2010

# Dynamic Quay Crane Allocation

Yan ZHANG
*Singapore Management University*, yan.zhang.2008@smu.edu.sg

# DYNAMIC QUAY CRANE ALLOCATION

ZHANG YAN

SINGAPORE MANAGEMENT UNIVERSITY
2010

# Dynamic Quay Crane Allocation

by
ZHANG Yan

Submitted to Lee Kong Chian School of Business
in partial fulfillment of the requirements for the Degree of
Master of Science in Operations Management

Supervisor: Prof LIM Yun Fong

Singapore Management University
2010

**Abstract**

Dynamic Quay Crane Allocation

by

ZHANG Yan

We introduce simple rules for quay cranes to handle containers along a berth where vessels arrive continuously in time. We first analyze a model where workload is continuous. Our analysis shows that if the system is configured properly, it will always converge to a state with the maximum possible throughput regardless of external disruptions or changes in workload. Numerical simulations based on a discrete workload model suggest that, by following the same rules, the system can still converge to state with throughput that is very close to its upper bound.

Key words: quay crane scheduling; container terminals; bucket brigades; self-organizing systems

# Contents

**Acknowledgements**

I am grateful that I will be graduating as a master student from Singapore Management University. The past two years time in Singapore greatly enriched my knowledge and my life.

I would like to thank all professors from Operations Management Program in Lee Kong Chian School of Business, who have guided and instruct me during courses and seminars. I would never forget what I have learnt from them. Not only knowledge but also how to do things correct.

Particularly I would like to convey my sincere thanks to Professor LIM Yun Fong; I have learnt much from working with him during my thesis writing processes. It would be impossible for me to finish my thesis without his support and guidance. I also would like to thank Dr. Brian Rodrigues and Dr. Moosa Sharafali as my thesis committee members who provided many valuable advices for my thesis.

I am also indebted to those classmates and friends who helped me during my master study period. There are disappointing, struggling but there are also happiness with them. Congratulations to all of us.

Finally, I would like to thank my family who are always supportive and encourage warmly me to move forward.

# 1 Introduction

As globalization shapes the world rapidly, deep-sea maritime transportation becomes increasingly important as a key component in the global supply chain. The throughput of world container ports grew by 4% to reach 506.9 million TEUs (Twenty-foot Equivalent Units) in 2008 (UNCTAD 2009). The top three busiest container ports in the world: Singapore, Shanghai, and Hong Kong handle 29.9, 28.0, and 24.2 million TEUs, respectively, in 2008. These ports not only compete with each other, but also with new ports from emerging economies with significantly lower operations costs. As a result, container ports are keen to improve their productivity by introducing new methodologies to their operations to increase their competitiveness.

Typically, each container port consists of several container terminals. For example, the port of Singapore contains four container terminals. The number of berths in each terminal ranges from 8 to 23. Within a container terminal there are two types of container movements: import and export. Figure 1 shows that after a vessel reaches a container terminal, quay cranes unload the import containers onto internal trucks, which carry them to their assigned storage locations at the yard. These containers are then unloaded to the yard by yard cranes. On the other hand, export containers arrive at the terminal by external trucks. These containers are stored in the yard until the vessels that transport them to the next destinations arrive. The export containers are then loaded by yard cranes onto internal trucks that carry them to a berth. At the berth, the export containers are loaded to the corresponding vessels by quay cranes.

Quay cranes are standard equipment for handling containers at the berth. They are the most expensive equipment in a container terminal. Since ports are constrained by limited budget on new equipment, it is crucial to operate the quay cranes

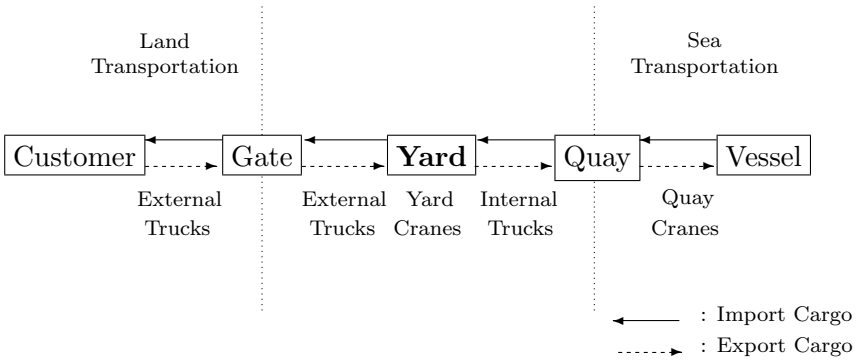



Figure 1: **Typical operations of a container terminal.** This figure illustrates typical import and export processes in a container terminal.

efficiently. The operations of quay cranes are subject to two constraints: (1) The *non-crossing constraint* requires the cranes to maintain in a fixed sequence. This is because the cranes along the same berth share a common rail. (2) The *minimum separation constraint* requires the cranes to keep a minimum distance from each other due to safety reasons.

We assume a vessel is already assigned a berthing position when it arrives at a container terminal. The vessel is moored at the assigned position and waits to be served by quay cranes. Given limited space and resources, the container terminal can only serve a finite number of vessels at a time. For ports that are congested, it is generally very expensive for vessels to stay at their berthing positions. For example, for a moderately large vessel with length 1,089 feet, the daily charges for berthing at the ports of Hong Kong and Houston can be as high as US$17,637 and US$10,380 respectively. Thus, it is important to fully utilize the quay cranes and maximize the number of containers handled per unit time. In fact, the turnover time per vessel is one of the most important measures of a port's service quality. In the *dynamic quay-crane allocation problem*, a fixed number of quay cranes at a berth with limited length serve a series of vessels that arrive sequentially at the berth over time. The objective is to maximize the long-run average throughput (number of containers handled per

unit time) subject to the non-crossing and minimum separation constraints.

The dynamic quay-crane allocation problem is further complicated by uncertainty in the terminal operations even if the arrival time of each vessel is given. Sources of uncertainty include weather conditions, break down of equipment, emergency halting due to safety reasons, mislabeled or lost containers and traffic congestion within the terminal.

Most papers in the literature solve the quay-crane allocation problem in a static setting in which a given number of vessels at a berth are pre-assigned to a set of quay cranes during a planning horizon. The problem is typically formulated as a machine scheduling problem (see, for example, Daganzo(1989) and Lim et al. (2007)). If any system parameter changes (for example, a vessel's arrival is delayed), the problem needs to be resolved. Apart from computational complexity, the dynamic nature of the problem causes these static methods not optimal and tedious to implement over an extended period of time. As a result, we need a new approach to handle the dynamic nature of the quay-crane allocation problem.

Inspired by the ideas of bucket brigades for dynamic assembly line balancing (Bartholdi and Eisenstein 1996a), we introduce simple rules for quay cranes to share work so that the long-run average throughput of a berth is maximized. Bucket brigades are a way to coordinate workers on an assembly line to maximize the production efficiency. To form a bucket brigade, each worker follows a simple rule: Continue to assemble your item (an instance of the product) along the line until either your colleague downstream takes over it or you complete it if you are the last worker of the line; then you walk back to get more work, either from your colleague upstream or from a buffer at the start of the line if you are the first worker.

If workers are sequenced from slowest to fastest according to their processing

rates in the direction of production flow, then a bucket brigade will *self-balance* such that every worker repeatedly covers an appropriate portion of work content on the line and the system's throughput is maximized (Bartholdi and Eisenstein 1996a). Bucket brigades are widely adopted to coordinate workers in order-picking in distribution centers (Bartholdi and Eisenstein 1996b and Bartholdi et al. 2001). They are also used in manufacturing for producing garments, packaging cellular phones, and assembling tractors, large-screen televisions, and automotive electrical harnesses (Bartholdi and Eisenstein 1996a, b, Bartholdi and Eisenstein 2005, and Villalobos et al. 1999a, b).

Bucket brigades are effective in coordinating workers on assembly lines for the following reasons: (1) The rule is simple and it is easy to implement. (2) Since they can self-balance, one needs neither a work-content model nor computation for work balance, which are required by any static work-allocation strategy. (3) Workers dynamically and constantly balance their work and so, the system can restore balance from temporary disruptions and is adaptive to changes in work content.

To effectively coordinate quay cranes at a berth, one needs to control the processing rate (number of containers handled per unit time) of each crane. After a vessel arrives at the berth, import containers are unloaded from the vessel while export containers are loaded to the vessel. In practice, the bottleneck of these unloading and loading processes lies on the internal trucks that transport these containers between the berth and the yard. Due to traffic conditions in the yard, the quay cranes often wait for the internal trucks to arrive. By allocating the internal trucks to the quay cranes, we can control the processing rates of the cranes. By adopting the ideas of bucket brigades in the quay-crane operations, the cranes can share their work in an efficient manner, without complex computation. Even with disruptions, the system

can constantly attain its maximum throughput as it self-balances.

After we review the related literature, we propose a new methodology for the dynamic quay-crane allocation problem based on a model with continuous workload. We analyze the dynamics and determine the long-run average throughput of the system. We then describe a model with discrete workload that is suitable for container terminals and evaluate the performance of our method based on this model through numerical simulations.

# 2 Literature review

Daganzo (1989) is the first to address the quay-crane scheduling problem. The author assumes the crane movement time is negligible compare to the time to handle containers. A ship of containers are separated into holds that are considered as jobs. To maximize cost savings, the author formulates the static version of the quay-crane scheduling problem as a mixed-integer program. He also develops several heuristic principles for dynamic scheduling in reality. His model does not consider the non-crossing and minimum separation constraints. Only small-scale problem instances are solved in his numerical experiments.

Peterkofsky and Daganzo (1990) decompose the static quay-crane scheduling problem into two parts: departure scheduling and crane allocation. The authors formulate the problem as a maximum flow problem and solve it by a branch-and-bound method. Their algorithm is able to solve larger problem instances than Daganzo (1989).

Kim and Park (2004) solve the quay-crane scheduling problem as an m-parallel machine scheduling problem. They consider the non-crossing constraint and intro-

duce a branch-and-bound method to solve the problem. The computational time of their method increases rapidly with problem size. They also propose a greedy randomized adaptive search procedure to reduce the computational time.

Lim et al. (2004) solve the quay-crane scheduling problem to maximize throughput using dynamic programming. They consider the non-crossing constraint, the minimum separation constraint, and the job-separation constraint. The authors assume each crane can only process one job and each job can only be served by one crane. Numerical experiments suggest that the squeaky wheel optimization with local search performs the best among various heuristics. The authors claim that their solution can be recalculated quickly to handle dynamic situations due to changes such as early completion of a vessel's jobs.

Zhu and Lim (2006) consider the quay-crane scheduling problem with an objective to minimize the latest completion time. The authors assume that the crane processing rates are constant, all jobs are non-preemptive, and cranes cannot cross each other. They formulate the problem as an integer program. They first solve it using CPLEX and then improve the quality of a solution using a branch-and-bound algorithm. The authors also propose a simulated annealing algorithm to obtain near optimal solutions.

Lim et al. (2007) consider the quay-crane scheduling problem as an m-parallel machine scheduling problem and propose several heuristics to solve it. They consider the non-crossing constraint and assume non-preemptive jobs. The authors decompose the problem into two parts: (1) crane allocation to jobs, and (2) time allocation for each crane. They prove that given a crane allocation the optimum time allocation for each crane can be found easily. As a result, they focus on finding the optimal crane allocation. Numerical results suggest that a simulated annealing algorithm performs

significantly better than other methods in terms of computational time.

Lee et al. (2008) consider the quay-crane scheduling problem with the non-crossing constraint. They divide a vessel into holds and assume each crane can only work on one hold at a time. They ignore the travel time between holds. The authors formulate the problem as a mixed-integer program and solve it using a genetic algorithm.

Liu et al. (2006) study the dynamic quay-crane scheduling problem with the non-crossing and minimum separation constraints. To minimize the maximum relative tardiness of vessel departures, the authors formulate a mixed-integer linear programming model. Since the computational time increases rapidly with problem size, they decompose the problem into two parts: the vessel-level problem and the berth-level problem. They improve the results by allowing cranes to share jobs. They also shorten the computation to a reasonable amount of time by using heuristics.

Table 1 summarizes the quay-crane scheduling literature. Most authors assume jobs are non-preemptive. Note that only three papers consider the minimum separation constraint. Bierwirth and Meisel (2009) give an excellent survey on the seaside operations planning of container terminals. The authors divide the problem into three parts: the berth allocation problem, the quay-crane assignment problem, and the quay-crane scheduling problem. They describe the literature of each problem in detail.

Bartholdi and Eisenstein (1996a) provide the first theoretical analysis on bucket brigade assembly lines. They assume deterministic work content and each worker has a finite, constant work velocity and an infinite walk-back velocity. Workers cannot overtake each other and so they remain in a fixed sequence along the line. The authors prove that if workers are sequenced from slowest to fastest according to their

Table 1: **Summary of quay-crane scheduling literature.**

| **Paper** | **Objective** | **Method** | **Preemptive** | **Non-crossing** | **Separation** |
|---|---|---|---|---|---|
| Daganzo(1989) | Max cost savings | MIP | Y* | N | N |
| Peterkofsky & Daganzo (1990) | Min delay costs | Brand & Bound | Y | N | N |
| Kim & Park (2004) | Min the latest completion time | greedy randomized adaptive search procedure | N | Y | Y |
| Lim et al. (2004) | Max throughput | squeaky wheel optimization | N | Y | Y |
| Zhu & Lim (2006) | Min the latest completion time | simulated annealing | N | Y | N |
| Lim et al. (2007) | Min the latest completion time | simulated annealing | N | Y | N |
| Lee et al. (2008) | Min the latest completion time | genetic algoritm | N | Y | N |
| Liu et al. (2006) | Min maximum relative tardiness of vessel departures | MILP decomposition | Y | Y | Y |

* Y: Yes, N: No

work velocities in the direction of production flow, the hand-off points between any two neighboring workers will converge to a *fixed point* (see, for example, Alligood 1996). Furthermore, if the work content is uniformly and continuously distributed on the line, then the system's throughput attains the maximum possible value.

Bartholdi et al. (1999) describe all possible dynamics of two- and three-worker bucket brigades with workers not necessarily sequenced from slowest to fastest. Bartholdi et al. (2001) investigate the behavior of bucket brigades when the work content is stochastic. Bartholdi and Eisenstein (2005) consider each worker spend a constant walk-back time and a constant hand-off time to get work from his colleague. Bartholdi et al. (2009) assume each worker has a constant walk-back velocity and they are allowed to overtake or pass each other. The authors show that the system may behave chaotically if it is not configured properly. Lim and Yang (2009) study policies that

maximize the throughput of bucket brigades on discrete work stations.

# 3 A continuous model

Consider a berth that serves a sequence of vessels shown in Figure 2. Cargos on each vessel are partitioned into *bays* along the vessel's longitudinal axis. Each bay stretches across the width of the vessel. After all the import cargos are unloaded from a bay, export cargos are then loaded to the same bay. We assume both import and export cargos of each bay are predetermined.

Define a *job* as the unloading and loading operations for a set of adjacent bays on the vessels. A job may comprise bays from different vessels that are adjacent to each other along the berth. We assume the workload of each job is a constant that we normalize to 1 and it is continuously and uniformly distributed on its bays. We also assume each job is preemptive so that a crane can take over another crane's job. Although this workload model is more appropriate for bulk cargos (such as coal, minerals, grains, and dry chemicals), it serves as a continuous approximation of containers. The long-run average throughput derived from this model serves as an upper bound of the throughput of a more realistic, discrete workload model discussed in the next section.

Define the direction from left to right along the berth as *forward* and the reverse direction as *backward*. Each job is processed in the forward direction from its left-most bay to its right-most bay. All the unloading and loading operations for each bay are done before the next bay is processed. A vessel leaves the berth after all its jobs are processed and a new vessel is assigned to the corresponding berth position. We assume the port is sufficiently congested that such a new vessel is always available.

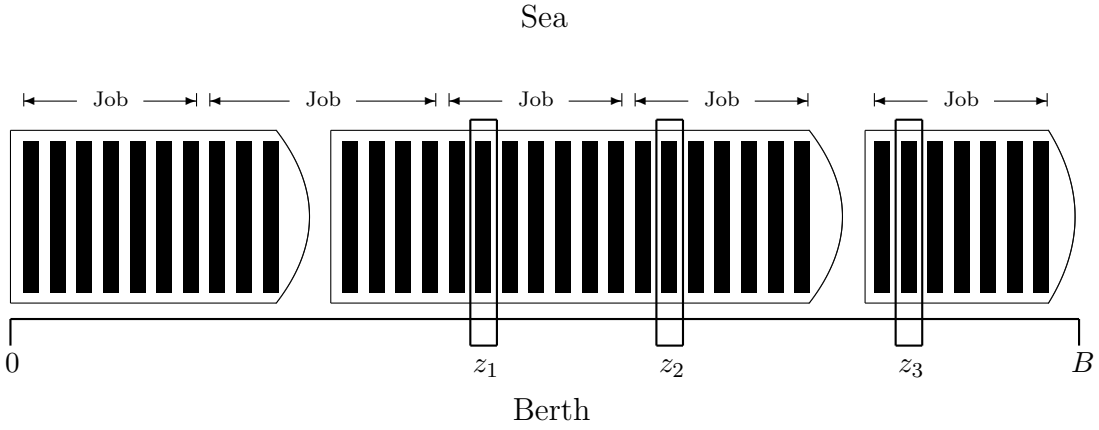



Figure 2: **Jobs on vessels.** A job is defined as the unloading and loading operations for a set of adjacent bays on the vessels. Note that a job may comprise bays from different vessels. The location of quay crane $i$ is denoted as $z_i$.

This assumption is reasonable for those congested ports such as the port of Los Angeles, which constantly faces a long queue of vessels waiting to be served.

Quay cranes are sequenced along the berth. Since cranes of the same berth move on a common rail track, they cannot pass each other and therefore maintain a fixed ordering along the berth. We first index the quay cranes from 1 to $n$ in the forward direction. Due to the allocation method that we describe later, we may permute quay cranes' indices if it is necessary. Cranes $i-1$ and $i+1$ are called the *predecessor* and *successor*, respectively, of crane $i$.

Define the left-most point of the berth as the origin. Let $B$ be the length of the berth and $z_i$ be the location of crane $i$ along the berth. Both $B$ and $z_i$ can be expressed in number of bays. Due to safety reasons, the quay cranes cannot get too close to each other. Each crane must maintain a minimum separation distance $d$ from its immediate neighbors. As a result, the condition $z_j \geq z_i + d$ must hold if crane $j$ is on the right of crane $i$ along the berth.

We assume each job can be processed by at most one crane at a time. Each

crane $i$ processes jobs with a finite, constant rate $v_i$. Through our conversations with port executives, this rate is mainly determined by the number of internal trucks that support the crane. This is because the time to unload or load a container is generally smaller than the time to wait for an internal truck as the trucks are often stuck in traffic. Thus, to increase the processing rate of a crane, one can assign more internal trucks to the crane. We assume the workload of each job is sufficiently large that the time for a crane to move from one job to another is negligible compared to the time to process a job. This assumption can be satisfied by assigning sufficient bays to a job.

We initialize the system by assigning $n$ jobs from the right-most end of the berth to the $n$ cranes such that the $i$-th job from the right-most end of the berth is assigned to crane $n+1-i$. All cranes process their jobs by serving one bay after another in the forward direction. Each crane $i$ continues its job until crane $n$ completes a job and the system *resets* itself: Crane $n$ moves backward to take over work from crane $n-1$, which in turn moves backward to take over work from crane $n-2$ and so on, until crane 1 moves backward and initiates a new job. If crane $i < n$ completes a job, then it remains idle until its successor takes over its job.

We say the system is in a *normal state* when the cranes are sequenced from 1 to $n$ in the forward direction and crane 1 is not processing the left-most job of the berth. Each crane $i$ independently follows the rules below in the unloading and loading operations when the system is in a normal state:

**Work forward:** Continue to process your job in the forward direction until

1. your successor takes over your job, then **move backward**; or
2. you complete your job. If you are crane $n$, then **move backward**. Otherwise, **wait**.

**Move backward:** Take over work from your predecessor or initiate a new job if you are crane 1, then **work forward**.

**Wait:** Remain idle until your successor takes over your job, then **move backward**.

The waiting rule has the potential of wasting crane capacity because it requires cranes to remain idle. However, as we will see from our analysis below, the waiting rule will never be invoked in the long run if the system is properly configured.

Note that when crane 1 reaches the left-most job of the berth, it can no longer move backward to initiate a new job. We can resolve this issue by permuting the crane indices. Define a *permutation function* $\sigma(i)$ as

$$\sigma(i) \quad = \quad (i \bmod n) + 1. \tag{1}$$

The function $\sigma(i)$ performs a cyclic permutation on the crane indices. For example, if $n = 3$, $\sigma(1) = 2$, $\sigma(2) = 3$, and $\sigma(3) = 1$.

The system resets itself in a different way whenever crane $n$ finishes a job and there is a crane at the left-most job of the berth (either the crane is processing or it has completed the left-most job): The system permutes the crane indices by applying the function $\sigma(i)$ to them and crane 1 moves forward to initiate a new job on the right of the berth immediately after the permutation.

We illustrate this type of resets using an example in Figure 3. When crane 3 completes a job while crane 1 is processing or has completed the left-most job of the berth, the system resets itself by first applying the function $\sigma(i)$ on all the crane indices. As a result, crane 1 becomes crane 2, crane 2 becomes crane 3, and crane 3 becomes crane 1. Since crane 1 is now the right-most crane of the system after the permutation and its job is already completed, it immediately moves forward to initiate a new job at the right end of the berth. Note that we assume such a new job is always available after each permutation. The processing rates of the cranes are adjusted according to their new indices after each permutation.

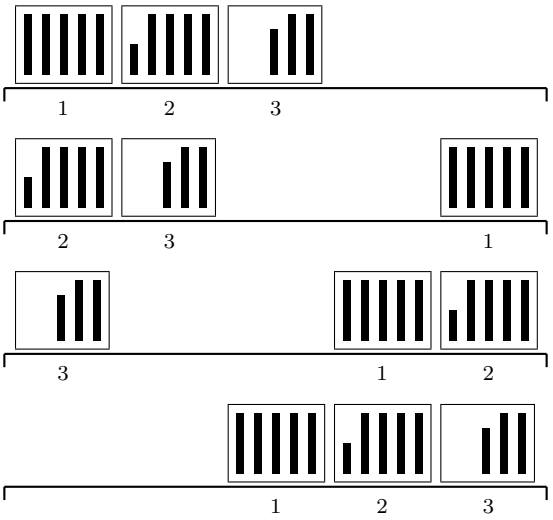



Figure 3: **Permuting crane indices.** This figure shows the indices of cranes after each permutation for a system with three cranes. The columns in each job represent its remaining workload schematically.

Similarly, when crane 3 completes a job while crane 2 is processing or has completed the left-most job of the berth, the system resets itself by applying the function $\sigma(i)$ on all the crane indices. Immediately after the permutation, crane 1 moves forward to initiate a new job immediately on the left of crane 2, which is now on the right of the berth. The system continues to permute the crane indices whenever crane $n$ finishes a job until the cranes are sequenced from 1 to $n$ in the forward direction again. Figure 3 shows a series of resets before the system restores to a normal state.

Since cranes process jobs in the forward direction along the berth and they must maintain a minimum separation distance from their immediate neighbors, a crane $i$ is *blocked* when it catches up with an immediate neighbor crane $j$ on the right. In this case, crane $i$ remains idle until crane $j$ finishes processing its current bay and both cranes proceed simultaneously to their next respective bays. Since crane 1 may be on the right of crane $n$ after a permutation (see Figure 3), we assume the berth

is sufficiently long that crane $n$ will never be blocked by crane 1 in such a situation. As a result, crane $n$ is never blocked and crane $i < n$ can only be blocked by its successor.

The following lemma shows that if cranes are indexed from slowest to fastest according to their processing rates, then they will not be blocked in the long run.

**Lemma 1.** *If $v_1 \leq v_2 \leq \cdots \leq v_n$, then blocking is transient.*

*Proof.* See Appendix A.1. □

According to the rules followed by the cranes, a crane remains idle if it completes a job before its successor takes over the job. We will show that this waste of capacity ceases after a transient period. Let $x_i$ denote the fraction of workload completed on the job of crane $i$. We say crane $i$ *overtakes* its successor if $x_i > x_{i+1}$. Lemma 2 shows that if cranes are indexed from slowest to fastest according to their processing rates, then overtaking is transient.

**Lemma 2.** *If $v_1 \leq v_2 \leq \cdots \leq v_n$, then overtaking will cease after $n-1$ resets.*

*Proof.* See Appendix A.2. □

Lemmas 1 and 2 imply that after a transient period all cranes are busy and there will be no blocking or overtaking.

To analyze the dynamics of the system, we trace the fraction of workload completed on each crane's job immediately before each reset. Let $\mathbf{x}^t = (x_1^t, x_2^t, \ldots, x_n^t)$ where $x_i^t \in [0,1], i = 1, \ldots, n$, denote the fraction of workload completed on the job of crane $i$ immediately before reset $t$. Since the system resets itself whenever crane $n$ finishes a job, we have $x_n^t = 1$ for all $t$. For convenience, we define $x_0^t = 0$ for all $t$. Let $f$ be a function defined implicitly by the rules followed by the cranes such that

$\mathbf{x}^{t+1} = f(\mathbf{x}^t)$. We say $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is a fixed point if $\mathbf{x}^* = f(\mathbf{x}^*)$. Lemma 3 identifies a fixed point for the quay-crane system.

**Lemma 3.** *There exists a unique fixed point* $\mathbf{x}^*$ *for the quay-crane system, where*

$$x_i^* = \frac{\sum_{j=1}^{i} v_j}{\sum_{j=1}^{n} v_j}, \qquad i = 1, \ldots, n.$$

*Proof.* See Appendix A.3. □

Lemma 4 determines the average throughput of the quay-crane system on its fixed point.

**Lemma 4.** *The average throughput of the quay-crane system on its fixed point* $\mathbf{x}^*$ *is*

$$\rho \;=\; \sum_{j=1}^{n} v_j.$$

*Proof.* See Appendix A.4. □

Theorem 1 shows that if the cranes are indexed from slowest to fastest according to their processing rates, then the system will converge to the fixed point $\mathbf{x}^*$.

**Theorem 1.** *The quay-crane system converges to the fixed point* $\mathbf{x}^*$ *if* $v_1 \leq v_2 \leq \cdots \leq v_{n-1} < v_n$.

*Proof.* See Appendix A.5. □

Note that the last inequality in Theorem 1 must be strict to ensure convergence. Lemma 4 shows that the system attains its maximum possible throughput on the fixed point $\mathbf{x}^*$. Theorem 1 shows that if the cranes are indexed from slowest to fastest then the system will attain its maximum possible throughput.

# 4 A discrete model

The continuos workload model discussed above serves as a continuous approximation to the cargos in container terminals. In practice, each job in a container terminal comprises unloading and loading operations of discrete containers. Figure 4 shows a vessel where containers are stored in 12 bays along its longitudinal axis, in 8 *lanes* across its width, and up to 4 *tiers*.

If the system is in a normal state, each crane takes over its predecessor's job during a reset. However, when a crane is handling (unloading or loading) a container, it cannot be preempted by another crane. As a result, each crane can take over a job from its predecessor only after the latter has finished handling its current container. We adopt a synchronous policy such that all cranes take over jobs from their predecessors simultaneously.

If the system is not in a normal state, it permutes crane indices and crane 1 moves to the right of the berth during a reset. The cranes start processing jobs with rates according to their new indices afterwards. Similarly, we adopt a synchronous policy such that the cranes' indices are permuted simultaneously.

Suppose a reset begins at time $s$ (that is, crane $n$ finishes a job at time $s$). Let $\tau_i$ be the time when crane $i$ finishes handling its latest container that is initiated before time $s$, for $i = 1, \ldots, n$. By definition, we have $\tau_n = s$. Note that if crane $i < n$ finishes a job before time $s$, then $\tau_i < s$. Define $\bar{\tau} = \max_{1 \le i \le n} \tau_i$. We require each crane $i$ with $\tau_i < \bar{\tau}$ stays idle in the time interval $[\tau_i, \bar{\tau})$. If the system is in a normal state, all cranes take over jobs from their predecessors simultaneously at time $\bar{\tau}$. As a result, the reset is not instantaneous as it begins at time $s$ and ends at time $\bar{\tau}$. In contrast, each reset is instantaneous in the continuous model. On the other hand, if the system is not in a normal state, the crane indices are permuted at time $s$.

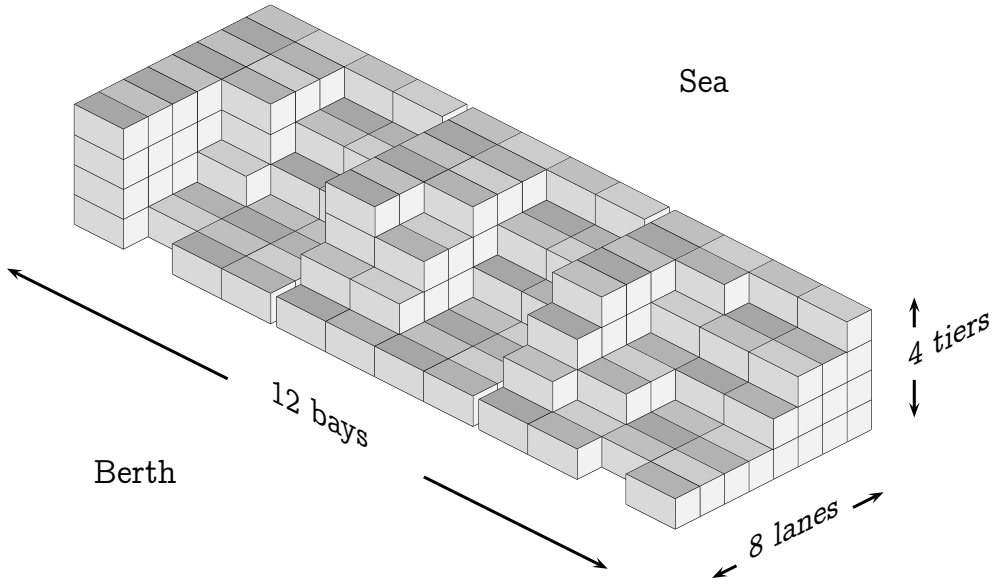



Figure 4: **Discrete containers on a vessel.** Containers on the vessel are stored in 12 bays along its longitudinal axis, in 8 lanes across its width, and up to 4 tiers.

Since a reset may require some cranes to be idle in this discrete workload model, the capacity of the system cannot be fully used. However, as we will see in our numerical simulations, the system can be configured such that its long-run average throughput based on this discrete model is generally very close to the system's maximum possible throughput.

# 5 Numerical results of the discrete model

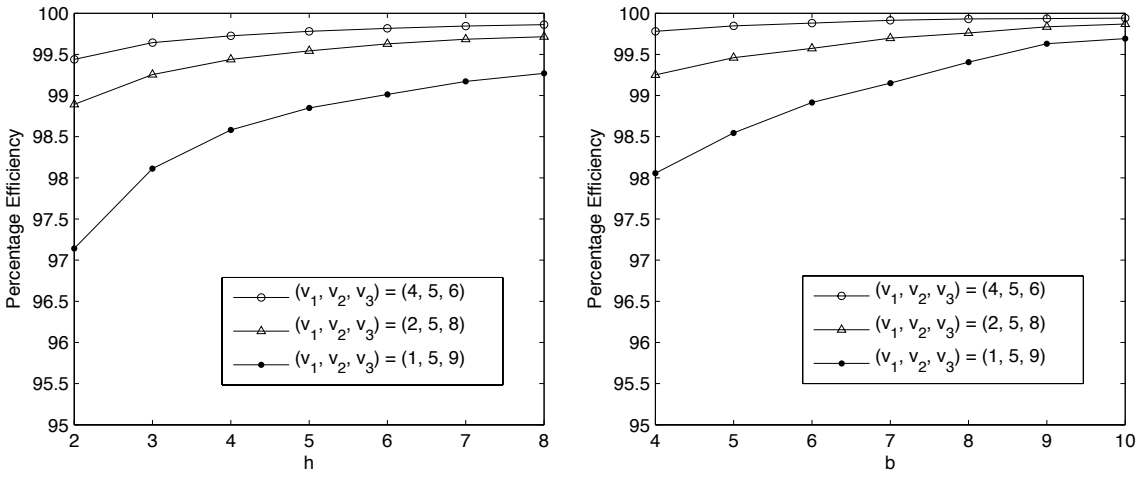
The analysis of the discrete model seems intractable for a general quay-crane system. In this section, we perform simulation studies to investigate the influence of various parameters on the system's throughput. We first study the impact of the job size, the difference in processing rate, and the number of cranes in the system before we study the impact of the ordering of processing rates.

We consider a berth that serves *batches* of vessels. Each batch of vessels comprises 40 bays and 18 lanes of containers. The containers can be stacked on top of each other up to $h$ tiers. We consider a stochastic problem in which the number of tiers of each stack of containers is a random integer variable uniformly distributed on $[0, h]$. We set the minimum separation distance $d = 4$ bays. We observe similar results for other values of $d$. We first assume there are three cranes in the system. The simulations are implemented in JAVA programming language and are run on a MACBOOK AIR computer with a 1.8GHz Intel Core 2 Duo CPU and 2GB of memory. Each data point in the figures below represents the average result of 10 simulation runs with each run generates 100 batches of vessels. Most of the data points can be obtained within seconds.

## 5.1 Impact of job size

We first investigate the performance of the system under different values of $h$ (the maximum number of tiers). Let $\rho$ denote the long-run average throughput of the system. Define *percentage efficiency* as $(\rho/\sum_{i=1}^{n} v_i) \times 100\%$. We set the number of bays per job $b = 4$. Figure 5(a) shows the percentage efficiency of the system for various $h$ under different combinations of crane processing rates. We use three different combinations of crane processing rates with a total rate of 15 containers per unit time. The cranes are indexed as 1 to $n$ from slowest to fastest for each combination of processing rates.

Our results suggest that the percentage efficiency increases with $h$ for each combination of crane processing rates. This is because as $h$ increases, each crane handles more containers before each reset. As a result, the cranes are more utilized and the average throughput increases.




(a) (b)

Figure 5: **Impact of job size.** (a) The percentage efficiency increases with $h$. (b) The percentage efficiency increases with $b$.

The easiest way to increase the average job size is to increase the number of bays per job $b$. Figure 5(b) shows the percentage efficiency of the system for various $b$ under different combinations of crane processing rates. We set $h = 5$ and the total processing rate is fixed at 15 containers per unit time. For each combination of crane processing rates, the percentage efficiency increases with $b$ due to the same reason as Figure 5(a).

Figure 5 suggests that the system is more efficient when the average job size gets larger. If each job contains more containers, the cranes spend more time on handling the containers rather than waiting in resets. Note that the percentage efficiency is at least 97% in Figure 5. This suggests that by following the simple rules in Section 3, the cranes can be coordinated to achieve a throughput level that is near the maximum possible for the system.

Note that for both Figures 5(a) and (b), the percentage efficiency increases as the processing rates of the cranes get closer. One may expect to attain 100% efficiency by making the cranes homogeneous. Unfortunately, this is not the case as found in the next section.

## 5.2 Impact of difference in processing rate

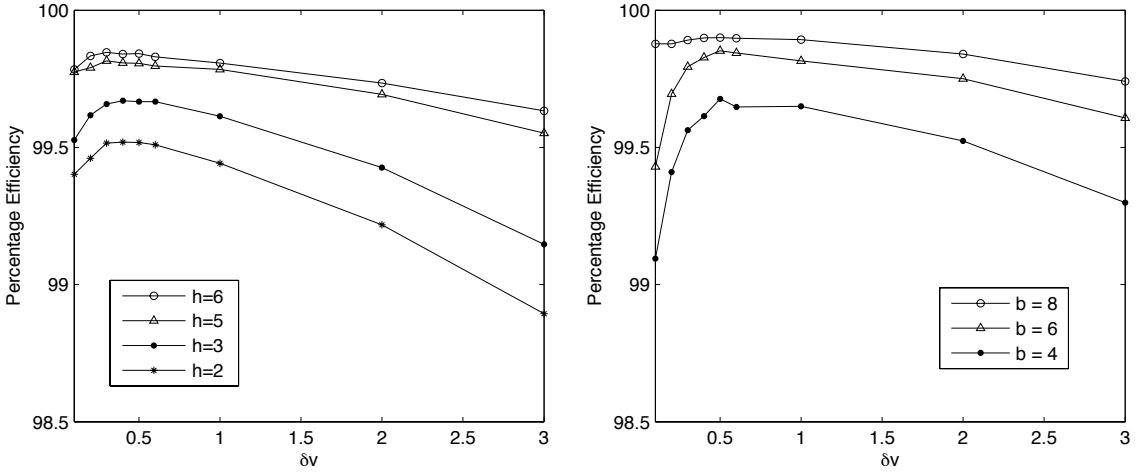
In this section, we set the processing rates $v_i = 5+(i-2)\delta v, i = 1, 2, 3$ and investigate the impact of the difference in processing rate $\delta v$ on the system's performance. Figure 6(a) shows the percentage efficiency for various $\delta v$ under different values of $h$. The percentage efficiency peaks at a certain value of $\delta v$ and it decreases as the crane processing rates deviate from each other. This is because during a reset in a normal state, the cranes generally wait for the final crane, which is often the slowest crane (crane 1), to finish handling its current container before they can take over jobs from their predecessors. As $\delta v$ increases, crane 1 gets slower, causing a longer average waiting time for other cranes in each reset.

On the other hand, as the crane processing rates get too close to each other, it is likely for the cranes to be blocked by their successors. This causes the percentage efficiency to drop.

Figure 6(b) shows the percentage efficiency for various $\delta v$ under different values of $b$. It shows a similar result: The percentage efficiency first increases and then drops as $\delta v$ increases. Both Figures 6(a) and (b) suggest that, given a fixed total capacity, the processing rates of the cranes should not be too similar or too different from each other to make the system efficient.

## 5.3 Impact of number of cranes

We examine the impact on the system's performance by varying the number of cranes $n$ from 2 to 6 with a total processing rate of 18 containers per unit time. We set $h = 5, b = 4$, and $v_i = 18/n + [i - (n+1)/2]\delta v$, for $i = 1, \ldots, n$. Figure 7(a) suggests that the percentage efficiency decreases with the number of cranes. This is because




(a) (b)

Figure 6: **Impact of difference in processing rate.** The percentage efficiency first increases and then decreases with $\delta v$.

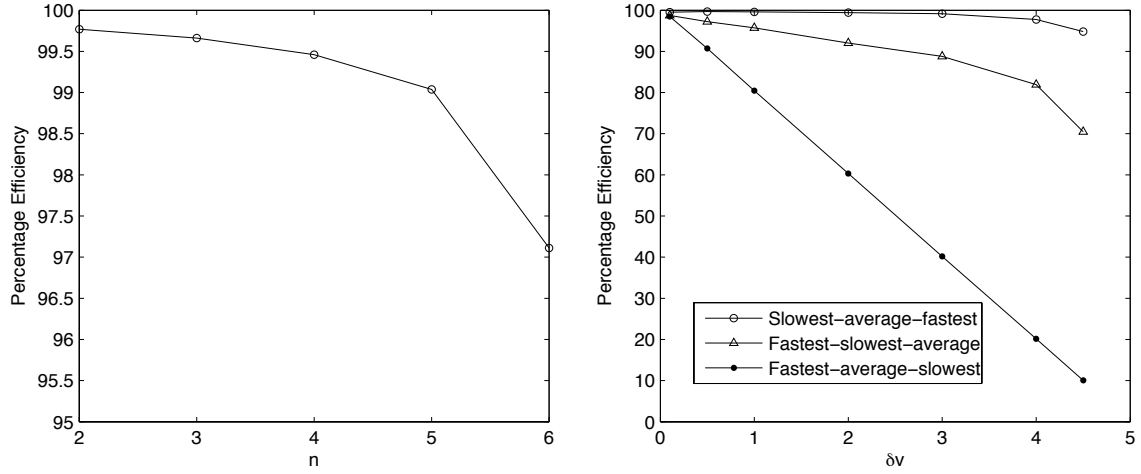
as the number of cranes increases, more take-overs of jobs are required for a reset. This generally leads to a larger $\bar{\tau}$ for each reset, causing more waste of crane capacity.

## 5.4 Impact of processing rate ordering

To study the impact of different orderings of crane processing rates, we consider a system of three cranes with rates $5 - \delta v$, 5, and $5 + \delta v$. Figure 7(b) compares the system's performance under three different orderings of processing rates with $\delta v$ ranges from 0.1 to 4.5.

The slowest-to-fastest ordering outperforms other orderings. The fastest-slowest-average ordering is more efficient than the fastest-to-slowest ordering. The performance of the fastest-to-slowest ordering declines linearly as $\delta v$ increases. This is because in the fastest-to-slowest ordering all the cranes are constantly blocked by the slowest crane (crane $n$) and thus, effectively they process their jobs with the slowest crane's rate, which decreases linearly with $\delta v$.

We have also examined the performance of other orderings beside the three orderings in Figure 7(b). We observe that the percentage efficiency of the system is mainly




(a) (b)

Figure 7: **Impact of number of cranes and ordering of processing rates.** (a) Given a fixed total processing rate, introducing more cranes to the system causes the percentage efficiency to drop. (b) The slowest-to-fastest ordering produces at a throughput level near the maximum possible for the system. The performance of the reverse ordering declines linearly as $\delta v$ increases.

determined by the last crane (crane $n$): All orderings with the same last crane have similar throughput. The slowest-to-fastest ordering remains to be the most efficient among all orderings of processing rates for all $\delta v$.

# 6 Conclusions

A key objective of container terminals is to serve arriving vessels swiftly. This requires effective allocation of quay cranes to maximize the number of containers handled per unit time, subject to the non-crossing and the minimum separation constraints among the cranes. The problem is further complicated by uncertainty in the arrivals of vessels, in the loading and unloading operations of containers, and in the traffic within the terminals.

Existing methods in the literature assume a static setting where all system's parameters are given. Most papers model the problem as a machine scheduling

problem and solve it using mathematical programming methods or heuristics. They handle uncertainty by recomputing the solution when changes in system's parameters occur. As a result, this approach is complex and onerous to implement in practice. Furthermore, only a few papers consider the minimum separation constraint, which is an important requirement in the operations of container terminals.

We propose simple rules for quay cranes to share work in a dynamic setting. These rules are easy to implement and do not require any computation to allocate the cranes. The non-crossing and the minimum separation constraints can be fulfilled naturally in our approach. Most importantly, the system is able to absorb uncertainty due to its dynamics: If we configure the system properly, it can always restore to a state with high throughput after disruptions or changes.

For workload that is continuously and uniformly distributed on the vessels (such as bulk cargos), we show that the system always converges to a state with the system's maximum possible throughput if the cranes are indexed from slowest to fastest according to their processing rates. This is appealing because the system can always restore efficiency after disruptions or changes such as delay in vessel arrivals, mishandling of cargos, bad weather conditions, or traffic congestion in the terminals. We then apply this result to the discrete workload model that is suitable for container terminals.

For the discrete workload model, we perform numerical experiments to examine the system's throughput when the cranes follow the simple rules proposed. We assume the number of tiers in each stack of containers are stochastic. We investigate the impact of job size, the difference in crane processing rate, the number of cranes, and the ordering of crane processing rates.

The average throughput of the system increases with the job size, which can

be scaled up by increasing the number of tiers in each stack of containers or the number of bays in each job. We observe that the difference in processing rate among the cranes has a profound impact on throughput, which first increases and then decreases with the difference in processing rate. This suggests that a manager should avoid adopting uniform rates or very different rates for the cranes. We also find that increasing the number of cranes will increase the number of take-overs during a reset, causing the average throughput to drop. Finally, it is important to index the cranes from slowest to fastest as our numerical simulations suggest that other orderings may cause blocking and thus, the cranes may not fully use their capacity.

Our approach is appealing to container port operators as it requires neither a major reconfiguration of the system nor expensive investment on facility. Since we do not require sophisticated software to coordinate the cranes, no significant maintenance cost is incurred and integration with other operations of the terminals is also not difficult. Furthermore, we do not need precise information about arrival and departure times of vessels because the cranes allocate themselves dynamically by following the rules. All we need is to index the cranes from slowest to fastest so that the system can constantly restore efficiency as it faces uncertainty in the operations.

# References


Alligood, K.T., T.D. Sauer, J.A. Yorke. 1996. *Chaos: An Introduction to Dynamical Systems*. Springer, New York. ISBN 0-387-94677-2.

Bartholdi, J.J. III, L.A. Bunimovich, D.D. Eisenstein. 1999. Dynamics of two- and three-worker "bucket brigade" production lines. *Oper. Res.* **47**(3) 488–491.

Bartholdi, J.J. III, D.D. Eisenstein. 1996a. A production line that balances itself.

*Oper. Res.* **44**(1) 21–34.

Bartholdi, J.J. III, D.D. Eisenstein. 1996b. The bucket brigade web page. URL `http://www.BucketBrigades.com`.

Bartholdi, J.J. III, D.D. Eisenstein. 2005. Using bucket brigades to migrate from craft manufacturing to assembly lines. *Manufacturing Service Oper. Management* **7**(2) 121–129.

Bartholdi, J.J. III, D.D. Eisenstein, R.D. Foley. 2001. Performance of bucket brigades when work is stochastic. *Oper. Res.* **49**(5) 710–719.

Bartholdi, J.J. III, D.D. Eisenstein, Y.F. Lim. 2009. Deterministic chaos in a model of discrete manufacturing. *Naval Res. Logist.* **56**(4) 293–299.

Bierwirth, C., F. Meisel. 2009. A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **202**(3) 615-627.

Daganzo, C.F.. 1989. The crane scheduling problem. *Transport. Res.* **23B**(3) 159-176.

Kim, K.H., Y.M. Park. 2004. A crane scheduling method for port container terminals. *Eur. J. Oper. Res.* **156** 752-768.

Lee, D.-H., H.Q. Wang, L. Miao. 2008. Quay crane scheduling with non-interference constraints in port container terminals. *Trasport. Res.* **44E** 124-135.

Lim, A., B. Rodrigues, F. Xiao, Y. Zhu. 2004. Crane scheduling with spatial constraints. *Naval Res. Logist.* **51**(3) 386-406.

Lim, A., B. Rodrigues, Z. Xu. 2007. A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Res. Logist.* **54** 115-127.

Lim, Y.F., K.K. Yang. 2009. Maximizing throughput of bucket brigades on discrete work stations. *Production Oper. Management* **18**(1) 48–59.

Liu, J., Y. Wan, L. Wang. 2006. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Res. Logist.* **53** 60-74.

Peterkofsky, R.I., C.F. Daganzo. 1990. A branch and bound solution method for the crane scheduling problem. *Transport. Res.* **24B**(3) 139-172.

UNCTAD. 2009. *Review of Maritime Transport.* United Nations Conference on Trade and Development.

Villalobos, J.R., L.F. Muñoz, L. Mar. 1999a. Assembly line designs that reduce the impact of personnel turnover. *Proc. of IIE Solutions Conference*, Phoenix, AZ.

Villalobos, J.R., F. Estrada, L.F. Muñoz, L. Mar. 1999b. Bucket brigade: A new way to boost production. *Twin Plant News* **14**(12) 57–61.

Zhu, Y., A. Lim. 2006. Crane scheduling with non-crossing constraint. *J. Oper. Res. Soc.*, **57**(12) 1464-1471.

# A Technical details

## A.1 Proof of Lemma 1

*Proof.* We prove by induction. According to our assumption, crane $n$ is never blocked. If crane $n-1$ is blocked by crane $n$, the former remains idle until the latter completes its current bay. After that both cranes process their next respective bays with their processing rates. Since $v_{n-1} \leq v_n$ and the workload of each bay is uniform, crane $n-1$ will never be blocked by crane $n$ again. Therefore, blocking is transient for crane $n-1$.

Assume blocking is transient for crane $j+1$. Crane $j$ will never be blocked by crane $j+1$ after a transient period because $v_j \leq v_{j+1}$ and the workload of each bay is uniform. This completes the proof. □

## A.2 Proof of Lemma 2

*Proof.* We prove by induction. It is trivial that after 1 reset crane 1 will never overtake crane 2 because $v_1 \leq v_2$. Assume there is no overtaking among cranes $1, 2, \ldots, j$ after $j-1$ resets. After $j$ resets, crane $j+1$ receives a job from crane $j$. Since there is no overtaking among cranes $1, 2, \ldots, j$, $x_j \leq x_{j+1}$ after the $j$-th reset. Crane $j$ will not overtake crane $j+1$ afterwards because $v_j \leq v_{j+1}$. Thus, there is no overtaking among cranes $1, 2, \ldots, j+1$ after $j$ resets. This shows that for a system with $n$ cranes overtaking will cease after $n-1$ resets. □

## A.3 Proof of Lemma 3

*Proof.* On the fixed point each crane $i$ repeats a fixed portion of work for each job processed: It relinquishes a job to its successor at point $x_i^*$ and then takes over work

from its predecessor at point $x^*_{i-1}$. The fixed point $\mathbf{x}^*$ can be found by solving the following equations:

$$\frac{x^*_1}{v_1} = \frac{x^*_i - x^*_{i-1}}{v_i}, \quad i = 2, \ldots, n.$$

Lemma 3 follows by simple algebra. □

## A.4 Proof of Lemma 4

*Proof.* On the fixed point the system completes a job every time crane $n$ completes its portion of work. The average throughput is

$$\begin{aligned} \rho &= \frac{1}{(1 - x^*_{n-1})/v_n}; \\ &= \sum_{j=1}^{n} v_j. \end{aligned}$$

□

## A.5 Proof of Theorem 1

*Proof.* Lemmas 1 and 2 show that after a transient period there is no blocking and overtaking in the system. Consider the system after blocking and overtaking have ceased. Iteration $t$ corresponds to the time period between $t$-th and $(t+1)$-st resets. In iteration $t$, each crane $i$ takes over a job from its predecessor at point $x^t_{i-1}$ and relinquishes the job to its successor at point $x^{t+1}_i$. The time spent by crane $i$ on this job is equal to the duration of iteration $t$. Thus, we have

$$\frac{x^{t+1}_i - x^t_{i-1}}{v_i} = \frac{1 - x^t_{n-1}}{v_n}, \quad i = 1, \ldots, n.$$

It follows by simple algebra that for $i = 1, \ldots, n$,

$$\begin{aligned} x_i^{t+1} - x_{i-1}^{t+1} &= x_{i-1}^t - x_{i-2}^t + \frac{v_i - v_{i-1}}{v_n}\left(1 - x_{n-1}^t\right), \\ \frac{x_i^{t+1} - x_{i-1}^{t+1}}{v_i} &= \left(\frac{v_{i-1}}{v_i}\right)\frac{x_{i-1}^t - x_{i-2}^t}{v_{i-1}} + \left(1 - \frac{v_{i-1}}{v_i}\right)\frac{1 - x_{n-1}^t}{v_n}. \end{aligned} \tag{2}$$

Let

$$y_i^t = \frac{x_i^t - x_{i-1}^t}{v_i}, \quad i = 1, \ldots, n.$$

Equations (2) become

$$y_i^{t+1} = \left(\frac{v_{i-1}}{v_i}\right) y_{i-1}^t + \left(1 - \frac{v_{i-1}}{v_i}\right) y_n^t, \quad i = 1, \ldots, n.$$

These equations can be expressed as a linear system

$$\mathbf{y}^{t+1} = A\mathbf{y}^t,$$

where $\mathbf{y}^t = (y_1^t, y_2^t, \ldots, y_n^t)^T$ and $A$ is a transition matrix of a finite state Markov chain. Since the Markov chain is irreducible and aperiodic (see Ross 1996), $A^t \to A^*$ as $t \to \infty$. Thus, the orbit $\mathbf{y}^0, \mathbf{y}^1, \mathbf{y}^2, \ldots$ converges to the unique fixed point $\mathbf{y}^*$. It can be shown that $\mathbf{x}^t$ converges to the fixed point $\mathbf{x}^*$ by simple algebra. □

# B Terminal Operations Survey

With the financial crisis drag down the economic growth all over the world in 2008, the world's container throughput still achieve approximately 4% growth.

Singapore retained its lead as the busiest port in the world in terms of the total number of TEU moves, growing at just over 7% compared to the previous year. Shanghai matched this growth rate and maintained its position in second place. This was a much lower growth rate than the 20 per cent experienced over the last few years. The gap between Singapore and Shanghai widened slightly in 2008 to 1.9 million TEUs, from 1.7 million in the previous year, despite extra capacity with the completion of the third-phase expansion of Yangshan port(located off Shanghai).

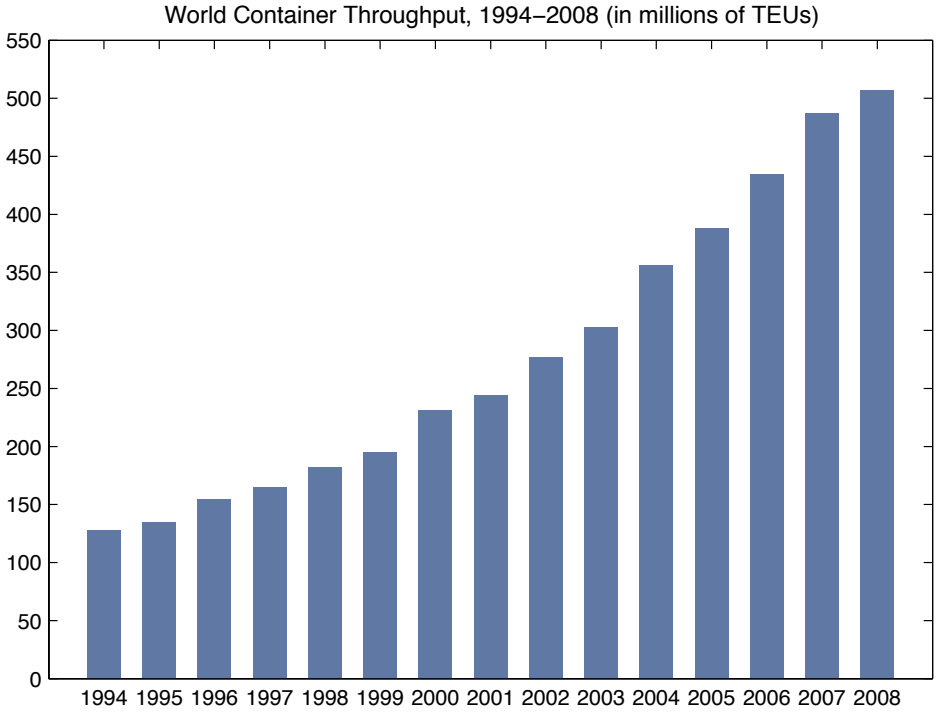



Figure 8: **World Container Throughput, 1994-2008**

source: REVIEW OF MARITIME TRANSPORT, UNCTAD

From Figure 8 we can clearly see the trend from the yearly statistics of the world's container throughput from year 1994 to 2008.

Table 2 listed top ten world's container ports according to their total number of TEU moves. We notice that most of them are from Asia.

Table 2: Top 10 World's Container Ports, 2008 - 2006

| No. | Port | Country | 2008 | 2007 | 2006 |
|---|---|---|---|---|---|
| 1 | Singapore | Singapore | 29,918 | 27,932 | 24,792 |
| 2 | Shanghai | China | 27,980 | 26,150 | 21,710 |
| 3 | Hong Kong | China | 24,248 | 23,881 | 23,539 |
| 4 | Shenzhen | China | 21,414 | 21,099 | 18,469 |
| 5 | Busan | South Korea | 13,425 | 13,270 | 12,039 |
| 6 | Dubai Ports | United Arab Emirates | 11,827 | 10,653 | 8,923 |
| 7 | Ningbo | China | 11,226 | 9,349 | 7,068 |
| 8 | Guangzhou | China | 11,001 | 9,200 | 6,600 |
| 9 | Rotterdam | Netherlands | 10,784 | 10,791 | 9,655 |
| 10 | Qingdao | China | 10,320 | 9,462 | 7,702 |

## B.1 Port of Singapore

PSA Singapore Terminals is the worlds largest container transshipment hub, handling about one-fifth of the world's total container transshipment throughput, and 6% of global container throughput. It has a network of 200 shipping lines serving 600 ports in 123 countries. It was also the busiest port in terms of total cargo tonnage handled until 2005, when it was surpassed by the Port of Shanghai.

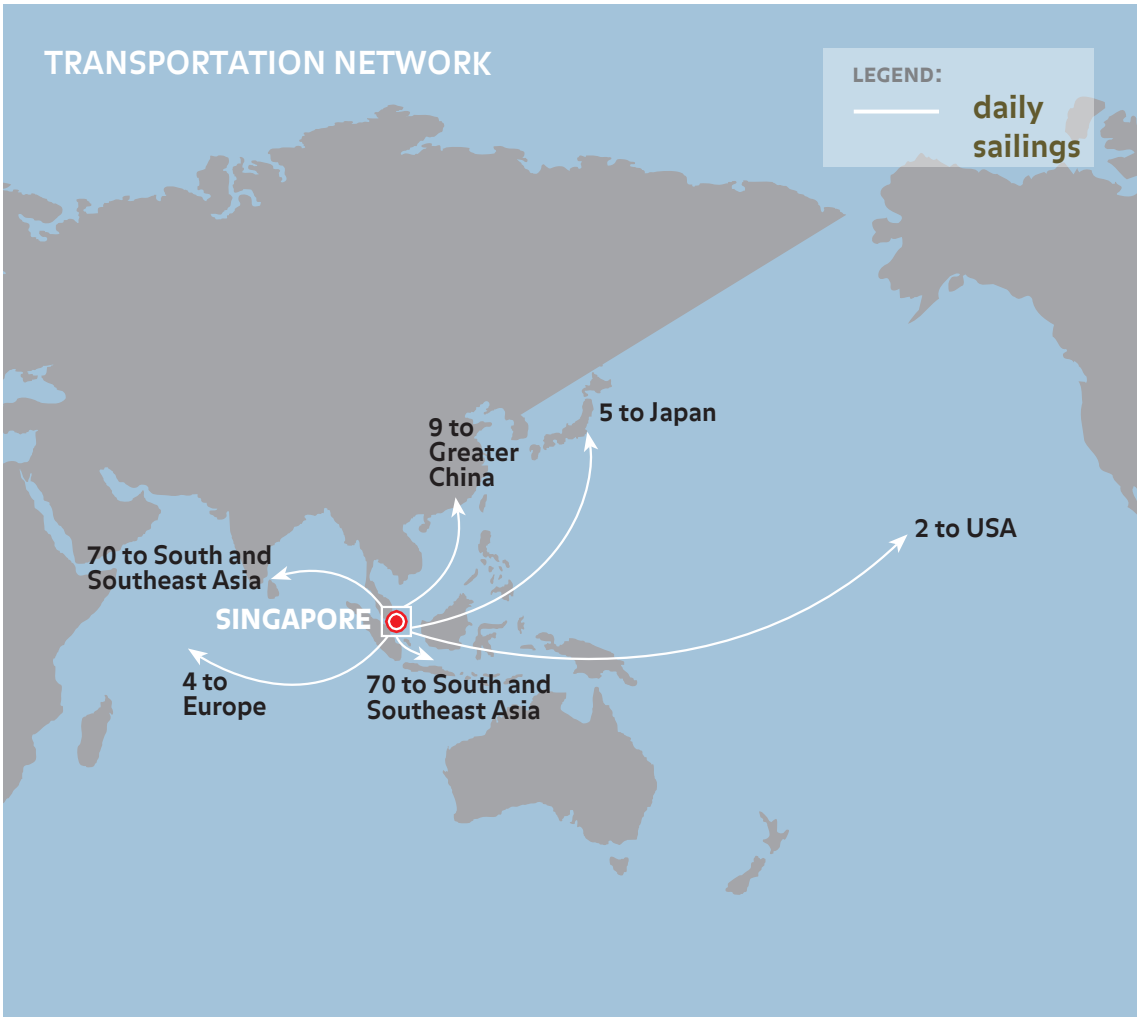



Figure 9: **Singapore Port Daily Sailings**

source: www.internationalpsa.com/factsheet/pdf/Singapore.pdf

Singapore terminal handled 29.918 million TEUs in 2008. Although facing the global financial crisis impact in 2009, they still handled around 23.5 million TEUs containers until November 2009. Figure 9 reveals the daily sailings destination from port of Singapore.

In Singapore, PSA Singapore Terminals operates four container terminals at Tanjong Pagar, Keppel, Brani and Pasir Panjang, with a total of 54 container berths when completed. They operate as one seamless and integrated facility.


Figure 10: **Singapore Port Terminals Layout** source: same as Figure 9

Noteworthy is Pasir Panjang Terminal, PSAs most advanced terminal, equipped with berths up to 16 metres deep with quay cranes able to reach across 22 lanes of containers to accommodate the worlds largest container ships. The terminals bridge crane system allows each operator to handle up to six cranes, as opposed to one previously. PSA Singapore Terminals is developing 23 container berths at the Port of Singapore's Pasir Panjang terminal, and another 16 berths will be added by 2018, bringing total container-handling capacity at the Port of Singapore for 50 million TEUs per year.

PSA Singapore Terminals also manages two multi-purpose terminals at Pasir Panjang at the Sembawang Wharves. These Port of Singapore terminals offer many port-related services including warehouses, open storage, and facilities for breakbulk and specialized cargos. The Port of Singapore's Asia Automobile Terminal (Singapore) makes PSA terminals a fast-growing major automotive transshipment hub for

the region.


Figure 11: **Pasir Panjang Terminal** source: same as Figure 9

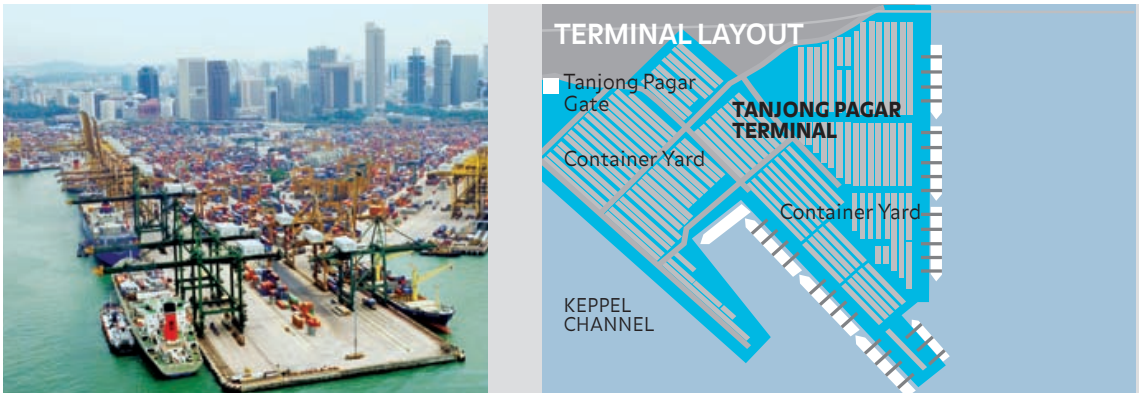



Figure 12: **Tanjong Pagar Terminal** source: same as Figure 9

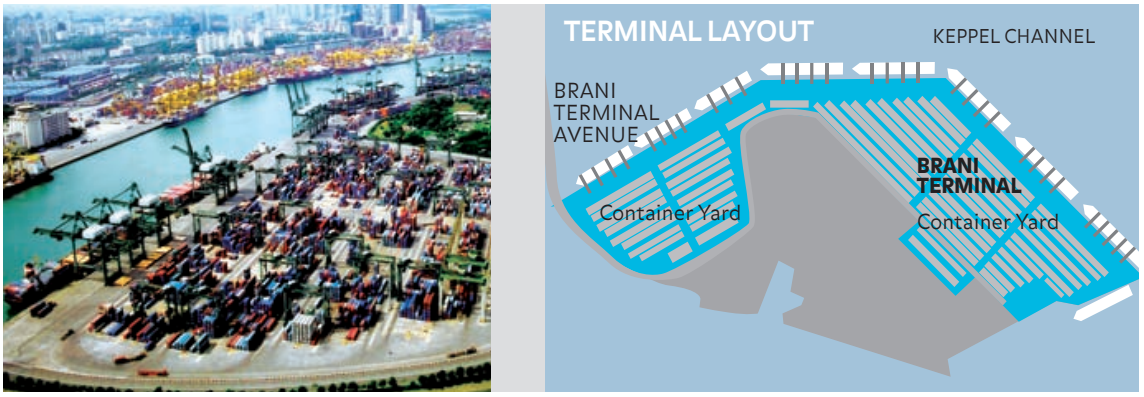



Figure 13: **Brani Terminal** source:same as Figure 9

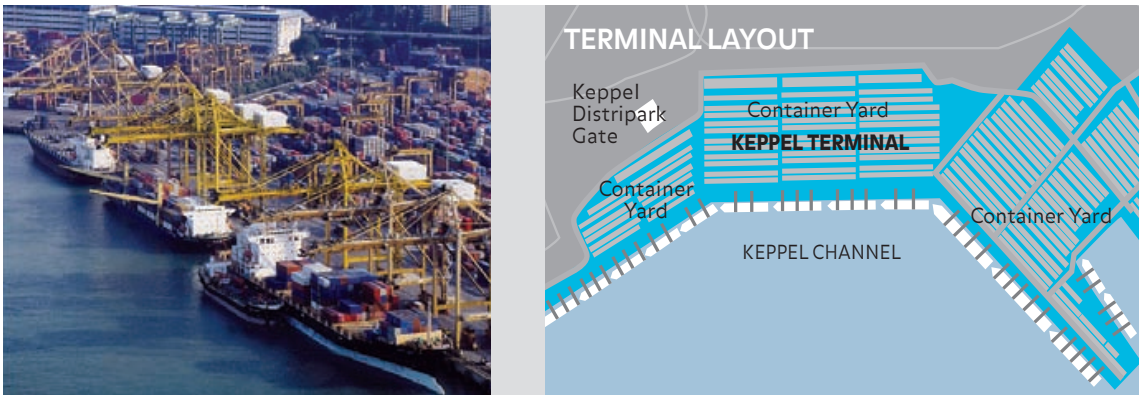



Figure 14: **Keppel Terminal** source: same as Figure 9

PSA International Pte Ltd, formerly Port of Singapore Authority is the second largest port operator in the world. The company's flagship operations are PSA Singapore Terminals, PSA HNN and PSA Marine. In total, PSA operates 28 port projects in 16 countries across Asia, Europe and the Americas, with a global capacity of 111 million TEUs over 66km of quay length.

Table 3: Container Terminals in Port of Singapore

| Terminal | Pasir Panjang | Tanjong Pagar | Keppel | Brani | Total |
|---|---|---|---|---|---|
| Area (ha) | 335 | 85 | 100 | 80 | 600 |
| Quay Length (m) | 7,900 | 2,300 | 3,200 | 2,600 | 16,000 |
| Maximum Depth at Chart Datum (m) | 16 | 14.6 | 14.6 | 15 | - |
| Container Berths | 23 | 8 | 14 | 9 | 54 |
| Quay Cranes | 87 | 29 | 42 | 32 | 190 |

Table 3 summarized the facilities situations in four terminals.

## B.2 Port of Hong Kong

Hong Kong is one of several hub ports serving the South-East and East Asia region, and is an economic gateway to mainland China. Hong Kong set a record in its container throughput in 2008 by handling 24.494 million TEUs, maintaining its status as the largest container port serving southern China and one of the busiest ports in the world. Some 217,360 vessels arrived in Hong Kong during year 2008, carrying 259 million tonnes of cargos and about 24 million passengers. According to the statistics from the port, up to November 2009, there are totally 18.973 million TEUs containers handled in Hong Kong, which show a big drop from the impact of the global financial crisis.

The Port of Hong Kong contains almost 7.7 thousand meters of quays at the Kwai Chung and Stonecutters terminals, about 7.0 thousand meters of quays at public cargo working areas, and 31 mooring buoys for ocean-going vessels. Three public passenger ferry terminals serve over 20 million passengers per year traveling to and from mainland China and Macau.

Table 4: Container Terminals in Port of Hong Kong

| | |
|---|---|
| Container Terminal | 9 |
| Area (ha) | 279 |
| Quay Length (m) | 7,804 |
| Maximum Depth at Chart Datum (m) | 12.5 - 15.5 |
| Container Berths | 24 |
| Quay Cranes | 97 |
| Terminal Operators | 5 |

Hong Kong is now the third biggest container terminal in the world, after Singapore and Shanghai. Because of rapid development in China, it faces the competition from other ports located in southern China like Shenzhen and Guangzhou which are not constrained by their land limitation.

Figure 15: **Port of Hong Kong** source:http://www.shhsfy.gov.cn

Hong Kong possesses a world-class shipping register. Hong Kong has built up this reputation by its rich heritage of maritime services, which are well-known for their efficiency, competitiveness and professionalism. As many internationally renowned shipowners are operating their ships in Hong Kong, controlling or managing about 8 per cent of the world's total tonnage, Hong Kong is among the top ten largest maritime centres in the world.

The Port of Hong Kong contains ample facilities for the repair, maintenance, drydock, and slipping of vessels. There are three floating docks off Tsing Yi Island, the largest with capacity to lift up to 46 thousand tons. There are many smaller shipyards

throughout the Port of Hong Kong that repair vessels and build specialized craft.

From Tables 4 and 3 we can observe that the area for both ports are really narrow and small compared with other ports like port of Rotterdam and Port of LA which have more than one thousand ha terminal area. From this comparison, we can imagine how busy those ports are. The operation efficiency becomes crucial for these ports.

## B.3 Port of Rotterdam

As the biggest seaport in Europe, Rotterdam handled 421.1 million tons of cargo in 2008, including almost 313 million tons of imports and over 108 million tons of exports. Within these total were 288.9 million tons of bulk cargos and 132.2 million tons of general cargos. This is far more than the second port in Europe, Antwerp, only handled 189.5 million tons of cargos the same year. Port of Rotterdam used to be the world's busiest port, now overtaken by Asian ports like Singapore and Shanghai.

The Port of Rotterdam covers a total area of over 10.5 thousand hectares, including 5.3 thousand hectares of industrial sites and 5.3 thousand hectares of infrastructure and water surface. The port is 40 kilometers long and contains 89 kilometers of quays and 1500 kilometers of pipelines.

The Port of Rotterdam has over 90 terminals specializing in different types of cargos. There are 35 terminals for liquid bulk cargos, 17 multi-purpose terminals, and 15 dry bulk terminals. While most imports handled in the Port of Rotterdam in 2008 were bulk goods (246 million tons of import bulk versus 43 million tons of export bulk). The Port of Rotterdam contains nine container terminals for deep-sea, short-sea, and inland shipping. It also has seven roll-on/roll-off terminals. It contains

Figure 16: **Port of Rotterdam** source: http://www.portofrotterdam.com/

three juice terminals and two fruit terminals. The Port of Rotterdam also has one terminal each for steel and paper, cars, and cruise vessels. The steel and paper terminal is an all-weather terminal. The Port of Rotterdam distributes cargoes to inland Europe's huge market of consumers through five transportation modes: roads, railways, pipelines, inland ships, and coastal ships. All industrial and economic centers in Western Europe can be reached from the Port of Rotterdam within 24 hours of arriving at port. The Port of Rotterdam handles over 400 million tons of cargos every year. Over 500 liner services make scheduled calls at the Port of Rotterdam and connect it with more than a thousand ports around the world. It is a gateway to a market of more than 500 million European consumers.

Table 5: Container Terminals in Port of Rotterdam

| | |
|---|---|
| Container Terminal | 9 |
| Area (ha) | 5,257 |
| Quay Length (m) | 89,000 |
| Maximum Depth at Chart Datum (m) | 24 |
| Container Berths | 23 |
| Quay Cranes | 103 |

## B.4 Port of Los Angeles

The Port of Los Angeles is located at the second biggest city and is one of its busiest seaports of USA. In 2008, the Port of Los Angeles served 2370 vessels carrying a total of 170 million metric revenue tons of cargos, including 8.3 million TEUs of containerized cargos and almost 163.4 thousand automobiles. That year, the Port of Los Angeles also welcomed 1.2 million cruise passengers.

The Port of Los Angeles covers over three thousand hectares (over 1.7 thousand hectares of land and almost 1.3 thousand hectares of water surface). It stretches across 69 kilometers of waterfront with water depths of up to 16.2 meters. The Port of Los Angeles handles about 190 million metric tons of cargo each year. The Port of Los Angeles contains 69 container cranes, including 20 post-Panamax Plus cranes. It also contains 17 marinas with 3800 slips for recreational boats. The World Cruise Center in the Port of Los Angeles is the country's most secure cruise passenger complex.

The Port of Los Angeles contains 27 cargo terminals that handle containers, liquid and dry bulk, breakbulk, and automobiles and support 270 berths. Terminals in

Table 6: Container Terminals in Port of Los Angeles

| # | Terminal | Area (acres) | Quay Length | Berths | Quay Cranes |
|---|---|---|---|---|---|
| 1 | Berth 100 | 75 | 1,200' | 1 | 4 |
| 2 | Berth 121-131 | 186 | 3,500' | 4 | 5 |
| 3 | Berth 135-139 | 173 | 4,050' | 5 | 12 |
| 4 | Berth 206-209 | 86 | 2,180' | 1-2 | 8 |
| 5 | Berth 212-225 | 185 | 5,800' | 5 | 8 |
| 6 | Berth 226-236 | 205 | 4,700' | 3 | 8 |
| 7 | Berth 302-305 | 292 | 4,000' | 4 | 12 |
| 8 | Berth 401-406 | 484 | 7,190' | 6 | 14 |
| | Total | 1,686 | 32,620' | 29-30 | 71 |

the Port of Los Angeles specialize in containers (8 terminals), liquid bulk cargo (7 terminals), breakbulk (3 terminals), dry bulk (2 terminals), passengers and ferries (2 terminals), and automobiles (1 terminal). The Port of Los Angeles also has four warehouse terminals.

With increasingly large container vessels entering global service and worldwide cargo activity continuing to grow, the Port of Los Angeles is aggressively meeting the challenge by handling ships of virtually any size, and it is growing green by moving greater cargo volumes in an environmentally responsible manner with a broad spectrum of clean technology programs. At the same time, the Port is assertively engaged in minimizing adverse road and rail traffic impacts and in ensuring the utmost safety and security in operations.

The top trading partners of the port of LA in 2004 were China ($68.8 billion),

Japan ($24.1 billion), Taiwan ($10.8 billion), Thailand ($6.7 billion) and South Korea ($5.6 billion).

The most imported types of goods were (in order): furniture; apparel; toys and sporting goods; vehicle and vehicle parts; and electronic products.

From 2002 to the present, the Port has had a large backlog of ships waiting to be unloaded at any given time. Many analysts believe that the Port's traffic may have exceeded its physical capacity as well as the capacity of local freeway and railroad systems. The chronic congestion at the Port is beginning to cause ripple effects throughout the American economy and is disrupting Just In Time inventory practices at many companies.

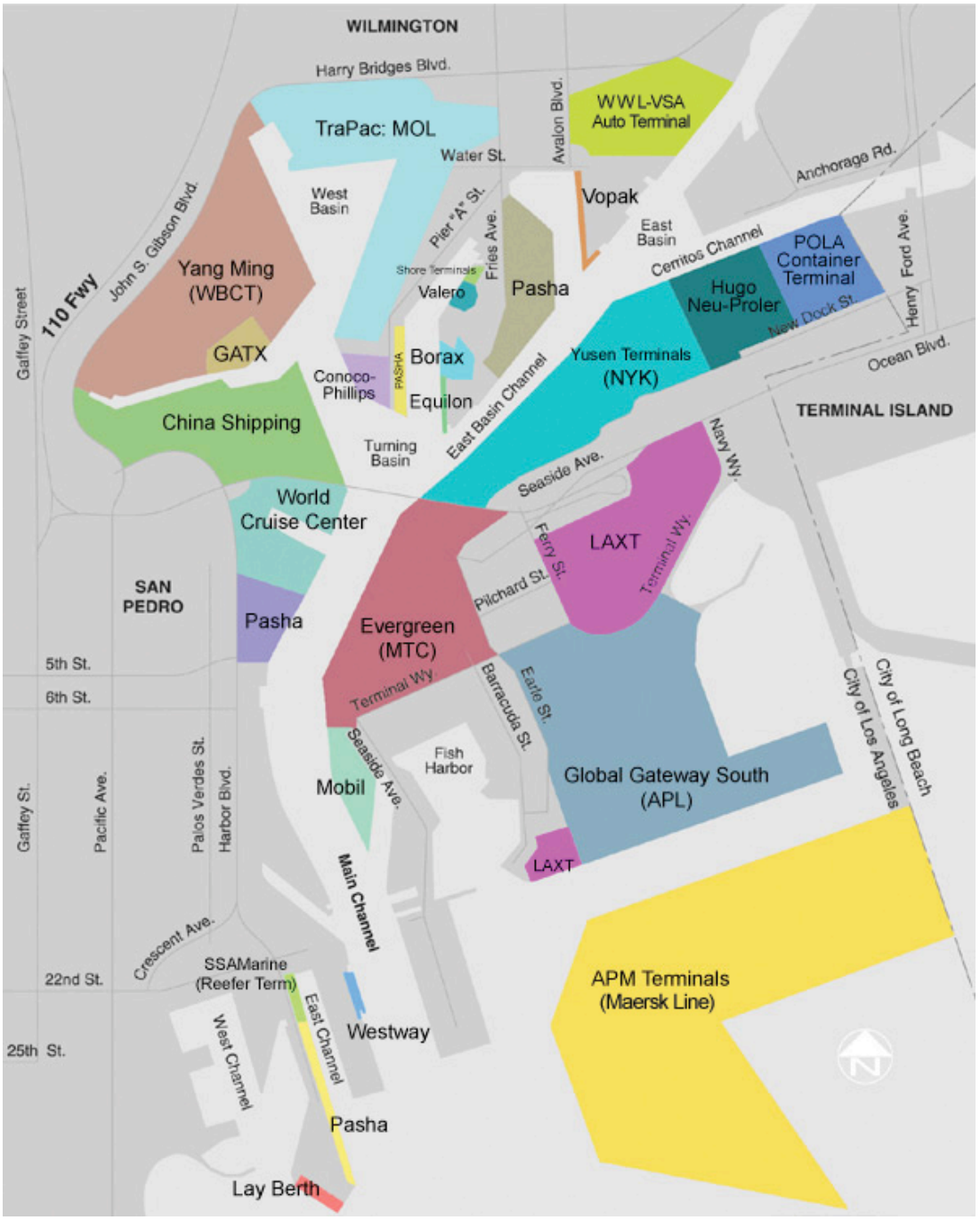



Figure 17: **Port of Los Angeles Terminal Layout**
source: http://www.mxsocal.org/portmaps.aspx