International Journal of
Transportation Science and Technology
volume 1 · number 2 · 2012

An Agent-Based Solution Framework for Inter-Block Yard Crane Scheduling Problems

by
Omor Sharif, Nathan Huynh, Mashrur Chowdhury, Jose M. Vidal

ISSN 1475 472 X **Multi-Science Publishing Co. Ltd**

An Agent-Based Solution Framework for Inter-Block Yard Crane Scheduling Problems

Omor Sharif[a], Nathan Huynh[a,*], Mashrur Chowdhury[b], Jose M. Vidal[c]
[a]Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC 29208, USA
[b]Glenn Department of Civil Engineering, Clemson University, Clemson, SC 29634, USA
[c]Department of Computer and Science Engineering, University of South Carolina, Columbia, SC 29208, USA
*Corresponding Author: Telephone: (803) 777-8947; Fax: (803) 777-0670.
Email addresses: omor.sharif@gmail.com (Omor Sharif), huynhn@cec.sc.edu (Nathan Huynh), mac@clemson.edu (Mashrur Chowdhury), jmvidal@gmail.com (Jose M. Vidal).

ABSTRACT
The efficiency of yard operations is critical to the overall productivity of a container terminal because the yard serves as the interface between the landside and waterside operations. Most container terminals use yard cranes to transfer containers between the yard and trucks (both external and internal). To facilitate vessel operations, an efficient work schedule for the yard cranes is necessary given varying work volumes among yard blocks with different planning periods. This paper investigated an agent-based approach to assign and relocate yard cranes among yard blocks based on the forecasted work volumes. The goal of our study is to reduce the work volume that remains incomplete at the end of a planning period. We offered several preference functions for yard cranes and blocks which are modeled as agents. These preference functions are designed to find effective schedules for yard cranes. In addition, we examined various rules for the initial assignment of yard cranes to blocks. Our analysis demonstrated that our model can effectively and efficiently reduce the percentage of incomplete work volume for any real-world sized problem.

Keywords: Yard crane scheduling, container terminals, multi-agent systems, deferred acceptance algorithm.

1. INTRODUCTION

The importance of marine container terminals in international trade has been well documented (Vis and de Koster, 2003; Steenken et al., 2005; Stahlbock and VoB, 2008). Previous studies have also reported planning and operational challenges that port authorities and terminal operators have to contend with at marine container terminals (Murty et al. (2005), Rashidi and Tsang (2006), Henesey (2006)). Capacity constraints, lack of adequate decision making tools, congestion, and environmental concerns are some of the issues facing terminals today. Various operations research techniques, automated equipment, and information technology have been applied in an effort to improve the efficiency of various terminal operations with limited resources and high workloads. An important problem that has been studied extensively is how to expedite vessel operations (Steenken et al., 2005). To this end, researchers have investigated the quay crane scheduling problem, transporter scheduling problem, and yard crane scheduling problem (Rashidi and Tsang, 2006). This study focuses on the yard crane scheduling problem. It deals with the planning problem of appropriate locations for yard cranes and where to move them during vessel operations to facilitate the unloading of import containers and loading of export containers.

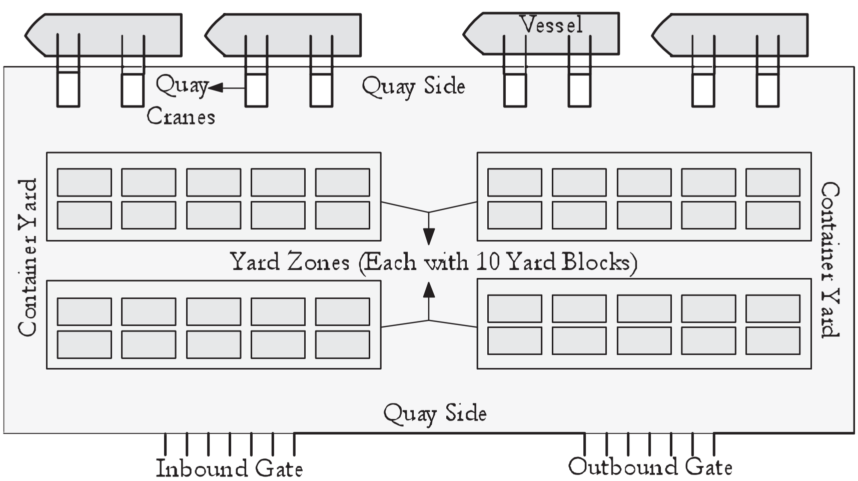


Figure 1. Illustration of a container yard layout (not to scale).

The container yard at a marine terminal serves as a buffer for import containers before they are picked up by a drayage truck and for export containers before they are loaded onto a vessel. Figure 1 depicts a simplified layout of a marine container terminal- the land side gates for external truck operations, the yard for container storage and the quay side for vessel operations. The yard is typically made up of multiple yard zones and each zone with multiple yard blocks. A typical block at a U.S. terminal is forty 40-foot bays long, 6 rows wide and 4 containers high (Huynh and Vidal, 2012). There are two types of yard cranes, rubber tired and rail mounted. Rail-mounted yard cranes move

Figure 2. A rubber-tired yard crane at work.

on rails and they can only travel in one direction (along the length of block). Rubber tired yard cranes are more flexible in that they can move in both directions. They are the more popular choice among U.S. terminals. The yard crane scheduling problem addressed in this study assumes the use of rubber-tired yard cranes. A rubber tired yard crane is shown in Figure 2.

During vessel operations, the yard cranes need to be able to keep up with the quay cranes as they load and unload containers from the vessel. The amount of work in each block depends on whether the vessel is being unloaded or loaded. During the unloading operation, import containers are unloaded from the vessel and stored in designatedimport yard blocks, and during the loading operation, export containers are retrieved from specific exportyard blocks and loaded onto the vessel. Typically, a number of unloading and loading operations will take place during vessel operations that will require the yard cranes to be in various yard blocks. Thus, an efficient yard crane deployment plan is necessary to address variable work volume and is imperative in lowering the vessel turn time. Vessel turn time refers to the time a vessel spent at a terminal while awaiting theunloading and loadings of containers. The vessel turn time is one of the chief indicators of a terminal's productivity and competitiveness. The inter-block yard crane deployment problem is as follows: given the forecasted work volume of each block in each period of a day, assign the yard cranes among blocks dynamically so that the total incomplete work volume in the yard is minimized. This scheduling problem historically has been addressed by mathematical optimization programs (A review is available in Section 2). In contrast, this study utilizes an agent-based approach. Agent-based modeling is a decentralized and a relatively new research field within the realm of artificial intelligence. Researchers and practitioners in many disciplines, from biology to economics, have developed agent-based models, and the number of applications continues to rise (Bernhardt, 2007). While agent-based modeling and Multi-Agent Systems (MAS) have been applied in many different disciplines, they are relatively unexplored in the area of port operations.

There has been no study using a decentralized approach to solve the inter-block crane deployment problem. The typical advantages of adopting an agent-based approach over classical optimization include the capability of solving problems of large sizes through discretization, producing a time efficient solution, obtaining a solution adaptive to changes in a dynamic system, and obtaining a robust system with better computational stability (Davidsson et al., 2003). Additionally, in recent years integrative modeling for container terminals are being emphasized which is based upon the fact that various processes in a terminal are interconnected and improved terminal performance cannot necessarily be achieved by treating the processes separately (Stahlbock and VoB, 2008). Therefore, there is a need to integrate models of various terminal processes with one another. Only a few studies have attempted such an integrated approach. However, the scope of these studies is limited in that only a few selected operations are considered together, and the focus is primarily on the quayside processes. More decision tools need to be developed and integrated. Multiagent systems approach hasbeen proposedas a tool in integrated decisions frameworksin works by Thurston and Hu (2002), Henesey (2006), Franz et al. (2007).Our model may serve as a component tool in an integrated multi-agent model. In past work, we have developed agent-based models related to gate operations and real-time yard crane scheduling (Sharif et al. (2011), Huynh and Vidal (2010)). The model described in this study can be combined with these previously developed models to develop an integrated model. Such an integrated model is not viable using traditional optimization techniques due to computational complexity. Also, our agent-based approach to the yard crane deployment model is easier to understand because it uses simple intuitive preference functions for agents. These preference functions are designed to quickly find effective schedule for cranes. In addition, we examined various rules for the initial assignment of cranes to blocks. Furthermore, in this study, we assessed some simple strategies for an initial assignment of cranes to yard blocks to provide guidance for the terminal operator or stevedore.

2. RELATED STUDIES

There is a vast amount of literature in the area of marine container terminal modeling. With container terminal operations becoming more and more important, an increasing number of publications on container terminals have appeared in the literature. A survey of container terminals related research can be found in several sources: Vis and de Koster (2003), Steenken et al. (2005), Stahlbock and VoB (2008), Crainic and Kim (2007), Murty et al. (2005), Rashidi and Tsang (2006), Vacca et al. (2007), Henesey (2006). A comprehensive review is beyond the scope of this paper.

Yard operations, in general, involves two classes of decision problems, namely, 1) storage space assignment problem - the objective is to determine an optimum space allocation such that handling and re-handling of containers is kept at a minimum and traveling time of vehicles is minimized. Types of operations such as 'wheeled' and 'stacking' operations, land utilization, and efficient accessibility of containers are some factors generally taken into account (Vis and de Koster, 2003); and 2) scheduling problem of yard equipment such as yard cranes used for container storage, retrieval and reshuffling operations - the objective is to maximize utilization of cranes and minimize unfinished work volume, completion time of tasks and waiting time of transport vehicles in a planning period. The review in this section is primarily focused on literature addressing the latter problem i.e. yard crane scheduling at marine container terminals. Existing studies involving scheduling of yard cranes can further be divided into two subgroups, as they consider decision making at two different levels: 1) assigning yard cranes to blocks (crane-to-block) and 2) assigning yard cranes to trucks (crane-to-truck). The 'crane-to-block', also frequently called 'inter-block yard crane deployment problem', involves the dynamic allocation of yard cranes to various storage blocks. This subgroup aims at optimizing the movement of yard cranes among blocks. In contrast, the 'crane-to-truck' involves determining the optimal sequence of handling of individual containers i.e. serve individual trucks, which deals with the real-time bay to bay movement of cranes. Although both 'crane-to-block' and 'crane-to-truck' are decision problems considered at the operational level (i.e. short-term planning), the former usually needs to be solved at longer periodic intervals (typically every several hours), while the latter typically needs to be solved at shorter periodic intervals (typically several times within an hour) or in real-time.

One of the earliest studies on inter-block yard crane deployment is by Zhang et al. (2002). The focus of the study was to, given the forecasted work volume of yard blocks in each period of a day, find the times and routes of crane movements among yard blocks so that the total delayed work volume in the yard is minimized." They formulated a Mixed Integer Program (MIP) for dynamic deployment of yard cranes and solved the program using Lagrangean relaxation. In another study by Cheung et al. (2002), they addressed the deployment problem with the same objective. The formulation was also an MIP which was shown to be NP-hard and a new solution approach called 'successive piecewise-linear approximation method' was developed, which is more effective and efficient than the Lagrangean decomposition. Linn et al. (2003) presented an algorithm and a mathematical model for the optimal yard crane deployment. The effectiveness of the model was tested against a set of actual operation data collected from a major container terminal in Hong Kong. Linn and Zhang (2003) developed a least cost heuristic algorithm to find a near optimal solution of practical size crane deployment problems. Yan et al. (2008) presented two heuristic algorithms, the hill-climbing algorithm and the best-first-search algorithm, to overcome the NP-hardness of the deployment problem. He et al. (2010) employed a hybrid algorithm, which combines heuristic rules and parallel genetic algorithm. A simulation model was also developed to evaluate their approach.

Kim et al. (2003) used simulation to study various truck serving rules for yard cranes to minimize truck delay. The sequencing rules comprise dynamic programming, first-come-first- served, unidirectional travel, nearest-truck-first-served, shortest-processing time rule, and a rule set from reinforcement learning. Ng and Mak (2005) studied the problem of scheduling a yard crane to handle a given set of jobs with different ready times. They proposed a branch and bound algorithm to solve an MIP that finds an optimal schedule that minimizes the sum of truck waiting times. In a follow-up study by Ng (2005), the author extended his previous work to deal with multiple yard cranes instead of a single yard crane. His model accounted for interference among cranes which may occur when they are sharing a single bi-directional traveling lane. An integer program was proposed and a heuristic was developed to solve the model. Lee et al. (2007) studied the scheduling of a two yard crane system which serves the loading operations of one quay crane at two different container blocks, so as to minimize the total loading time at stack area. A simulated annealing algorithm was developed to solve the proposed mathematical model. Li et al. (2009) developed a crane scheduling model where operational constraints such as fixed yard crane separation distances and simultaneous container storage/retrievals are considered. The model was solved using heuristics and a rolling-horizon algorithm. Huynh and Vidal (2010) introduced an agent-based approach to schedule yard cranes with a specific focus on assessing the impact of different crane service strategies on drayage operations. In their work, they modeled the cranes as utility maximizing agents and developed a set of utility functions to determine the order in which individual containers are handled.

In this paper we address the crane-to-block level of decision making which isknown asthe 'inter-block crane deployment problem'. The contributionsof this study to the literature are: 1) provides an agent based framework for solving the inter-block crane deployment problem, 2) provides an approach that effectively minimizes the percentage of incomplete work volume, 3) provides a scalable and time efficient approach, and 4) provides various strategies of initial assignment of yard cranes.

3. METHODOLOGY

This section provides details regarding our assumptions and inter-block crane deployment model.

3.1. Assumptions

The assumptions are as follows.

- The total operational hours of a container terminal is divided into several shifts or planning periods. The planning periods can be of equal or different time lengths.
- A forecast of the work volumes in the yard blocks are known at the beginning of the planning period.
- For the safety of crane operations, at any time at most two cranes can work in the same block.
- An idling yard crane at some block can be relocated to assist another block, but such transfer is only allowed once per planning period. This assumption is to avoid traffic congestion in the yard area.
- For each block *the initial work volume* at the beginning of a planning period is the work volume forecast for that period plus incomplete work volume from the previous period.

- The transport vehicles (Internal Trucks or Automated Guided Vehicles) moving between storage yard and quay cranes does not introduce delay in yard cranes' operation.
- The capacity of yard cranes are identical and equal to length of planning period (measured in time units).
- When thenumber of container moves in a yard block is known, Equation 1 is used to obtain workvolume in time units for that block.The parameter 'Average time units required per move' required in Equation 1 can be estimated by a container terminal (i.e. 'total time required by yard cranes to handle a number of containers' divided by 'total number of containers handled' gives how much time on average is required per move).

$$\text{Work volume} = \text{Average time units required per move} \times \text{Number of container moves} \quad (1)$$

The transfer time of a yard crane between two blocks are calculated in the following manner. If a yard crane is relocated to a block for which it needs to travel in a longitudinal direction with respect to its current location, the transfer time is 10 minutes for each block traversed. For example, in Figure 3, if a yard crane is relocated from block *B2* to *B8*, it takes (3 × 10) or 30 minutes to complete the transfer. If a yard crane is relocated to a block for which it also needs to travel in a transverse direction with respect to its current location, the transfer time is 10 minutes for each block traversed plus 5 minutes for an additional two 90 degree turns of the crane wheels. If a yard crane is relocated from block *B2* to *B5*, it takes (5 + 2 × 10) or 25 minutes to complete the transfer.

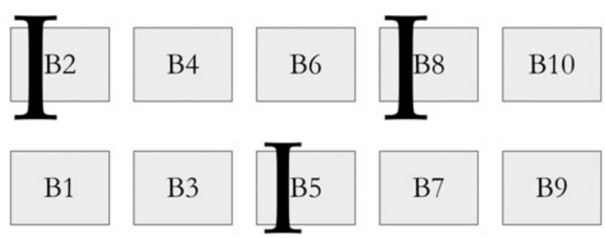


Figure 3. Illustrating the transfer time of cranes.

3.2. Initial Assignment of yard cranes

At the beginning of a planning period, the container yard manager must decide on the initial distribution of yard cranes among the yard blocks. The initial assignment of yard cranes can be simply random or uniformly distributed among the blocks. However, a more reasonable assignment will be that based on work volume forecasts in the blocks at the beginning of a planning period. The studies in literature dealing with the inter-block deployment problem assumes that an initial assignment of the cranes are given or known (which, in reality, is usually based on the judgment of the yard manager).

However, in our study we have investigated some intuitive strategies that can be employed for the initial assignment. These strategies are presented in this section. Note thata good strategy shall be tailored to achieve objectives such as: 1) assign cranes to blocks where they are most needed i.e. based on work volume and 2) reduce the number offuture inter-block crane transfers during operation to prevent loss of time and crane productivity. We address these goals in three approaches namely 'high to low work volume', 'crane at each block' and 'reduce transfers'.

For illustration we use the following variable definitions-
T_c ≡ Capacity of a crane c in timeunits (length of planning period)
IW_b ≡ Initial work volume of a block b at the beginning of a planning period
$NC_b^{initial}$ ≡ Number of cranes initially assigned to a block b at the beginning of a planning period
$NC_b^{current}$ ≡ Number of cranes currently assigned to a block b i.e. its value may change over operational hours
NC_b^{max} ≡ Maximum number of cranes that can work in a block b simultaneously (The value of NC_b^{max} is set to 2 in our model)

High to low work volume

In this strategy, a list of blocks is generated by sorting the blocks in the order of decreasing IW_b. Thus, the topmost item of the list has the maximum work volume and the bottommost item has the minimum work volume. Then, cranes are assigned to blocks according to their order in the list starting with the topmost item. Once a block is assigned NC_b^{max}cranes, the next block in the list is considered for assignment, and the process is continued until all available cranes are assigned. Note that this simple strategy does not directly take into account the actual value of work volumes but only their relative order in the list.

Crane at each block

In this strategy three possible scenarios are considered. Let the total number of cranes ben_c and the total number of blocks ben_b.

- If $n_c = n_b$, assign a single crane at each block.
- If $n_c < n_b$, generate a list of blocks sorted in order of decreasing IW_b. Then assign a single crane to each of the top n_c blocks from that list.
- If $n_c > n_b$, first assign a single crane in each block. Then calculate the incomplete work volume for all blocks using Equation 2.

$$\textit{Incomplete work volume of block } b = IW_b - T_c \tag{2}$$

Now, create a list of blocks sorted in decreasing order of *incomplete work volume* as found from Equation 2.Next assign a single crane to each of top $n_c - n_b$ blocks from that list.

Reduce transfers

This strategy assigns cranes to blocks in the following steps-

- Find the blocks that satisfy Equation 3. For these blocks, assign NC_b^{max} cranes in each block.

$$IW_b \geq NC_b^{max} \times T_c \tag{3}$$

- Next, find the blocks that satisfy Equation 4. For these blocks, assign a single crane in each block.

$$IW_b < NC_b^{max} \times T_c \quad \text{and}$$
$$IW_b \geq T_c \tag{4}$$

- Calculate the incomplete work volume for the blocks using the following Equation 5.

$$\textit{Incomplete work volume of block } b = IW_b - T_c \times NC_b^{current} \tag{5}$$

Then, find the blocks that satisfy the following condition in Equation 6.

$$NC_b^{current} < NC_b^{max} \quad \text{and}$$
$$\textit{Incomplete work volume of block } b \geq 0.7 \times T_c \tag{6}$$

Next, create a list of these blocks sorted in order of decreasing work volume as found in Equation 5. Assign a single crane to each block from top of that list and continue until there are cranes available. Note that the factor ‘0.7’ in Equation 6 is a measure of how much need there is for a block to have an additional crane. This factor should be between 0.66 to 1.0. For our study we used 0.7 which produces the best results in our model.

- If additional unassigned cranes are available, compute the revised incomplete work volume for blocks using Equation 5. Then create a list of these blocks sorted inorder of decreasing work volume for which Equation 7 holds true.

$$\textit{Currently assigned number of cranes} < NC_b^{max} \tag{7}$$

Next assign a single crane to each block starting from the top of that list and continue until there are cranes available. Repeat this step until all remaining cranes are assigned.

3.3. Pre-Analysis steps

Once a decision on the initial assignment is made, the cranes will be appointed to their designated blocks at the beginning of the planning period. The next step is todetermine the inter-block crane transfers during operations of the planning period. However, before we move to that step of the analysis, we can exclude some blocks and cranes from that step. This simplification is a set of reasonable assumptions, inclusions and exclusion procedures that are also used in previous studies. These procedures effectively reduce the size of the crane deployment problem. The following paragraphs describe how the blocks and cranes are identified to be *excluded* from or *included* into further analysis.

- We *exclude* a block if it has the maximum number of cranes initially assigned to it and its work volume equals or exceeds the capacity of those initially assigned cranes. Clearly the block cannot accommodate any additional cranes and its currently assigned cranes need to stay at that block till the end of the planning period. Thus, we can also *exclude* the cranes initially appointed at the block from analysis for they will not be transferred to other blocks.
- We *exclude* a block if its work volume is equal to the capacity of the initially assigned crane or cranes since the block does not require any additional cranes. Also, we can *exclude* the cranes appointed to that block from analysis since the crane or cranes will not be transferred to other block.
- We *exclude* a block if its work volume is less than the capacity of its initially assigned crane or cranes, obviously the block does not require any additional cranes. However, thecranes have extra capacity left after finishing work in that block and can be transferred to help out in other blocks. Therefore we *include* these cranes in further analysis. The extra capacity of the cranes depends on how much work volume each crane shares in that block. If we limit the sharing to a minimum we can save time spent on transfers. The extra capacity of a crane *c* initially assigned at a block *b* can be computed using Equation 8.

$$E\,(c) = T_c \times NC_b^{initial} - IW_b \tag{8}$$

- We *include* a block if it has less than the maximum number of cranes that can be initially assigned to it and its work volume exceeds the capacity of initially assigned cranes. Under this situation the block needs help and can accommodate additional crane or cranes transferred to it from other blocks. However, the cranes that are already located in this block are needed for the entire period and thus we can *exclude* them from further analysis. The amount of help needed or the incomplete work volume of a block *b* can be computed using Equation 9.

$$H\,(b) = IW_b - T_c \times NC_b^{current} \tag{9}$$

After the above steps are carried out the deployment problem will consist of a set of blocks needing help that can accommodate additional cranes and set of cranes with extra capacity available for helping out other blocks.

3.4. Dynamic deployment of cranes

A formal description of the problem is given here. Let us consider that the deployment problem consists of a set of blocks B and a set of cranes C. Each block $b \in B$ has a strict preference ordering over the cranes in C and each crane $c \in C$ has a strict preference ordering over the blocks in B. The preference ordering of a block i is denoted as $\succ_b^i$ and $c_x \succ_b^i c_y$ means block i ranks crane x above crane y. Similarly, the preference orderingfor crane j is $\succ_c^j$. We want a matching between agents in B and C,we want the matching to subject to these constraints 1) each crane can be matched/assigned to at most one block, and 2) one block can be matched/assigned to one or more cranes but not exceeding NC_b^{max}.

3.4.1 Preference functions for agents

We present in this section some strategies to generate preference orderings for agents. Once the preferences are available they can be used in subsequent application of the algorithm. Note that, a good preference strategy encourages 'crane-block' matchings that will likely minimize the total incomplete work volume at the end of a planning period. We have investigated four different strategies, namely, 'Minimum transfer time', 'Positive difference', 'Absolute difference' and 'Absolute difference squared distance'.

Minimum transfer time

In this strategy, the preference orderings for cranes and blocks are determined by transfer time required to relocate a crane c from its origin block o to destination block d which we denote as TT_c^{od}. This is a greedy approach that only considers transfer time. In contrast to other strategies we investigated, it does not take into account the extra capacity of a crane or the amount of help needed by a block. A crane simply prefers to be transferred to a block that is closest to its current block and a block prefers to attract a crane that is currently located in a block closest to it. The preference for a crane c over a block b is computed as in Equation 10-

$$\succ_c = TT_c^{od} \tag{10}$$

The preference for a block b over a crane c is computed as in Equation 11-

$$\succ_b = TT_c^{od} \tag{11}$$

Using the above equations, the preference ordering for a crane or a block agent over agents in B or C respectively can be obtained by sorting agents yielding the minimum to maximumvalue. Any tie is broken arbitrarily or randomly. If the extra capacity of a crane is less than or equal to the transfer time to a block then the transfer is invalid and both the crane and block pair will remove each other from the preference ordering.

Positive difference

In this strategy, the preference ordering for cranes and blocks is determined by (1) transfer time required to relocate a crane from its origin block to destination block, (2) extra capacity of a crane (3) the amount of help needed by a block. The preference for a crane c over a block b is computed using Equation 12.

$$\succ_c = E(c) - H(b) - TT_c^{od} \tag{12}$$

The preference for a block b over a crane c is computed using Equation 13.

$$\succ_b = E(c) - H(b) - TT_c^{od} \tag{13}$$

Using the above equations, the preference ordering for a crane or a block agent over agents in B or C respectively can be obtained by sorting agents yielding maximum to minimum value. Any tie is broken arbitrarily or randomly. A positive preference value implies that if the crane is relocated to a block, it will finish all the unfinished work there and will have some idle time. A negative preference value implies that if the crane is relocated to a block, it will only finish a portion of the unfinished work with no idle time. The idea underlying this preference function is to finish all work in a block, however a large positive value implies significant unused crane time. If the extra capacity of a crane is less than or equal to transfer time to a block then the transfer is invalid and both the crane and block pair will remove each other from the preference ordering.

Absolute difference

In this strategy, similar to 'positive difference', the preference ordering for cranes and blocks is determined by (1) transfer time required to relocate a crane from its origin block to destination block, (2) extra capacity of a crane (3) the amount of help needed by a block. However, we take the absolute of preference values. The preference for a crane c over a block b is computed using Equation 14.

$$\succ_c = \left|E(c) - H(b) - TT_c^{od}\right| \tag{14}$$

The preference for a block b over a crane c is computed using Equation 15.

$$\succ_b = \left|E(c) - H(b) - TT_c^{od}\right| \tag{15}$$

Using the equations, the preference ordering for a crane or a block agent over agents in B or Crespectively can be obtained by sorting agents yielding minimum to maximumvalue. Any tie is broken arbitrarily or randomly. A small preference value implies that aftertransfer time is deducted from the extra capacity of a crane it closely matches to the help needed by a block. A large preference value implies a large difference, that is, if the crane is relocated to a block, it will either finish all incomplete work but at the cost of significant idling or can only finish a small portion of the incomplete work volume with no idling. The idea underlying this preference function is to encourage matching of a crane and a block pair for which extra capacity is close to incomplete work volume. If the extra capacity of a crane is less than or equal to transfer time to a block then the transfer is invalid and both the crane and block pair will remove each other from preference ordering.

Absolute difference squared distance

In this strategy, similar to 'absolute difference', the preference ordering for cranes and blocks is determined by (1) transfer time required to relocate a crane from its origin block to destination block, (2) extra capacity of a crane (3) the amount of help needed by a block. However, we take the square of the transfer time to accentuate its effect on the preference values. The preference for a crane *c* over a block *b* is computed using Equation 16.

$$\succ_c = \left|E(c) - H(b) - (TT_c^{od})^2\right| \quad (16)$$

The preference for a block *b* over a crane *c* is computed using Equation 17.

$$\succ_b = \left|E(c) - H(b) - (TT_c^{od})^2\right| \quad (17)$$

Using the above equations, the preference ordering for a crane or a block agent over agents in *B* or *C* respectively can be obtained by sorting agents yielding maximum to minimum value. Any tie is broken arbitrarily or randomly. The idea underlying this preferencefunction is the same as 'absolute difference', that is, to encourage matching of a crane anda block pair for which extra capacity is close to the incomplete work volume. However, since long transfer time of a crane translates to a high loss in the crane's productivity, we aim to discourage moves involving long transfer times using a squared value. If the extra capacity of a crane is less than or equal to the transfer time to a block then the transfer is invalid and both the crane and block pair will remove each other from the preference ordering.

3.4.2. An algorithm to assign cranes to blocks

Since we are interested in assigning the cranes to blocks, we can view the problem as if we are 'matching' cranes with blocks (or vice versa). In other words, a crane matched to a block is essentially assigning the crane to block. The reason why we want to treat the task of assignment as a 'matching problem' is because then we can use an algorithm that is similar in construction as 'deferred acceptance algorithm' (DAA). The DAA is a matching model first introduced by Gale and Shapley (1962) in their famous paper "College admission and stability of marriage". Since the paper was published it has generated numerous follow up studies by researchers in Economics and Computer science. DAA has been applied to various real world matching problems such as assigning students to schools, people to jobs, nurses to residencies etc. DAA is able to find a match between two sets of agents in a two-sided market, where each set of agents have preferences over the other set of agents to which they wish to match. The basic idea of DAA is that the agents from one side of the market propose, in their order of preferences, to the agents on other side of market. Then the set of agents receiving the proposals review and reject (also in their order of preferences) and final acceptance is deferred until the last step of the DAA. For detailed information regarding the algorithm and various relevant theoretical results to date see Roth (2008). DAA have two versions depending on which side of the market are proposing. Another variation of the model is 'one-to-one' matching vs. 'many-to-one' matching. A marriage model is an example of a one-to-one matching since each man is matched to at most one woman or vice versa, whereas a college admission model is a many-to-one matching problem since each student can be matched to at most one college but a college can be matched to more than just one student. An important note is that matching algorithms are generally studied in context of their 'stability' property, which is not in our interest. DAA does not provide any mechanism to generate preferences for agents, it is assumed that true preferences of agents are known. We assume that the preferences functions we offeredforthe cranes and blocks are their true preferences.

For the inter-block deployment problem we use an algorithm similar to many-to-one matching version of DAA because in a planning period a crane can be matchedto only one block while a block can be matched to more than one crane. We present two versions of our algorithm here, namely (1) crane proposing version (2) block proposing version. For illustration of algorithm, we define 'quota' q_i for a block *i* as the maximum number of cranes a block can hold at some timeminus the number of cranes currently located in the block i.e. $(NC_b^{max} - NC_b^{current})$. From the previous section we knowthat, each crane has strict preferences defined over the set of blocks, and each block has strict preferences defined over the set of cranes, and a matching is to be determined that will assign each crane *j* to no more than one block, and each block *i* to no more than $q_i$cranes.

Crane proposing version

Consider the following steps.

1. Each crane *j* proposes to first block *i* from its preference list (if crane *j*'s preference list is not empty).

2. Each block *i* receiving more than q_i proposals, 'holds' the most preferred q_i cranes and rejects all others.

n. Each crane *j* rejected at step *n – 1* removes the block *i* rejecting the crane from its preference list. Then the rejected crane *j* makes a new proposal to its next most preferred block *i* who hasn't yet rejected it. (if crane *j* 's preference list is not empty). Go to step *n – 1*.

Stop: when no further proposals are made, that is, no cranes are rejected or the rejected cranes preference list is empty.

Finally, match the blocks to the cranes whose proposals they are holding. (if any)

Block proposing version

Consider the following steps.

1. Each block *i* proposes to its most preferred q_i cranes from its preference list (if block *i*'s preference list is not empty).

2. Each crane j who received at least one proposal, 'holds' the most preferred block and rejects all others.
n. Each block j who is rejected by one or more cranes at step $n - 1$ will remove those cranesfrom its preference list. Let, the number of rejections a block j has received is r_j. The rejected block$j$ makes a newproposal to its next $r_j$preferred cranes to whom the block has not proposed already (if crane j's preference list is not empty). Go to step $n - 1$.
Stop: when no further proposals are made, that is, no blocks are rejected or the rejected blocks' preference lists are empty.

Finally, match the cranes to the blocks whose proposals they are holding. (if any)

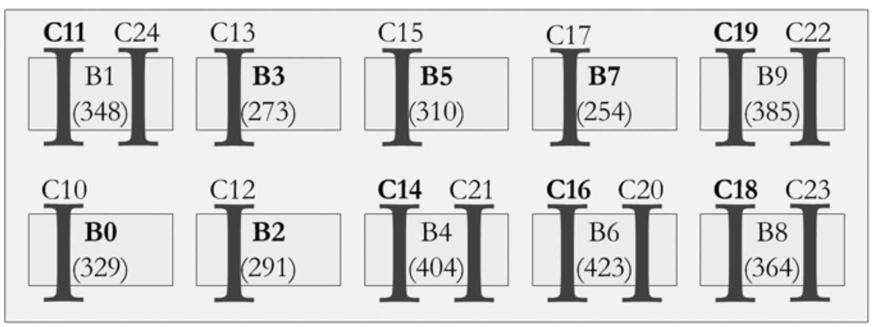
3.5. A sample example

In this section we review how our model solves a sample crane deployment scenario. The scenario is illustrated in Figure 4 that considers a problem with 10 yard blocks (rectangles) and15 yard cranes (*I* shaped footprints) and the layout is as shown. Block IDs are preceded with the letter 'B' and crane IDs are preceded with the letter 'C'. The work volume for a block at the beginning of a planning period in minutes is shown within the parenthesis. The length of planning period is 4 hours or 240 minutes. The initial distribution of 15 yard cranes to 10 blocks is obtained using the 'reduce transfers' assignment strategy, which is the location of cranes at the beginning of the planning period as shown in Figure 4a. Then we run the pre-analysis steps to find out the blocks and cranes that will participate in further analysis. There are 5 cranes with available extra capacity and they are C11, C14, C16, C18 and C19 (shown in bold face). Also there are 5 blocks that needs help and they are B0, B2, B3, B5 and B7 (in bold face). Now we generate preference lists for these cranes and block agents using 'minimum transfer time' strategy. The lists are:

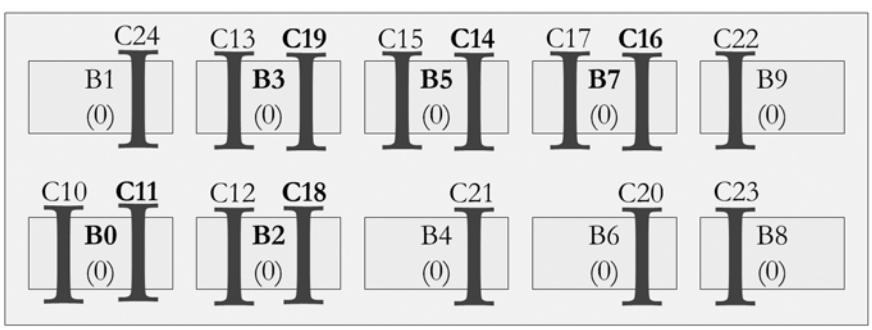
$\succ_{CRANE11}$: block 0 $\succ$ block 3 $\succ$ block 2 $\succ$ block 5 $\succ$ block 7
$\succ_{CRANE16}$: block 7 $\succ$ block 5 $\succ$ block 2 $\succ$ block 3 $\succ$ block 0
$\succ_{CRANE14}$: block 5 $\succ$ block 2 $\succ$ block 3 $\succ$ block 7 $\succ$ block 0
$\succ_{CRANE19}$: block 7 $\succ$ block 5 $\succ$ block 3 $\succ$ block 2 $\succ$ block 0
$\succ_{CRANE18}$: block 7 $\succ$ block 5 $\succ$ block 2 $\succ$ block 3 $\succ$ block 0
$\succ_{BLOCK3}$: crane 11 $\succ$ crane 14 $\succ$ crane 16 $\succ$ crane 19 $\succ$ crane 18
$\succ_{BLOCK0}$: crane 11 $\succ$ crane 14 $\succ$ crane 16 $\succ$ crane 18 $\succ$ crane 19
$\succ_{BLOCK2}$: crane 14 $\succ$ crane 11 $\succ$ crane 16 $\succ$ crane 18 $\succ$ crane 19
$\succ_{BLOCK5}$: crane 14 $\succ$ crane 16 $\succ$ crane 11 $\succ$ crane 19 $\succ$ crane 18
$\succ_{BLOCK7}$: crane 16 $\succ$ crane 19 $\succ$ crane 14 $\succ$ crane 18 $\succ$ crane 11

Next we apply the algorithm in Section 3.4.2 to solve for matching using the crane proposal version which yield the following matching- (block 0, crane 11); (block 3, crane 19); (block 2, crane 18); (block 7, crane 16); (block 5, crane 14). The final locations of the cranes after transfer to their matched blocks are shown in Figure 4b. For all blocks the work volume is zero at the end of the planning period. Thus percentage incomplete workload is also zero. If we used the mathematical program proposed by

Linn et al. (2003) we would obtain the same results. For the majority of the cases our model is able to find optimal or near-optimal solutions. This is evident from Tables 2 and 3 where percentage incomplete work volume found by our model is close to mathematical program.Note that, some of the preferences in this example are symmetric, that is, the crane's first choice is a block whose first choice is the crane (e.g. Block 0 and Crane 11; Block 7 and Crane 16; Crane 14 and Block 5). It may appear that we do not need the algorithm to find the crane-block pairs, since we can just put the cranes in their preferred blocks. However this is not always the case. Notice that Block 3 and Block 0 both wants Crane 11; Block 0 gets it because it is the first item of Crane 11's list; Block 3 ends up getting 4th choice). This example uses the same preference strategies for block and crane agents. However, we can pick different preference strategies; for example, the cranes may use 'minimum transfer time', whereas the blocks may use 'absolute difference'.



(a) Initial state of yard at the beginning of planning period



(B) Final state of yard at the end of planning period

Figure 4. A sample craned employment scenario.

4. IMPLEMENTATION AND RESULTS FROM EXPERIMENTS

The aforementioned methodologies were implemented in Netlogo, a multi-agent simulation framework (Wilensky,1999). Netlogo facilitates experimentation and evaluation of the proposed paradigm. It provides many useful primitives (i.e. procedural commands) that are particularly suitable for this implementation. In our framework, blocks and cranes are modeled as stationary and mobile agents, respectively. Figure 5 shows a screenshot of our model and graphical user interface (GUI). As shown, the model provides several sliders for ease of changing various parameters. The parameters that could be changed directly on the GUI include the number of blocks, number of cranes, work volume level, selection of initial crane assignment strategy, and preference functions for agents.

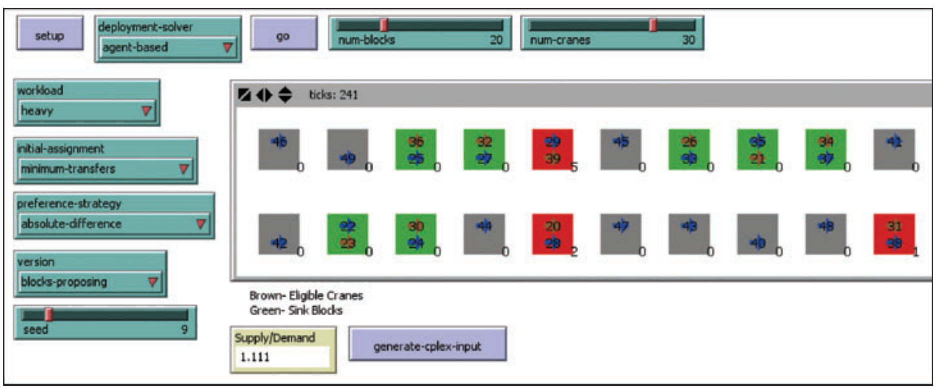


Figure 5. A screen shot of the agent-based model.

The implementation consists of the following steps in sequence:

1. Setup the layout of blocks
2. Choose a work volume level for the planning period
3. Generate work volume for each block
4. Create cranes
5. Choose initial crane assignment strategy
6. Assign the cranes to blocks
7. Compute inter-block travel time and the transfer time matrix
8. Run the pre-analysis steps to filter blocks and cranes that will participate in further analysis
9. Choose between block proposal and crane proposal version
10. Run the chosen version and solve for matching
11. Reassign cranes according to the solution
12. Update graphics and record results

We tested our model against various real-world sized crane deployment problems.

The test parameters that were used for experiments are shown in Table 1. Some of these parameters describing the size of the problem are varied over realistic ranges. Other parameters are various options to modify the steps of analysis. Fifty replications are run for combinations of the test parameters. The performance measures recorded were total incomplete work volume and total crane idling at the end of the planning period. For comparative purposes, we also performed a run where a centralized mathematical program developed by Linn et al. (2003) is used to solve the test problems. As shown in Table 1, the highest number of blocks in our experiment is 30, a case which is comparable to a fairly large real world problem. As the number of blocks becomes large in a container terminal, the yard area is partitioned into multiple yard zones, each yard zone consisting of a group of blocks. In practice the yard cranes are not moved from one zone to another zone in a planning period. The numbers of cranes were set to be equal to the number of blocks or 50% higher than the number of blocks. To assign the work volume to blocks in a planning period, various assignment procedures or work volume severity (supply-demand ratio) is assumed. Work volume condition has two parts 1) total work volume of blocks compared to capacity of all cranes and 2) distribution of work volume among the blocks. Table 1 lists three different work volume conditions-moderate, heavy and above capacity. *Moderate* work volume condition implies that the total work volume is 60% of the total available crane capacity. Total available crane capacity is the number of cranes times the capacity of a single crane. The distributions of work volume among the blocks are such that work volume can be 40% higher or lower than the average work volume per block. Average work volume per block is the total work volume divided by the number of blocks. *Heavy* work volume conditions imply that the total work volume is 90% of the total available crane capacity. The distributions of work volume among the blocks are such that the work volume can be 20% higher or lower than the average work volume per block. *Above capacity* work volume condition implies that the total work volume is 110% of the available crane capacity. The distributions of work volume among the blocks are such that work volume can be 40% higher or lower than the average work volume per block. The rest of the parameters of Table 1 (i.e. initial assignment, preference strategy and version) and their values are as described in Section 3.

Table 1. Values of parameters used in experiments.

Parameter	Value	Unit
Number of blocks	10, 20, 30	Nos
Number of cranes	1 or 1.5 × Number of blocks	Nos
Length of planning period	240	Minutes
Work volume	(1) Moderate (2) Heavy (3) Above capacity	Minutes
Initial assignment	(1) Random (2) Crane at each block (3) High to low work volume (4) Reduce transfers	–
Preference strategy	(1) Minimum transfer time (2) Absolute difference (3) Positive difference (4) Absolute inverse squared distance	–
Version	(1) Blocks proposing (2) Cranes proposing	–

In Tables 2 and 3 the percentage incomplete work volumes are listed for various cases in the experiments. The column headings 'M', 'H' and 'AC' refer to 'medium', 'heavy' and 'above capacity' work volume conditions. The percentage of incomplete work volume at the end of a planning period can be computed as in Equation 18.

$$\textit{Percentage incomplete work volume} = \frac{\textit{Initial Work Volume - Finished Work Volume}}{\textit{Initial Work Volume}} \times 100\% \tag{18}$$

We show results for those combinations of parameters that provide the best performance from our model i.e. minimize the total incomplete work volume of blocks at the end of the planning period. The results in Table 2 and 3 are based on 'Reduce transfers' chosen as initial assignment since this strategy producesthe best results. Also,the results are not influenced by the 'version' of algorithm such as 'Cranes Proposing' or 'Block Proposing' because as long as we use the same preference strategy for block agents and crane agents, the results will be indifferent. For 'medium' condition the percentage incomplete work volume is always zero no matter what preference functions we use or the size of the problem. For 'heavy' conditions, the percentage incomplete work volume is also very low and always less than or equal to 1% remaining unfinished. For 'above capacity' conditions, the percentage of incomplete work volume is also very promising, within 3% of the optimal solution found by mathematical program. Note that in the above capacity condition demand exceeds supply, therefore it is not possible for the cranes to complete all work volumes. In fact, even if we disregard the time loss by transferring cranes among blocks, it can be easily shown that there will always remain at least 9.1% of the work volume incomplete. For this case the 'minimum transfer time' preference function appears to be consistently the best strategy.

Table 2. Percentage in complete work volume:
Case I-average number of cranes per block=1.0

Number of Blocks		10			20			30	
Number of Cranes		10			20			30	
Work Volume	M	H	AC	M	H	AC	M	H	AC
Minimum Transfer Time	0	0.13	11.34	0	0.10	11.14	0	0.12	11.28
Absolute difference	0	0.16	12.16	0	0.13	12.09	0	0.11	12.16
Positive difference	0	0	13.55	0	0.05	13.08	0	0.04	12.86
Absolute difference squared distance	0	0.13	11.80	0	0.08	11.69	0	0.10	11.81
Mathematical Program	0	0	10.57	0	0	10.57	0	0	10.60

Table 3. Percentage incomplete work volume: Case II-average number of cranes per block = 1.5

Number of Blocks		10			20			30	
Number of Cranes		15			30			45	
WorkVolume	M	H	AC	M	H	AC	M	H	AC
Minimum Transfer Time	0	0.02	9.79	0	0.20	10.00	0	0.48	10.04
Absolute difference	0	0.07	10.19	0	0.18	10.45	0	0.43	10.48
Positive difference	0	0.14	10.40	0	0.53	10.73	0	1.00	10.71
Absolute difference squared distance	0	0.01	10.06	0	0	10.31	0	0.01	10.38
Mathematical Program	0	0	9.69	0	0	9.77	0	0	9.79

Our model was created using Netlogo version 4.1.3 running on a personal computer with 2.57GHz Centrino dual-core CPU and 4 Gigabytes of RAM. The experiments were run using the 'BehaviorSpace', a tool integrated with NetLogo. The computation time to find a solution using our deployment algorithm is very short; a problem with 10 blocks can be solved in less than a second, and a problem with 30 blocks can solved in less than 3 seconds.

5. CONCLUSION AND FUTURE WORK

This paper presented a study on the inter-block crane deployment problem in a marine container terminal. The deployment problem is an integral part of the daily decision making for terminal operators and stevedores. The goal of our study was to best utilize the capacity of cranes to minimize the work volume that remains incomplete. We explored various strategies for how to assign the cranes among blocks at the beginning of a planning period based on the work volume forecast. Adopting an agent-based approach, we presented preference functions to generate preferences for crane and block agents. These preference functions are intuitive and constructed based on the parameters that influence the best utilization of cranes' capacities. We applied the deferred acceptance algorithm based on these preferences of agents to establish effective relocations of cranes during a planning period. The results showed that our model provides an excellent solution in short time for a range of work volume conditions with high variation. In 'medium' condition all work can be finished within planning period, in 'heavy' condition the percentage remaining incomplete is less than or equal to 1%, in 'above capacity' condition the percentage remaining incomplete is within 3% of the optimal. Our model is scalable to large sized problems; a test case with 30 blocks can be solved within 3 seconds.

There are a number of ways in which this work could be extended. In future work, we plan to consider relocating cranes multiple times within a planning period. In this study, we limited the relocation of yard cranes to once per crane per planning period. In addition, we plan to extend the model to include forecasts for multiple planning periods in making deployment decisions.Anotherdirection this research could be taken is to solve the integrated problem involving the inter-block crane scheduling with quay crane scheduling and/or drayage truck scheduling.

REFERENCES

Bernhardt, K., 2007. Agent-based modeling in transportation. *Artificial Intelligence in Transportation*, 72.

Cheung, R., Li, C., Lin, W., 2002. Interblock crane deployment in container terminals. *Transportation Science* 36 (1), 79.

Crainic, T., Kim, K., 2007. Intermodal transportation. *Transportation* 14, 467–537.

Davidsson, P., Johansson, S., Persson, J., Wernstedt, F., 2003. Agent-based approaches and classical optimization techniques for dynamic distributed resource allocation: A preliminary study. *AAMAS'03 workshop on Representations and Approaches for Time-Critical Decentralized Resource/Role/Task Allocation*. Citeseer.

Franz T, Voß S, Rölke H (2007) Market-mechanisms for integrated container terminal management. In: *Bertram V (ed) Proceedings of the 6th international conference on computer and IT applications in the maritime industries*, COMPIT'07, Cortona, April 23–25. INSEAN, pp 234–248

Gale, D., Shapley, L., 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69 (1), 9–15.

He, J., Chang, D., Mi, W., Yan, W., 2010. A hybrid parallel genetic algorithm for yard crane scheduling. *Transportation Research Part E: Logistics and Transportation Review* 46 (1), 136–155.

Henesey, L., 2006. *Multi-agent systems for container terminal management*. Citeseer.

Huynh, N., Vidal, J., 2010. An agent-based approach to modeling yard cranes at seaport container terminals. In *proceeding of the Symposium on Theory of Modeling and Simulation*.

N. Huynh and J.M. Vidal, 2012.A Novel Methodology for Modeling Yard Cranes at Seaport Terminals to Support Planning and Real-Time Decision Making. *International Journal of Six Sigma and Competitive Advantage*. 7(1), 62-91

Kim, K. H., Lee, K. M., Hwang, H., 2003. Sequencing delivery and receiving operations for yard cranes in port container terminals. *International Journal of Production Economics* 84 (3), 283–292.

Lee, D., Cao, Z., Meng, Q., 2007. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *International Journal of Production Economics* 107 (1), 115–124.

Li, W., Wu, Y., Petering, M. E. H., Goh, M., Souza, R. d., 2009. Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research* 198 (1), 165–172.

Linn, R., Liu, J., Wan, Y., Zhang, C., Murty, K. G., 2003. Rubber tired gantry crane deployment for container yard operation. *Computers and Industrial Engineering* 45 (3), 429–442.

Linn, R., Zhang, C., 2003. A heuristic for dynamic yard crane deployment in a container terminal. *IIE transactions* 35 (2), 161–174.

Murty, K., Liu, J., Wan, Y., Linn, R., 2005. A decision support system for operations in a container terminal. *Decision Support Systems* 39 (3), 309–332.

Ng, W. C., 2005. Crane scheduling in container yards with inter-crane interference. *European Journal of Operational Research* 164 (1), 64–78.

Ng, W. C., Mak, K. L., 2005. Yard crane scheduling in port container terminals. *Applied Mathematical Modelling* 29 (3), 263–276.

Rashidi, H., Tsang, E., 2006. Container terminals: Scheduling decisions, their formulation and solutions. Submitted to the *Journal of Scheduling*.

Roth, A., 2008. Deferred acceptance algorithms: History, theory, practice, and open questions. International *Journal of Game Theory* 36 (3), 537–569.

Sharif, O., Huynh, N., Vidal, J.M. 2011. Application of El Farol model for managing marine terminal gate congestion. *Research in Transportation Economics*. 32 (1), 81-89.

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

Steenken, D., VoSS, S., Stahlbock, R., 2005. *Container terminal operation and operations research a classification and literature review*. pp. 3–49.

Thurston T, Hu H (2002) Distributed agent architecture for port automation. In: *Proceedings of the 26th annual international computer software and applications conference (COMPSAC'02), Oxford,* August 26–29. IEEE Computer Society, Los Alamitos, pp 81–87

Vacca, I., Bierlaire, M., Salani, M., 2007. Optimization at container terminals: status, trends and perspectives. In: *7th Swiss Transport Research Conference*, Monte Verita/Ascona. Citeseer, pp.12–14.

Vis, I. F. A., de Koster, R., 2003. Transshipment of containers at a container terminal: An overview.European *Journal of Operational Research* 147 (1), 1–16.

Wilensky, U., 1999. *Netlogo itself: Netlogo*. *Center for Connected Learning and Computer-BasedModeling*, Northwestern University, Evanston, IL.

Yan, W., Huang, Y., He, J., 2008. An investigation into dynamic yard crane deployment and comparisons between hill-climbing and best-first-search algorithms. *International Journal of Computer Applications in Technology* 32 (4), 254–264.

Zhang, C., Wan, Y., Liu, J., Linn, R. J., 2002. Dynamic crane deployment in container storage yards. Transportation Research Part B: *Methodological* 36 (6), 537–555.