem into its inverse. (If one were to videotape the process of unloading and loading the row, and then play this recording in reverse, the reversed video would display a sequence of operations with the same total time as for a problem in which the role of loads and unloads is switched.) □

The results from Equation (5), Lemma 2, and Lemma 3 can be neatly summarized if we define $u' = \min_c(u_c)$, $l' = \min_c(l_c)$, $u^* = \max_c\{u_c\}$, and $l^* = \max_c\{l_c\}$. The following is true:

Theorem 1. $\max\{\Lambda + u', Y + l'\} \leq w^* \leq w_G \leq \max\{\Lambda + u^*, Y + l^*\}$.

For large ships where $\{\Lambda, Y\} \gg (u^*, l^*)$, the upper and lower bounds are very close to each other. In fact, if u_c and l_c are bounded by a constant (stack size), then the gap between the upper and lower bound vanishes as the number of stacks (problem size) tends to infinity. We now consider ships with deck hatches.

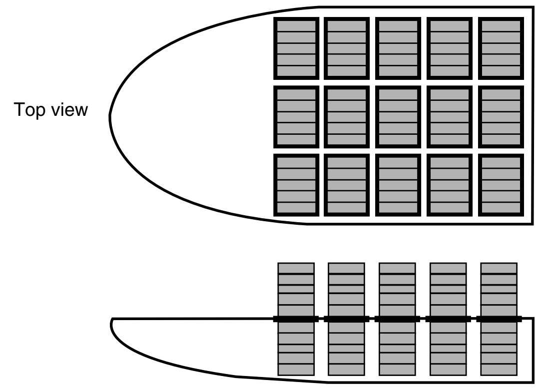

Figure 5 **Hatched Ship**
Note. Top and side views of a ship with hatch coverings.

6. Deck Hatches

Today most container ships have hatch coverings, as shown in Figure 5. These are large steel plates that separate above-deck and below-deck storage. They normally cover one-third of a single row. Figure 5 displays a vessel three hatches wide, with five stacks of containers above and five stacks below each hatch. A typical stack includes up to eight containers above deck and a similar number below. Hatches change the nature of the problem, because the stacks cannot be handled without interruption. To access the containers below a hatch all containers must be unloaded from above the hatch, and before loading containers atop a hatch all containers below the hatch must be loaded.

We propose a decomposition algorithm that reduces the problem of handling a hatched row to a sequence of problems already addressed in this paper. As in the hatchless case, each stack on the row is given an initial label. To carry out the strategy it is necessary to

1. Order the hatches using a greedy strategy using the same method as for the hatchless case. Treat the hatches as stacks, considering only the containers atop the hatches.
2. Order the stacks within each hatch using a greedy strategy, considering only the containers below deck.

The algorithm is then as follows:

1. Apply any efficient strategy with $\Pi = \Pi'$ to the containers above deck (e.g., the greedy strategy or Johnson's rule), treating hatches as stacks and pausing each time all containers above hatch h have been removed.
2. During the hth pause, unload and load the containers below the hth hatch using any efficient strategy with $\Pi = \Pi'$.

This method may not provide the fewest cycles to complete a row but is efficient, easy to implement, and yields simple performance formulae by building on Theorem 1. The theorem is useful because each piece below a hatch can be viewed as a hatchless row, and the containers above a hatch as a stack of a hatchless row. These analogies allow us to use the analysis of the hatchless ship to develop bounds for the hatched case. First it is necessary to define some notation.

- h—hatch index
- S_h—the set of stacks for hatch h
- N_h—the number of stacks above or below hatch h
- H—the set of hatches
- $\bar{u}_{hc}$—the number of containers to unload below hatch h in stack $c \in S_h$
- $\underline{u}_{hc}$—the number of containers to unload above hatch h in stack $c \in S_h$
- $\bar{l}_{hc}$—the number of containers to load below hatch h in stack $c \in S_h$
- $\underline{l}_{hc}$—the number of containers to load above hatch h in stack $c \in S_h$
- $\bar{u}_h = \sum_{c\in S_h} \bar{u}_{hc}$—containers to unload below hatch h
- $\underline{u}_h = \sum_{c\in S_h} \underline{u}_{hc}$—containers to unload above hatch h
- $\bar{l}_h = \sum_{c\in S_h} \bar{l}_{hc}$—containers to load below hatch h
- $\underline{l}_h = \sum_{c\in S_h} \underline{l}_{hc}$—containers to load above hatch h
- $\bar{Y} = \sum_{h\in H}\sum_{c\in S_h} \bar{u}_{hc}$—containers for unloading below deck
- $\underline{Y} = \sum_{h\in H}\sum_{c\in S_h} \underline{u}_{hc}$—containers for unloading above deck
- $\bar{\Lambda} = \sum_{h\in H}\sum_{c\in S_h} \bar{l}_{hc}$—containers for loading below deck
- $\underline{\Lambda} = \sum_{h\in H}\sum_{c\in S_h} \underline{l}_{hc}$—containers for loading above deck
- w_A—the number of cycles above deck
- w_B—the number of cycles below deck

Theorem 2. *An upper bound on the optimum number of cycles for the hatched case is*

$$\sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max\{\underline{\Lambda}, \underline{Y}\} + \max_h\{\underline{u}_h, \underline{l}_h\} + \max_{c\in S_h}\{\bar{u}_{hc}, \bar{l}_{hc}\}.$$

Proof. From Theorem 1 the number of cycles above deck, w_A, is bounded by

$$w_A \le \max\{\underline{\Lambda}, \underline{Y}\} + \max_h\{\underline{u}_h, \underline{l}_h\}. \tag{8}$$

Likewise, the number of cycles below deck, w_B, is bounded by

$$w_B = \sum_{h\in H} w_{B,h} \le \sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max_{c\in S_h}\{\bar{u}_{hc}, \bar{l}_{hc}\}. \tag{9}$$

Obviously, then, the total number of cycles with the algorithm satisfies

$$w_A + w_B \le \sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max\{\underline{\Lambda}, \underline{Y}\} + \max_h\{\underline{u}_h, \underline{l}_h\} + \max_{c\in S_h}\{\bar{u}_{hc}, \bar{l}_{hc}\}, \tag{10}$$

which is what we set out to prove. □

Theorem 3. *A lower bound on the optimum number of cycles for the hatched case is*

$$\sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max\{\underline{\Lambda}, \underline{Y}\} + \min_h\{\underline{u}_h, \underline{l}_h\} + \min_{c\in S_h}\{\bar{u}_{hc}, \bar{l}_{hc}\}.$$

Proof. We know from Theorem 1, but treating each hatch as a stack, that a lower bound on the number of cycles above deck is

$$\max\{\underline{\Lambda}, \underline{Y}\} + \max_h\{\underline{u}_h, \underline{l}_h\}. \tag{11}$$

We also know from Theorem 1, but treating each hatch as a vessel, that a lower bound on the number of cycles below deck is

$$\sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max_{c\in S_h}\{\bar{u}_{hc}, \bar{l}_{hc}\}. \tag{12}$$

The sum of these two expressions is the expression in the theorem statement. Because it must be a lower bound to the total number of cycles, above and below deck, the theorem is proven. □

Clearly, for rows where $\sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max\{\underline{\Lambda}, \underline{Y}\} \gg \max_h\{\underline{u}_h, \underline{l}_h\} + \max_{c\in S_h}\{\bar{u}_{hc}, \bar{l}_{hc}\}$, both the upper and lower bounds are close to the solution provided by the greedy strategy and $\sum_{h\in H}\max\{\bar{u}_h, \bar{l}_h\} + \max\{\underline{\Lambda}, \underline{Y}\}$ provides a reasonable estimate for the number of cycles required to unload and load a row with deck hatches. As with the hatchless case, the gap between the upper and lower bound is quite small, and decreases with the size of the row.

If one double cycles only below deck, as is current practice, the benefits of double cycling will be reduced by roughly the ratio of containers unloaded and loaded above deck to containers unloaded and loaded both above and below deck. Results comparing double cycling without hatches, to double cycling only below deck, are shown in §7.1.

7. Evaluation

This section addresses the magnitude of double-cycling benefits. We present tools to convert benefits from number of cycles to an amount of time, and compare the ensuing results to data collected in a real-world trial of double cycling. We also consider the financial impact of double cycling, and present results that estimate benefits for current and future vessels.

7.1. Evaluation of the Reduction in the Number of Cycles

We evaluate here the reduction in the number of cycles achieved by double cycling as predicted by our formulae, and compare it with the reduction achieved using Johnson's rule and the greedy strategy. To understand the benefits on a larger scale a computer program was used to generate problem

Table 1 Parameter Settings Used to Generate Vessel Data

Parameter	Setting
Number of stacks	Varies (as indicated in the figure)
Beta distribution parameters	5 runs with $p_i = 1$, $q_i = 1$, $p_e = 1$, $q_e = 2$ 5 runs with $p_i = 1$, $q_i = 1$, $p_e = 2$, $q_e = 1$ 10 runs with $p_i = 1$, $q_i = 1$, $p_e = 2$, $q_e = 2$ 5 runs with $p_i = 2$, $q_i = 2$, $p_e = 1$, $q_e = 1$ 10 runs with $p_i = 2$, $q_i = 2$, $p_e = 2$, $q_e = 2$ 5 runs with $p_i = 2$, $q_i = 2$, $p_e = 2$, $q_e = 1$
Maximum number of imports in one stack	20
Maximum number of exports in one stack	20

instances and calculate the number of moves for each algorithm. Comparisons were made for ships without deck hatches and also for hatched ships when double cycling only below deck. The number of containers to unload and load in each stack, u_c and l_c, were determined with independent draws of beta random variables with parameters: (p_i, q_i) for imports and (p_e, q_e) for exports. Because beta random variables have a range between zero and one, each sampled value was multiplied by the maximum stack height and then rounded down to the nearest integer. These values were, in general, different for imports and exports. Some stacks could have zero containers. Parameter settings used to generate the figures are shown in Table 1. The results are shown in Figures 6 and 7. Each data point represents the average value of 40 generated vessels.

As expected, the benefits using the greedy strategy are smaller than the benefits using the optimal strategy, but the difference is small and both are

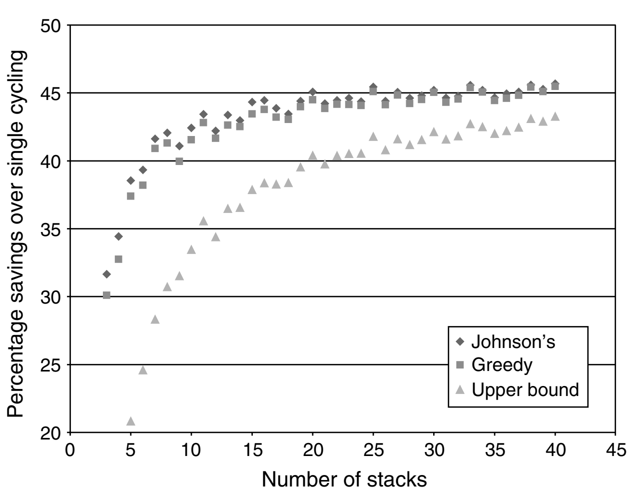

Figure 6 Performance Comparison of Greedy Strategy and Johnson's Rule to Single Cycling for Vessels Without Deck Hatches

Note. Each data point shows the percentage savings over single cycling and is the average result for 40 generated vessels.

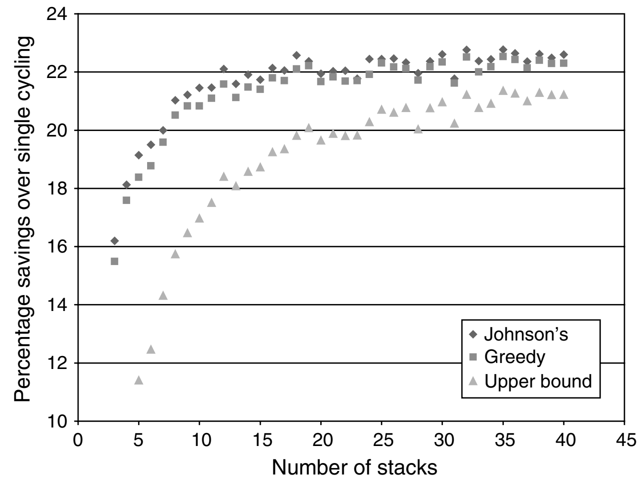

Figure 7 Comparison of the Greedy Strategy, Proximal Strategy, and Johnson's Rule to Single Cycling When Double Cycling Only Below Deck

Note. Each data point shows the percentage savings over single cycling and is the average result for 40 generated vessels.

very close to the (smaller) benefits predicted with the upper bound formula. For a row of 20 stacks, there is a 45% reduction in number of moves over single cycling for the optimal strategy, a 44% reduction using the greedy strategy, and a 40% reduction predicted by the upper bound formula. Notice that the savings range in the figures has been reduced to allow closer comparison of the values, and that benefits above 35% are commonplace.

Figure 7 shows the percentage savings over single cycling predicted for hatched ships with the upper bound formula, and those achieved with Johnson's rule and the greedy algorithm, when double cycling only below deck. Here we assume all containers are removed from atop hatches, all hatch coverings are removed, all containers below deck are unloaded and loaded using a double-cycling algorithm, hatch coverings are replaced, and containers above deck are loaded. Notice again that the scale of the axis has been adjusted for closer comparison of the strategies. For a vessel with 20 stacks per row, the benefits of both the greedy and optimal strategies are close to 22%. A 20% reduction is predicted by the upper bound formula. The results of these experiments indicate that double cycling can reduce the number of cycles significantly, and that the improvement is insensitive to the algorithm used.

7.2. Time Savings

We now examine how the reduction in number of cycles translates into decreased operating time. Whereas the number of cycles required to turn around the vessel is a relevant metric, the real benefit comes from reducing operational time consumed by the

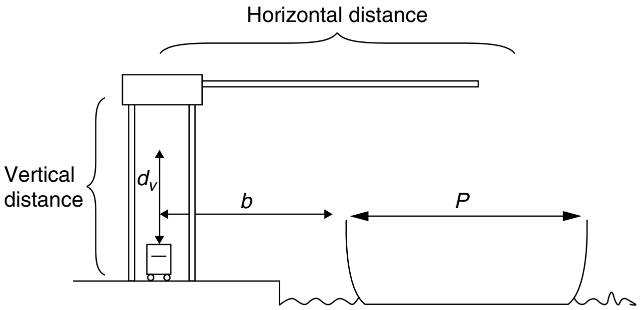

Figure 8 Horizontal and Vertical Motion of the Crane

unloading and loading processes. We will use the following notation. Please refer to Figure 8.

- W—average time saved for each replacement of two single cycles by one double cycle
- S_r—number of cycles required to turn-around row $r \in \{1, \ldots, R\}$ using single cycling
- D_r—number of cycles required to turn-around row $r \in \{1, \ldots, R\}$ using double cycling
- d_r—number of cycles moving two containers while operating on row $r \in \{1, \ldots, R\}$
- V_h—hoist speed of the trolley when not moving a container
- V_l—speed of the crane when moving lengthwise along the vessel
- V_t—horizontal travel speed of the trolley when not moving a container
- d_V—vertical distance from the apron to the maximum height a container can reach
- d_L—lateral distance between two rows of the vessel
- b—horizontal distance from landside vehicle to the landside edge of the vessel
- P—width of the vessel
- T_r—time required to position landside vehicle after departure of previous vehicle

Consider the time taken by the same two containers with single and double cycling. For each double cycle we save some empty-crane travel relative to the two corresponding single cycles, but we also experience a slight landside-repositioning penalty. The time penalty, T_r, is incurred because after dropping a container for unloading onto a landside vehicle, the crane must wait for a container for loading to be positioned below the crane. With single cycling, this can be done simultaneously with other crane operations.

The total distance travelled by the crane is reduced by one complete empty cycle (without moving a container) between the apron and the position above either the container to load or the container to unload, whichever is closer. Therefore, the average time saved by a double cycle, W, satisfies

$$2\left[\max\left(\frac{d_V}{V_h}, \frac{b}{V_t}\right) + \frac{(1/3)P}{V_t}\right] - T_r < W < 2\left[\frac{d_V}{V_h} + \frac{b}{V_t} + \frac{(1/2)P}{V_t}\right] - T_r, \qquad (13)$$

where the quantities in brackets are low and high estimates of the time for a one-way (empty) move. One reason for the interval estimation is that an unspecified amount of horizontal and vertical motion may take place simultaneously. For the lower bound we assume all horizontal and vertical motions are carried out simultaneously. For the upper bound we assume they are carried out separately. We also consider a range for the horizontal distance saved. The one-way savings is the average distance between the apron and the closer of the two containers. This is optimistically considered to be $(b + \frac{1}{2}P)$ for the upper bound (as if the two containers were always next to each other), and pessimistically assumed to be $(b + \frac{1}{3}P)$ for the lower bound (assuming a uniform distribution of locations). The average time saved by double cycling on a single row is the product of W and the number of double cycles: $d_r = S_r - D_r$.

After completing loading and unloading operations on a row, the crane moves laterally along the vessel to the next row. If there are R rows on a vessel, the time consumed with lateral motion is $2(R-1)d_L/V_l$ if the vessel is unloaded completely before any containers are loaded (which is common practice), and $(R-1)d_L/V_l$, exactly half, if the crane double cycles each row before moving on to the next row. We can now compare the results of this analysis to empirical data, collected during a double-cycling trial at the Port of Tacoma, Washington.

7.3. Validation

In June 2003, the Center for the Commercial Deployment of Transportation Technologies, Transystems, the Port of Tacoma, and Washington United Terminals worked together on a full-scale demonstration of the efficient marine terminal concept. Double cycling of container cranes is a key element of this concept. On June 28th, one bay of a Hanjin vessel was loaded and unloaded simultaneously using double cycling (TranSystems Corporation 2003). Double cycling occurred below deck only. During this trial the adjusted average time for a single cycle was 1 minute 45 seconds, and for a double cycle it was 2 minutes 50 seconds. Thus, double cycling saved 40 seconds per pair of containers that were double cycled. We now compare the difference in these empirical cycle times to the differences obtained using the expressions developed above. Parameter values used for the time savings analysis (based on the trial at Tacoma) are given in Table 2 (Garcia 2003–2005).

Table 2 Parameter Values for Evaluation, Based on Port of Tacoma Trial

Parameter	Value
V_h	300 feet per minute
V_t	500 feet per minute
P	130 feet
d_V	75 feet
b	60 feet
T_r	15 seconds

Source. Ward (2003–2005) and TranSystems Corporation (2003).

The lower bound to the time saved is 25.4 seconds. The upper bound to the time saved is 45 seconds. The empirical difference was 40 seconds. We expect the empirical difference to be closer to the upper bound, because, in practice, a significant amount of the vertical trolley travel and horizontal trolley travel takes place separately.

Clearly, the specific results depend on the parameters of each crane, vessel, and container arrangement, but we have demonstrated in this case that our formula matches empirical data. With the parameters of Table 2 a 21% reduction in the number of cycles decreases operating time by approximately 8%, and a 35% reduction would decrease it by 13%.

We have formulated the problem of minimizing the number of cycles required to turn around the vessel. From these results, we have developed a method for converting benefits from number of cycles to an amount of time, assuming constant cycle times for single and double cycling. Cycle times turn out to be nearly constant (insensitive to the position of the stacks), so our approach approximately minimizes the operational time.[1]

7.4. Economic Impact

In this section, we provide simple an estimate of the economic benefit of double cycling. This estimate is based on data from a specific West Coast terminal. Ports vary distinctly in their ownership and fee structures, so these results may not be relevant for all terminals. We assume that freed resources can be usefully employed. In practical terms, this means that the released capacity can be used to move additional containers. This assumption reflects current market conditions, where demand is expected to exceed capacity during the peak season.

We consider the economic impact of a 10% reduction in operating time, because this is typical for a medium-size vessel when double cycling below deck. We assume a vessel, capable of carrying 6,000 TEUs, unloads and loads 1,500 containers in 50 hours using single cycling and in 45 hours using double cycling. We compare the main benefits in dollars per container moved. Detailed assumptions and data sources can be found in Goodchild (2005). The results are shown in Table 3.

The value of the double-cycling benefits are significant, but the beneficiaries include parties who are not responsible for its implementation. If a larger portion of the benefits were experienced by those responsible for its implementation, we might see more widespread use of the technique.

Although double cycling will not eliminate current port congestion, it can be implemented quickly and, in conjunction with other measures, can ease congestion before more long term infrastructure projects come on line. Any amount of double cycling will reduce the number of cycles required to turn around the vessel. The next step in this research is to understand how other port resources, such as landside equipment, gate time, and rail capacity are affected by double cycling. For example, double cycling while unloading and loading the vessel creates an opportunity to double cycle landside equipment. Chassis used to deliver containers to the apron can then carry an unloaded container to local storage. Typically, these chassis return to the local storage empty. In an effort to understand why the implementation of double cycling has been so slow, we should also look at the economic costs and benefits of double cycling with a more systematic approach, to understand how different parties may be encouraged to work together to implement double cycling.

Appendix

The double-cycling problem can be formulated as the two-machine flow shop problem below. We use the following notation:

- u_c—number of containers to unload in stack $c \in S$
- l_c—number of containers to load in stack $c \in S$
- FU_c—completion time of unloading $c \in S$
- FL_c—completion time of loading $c \in S$
- w—maximum completion time
- X_{kj}—binary variable to for ordering of unloading jobs (1 if $j \in S$ is unloaded after $k \in S$ and 0 otherwise)
- Y_{kj}—binary variable to for ordering of loading jobs (1 if $j \in S$ is loaded after $k \in S$ and 0 otherwise)
- M—a large number

The scheduling problem (SP) is to minimize the maximum completion time of all jobs subject to constraints. The result is to uniquely identify the permutations Π and Π',

[1] If cycle times depended heavily on the position of the stacks, benefits could arise from strategies that would handle together proximal stacks. The optimization problem becomes more complex and does not lead to simple formulae. These strategies have been examined in Goodchild (2005), but their main benefit appears to be implementation simplicity.

Table 3 Comparison of the Approximate Economic Benefits of Double Cycling

Resource	Savings ($/container moved)	Extra cost ($/container moved)	Beneficiary	Labor involved (effect)
Vessel	$40.00	$13.00	Vessel operator	Ship crew (spend less time in port)
Crane	$22.00	$0.00	Terminal operator	Crane operator and Stevedores (more stress due to less time with each container but can be compensated with better pay)
Berth	$22.00	$0.00	Port authority	Landside labor force (work changes little in character but becomes more productive and could be compensated with better pay)
Total	$85.60	$13.00		

Note. Based on data from a specific West Coast terminal.

and a feasible set of job start and end times. It is assumed that the process starts at time zero. The formulation is

$$\text{(SP)} \quad \text{minimize} \quad w \tag{14a}$$

$$\text{subject to} \quad w \geq FL_c \quad \forall c \in S, \tag{14b}$$

$$FL_c - FU_c \geq l_c \quad \forall c \in S, \tag{14c}$$

$$FU_k - FU_j + MX_{kj} \geq u_k \quad \forall j, k \in S, \tag{14d}$$

$$FU_j - FU_k + M(1 - X_{kj}) \geq u_j \quad \forall j, k \in S, \tag{14e}$$

$$FL_k - FL_j + MY_{kj} \geq l_k \quad \forall j, k \in S, \tag{14f}$$

$$FL_j - FL_k + M(1 - Y_{kj}) \geq l_j \quad \forall j, k \in S, \tag{14g}$$

$$FU_c \geq u_c \quad \forall c \in S, \tag{14h}$$

$$X_{kj}, Y_{kj} = 1, 0 \quad \forall j, k \in S. \tag{14i}$$

These constraints completely define the double-cycling problem. Constraints (14b) ensure that the makespan is greater than or equal to the completion of loading of all stacks. Constraints (14c) ensure that stacks are only loaded after all necessary stacks have been unloaded. Constraints (14d), (14e), and (14i) ensure that every stack is unloaded after the previous one in Π' has been unloaded. This is achieved by specifying for every pair of stacks (j, k) that either stack k is unloaded before stack j (if $X_{kj} = 1$) or the reverse (if $X_{kj} = 0$), and that the time difference between the two events is large enough to unload the second of the two stacks. Constraints (14f), (14g), and (14i) are equivalent to (14d), (14e), and (14i) but for loading jobs. Constraints (14h) ensure that each unloading completion time allows for enough time to at least unload that stack.

References

Bendall, H. B., A. F. Stent. 1996. Hatchcoverless container ships: Productivity gains from a new technology. *Maritime Policy Management* **23**(2) 187–199.

Castilho, B. D., C. F. Daganzo. 1993. Handling strategies for import containers at marine terminals. *Trans. Res. Part B* **27** 151–166.

Christiansen, M., K. Fagerholt, B. Nygreen, D. Ronen. 2005. Maritime transportation. C. Barnhart, G. Laporte, eds. *Transportation Handbooks in Operations Research and Management Science*. North-Holland, Amsterdam, Netherlands.

Crainic, T. G., K. H. Kim. 2005. Intermodal transportation. C. Barnhart, G. Laporte, eds. *Transportation Handbooks in Operations Research and Management Science*. North-Holland, Amsterdam, Netherlands.

Daganzo, C. F. 1989. The crane scheduling probem. *Trans. Res. Part B* **23** 159–175.

Daganzo, C. F. 1990. The productivity of multipurpose seaport terminals. *Trans. Sci.* **24**(3) 205–216.

Garcia, B. 2003–2005. Personal communications. Transystems Corportation.

Goodchild, A. V. 2005. Crane double cycling in container ports: Algorithms, evaluation, and planning. Ph.D. thesis, University of California at Berkeley, Department of Civil and Environmental Engineering, Berkeley, CA.

Imai, A., E. Nishimura, S. Papadimitriou. 2001. The dynamic berth allocation problem for a container port. *Trans. Res. Part B* **35** 401–417.

Johnson, S. M. 1954. Optimal two- and three-stage production schedules with setup times included. *Naval Res. Logist. Quart.* **1** 61–68.

Kim, K. H., J. W. Bae. 1999. A dispatching method for automated guided vehicles to minimize delay of containership operations. *Internat. J. Managment Sci.* **5** 1–25.

Kim, K. H., H. B. Kim. 2002. The optimal sizing of the storage space and handling facilities for import containers. *Trans. Res. Part B* **36** 821–835.

Kim, K. H., J. S. Kang, K. R. Ryu. 2004. A beam search algorithm for the load sequencing of outbound containers in port container terminals. *Oper. Res. Spectrum* **26**(10) 93–116.

Lai, K. K., J. Leung. 2000. Analysis of gate house operations in a container terminal. *Internat. J. Modeling Simulation* **20** 89–94.

Mongelluzzo, B. 2005a. L.A. won't see peak repeat. *J. Commerce* **6**(24) 12.

Mongelluzzo, B. 2005b. Ports rush to keep up. *J. Commerce* **6** 26–30.

Park, Y. M., K. H. Kim. 2003. A scheduling method for berth and quay cranes. *Oper. Res. Spectrum* **25** 1–23.

Peterofsky, R. I., C. F. Daganzo. 1990. A branch and bound solution method for the crane scheduling problem. *Trans. Res. Part B* **24** 159–172.

Schonfeld, P., O. Sharafeldien. 1985. Optimal berth and crane combinations. *J. Waterway, Port, Coastal Ocean Engrg.* **111** 1060–1072.

Taleb-Ibrahimi, M., B. Castilho, C. F. Daganzo. 1993. Storage space vs. handling work in container terminals. *Trans. Res. Part B* **27** 13–32.

TranSystems Corporation. 2003. Efficient marine terminal full scale demonstration. http://www.ccdott.org.

Vis, I. F. A., R. de Koster, K. J. Roodbergen. 2001. Determination of the number of automated guided vehicles required at a semi-automated container terminal. *J. Oper. Res. Soc.* **52** 409–417.

Ward, T. 2003–2005. Personal communications. JWD Group.