8-2001

Analysis of a New Vehicle Scheduling and Location Problem

Ebru K. BISH
Virginia Polytechnic Institute and State University

Thin Yin LEONG
Singapore Management University, tyleong@smu.edu.sg

Chun-Lun LI
Hong Kong University of Science and Technology

Jonathan W. C. NG
University of Hong Kong

David SIMCHI-LEVI
Massachusetts Institute of Technology

DOI: https://doi.org/10.1002/nav.1024

Published in Naval Research Logistics, Volume 48, Issue 5, August 2001, Pages 363-385
https://doi.org/10.1002/nav.1024

Analysis of a New Vehicle Scheduling and Location Problem

Ebru K. Bish,[1] Thin-Yin Leong,[2] Chung-Lun Li,[3] Jonathan W. C. Ng,[4] David Simchi-Levi[5]

[1] *Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061*

[2] *PSA Corporation Ltd., Singapore*

[3] *Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Hong Kong, People's Republic of China*

[4] *Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong, People's Republic of China*

[5] *Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts*

Abstract: We consider a container terminal discharging containers from a ship and locating them in the terminal yard. Each container has a number of potential locations in the yard where it can be stored. Containers are moved from the ship to the yard using a fleet of vehicles, each of which can carry one container at a time. The problem is to assign each container to a yard location and dispatch vehicles to the containers so as to minimize the time it takes to download all the containers from the ship. We show that the problem is NP-hard and develop a heuristic algorithm based on formulating the problem as an assignment problem. The effectiveness of the heuristic is analyzed from both worst-case and computational points of view.
Naval Research Logistics 48: 363–385, 2001

1. INTRODUCTION

In the last few years we have seen the breakdown of many trade barriers and the globalization of trade. These developments have increased the importance of logistics and transportation, and in particular the importance of marine transportation systems. These systems include a network of terminals around the globe that allow manufacturers and shippers to deliver goods quickly to their customers. These terminals serve as hubs for the transshipment of containers from ship to ship or to other modes of transportation, e.g., rail and trucks.

It is well known that in today's transportation industry, speed is everything, and it is no different in marine transportation. Thus, an efficient terminal is one that allows speedy transshipment of

Correspondence to: E.K. Bish
Research supported in part by the PSA Corporation Ltd., Singapore.

containers to and from the ships. Such a speedy operation is important to both the carrier, since it provides significant operational efficiencies, and to the terminal which can handle a large number of ships per day. Unfortunately, in many regions around the globe, the terminals are now working at, or close to, capacity and there is a significant pressure from the political and business sectors to increase terminal throughput and, in particular, *decrease ship turnaround time at the terminal*. In most cases, this requires the development of methodology and tools which will allow the efficient coordination of activities within the terminal area.

When a ship arrives at the terminal, containers are first discharged from the ship onto vehicles by the *quay cranes*; the vehicles then transport the containers to various storage locations. The terminal handles a (typically small) number of ships at a time and each ship is served by a number of quay cranes. A few hours before the arrival of an incoming ship, the terminal receives detailed information about its contents, i.e., containers that are to be discharged into the yard. This information allows the terminal dispatchers to generate the so-called *crane job sequence—*a detailed sequence for each quay crane serving the ship, specifying the order of the containers that are to be discharged from the ship. Thus, at any point in time, the quay crane operator has information on the next container he/she is going to work on. For each such container, the crane list will identify a number of potential storage locations in the yard, typically two to four.

It is of no surprise that managing, controlling and operating such a system is very complex. At the operational level the questions are clear: How should vehicles be dispatched to containers, what is an optimal storage location for each container discharged from the ship, how should vehicles be routed in the terminal area, and what is an effective traffic control mechanism?

Evidently, these issues are interrelated. Unfortunately, solving a single integrated model that addresses all the operational decisions is well beyond today's computing capability. For that reason, in this research we decompose the problem into several related models: dispatching vehicles to containers, assigning discharged containers to specific locations, and routing vehicles (see [3]). Our objective is to analyze each model separately in order to gain an insight into the system. In this paper we focus on the problem of assigning discharged containers to yard locations and dispatching vehicles to containers.

Specifically, we consider a single quay crane with a list containing n containers to be unloaded from the ship. We refer to each container as a *job*. Each unloaded job has to be taken to a location in the storage yard, where it will be stored until its transfer to another ship. Associated with each unloaded job is a set of locations it can be assigned to. These sets of locations for different jobs may overlap. Associated with the single quay crane are k vehicles, $k < n$, which unload and transfer the jobs from the ship to the yard. Each vehicle can carry one job at a time. After delivering the job to a location in the yard, the vehicle will return to the ship to serve other jobs.

Due to the large sizes of the containers, it is only practical for a crane to unload a container onto a vehicle; unloading to the ground would require another crane operation to take the container from the ground and load it onto a vehicle. This constraint requires a vehicle to be available by the crane throughout the unloading operation. The time required to unload a job by the quay crane is denoted by s. We assume that s is deterministic, and is the same for all jobs. Without loss of generality, we assume that each vehicle travels at unit speed, i.e., each vehicle travels one unit of distance per unit time, and all k vehicles are available at the ship at time 0.

As observed earlier, the crane job sequence is always determined before the unloading process starts. This predetermined sequence imposes start-start type of precedence constraints on the jobs. That is, a job cannot be unloaded by the quay crane until all the jobs preceding it in the crane job sequence are unloaded. This implies that a job cannot be taken by a vehicle before its predecessors are taken by the vehicles. We will number the jobs such that the ith job in the crane job sequence will be referred to as job i.

Let $N = \{1, 2, \ldots, n\}$ be the set of all the jobs and $L = \{l_1, l_2, \ldots, l_m\}$ be the set of all the locations in the yard. Let λ_j be the distance of location l_j from the quay crane ($j = 1, \ldots, m$). In practice, containers are not homogeneous in terms of weight and content. Thus, in some cases it may be desirable to store some containers in specific storage locations, taking into account factors such as the container's next destination, weight and contents. For $i = 1, \ldots, n$ and $j = 1, \ldots, m$, let

$$a_{ij} = \begin{cases} 1, & \text{if job } i \text{ \textit{can} be assigned to location } l_j, \\ 0, & \text{otherwise,} \end{cases}$$

$$X_{ij} = \begin{cases} 1, & \text{if job } i \text{ \textit{is} assigned to location } l_j, \\ 0, & \text{otherwise.} \end{cases}$$

Here, a_{ij} is a given parameter and X_{ij} is a decision variable. Since each job must be assigned a location, we have

$$\sum_{j=1}^{m} a_{ij} X_{ij} = 1 \quad (i = 1, \ldots, n). \tag{1}$$

Also, each location cannot be assigned to more than one job, which implies that

$$\sum_{i=1}^{n} a_{ij} X_{ij} \leq 1 \quad (j = 1, \ldots, m). \tag{2}$$

Thus, the vehicle-scheduling-location problem is to determine X_{ij} ($i = 1, \ldots, n$, $j = 1, \ldots, m$) which satisfies constraints (1) and (2), and to schedule the vehicles to carry the jobs to their assigned locations so as to *minimize the makespan*, which is the time the last vehicle returns to the ship after all n jobs are unloaded and are taken to their locations in the yard. We refer to this problem as the *vehicle-scheduling-location problem*.

When the assignment of jobs to locations is given, it is clear that the following *greedy algorithm* minimizes the makespan: "The next unloaded job in the crane sequence is taken by any waiting vehicle or is taken by the first arriving vehicle if no vehicle is waiting" (see Chen et al. [7]). Unfortunately, the vehicle-scheduling-location problem, where the assignment of jobs to locations is also to be determined, is significantly more difficult than the problem analyzed by Chen et al. Indeed, in Section 3, we analyze the computational complexity of the vehicle-scheduling-location problem and show that the problem is NP-hard. This implies that it is unlikely that an efficient, polynomial time algorithm will ever be developed for this problem. Thus, in Section 4. we present a heuristic algorithm, called the *Assignment Problem Based* (APB) heuristic, which, as its name implies, is based on formulating the problem as an assignment problem.

To analyze the performance of the APB heuristic, we need to define two performance measures. The first is the *asymptotic worst-case performance ratio* of a heuristic which is defined as the maximum deviation from optimality for all instances that are sufficiently "large." On the other hand, the *absolute worst-case performance ratio* of a heuristic is defined as the heuristic solution's maximum deviation from optimality for all instances, not necessarily those that are large.

Note that, in practice, the number of containers to be discharged from a ship is usually fairly large, ranging from 100 to 500 containers. Hence, we are especially interested in the performance of the heuristic for large problem sizes, which implies that the asymptotic analysis of the heuristic performance may prove to be helpful. Thus, in Sections 5. and 6. we characterize the absolute and the asymptotic worst-case performance of the APB heuristic. Computational experiments are conducted and the results are reported in Section 7.

In the next section we provide a brief overview of the related literature.

2. LITERATURE REVIEW

Problems associated with dispatching and routing vehicles and locating items or facilities arise frequently in *logistics systems* (see, for instance, Bramel and Simchi-Levi [5]). Thus, these problems have been extensively studied in the operations research/management science literature under different settings including, but not limited to, vehicle fleet management, truck routing, and warehouse management.

Unfortunately, most of this research is not directly applicable to a container terminal operation due to its unique characteristics. This, in turn, requires the development of algorithms that take into account the special characteristics and constraints associated with container terminals. In the following, we restrict the literature review to a review of research related to three topics: container terminals, material handling systems control, and resource-constrained scheduling. This literature deals with issues and problems similar to the one discussed in this paper. This review is not meant to be exhaustive, but rather indicative of the different areas related to the problem analyzed in the current paper.

2. 1. Container Terminals

As mentioned in the previous section, issues related to ship terminal operations have gained attention only recently due to the increased importance of marine transportation systems. Since the most important objective for a container terminal is to increase its throughput, or in particular, to decrease ship turnaround times (see, for instance, McKinsey and Company, Inc. [30]), most of the research has focused on that objective. Atkins [1] was one of the first to provide a detailed description of ship terminal operations, including comparisons of container handling equipment and storage yard strategies. Although there was some published work prior to the eighties that used operations research techniques to analyze terminal operations, it was not until the mid-eighties that researchers started focusing on terminal productivity issues (see Kiesling and Walton [20]).

Most of the literature on container terminals has used queuing theory to analyze terminal operations. These queuing models focus on strategic issues such as determining the equipment capacity, both on the water side (such as berth capacity), and on the land side (such as the number of quay cranes, vehicles, and yard cranes); see, for instance, Daganzo [11], Easa [13], Edmond [14], Gransberg and Basilotto [18], Jones [19], Kiesling and Walton [20], Kozan [28], Mettam [31], Miller [32], Noritake and Kimura [33, 34], Plumlee [36], Stavros [39], and Zoran [42]. Several researchers focus on the operational level issues, such as scheduling the cranes and determining storage locations for the unloaded containers. Kim and his colleagues [22, 24] analyze the problem of determining a storage location for each unloaded container so as to minimize the expected total number of rehandling and propose various approaches, including dynamic programming. Kim and Kim [23, 25] also analyze the problem of routing the cranes in the yard storage area; they formulate the problem as a Mixed Integer Problem and propose optimization algorithms. Kim [21] and Castilho and Daganzo [6] develop methodologies to estimate the number of relocation movements in the yard. Daganzo and Peterkofsky study how quay crane operating strategies affect berth throughput; they propose various heuristics and exact algorithms to solve the crane scheduling problem taking into account berth capacity limitations (see Daganzo [10, 12] and Peterkofsky and Daganzo [35]). Van Hee and Wijbrands [41] propose a decision support system for the capacity planning of container terminals; the decision support system combines several heuristic rules and analyzes the interactions between different elements of a container terminal.

Ship container terminals also share similarities with rail container transportation systems. Unfortunately, the literature that uses mathematical optimization techniques for rail transportation has also been limited (see Cordeau, Toth, and Vigo [9] for a review of optimization models for rail transportation). Powell and Carvalho [37] analyze the problem of dispatching flatcars (vehicles) to containers and trailers in a rail terminal; the model allows for a large variety of flatcars in the terminal, each with different capacities. Bostel and Dejax [4] analyze the operational problems in a rail terminal. They decompose operational decisions into two levels: (1) container location problem, and (2) equipment scheduling problem, and solve these problems separately. They formulate problem (1) as a *multicommodity network flow problem* with binary variables and apply heuristic algorithms due to the problem size. Kozan [29] analyzes the equipment assignment problem at a rail container terminal and proposes a simulation model that utilizes various heuristics.

In this research, we take a different approach from that of the previous research: We analyze the container location and vehicle dispatching problems simultaneously at a ship container terminal, while explicitly considering quay crane capacity and job precedence constraints; our objective is to develop algorithms that are easy to implement for large problem sizes, and whose effectiveness can be characterized analytically.

2. 2. Material Handling Systems Control

Vehicle dispatching and routing strategies have been extensively analyzed, mainly in the context of *Material Handling Systems* (MHS) control. Assuming a fixed shop layout with predetermined material handling flow paths and fixed transporter fleet size, the MHS control problem deals with the assignment of transportation equipment to service requests on the shop floor. Most of the related work has focused on *Automated Guided Vehicle Systems* (AGVS); see Co and Tanchoco [8] and King and Wilson [26] for extensive reviews.

Thus, an important problem in AGVS management is the vehicle dispatching problem. Indeed, vehicle dispatching problems have been analyzed under various assumptions including different carriers (unit-load or multiple-load), path layout and configurations, demand patterns (static or stochastic), and mostly in job shop environments (see, for instance, Bilge and Tanchoco [2], Egbelu [15], Egbelu and Tanchoco [16], Sinriech and Palni [38], and the references within).

A container terminal is similar to an AGVS with a single pickup point, many delivery points, and unit-load AGVs. Evidently, the single pickup point in the AGVS corresponds to the ship area in the container terminal, the delivery points correspond to the container locations and the unit-load AGVs in the AGVS correspond to vehicles in the container terminal.

The main difference between AGVS and a container terminal is that in a container terminal, containers have precedence and storage location constraints, and they compete for a set of *resources*, the cranes: Each container needs to be unloaded from ships by the cranes.

The crane constraint also relates the vehicle-scheduling-location problem to resource-constrained scheduling, discussed in the following section.

2. 3. Resource-Constrained Scheduling

Another problem that is of particular relevance to our research is the *machine interference problem*, which involves scheduling a set of jobs (each with a given processing time) on a set of machines. Before the processing of each job can start, the corresponding machine needs service (setup, loading, unloading, etc.) by an operator who can serve one machine at a time (see, for instance, Kravchenko and Werner [27], Stecke and Aronson [40], and the references within). This problem is mostly studied in robot-served manufacturing environments and in the textile industry. The vehicle-scheduling-location problem is similar to the machine interference problem: Crane operations represent the setups, crane represents the operator, each container corresponds to a job and vehicles to machines. However, the assignment of locations to containers (and thus, the job processing times in the machine interference problem) is part of the decision in the vehicle-scheduling-location problem, not a parameter as is the case for the machine interference problem.

3. COMPUTATIONAL COMPLEXITY

In this section we analyze the computational complexity of the vehicle-scheduling-location problem.

THEOREM 1: The vehicle-scheduling-location problem is NP-hard.

PROOF: We transform the Even–Odd Partition problem, which is known to be NP-complete [17], to the vehicle-scheduling-location problem. The Even–Odd Partition problem is defined as follows.

Even–Odd Partition: Given a set of $2r$ elements $Y = \{1, 2, \ldots, 2r\}$ and a size $x_i \in Z^+$ for each $i \in Y$, and $B = \frac{1}{2}\sum_{i\in Y} x_i$, does there exist $Y' \subset Y$ such that $\sum_{i\in Y'} x_i = \sum_{i\in Y\setminus Y'} x_i = B$ and exactly one element of $\{2i-1, 2i\}$ belongs to Y' for $i = 1, \ldots, r$?

Given an arbitrary instance of Even–Odd Partition, we construct a corresponding instance of the vehicle-scheduling-location problem with

$$
\begin{aligned}
&k = 2,\\
&n = 2r,\\
&m = 2r,\\
&s = 0,\\
&\lambda_i = x_i + 2B, \quad \text{for } i = 1, \ldots, 2r,\\
&a_{2i-1,2i-1} = a_{2i-1,2i} = a_{2i,2i-1} = a_{2i,2i} = 1, \quad \text{for } i = 1, \ldots, r \quad (\text{other } a_{ij} \text{ values are } 0).
\end{aligned}
$$

Let $M = 2(1 + 2r)B$. We set the crane job sequence in this problem instance as $(1, 2, \ldots, 2r)$. Note that in this construction, either $X_{2i-1,2i-1} = X_{2i,2i} = 1$ or $X_{2i-1,2i} = X_{2i,2i-1} = 1$, for $i = 1, \ldots, r$. We will show that there exists an even–odd partition of the set Y if and only if there exists a schedule for the vehicle-scheduling-location problem with a makespan no greater than M.

Suppose there exists an even-odd partition $\{Y', Y \setminus Y'\}$ of the set Y. Then, we will form a schedule with no inserted vehicle idle time as follows. For $j = 1, \ldots, 2r$, vehicle 1 will take a job to location l_j if $j \in Y'$, and vehicle 2 will take a job to this location if $j \in Y \setminus Y'$. Specifically, for $i = 1, \ldots, r$, we set $X_{2i-1,2i-1} = X_{2i,2i} = 1$ if $2i - 1 \in Y'$ and vehicle 1 finishes serving its previous job earlier than vehicle 2 or if $2i \in Y'$ and vehicle 2 finishes serving its previous job earlier than vehicle 1. Otherwise, we set $X_{2i-1,2i} = X_{2i,2i-1} = 1$. The makespan of this schedule is

$$\begin{aligned}
\max\left\{\sum_{j\in Y'} 2\lambda_j,\ \sum_{j\in Y\setminus Y'} 2\lambda_j\right\} &= \max\left\{\sum_{j\in Y'} 2(x_j+2B),\ \sum_{j\in Y\setminus Y'} 2(x_j+2B)\right\} \\
&= \max\left\{2(1+2r)B,\, 2(1+2r)B\right\} \\
&= M.
\end{aligned}$$

Conversely, suppose there exists a schedule for the vehicle-scheduling-location problem with makespan no greater than M. Note that $\sum_{j\in Y} 2\lambda_j = 2M$, that is, the total travel time of the two vehicles is equal to $2M$. This implies that the total travel time of each vehicle must be exactly M and there is no idle time for the vehicles. Next, we show that for $i = 1, \ldots, r$, jobs $2i-1$ and $2i$ must not be carried by the same vehicle. Suppose, to the contrary, that they are carried by the same vehicle, then let

$$i' = \min\{i \mid \text{jobs } 2i-1 \text{ and } 2i \text{ are carried by the same vehicle}\}.$$

Without loss of generality, we assume that jobs $2i'-1$ and $2i'$ are carried by vehicle 1. Then, by definition of i', among jobs $1, 2, \ldots, 2i'-2$, exactly half of them must be carried by vehicle 1, and vehicle 1 spends more than $4(i'-1)B$ units of time delivering these jobs (since $\lambda_j > 2B$, each job requires a round-trip travel time of more than $4B$). Furthermore, vehicle 1 spends more than $4B$ units of time to transport job $2i'-1$. Let $S_{2i'}$ denote the time that the crane starts unloading job $2i'$. Then, we have $S_{2i'} > 4i'B$ (see Fig. 1). On the other hand, because $\lambda_i = x_i + 2B$ and $\sum_{i=1}^{2r} x_i = 2B$ and there is no idle time in the vehicle schedule, vehicle 2 will finish transporting its first $i'-1$ jobs no later than time $4i'B$. In other words, vehicle 2 will finish transporting its first $i'-1$ jobs earlier than $S_{2i'}$. This implies that either there is an idle time on vehicle 2 after it finishes transporting the first $i'-1$ jobs (if vehicle 2 continues to transport some other jobs afterward) or the total travel time of vehicle 2 is less than that of vehicle 1 (if vehicle 2 only transports these $i'-1$ jobs), which is a contradiction. Hence, for $i = 1, \ldots, r$, jobs $2i-1$ and $2i$ must be carried by two different vehicles.

Recall that for $i = 1, \ldots, r$, jobs $2i-1$ and $2i$ must be assigned to locations $\{l_{2i-1}, l_{2i}\}$. For $i = 1, \ldots, r$, we assign element $2i-1$ to set Y' and element $2i$ to set $Y \setminus Y'$ if vehicle 1 carries a job to location l_{2i-1} and vehicle 2 carries a job to location l_{2i}; we assign element $2i$ to set Y' and element $2i-1$ to set $Y \setminus Y'$ if vehicle 1 carries a job to location l_{2i} and vehicle 2 carries a job to location l_{2i-1}. Thus, total travel time of vehicle 1 is $\sum_{j\in Y'} 2\lambda_j$ and the total travel time of vehicle 2 is $\sum_{j\in Y\setminus Y'} 2\lambda_j$. Therefore,

$$\sum_{j\in Y'} 2\lambda_j = \sum_{j\in Y\setminus Y'} 2\lambda_j = M,$$

implying

$$\sum_{j\in Y'} 2(x_j+2B) = \sum_{j\in Y\setminus Y'} 2(x_j+2B) = 2(1+2r)B,$$

which in turn implies that

$$\sum_{j\in Y'} x_j = \sum_{j\in Y\setminus Y'} x_j = B.$$

Hence, $\{Y', Y \setminus Y'\}$ is an even-odd partition of Y. This completes the proof of the theorem. □

Note that in the problem instance that we constructed in the proof of Theorem 1, the unloading time s is set equal to 0. This implies that the special case where $s = 0$ is also NP-hard (a discussion of this case appears in Section 5.). Furthermore, in the problem instance that we used in the above proof, two locations are allowed for each job (i.e., $\sum_{j=1}^{m} a_{ij} = 2$ for $i = 1, \ldots, n$). Thus, the special case where only a small number of possible locations are available for each job remains NP-hard.

4. AN ASSIGNMENT PROBLEM BASED HEURISTIC ALGORITHM

As proven in the previous section, the vehicle-scheduling-location problem is NP-hard, which indicates that the existence of an efficient algorithm to solve the problem optimally is unlikely. Thus, we need to develop an efficient heuristic to solve this problem.

Recall that λ_j is the distance of location l_j from the quay crane. Let $\lambda_{\max} = \max_{j=1,\ldots,m}\{\lambda_j\}$ and $\lambda_{\min} = \min_{j=1,\ldots,m}\{\lambda_j\} > 0$. Since the yard area is bounded, $\lambda_{\max}$ is also bounded. The following Assignment Problem (AP) formulation will be used in our heuristic solution procedure:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j X_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{m} a_{ij} X_{ij} = 1 \quad (i = 1, \ldots, n), \\
& \sum_{i=1}^{n} a_{ij} X_{ij} \le 1 \quad (j = 1, \ldots, m), \\
& X_{ij} = 0 \text{ or } 1 \quad (i = 1, \ldots, n;\ j = 1, \ldots, m).
\end{aligned} \tag{3}
$$

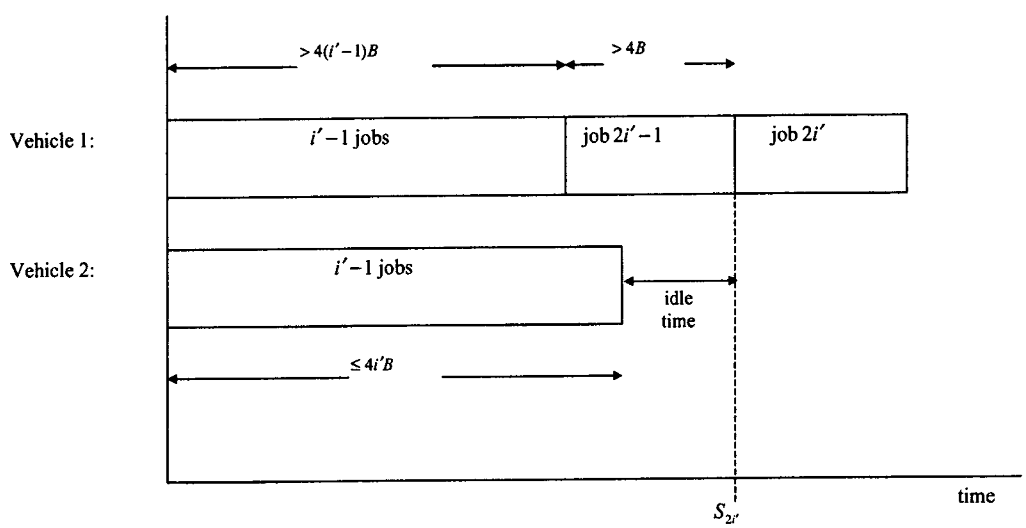


Figure 1. Vehicle schedule when jobs $2i' - 1$ and $2i'$ are both carried by Vehicle 1.

Note that the assignment problem differs from the vehicle-scheduling-location problem in that it completely ignores the vehicle schedule and assigns jobs to locations purely based on their travel distances from the ship.

Given any feasible solution X_{ij} $(i = 1, \ldots, n, j = 1, \ldots, m)$ to the AP, let $d_i = \sum_{j=1}^{m} \lambda_j X_{ij}$ for $i \in N$. That is, d_i is the distance between the ship and the location assigned to job i. Thus, any feasible job-location assignment will be represented by a *job distance vector* $\mathbf{d} = (d_1, d_2, \ldots, d_n)$. Since the unloading time of a job is s and the vehicles travel at unit speed, the total time for a vehicle to serve job i and return to the ship is $s + 2d_i$. Let $d_{\max} = \max_{i=1,\ldots,n}\{d_i\}$, which is bounded.

As observed earlier, for each feasible job-location assignment, i.e., a feasible solution to AP, with a job distance vector $\mathbf{d}$, there is a schedule that minimizes the makespan. Such a minimum makespan schedule corresponding to any given feasible job-location assignment is obtained by the Greedy Algorithm, where the next unloaded job in the crane job sequence is taken by the vehicle that arrives at the crane the earliest after finishing its previous job (see Chen et al. [7]).

For any given feasible job-location assignment with a job distance vector $\mathbf{d}$, we let $C_j^*(\mathbf{d})$ denote the time when vehicle j $(j = 1, \ldots, k)$ finishes serving all its jobs and returns to the ship in the minimum makespan schedule. Let $C_{\min}^*(\mathbf{d}) = \min_{j=1,\ldots,k}\left\{C_j^*(\mathbf{d})\right\}$ and $C_{\max}^*(\mathbf{d}) = \max_{j=1,\ldots,k}\left\{C_j^*(\mathbf{d})\right\}$. Thus, $C_{\max}^*(\mathbf{d})$ is the optimal makespan corresponding to a feasible job-location assignment with a job distance vector $\mathbf{d}$.

Consider the optimal solution X_{ij}^0 $(i = 1, \ldots, n, j = 1, \ldots, m)$ to the AP, and let $L^0 = \{l_j \mid \sum_{i=1}^{n} X_{ij}^0 = 1;\ j = 1, \ldots, m\}$, that is, L^0 is the set of locations that are assigned to jobs in this solution. The locations in set L^0 will be called *used locations*. Let $\mathbf{d}^0 = (d_1^0, d_2^0, \ldots, d_n^0)$ be the job distance vector corresponding to this optimal solution of the AP, that is, $d_i^0 = \sum_{j=1}^{m} \lambda_j X_{ij}^0$ for $i = 1, \ldots, n$. Let $d_{\max}^0 = \max_{i=1,\ldots,n}\{d_i^0\}$ and $D^0 = \sum_{i=1}^{n} d_i^0$.

Also, consider the job-location assignment in the optimal solution to the vehicle-scheduling-location problem. Let $\mathbf{d}^* = (d_1^*, d_2^*, \ldots, d_n^*)$ be the job distance vector corresponding to this optimal solution, and let $d_{\max}^* = \max_{i=1,\ldots,n}\{d_i^*\}$ and $D^* = \sum_{i=1}^{n} d_i^*$. Thus, the optimal makespan of the problem is given as $Z^* = C_{\max}^*(\mathbf{d}^*)$. By the optimality of the AP, we have

$$D^0 \leq D^*. \tag{4}$$

Since all the vehicles are identical, we may assume that if there is more than one vehicle waiting at the ship for the unloading of their jobs, then the vehicle that arrives at the crane the earliest takes the next job. In other words, we assume that the queuing discipline of the vehicles at the crane is first-come first-served.

Recall that for a given feasible job-location assignment, the Greedy Algorithm can be used to obtain a minimum makespan schedule for the vehicles to deliver the jobs. We now suggest the following Assignment Problem Based (APB) heuristic algorithm for solving the vehicle-scheduling-location problem.

STEP 1: Solve the Assignment Problem in (3) and assign locations to jobs based on the AP solution X_{ij}^0 $(i = 1, \ldots, n, j = 1, \ldots, m)$.

STEP 2: Apply the Greedy Algorithm to the jobs in the set N, with a job distance vector $\mathbf{d}^0 = (d_1^0, d_2^0, \ldots, d_n^0)$. In the Greedy Algorithm, the first k jobs are assigned, each to a single vehicle. We then assign the next job in the list to the first available vehicle that arrives at the crane after completing its previous job.

The idea of this heuristic is to decompose the vehicle-scheduling-location problem into two isolated steps, where the first step determines the location assignments by ignoring the vehicle schedule and the second step determines the vehicle schedule for the location arrangements obtained from the first step. Let Z^H denote the makespan of the schedule obtained by the APB heuristic, that is, $Z^H = C^*_{\max}(\mathbf{d}^0)$. In what follows we discuss some basic properties of the heuristic solution.

Recall that L is the set of all locations in the yard. We start by arranging all the locations in the set L in nondecreasing order of distances from the ship, and denote the ith location in this sequence as $l_{[i]}$ and its distance from the ship as $d_{[i]}$. We assume that $d_{[i]} \neq d_{[j]}$, for all $i, j \in L$ with $i \neq j$. However, the proofs of the following lemmas can easily be extended to the case where some locations have the same distances. Let $L(p) = \{l_{[1]}, l_{[2]}, \ldots, l_{[p]}\}$, for $p = 1, \ldots, m$. Consider the optimal solution X^0_{ij} ($i = 1, \ldots, n, j = 1, \ldots, m$) to the AP, with an objective value of D^0, and the corresponding set of locations L^0. We let $l_{[f]}$ be the location that is farthest away from the ship in the set L^0.

LEMMA 2: There exists no feasible solution to the AP which uses only locations in the set $L(f-1)$.

PROOF: Suppose, to the contrary, that there exists at least one feasible solution to the AP which uses only locations in the set $L(f-1)$. By the optimality of X^0_{ij} ($i = 1, \ldots, n$; $j = 1, \ldots, m$), the objective value of any such feasible solution must be greater than or equal to D^0. Also, by definition of $l_{[f]}$, we have $d_{[j]} < d_{[f]}$, for all $l_{[j]} \in L(f-1)$.

Suppose that job i_0 has been assigned to $l_{[f]}$ in the optimal solution X^0_{ij} ($i = 1, \ldots, n$, $j = 1, \ldots, m$) of the AP. In order to have a feasible solution in $L(f-1)$, job i_0 has to be assigned to another location in $L(f-1)$. There exists at least one location, say location $l_{[j_1]} \in L(f-1)$, to which job i_0 can be assigned, i.e., $a_{i_0,[j_1]} = 1$, since otherwise no feasible solution exists in $L(f-1)$. Clearly, location $l_{[j_1]}$ is used by another job, say job i_1, in the optimal solution X^0_{ij}, since otherwise a solution with less cost ($D^0 - d_{[f]} + d_{[j_1]} < D^0$) could be obtained by reassigning job i_0 to location $l_{[j_1]}$ with no conflicts.

If job i_0 is assigned to $l_{[j_1]}$, then job i_1 should be assigned to another location in $L(f-1)$ to obtain a feasible solution. All locations in $L(f-1)$ to which job i_1 can be assigned must be used by other jobs in the optimal solution X^0_{ij}, since otherwise a solution with less cost could be obtained by assigning job i_0 to $l_{[j_1]}$ and job i_1 to one of the unused locations in $L(f-1)$, say $l_{[j_2]}$, with no conflicts, and the cost would be $D^0 - d_{[f]} + d_{[j_2]} < D^0$. So job i_1 should be assigned to a used location $l_{[j_2]}$ in $L(f-1)$ when job i_0 is assigned to $l_{[j_1]}$ to obtain a feasible solution in $L(f-1)$.

We can repeat the same argument for all the jobs, that is, if job i_s is assigned to the location used by job i_{s+1} in the optimal solution X^0_{ij}, then job i_{s+1} should be assigned to another used location. Therefore, in this feasible solution to the AP, all n jobs are assigned to the used locations in $L(f-1)$. However, by definition of $L(f-1)$, there are only $n-1$ used locations in $L(f-1)$, which is a contradiction. Hence, a feasible solution in the set $L(f-1)$ does not exist, which completes the proof of the property. $\square$

The following corollary is a direct consequence of Lemma 2.

COROLLARY 3: $d^0_{\max} \leq d^*_{\max}$.

In the following sections, we use these properties to characterize the performance of the APB heuristic. Specifically, in Section 5. we analyze the case where crane unloading times are very small. Section 6. extends this analysis to the general case where crane unloading times can be any positive value.

5. THE ZERO UNLOADING TIME CASE

In this section we analyze the performance of the APB heuristic when the unloading time at the crane, s, is equal to 0. This case applies to the practical situation where the unloading time at the crane is much smaller than travel times from the ship to the yard locations. Unfortunately, the complexity analysis in Section 3. indicates that even the zero unloading time case is NP-hard.

To analyze the performance of the APB heuristic in this case, we start by introducing upper and lower bounds on the makespan of any solution to the problem in Lemma 4. These bounds will then be used to derive a lower bound on the optimal makespan and an upper bound on the heuristic makespan in Theorems 5 and 7, when we characterize the performance of the APB heuristic.

In the following analysis, we assume that the problem is feasible, i.e., given any arbitrary subset of jobs, there is always a feasible assignment of jobs to yard locations.

LEMMA 4: Given a job distance vector $\mathbf{d} = (d_1, d_2, \ldots, d_n)$, if $s = 0$, then the schedule obtained by the Greedy Algorithm satisfies the following condition:

$$\frac{1}{k}\sum_{i=1}^{n} 2d_i \leq C^*_{\max}(\mathbf{d}) \leq \frac{1}{k}\sum_{i=1}^{n-1} 2d_i + 2d_{\max}.$$

PROOF: The total distance traveled by all the k vehicles is $\sum_{i=1}^{n} 2d_i$. Thus, the makespan of the schedule, $C^*_{\max}(\mathbf{d})$, is bounded from below by $\sum_{i=1}^{n} 2d_i/k$. Now, consider the first $n-1$ jobs and let $\hat{\mathbf{d}} = (d_1, d_2, \ldots, d_{n-1})$ be the job distance vector of these jobs. Thus, $C^*_j(\hat{\mathbf{d}})$ is the time when vehicle j finishes serving all its jobs in the minimum makespan schedule for the first $n-1$ jobs, and $C^*_{\min}(\hat{\mathbf{d}}) = \min_{j=1,\ldots,k}\{C^*_j(\hat{\mathbf{d}})\}$. From the Greedy Algorithm and because $s = 0$, we have

$$C^*_{\max}(\mathbf{d}) \leq C^*_{\min}(\hat{\mathbf{d}}) + 2d_{\max}.$$

Since $s = 0$, there is no vehicle idle time in any schedule generated by the Greedy Algorithm. Hence, $\sum_{j=1}^{k} C^*_j(\hat{\mathbf{d}}) = \sum_{i=1}^{n-1} 2d_i$, which implies that

$$C^*_{\min}(\hat{\mathbf{d}}) = \min_{j=1,\ldots,k}\left\{C^*_j(\hat{\mathbf{d}})\right\} \leq \frac{1}{k}\sum_{j=1}^{k} C^*_j(\hat{\mathbf{d}}) = \frac{1}{k}\sum_{i=1}^{n-1} 2d_i.$$

Therefore,

$$C^*_{\max}(\mathbf{d}) \leq \frac{1}{k}\sum_{i=1}^{n-1} 2d_i + 2d_{\max}. \qquad \square$$

The next theorem states that the APB heuristic has an absolute worst-case performance ratio of 2. That is, the heuristic will generate a solution for the zero unloading time case with no more than 100% relative error.

THEOREM 5: If $s = 0$ then $Z^H/Z^* \leq 2$.

PROOF: In the optimal solution, the total distance traveled by all the k vehicles is $\sum_{i=1}^{n} 2d_i^* = 2D^*$. Thus, the makespan of the optimal schedule, $C_{\max}^*(\mathbf{d}^*)$, is bounded from below by $2D^*/k$. Therefore, by (4),

$$Z^* \geq \frac{2D^0}{k}. \tag{5}$$

Clearly, $Z^* \geq 2d_{\max}^*$. Hence, by Corollary 3,

$$Z^* \geq 2d_{\max}^0. \tag{6}$$

We thus have

$$\begin{aligned} Z^H &= C_{\max}^*(\mathbf{d}^0) \\ &\leq \frac{1}{k}\sum_{i=1}^{n-1} 2d_i^0 + 2d_{\max}^0 \quad \text{(by Lemma 4)} \\ &\leq \frac{2D^0}{k} + 2d_{\max}^0 \\ &\leq Z^* + Z^* \quad \text{[by (5) and (6)]}. \end{aligned}$$

Thus, $Z^H/Z^* \leq 2$. □

The next theorem indicates that this worst-case error bound is tight.

THEOREM 6: There exists an instance of the vehicle-scheduling-location problem with $s = 0$ in which the ratio Z^H/Z^* is arbitrarily close to 2.

PROOF: See Appendix A. □

Next we analyze the asymptotic worst-case performance of the APB heuristic. As mentioned previously, this case is of practical interest due to the large number of jobs involved in real cases. The next theorem states that the asymptotic worst-case performance ratio of the APB heuristic is 1 when $s = 0$. That is, when the crane unloading time is very small, the heuristic finds the schedule that minimizes the makespan for a large number of jobs. Thus, the following result is quite strong.

THEOREM 7: Given n jobs, let Z_n^* and Z_n^H denote the optimal makespan and the APB heuristic makespan, respectively. If $s = 0$ and k is constant, then

$$\lim_{n\to\infty} Z_n^H/Z_n^* = 1.$$

PROOF: From Lemma 4, we have

$$Z_n^H = C_{\max}^*(\mathbf{d}^0) \leq \frac{1}{k}\sum_{i=1}^{n-1} 2d_i^0 + 2d_{\max}^0$$

and

$$Z_n^* = C_{\max}^*(\mathbf{d}^*) \geq \frac{1}{k} \sum_{i=1}^{n} 2d_i^*.$$

Thus,

$$\frac{Z_n^H}{Z_n^*} \leq \frac{\sum_{i=1}^{n-1} 2d_i^0/k + 2d_{\max}^0}{\sum_{i=1}^{n} 2d_i^*/k} \leq \frac{2D^0/k + 2d_{\max}^0}{2D^*/k} = \frac{D^0}{D^*} + \frac{kd_{\max}^0}{D^*}.$$

Note that $Z_n^H \geq Z_n^*$ and $D^0 \leq D^*$ (from (4)). Hence,

$$1 \leq \frac{Z_n^H}{Z_n^*} \leq 1 + \frac{kd_{\max}^0}{D^*}.$$

Since $\lambda_{\min} > 0$, $D^* \to \infty$ as $n \to \infty$. This, together with the fact that $d_{\max}^0$ is bounded, implies that $Z_n^H/Z_n^* \to 1$ as $n \to \infty$. $\square$

The next section extends this analysis to the more general case.

6. THE GENERAL CASE

In this section we consider the general case where the unloading time at the quay crane is any positive value. Our objective is to characterize the performance of the APB heuristic for the general case. For this purpose, we again determine bounds on the solution values of the problem. Specifically, Lemma 8 provides a lower bound on the optimal makespan, whereas Lemma 9 provides an upper bound on the heuristic makespan. These bounds will then be used in Theorems 10 and 11 to analyze the absolute and asymptotic performance of the APB heuristic.

LEMMA 8: $Z^* \geq \max\left\{\frac{2D^0}{k} + \frac{ns}{k} + \frac{(k-1)s}{2},\, 2d_{\max}^0 + s,\, ns\right\}$.

PROOF: The total travel time and unloading time of the k vehicles in any feasible schedule is at least $2D^0 + ns$. A vehicle has to wait for another vehicle to finish unloading before it can perform its unloading. Since we are assuming that all the vehicles are available at the ship at time 0, the first vehicle does not wait for the crane, the second vehicle waits for s units of time, the third vehicle waits for $2s$ units of time, and so on. Thus, the total waiting time for the first k jobs is given by $\sum_{i=1}^{k}(i-1)s = k(k-1)s/2$. Therefore, the makespan of any feasible schedule must be no less than $\frac{1}{k}\left[2D^0 + ns + k(k-1)s/2\right]$, or, equivalently,

$$Z^* \geq \frac{2D^0}{k} + \frac{ns}{k} + \frac{(k-1)s}{2}.$$

Clearly, $Z^* \geq 2d_{\max}^* + s$, and by Corollary 3, we have

$$Z^* \geq 2d_{\max}^0 + s.$$

Finally, the total unloading time, at the ship, of all the jobs is ns, which implies that

$$Z^* \geq ns.$$

This completes the proof of the lemma. $\square$

LEMMA 9: $Z^H \le \frac{2D^0}{k} + \frac{(n-1)s}{k} + \frac{(k-1)s}{2} + \frac{(n-k-1)(k-1)s}{k} + s + 2d^0_{\max} + (k-1)s$.

PROOF: We first consider the first $n-1$ jobs. Let $\hat{\mathbf{d}}^0 = (d^0_1, \ldots, d^0_{n-1})$. The total travel time and unloading time of the vehicles spent on these $n-1$ jobs in the heuristic schedule is $2D^0 + (n-1)s$. As explained in the proof of Lemma 8, the total waiting time of the vehicles for the first k jobs is $\sum_{i=1}^{k}(i-1)s = k(k-1)s/2$. Since the waiting time of a vehicle for any particular job can be no more than $(k-1)s$, the total waiting time of the vehicles for jobs $k+1, k+2, \ldots, n-1$ is bounded from above by $(n-k-1)(k-1)s$. Thus,

$$\sum_{j=1}^{k} C^*_j(\hat{\mathbf{d}}^0) \le 2D^0 + (n-1)s + k(k-1)s/2 + (n-k-1)(k-1)s,$$

which implies

$$\begin{aligned} C^*_{\min}(\hat{\mathbf{d}}^0) = \min_{j=1,\ldots,k} \{C^*_j(\hat{\mathbf{d}}^0)\} &\le \frac{1}{k}\sum_{j=1}^{k} C^*_j(\hat{\mathbf{d}}^0) \\ &\le \frac{1}{k}\Big[2D^0 + (n-1)s + k(k-1)s/2 + (n-k-1)(k-1)s\Big]. \end{aligned}$$

The unloading time, travel time, and waiting time of job n is no greater than $s + 2d^0_{\max} + (k-1)s$. Therefore,

$$Z^H \le \frac{1}{k}\Big[2D^0 + (n-1)s + k(k-1)s/2 + (n-k-1)(k-1)s\Big] + s + 2d^0_{\max} + (k-1)s. \qquad \square$$

The next theorem states that the APB heuristic has an absolute worst-case ratio of 3. That is, it will generate a solution with no more than 200% error for the general case.

THEOREM 10: $Z^H/Z^* \le 3$.

PROOF: From Lemmas 8 and 9, we have

$$\begin{aligned} \frac{Z^H}{Z^*} &\le \frac{\frac{2D^0}{k} + \frac{(n-1)s}{k} + \frac{(k-1)s}{2}}{Z^*} + \frac{s + 2d^0_{\max}}{Z^*} + \frac{\frac{(n-k-1)(k-1)s}{k} + (k-1)s}{Z^*} \\ &\le 1 + 1 + \frac{\frac{(n-k-1)(k-1)s}{k} + (k-1)s}{ns} \\ &\le 3. \qquad \square \end{aligned}$$

The next theorem states that the asymptotic worst-case ratio of the APB heuristic is 2 in the general case.

THEOREM 11: $\lim_{n\to\infty} Z^H_n/Z^*_n \le 2$, where Z^*_n and Z^H_n denote the optimal makespan and the APB heuristic makespan, respectively.

PROOF: From Lemmas 8 and 9, we have

$$\begin{aligned}
\frac{Z_n^H}{Z_n^*} &\leq \frac{\frac{2D^0}{k}+\frac{(n-1)s}{k}+\frac{(k-1)s}{2}}{Z_n^*}+\frac{\frac{(n-k-1)(k-1)s}{k}+s+2d_{\max}^0+(k-1)s}{Z_n^*} \\
&\leq 1+\frac{\frac{(n-k-1)(k-1)s}{k}+s+2d_{\max}^0+(k-1)s}{ns} \\
&\leq 1+\frac{ns+2d_{\max}^0}{ns} \\
&= 2+\frac{2d_{\max}^0}{ns} \\
&\rightarrow 2, \quad \text{as } n\rightarrow\infty,
\end{aligned}$$

since $d_{\max}^0$ is bounded. □

Next, we consider the special case where all the locations in the storage yard are at least a certain distance away from the ship. Namely, $\lambda_{\min} \geq (k-1)s/4$, that is, the time required to travel from the ship to any location in the yard is no less than $(k-1)/4$ times the unloading time. This case is motivated by parameters typical of our industry partner. We will derive a tighter error bound for the APB heuristic in this case. In what follows, we first determine an upper bound on the vehicle waiting time at the cranes. We then use this result to derive an upper bound on the heuristic makespan.

LEMMA 12: Let W_i $(i=1,\ldots,n)$ be the vehicle waiting time at the crane corresponding to job i in the schedule obtained by the APB heuristic. If $\lambda_{\min} \geq (k-1)s/4$, then $\sum_{i=1}^{n-1} W_i \leq \frac{(n-1)(k-1)s}{2}$.

PROOF: See Appendix B. □

Observe that the upper bound on vehicle waiting times at the crane obtained in Lemma 12 is stronger than that used in Lemma 9. Recall that Lemma 9 uses $(k-1)s$ to bound the vehicle waiting time for each job $k+1,\ldots,n$. This can only be realized when each vehicle waits for all the remaining $k-1$ vehicles in the crane queue. Next we use Lemma 12 to derive an upper bound on the heuristic makespan for this special case.

LEMMA 13: If $\lambda_{\min} \geq (k-1)s/4$, then $Z^H \leq \frac{2D^0}{k}+\frac{(n-1)s}{k}+\frac{(n-1)(k-1)s}{2k}+s+2d_{\max}^0+(k-1)s$.

PROOF: By Lemma 12 and using the same argument as in the proof of Lemma 9. □

Now we can characterize the asymptotic performance of the APB heuristic for this special case.

THEOREM 14: If $\lambda_{\min} \geq (k-1)s/4$ and k is constant, then $\lim_{n\rightarrow\infty} Z_n^H/Z_n^* \leq \frac{3}{2}-\frac{1}{2k}$, where Z_n^* and Z_n^H denote the optimal makespan and the APB heuristic makespan, respectively.

PROOF: From Lemma 13,

$$\begin{aligned}
Z_n^H &\leq \frac{2D^0}{k}+\frac{(n-1)s}{k}+\frac{(n-1)(k-1)s}{2k}+s+2d_{\max}^0+(k-1)s \\
&= \frac{2D^0}{k}+\frac{(n-1)s}{k}+\frac{(k-1)s}{2}+\frac{(n-k-1)(k-1)s}{2k}+s+2d_{\max}^0+(k-1)s.
\end{aligned}$$

Hence, by Lemma 8, we have

$$\begin{aligned}
\frac{Z_n^H}{Z_n^*} &\leq \frac{\frac{2D^0}{k} + \frac{(n-1)s}{k} + \frac{(k-1)s}{2}}{Z_n^*} + \frac{\frac{(n-k-1)(k-1)s}{2k} + s + 2d_{\max}^0 + (k-1)s}{Z_n^*} \\
&\leq 1 + \frac{\frac{(n-k-1)(k-1)s}{2k} + s + 2d_{\max}^0 + (k-1)s}{ns} \\
&= \frac{3}{2} - \frac{1}{2k} + \frac{k^2+1}{2nk} + \frac{2d_{\max}^0}{ns} \\
&\to \frac{3}{2} - \frac{1}{2k}, \quad \text{as } n \to \infty,
\end{aligned}$$

since $d_{\max}^0$ is bounded. $\square$

It is interesting to find out whether this worst-case error bound is tight. A partial answer is provided in [3] who showed that the bound proved in Theorem 14 is tight at least for all odd values of k.

7. COMPUTATIONAL EXPERIMENTS

We now present the computational study carried out to analyze the empirical performance of the APB heuristic.

The terminal area considered in the numerical study is depicted in Figure 2. In this diagram, the black node represents the ship area, and each yard block in the yard area consists of a set of container storage locations. All vehicles in the system are identical, and vehicle travel times in the terminal area are deterministic and known.

In the computational study, we consider a ship with n discharged containers. A single quay crane and k vehicles are assigned to the ship. For each discharged container we generate 2–4

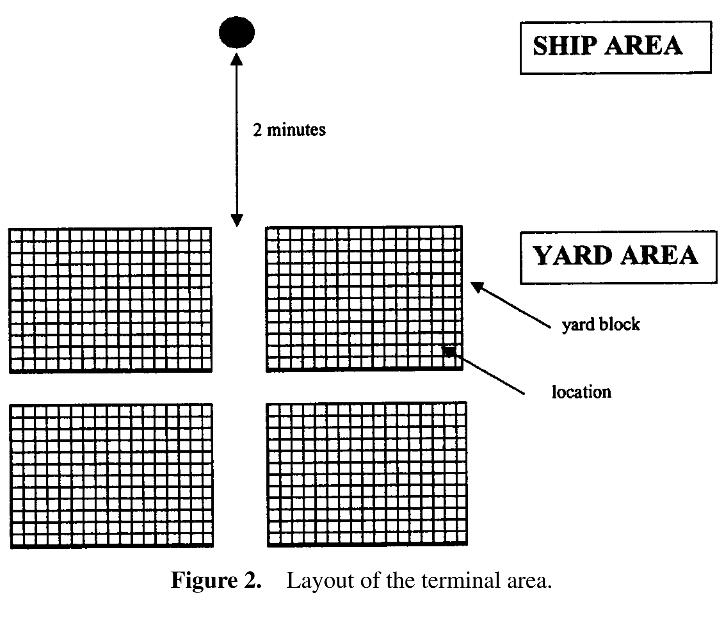


Figure 2. Layout of the terminal area.

Table 1. Average (maximum) deviations of Z^H from the lower bound.

n	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
20	3.7% (9.2%)	8.1% (16.9%)	11.7% (24.4%)	15.4% (31.0%)	21.7% (42.9%)	26.7% (48.0%)
30	2.8% (5.8%)	5.4% (11.2%)	9.6% (21.2%)	11.2% (27.8%)	15.1% (28.3%)	18.0% (35.0%)
50	2.4% (5.0%)	5.6% (9.2%)	8.6% (14.9%)	13.1% (19.3%)	9.9% (15.7%)	11.7% (19.2%)
100	1.9% (3.2%)	4.5% (6.8%)	7.0% (10.0%)	11.1% (17.2%)	15.9% (20.0%)	9.6% (16.0%)
500	1.5% (2.0%)	3.4% (4.2%)	5.8% (7.2%)	9.5% (11.4%)	14.1% (15.8%)	5.4% (7.8%)

possible storage locations in the yard area. The travel time of each location is generated from a uniform distribution between 2 and 20 minutes. The quay crane discharging time s is taken to be deterministic and is 1.5 minutes per job. These parameters are based on data we received from our industry partner, who is involved in managing a mega container terminal.

Before presenting the computational results, we remind the reader that in practice, n is typically between 100 and 500 while k is about 5. Of course, in the numerical study we tested the performance of the heuristic under a wide range of parameter values.

To calculate the effectiveness of the APB heuristic, we need to compare its makespan to the optimal one. Unfortunately, finding the optimal makespan requires solving a Mixed Integer Program (MIP), which can be quite time-consuming for large-size problems (see [3] for the MIP formulation). Consequently, we evaluate the performance of the heuristic relative to the lower bound on the optimal makespan developed in Lemma 8. These results are reported in Table 1.

Specifically, for each combination of n and k, we randomly generate 100 instances and determine the deviation of the heuristic makespan from the lower bound. Table 1 thus presents for each combination of n and k, *the average percentage deviation* of the heuristic makespan from the lower bound over all the 100 replications. The table also provides *the maximum percentage deviation* of the heuristic makespan from the lower bound over all replications. These numbers are given in parentheses.

Table 1 leads to the following conclusions: (i) For a given number of vehicles, the relative error generally decreases as n, the number of containers, increases. This is consistent with the asymptotic worst-case analysis results. (ii) All errors are reasonably small. Indeed, even for small number of discharged containers, say 20 containers, the error is at most 26.7% on the averages and no more than 48.0% over all instances. These results are consistent with our absolute worst-case analysis result provided by Theorem 10. (iii) More importantly, in practice n is between 100 and

Table 2. Average deviation of Z^H from Z^* and Z^* from the lower bound.

$n = 20$	Average Deviation of Z^H from Z^*	Average Deviation of Z^* from Lower Bound
$k = 2$	2.8%	0.9%
$k = 3$	2.0%	6.0%
$k = 4$	6.1%	5.3%
$k = 5$	8.3%	6.5%
$k = 6$	10.5%	10.1%
$k = 7$	13.3%	11.8%

500 while k is about 5. The numerical study shows that in this case the relative errors are fairly small; an error of at most 11.1% on the averages and at most 17.2% over all instances.

We must also note that all these percentage errors are determined relative to a lower bound on the optimal makespan, which may be far away from the optimal makespan. Hence, in Table 2 we analyze the effectiveness of the lower bound relative to the optimal makespan as well as the effectiveness of the heuristic relative to the optimal solution, for small-size problems with 20 jobs. This is done by comparing the heuristic, the lower bound, and the optimal makespan determined by solving the MIP formulation mentioned above. Specifically, the second column in Table 2 reports the average percentage deviation of the heuristic makespan from the optimal makespan over all 100 replications, whereas the third column reports the average percentage deviation of the optimal makespan from the lower bound over all 100 replications.

We have the following observations: (i) From the third column of Table 2, we observe that for a given number of jobs ($n = 20$), the quality of the lower bound deteriorates as the number of vehicles increases. This can be explained as follows. As the number of vehicles increases, vehicles may wait for longer times in the quay crane queue, thus increasing the optimal makespan. Unfortunately, the lower bound (see Lemma 8) does not include any vehicle waiting time component. (ii) By comparing the second column of Table 2 with the case of $n = 20$ in Table 1, we observe that the actual percentage errors reported in Table 2 are significantly less than those reported in Table 1. This suggests that the true performance of the APB heuristic is in fact better than the percentages given in Table 1, which are determined relative to the lower bound.

8. CONCLUSIONS

Our analysis of the vehicle-scheduling-location problem reveals that while the problem is indeed quite difficult, effective heuristics can be developed. Of course, in practice, other issues need to be incorporated into the analysis and addressed by the algorithms. These include identifying routes for each vehicle so as to avoid congestion, coordinating yard crane work load, etc. These are typically terminal specific issues that are addressed at a more detailed level. However, the insight and algorithm developed in this research should be at the core of any solution method used for container location.

APPENDIX A

PROOF OF THEOREM 6: Consider an instance with k vehicles, $n = 2k$ jobs, $m = 2k$ locations, $s = 0$, and the following parameters:

$$\begin{aligned}
&\lambda_i = k, \quad \text{for } i = 1, \ldots, k,\\
&\lambda_i = 1, \quad \text{for } i = k+1, \ldots, 2k,\\
&a_{ii} = 1, \quad \text{for } i = 1, \ldots, 2k,\\
&a_{k,2k} = a_{2k,k} = 1,
\end{aligned}$$

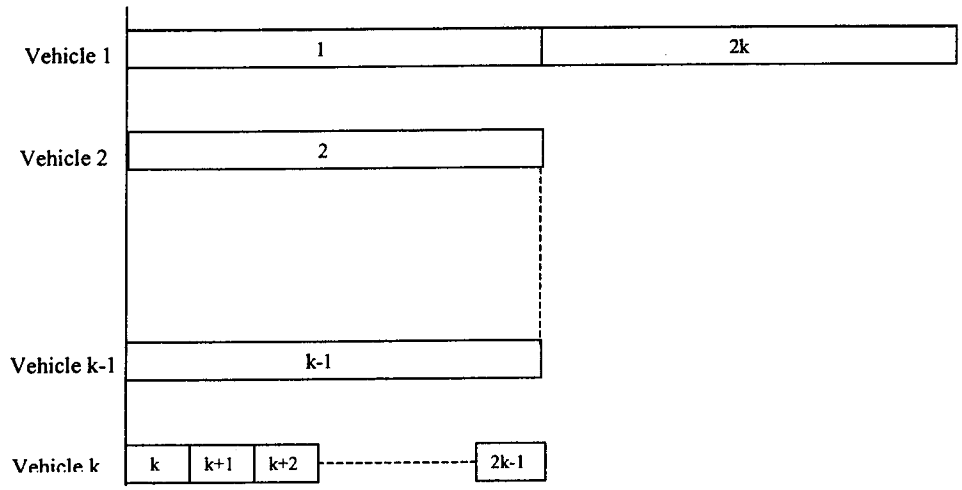
and all other a_{ij} values are 0.

There are two optimal solutions to the AP for this problem. One of the optimal solutions to the AP is given by the following assignments:

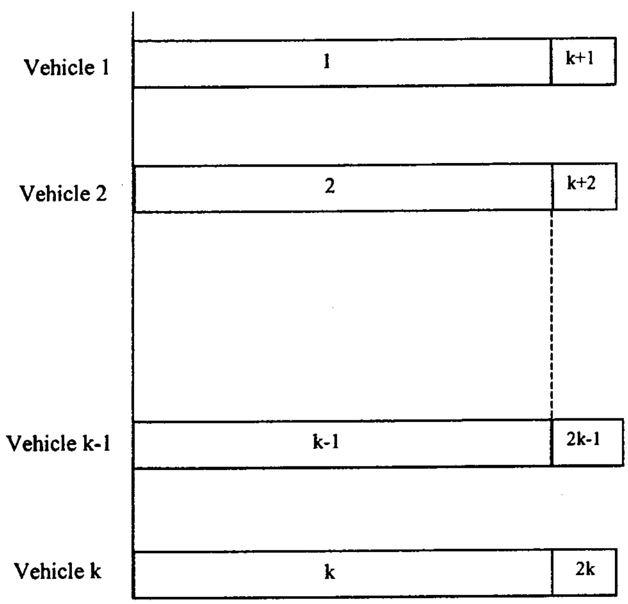
$$\begin{aligned}
&X^0_{k,2k} = X^0_{2k,k} = 1,\\
&X^0_{ii} = 1, \quad \text{for } i = 1, \ldots, k-1, k+1, \ldots, 2k-1.
\end{aligned}$$

Hence,

$$\mathbf{d}^0 = (\underbrace{k, \ldots, k}_{k-1}, \underbrace{1, \ldots, 1}_{k}, k).$$



(a) Vehicle schedule obtained by the APB Heuristic



(b) Optimal vehicle schedule

Figure 3. Example with $s = 0$.

When the Greedy Algorithm is applied to this assignment, we obtain the following schedule of jobs on the k vehicles:

Vehicle 1: jobs $1, 2k$ (total distance traveled = $4k$),

Vehicle 2: job 2 (total distance traveled = $2k$),

Vehicle 3: job 3 (total distance traveled = $2k$),

...

Vehicle $k-1$: job $k-1$ (total distance traveled = $2k$),

Vehicle k: jobs $k, k+1, \ldots, 2k-1$ (total distance traveled = $2k$).

Hence, $Z^H = C^*_{\max}(\mathbf{d}^0) = 4k$ [see Fig. 3(a)].

The optimal solution has the following assignments:

$$X^*_{ii} = 1, \quad \text{for } i = 1, \ldots, 2k.$$

Hence,

$$\mathbf{d}^* = (\underbrace{k, \ldots, k}_{k}, \underbrace{1, \ldots, 1}_{k}).$$

The optimal schedule of jobs on the k vehicles is:

Vehicle 1: jobs $1, k+1$ (total distance traveled = $2k+2$),

Vehicle 2: jobs $2, k+2$ (total distance traveled = $2k+2$),

...

Vehicle k: jobs $k, 2k$ (total distance traveled = $2k+2$).

Hence, $Z^* = C^*_{\max}(\mathbf{d}^*) = 2k+2$ [see Fig. 3(b)].

Therefore,

$$\frac{Z^H}{Z^*} = \frac{4k}{2k+2} \to 2, \quad \text{as } k \to \infty. \qquad \square$$

APPENDIX B

To prove Lemma 12, we first introduce some notation and discuss some properties on the vehicle waiting times.

Consider the schedule obtained by the APB heuristic, let S_i be the time that the crane starts unloading job i, for $i = 1, \ldots, n-1$. We define *uninterrupted job sequence* as a sequence of jobs that are served continuously by the crane, that is, with no idle crane time in between. Thus, if we ignore job n, then a given crane job sequence $(1, 2, \ldots, n-1)$ is partitioned into $u \geq 1$ uninterrupted job sequences

$$\Big\{(1, \ldots, i_1), (i_1+1, \ldots, i_2), \ldots, (i_{u-1}+1, \ldots, i_u)\Big\},$$

where $i_u = n-1$. Let $i_0 = 0$.

We consider the $j\text{th}$ ($j = 1, \ldots, u$) uninterrupted job sequence $(i_{j-1}+1, \ldots, i_j)$. Let q be the number of jobs in this sequence, that is, $q = i_j - i_{j-1}$. We have the following properties on this uninterrupted job sequence.

PROPERTY 1: *$W_{i_{j-1}+h} \leq (h-1)s$, for $h = 1, \ldots, q$.*

PROOF: Since the crane is available at time $S_{i_{j-1}+1}$, the vehicle that carries job $i_{j-1}+1$ does not need to wait for the crane, and for $h = 1, \ldots, q$, the vehicle that carries job $i_{j-1}+h$ will wait for the crane for no more than $(h-1)s$ units of time. $\square$

PROPERTY 2: *If $q > k$ and $\lambda_{\min} \geq (k-1)s/4$, then $W_{i_{j-1}+h} \leq (k-1)s/2$ for $h = k+1, \ldots, q$.*

PROOF: Suppose, to the contrary, that there exists a job $i_{j-1}+h$ ($k+1 \leq h \leq q$) with a vehicle waiting time greater than $(k-1)s/2$. Let

$$J = \Big\{i_{j-1}+h-k,\ i_{j-1}+h-k+1,\ \ldots,\ i_{j-1}+h-1\Big\},$$

which contains part of the $j\text{th}$ uninterrupted job sequence. Thus, the jobs in J are continuously unloaded by the crane during the time period $[S_{i_{j-1}+h-k}, S_{i_{j-1}+h}]$. Let v' be the vehicle that serves job $i_{j-1}+h$, and let W' be the waiting time of the vehicle for this job. Thus, vehicle v' arrives at the crane at time $S_{i_{j-1}+h} - W'$ and waits for W' time units before the crane becomes available to unload job $i_{j-1}+h$. No job is being unloaded onto vehicle v' during the time period $[S_{i_{j-1}+h} - W' - (k-1)s/2,\ S_{i_{j-1}+h}]$. Note that $S_{i_{j-1}+h} - W' - (k-1)s/2 < S_{i_{j-1}+h} - (k-1)s = S_{i_{j-1}+h-k+1}$. Therefore, at least two of the jobs in J are served by the same vehicle because (i) J has k elements, (ii) there are k vehicles, and (iii) no job in set J is served by vehicle v'. Let

$$g_1 = \max\left\{\hat{i} \in J \;\middle|\; \text{job } \hat{i} \text{ is served by the same vehicle as one of the jobs in } \{\hat{i}+1, \ldots, i_{j-1}+h-1\}\right\}.$$

Let v'' be the vehicle that serves job g_1. Let g_2 be the next job served by vehicle v'' after vehicle v'' serves job g_1. The time that vehicle v'' finishes carrying job g_1 to its location in the yard and returns to the ship is

$$\begin{aligned} S_{g_1} + s + 2d_{g_1} &\geq S_{i_{j-1}+h-k} + s + 2\lambda_{\min} \\ &= S_{i_{j-1}+h} - ks + s + 2\lambda_{\min} \\ &\geq S_{i_{j-1}+h} - (k-1)s/2 \\ &> S_{i_{j-1}+h} - W'. \end{aligned}$$

According to the first-come first-served queuing discipline, job g_2 should be served by vehicle v' instead of vehicle v'', which is a contradiction. This completes the proof of the property. $\square$

PROOF OF LEMMA 12: If $q \leq k$, then, by Property 1,

$$\sum_{h=1}^{q} W_{i_{j-1}+h} \leq \sum_{h=1}^{q} (h-1)s = \frac{q(q-1)s}{2} \leq \frac{q(k-1)s}{2}.$$

If $q > k$, then, by Properties 1 and 2,

$$\begin{aligned} \sum_{h=1}^{q} W_{i_{j-1}+h} &= \sum_{h=1}^{k} W_{i_{j-1}+h} + \sum_{h=k+1}^{q} W_{i_{j-1}+h} \\ &\leq \sum_{h=1}^{k} (h-1)s + \sum_{h=k+1}^{q} \frac{(k-1)s}{2} \\ &= \frac{k(k-1)s}{2} + \frac{(q-k)(k-1)s}{2} \\ &= \frac{q(k-1)s}{2}. \end{aligned}$$

Thus, in both cases,

$$\sum_{h=1}^{q} W_{i_{j-1}+h} \leq \frac{q(k-1)s}{2} = \frac{(i_j - i_{j-1})(k-1)s}{2}.$$

Applying this result to all the uninterrupted job sequences, we have

$$\sum_{i=1}^{n-1} W_i \leq \sum_{j=1}^{u} \frac{(i_j - i_{j-1})(k-1)s}{2} = \frac{(i_u - i_0)(k-1)s}{2} = \frac{(n-1)(k-1)s}{2}. \quad \square$$

ACKNOWLEDGMENT

The authors would like to thank an anonymous referee for his/her helpful comments and suggestions.

REFERENCES

[1] W.H. Atkins, Modern marine terminal operations and management, San Francisco, The Compage Company, 1983.

[2] U. Bilge and J.M.A. Tanchoco, AGV systems with multi-load carriers: Basic issues and potential benefits. J Manuf Syst 16(3) (1997), pp. 159–174.

[3] E.K. Bish, Theoretical analysis and practical algorithms for operational problems in a mega container terminal, Ph.D. Dissertation, Northwestern University, Evanston, IL, 1999.

[4] N. Bostel and P. Dejax, Models and algorithms for container allocation problems on trains in a rapid transshipment shunting yard, Transport Sci 32(4) (1998), pp. 370–379.

[5] J. Bramel and D. Simchi-Levi, The logic of logistics: Theory, algorithms, and applications for logistics management, Springer-Verlag, New York, 1997.

[6] B.D. Castilho and C.F. Daganzo, Handling strategies for import containers at marine terminals, Transport Res 27B(2) ((1993), pp. 151–166.
[7] Y. Chen, T. Leong, J.W.C. Ng, E.K. Bish, B.L. Nelson, and D. Simchi-Levi, Dispatching vehicles in a mega container terminal, Working paper, Northwestern University, Evanston, IL, 1998.
[8] C.G. Co and J.M.A. Tanchoco, A review of research on AGVS vehicle management, Eng Costs Prod Econ 21 (1991), pp. 35–42.
[9] J. Cordeau, P. Toth, and D. Vigo, A survey of optimization models for train routing and scheduling, Transport Sci 32(4) (1998), pp. 380–404.
[10] C.F. Daganzo, The crane scheduling problem, Transport Res 23B (1989), pp. 159–175.
[11] C.F. Daganzo, The productivity of multipurpose seaport terminals, Transport Sci 24(3) (1990), pp. 205–216.
[12] C.F. Daganzo, Crane productivity and ship delay in ports, Transport Res Record 1251 (1990), pp. 1–9.
[13] S.M. Easa, Approximate queuing models for analyzing harbor terminal operations, Transport Res 21B (1987), pp. 269–286.
[14] E. Edmond, Operation capacity of container berths for scheduled service by queuing theory, Dock Harbor Authority 56 (1975), pp. 230–234.
[15] P.J. Egbelu, Pull versus push strategy for automated guided vehicle load movement in a batch manufacturing system, J Manuf Syst 6(3) (1987), pp. 209–220.
[16] P.J. Egbelu and J.M.A. Tanchoco, Characterization of automated guided vehicle dispatching rules, Int J Prod Res 22(3), (1984), pp. 359–374.
[17] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, Freeman, New York, 1979
[18] D.D. Gransberg and J.P. Basilotto, Cost engineering optimum seaport capacity, Cost Eng 40 (1997), pp. 28–32.
[19] J.H. Jones, Ship turn-around time at the Port of Bangkok, J Waterways Harbors Div 94 (1968), pp. 135–148.
[20] M.K. Kiesling and M. Walton, Loading/unloading operations and vehicle queuing processes at container ports, Working paper, Texas A & M University, College Station, 1995.
[21] K.H. Kim, Evaluation of the number of rehandles in container yards, Comput Ind Eng 32(4) (1997), pp. 701–711.
[22] K.H. Kim and H.B. Kim, Segregating space allocation models for container inventories in port container terminals, Int J Prod Econ 59 (1999), pp. 415–423.
[23] K.H. Kim and K.Y. Kim, An optimal routing algorithm for a transfer crane in port container terminals, Transport Sci 33(1) (1999), pp. 173–176.
[24] K.H. Kim, Y.M. Park, and R. Kwang-Ryul, Deriving decision rules to locate export containers in container yards, Eur J Oper Res 124 (2000), pp. 89–101.
[25] K.Y. Kim and K.H. Kim, A routing algorithm for a single straddle carrier to load export containers onto a containership, Int J Prod Econ 59 (1999), pp. 425–433.
[26] R.E. King and C.M. Wilson, A review of AGV system design and scheduling, Prod Plan Control 2 (1991), pp. 44–51.
[27] S.A. Kravchenko and F. Werner, Parallel machine scheduling problems with a single server, Math Comput Model 26 (1997), pp. 1–11.
[28] E. Kozan, Comparison of analytical and simulation planning models of seaport container terminals, Transport Plan Technol 20 (1996), pp. 235–348.
[29] E. Kozan, Increasing the operational efficiency of container terminals in Australia, J Oper Res Soc 48(2) (1997), pp. 151–161.
[30] McKinsey and Company, Inc., Containerization: The key to low transport, A report for the British Transport Docks Board, London, 1967.
[31] J.D. Mettam, Forecasting delays to ships in port, Dock Harbor Authority 47 (1967), pp. 380–382.
[32] A.J. Miller, Queuing at single-berth shipping terminal, J Waterways Harbors Coastal Eng Div, 97 (1971), pp. 43–56.
[33] M. Noritake and S. Kimura, Optimum number and capacity of seaport berths, J Waterway Port Coastal Ocean Eng 109 (1983), pp. 323–339.
[34] M. Noritake and S. Kimura, Optimum allocation and size of seaports, J Waterway Port Coastal Ocean Eng 116 (1990), pp. 287–299.

[35] R.I. Peterkofsky and C.F. Daganzo, A branch and bound solution method for the crane scheduling problem, Transport Res 24B (1990), pp. 159–172.
[36] C.H. Plumlee, Optimum size seaport, J Waterways Harbors Div 92 (1966), pp. 1–24.
[37] W.B. Powell and T.A. Carvalho, Real-time optimization of containers and flatcars for intermodal Operations, Transport Sci 32(2) (1998), pp. 110–126.
[38] D. Sinriech and L. Palni, Scheduling pickup and deliveries in a multiple-load discrete carrier Environment, IIE Trans 30(11) (1998), pp. 1035–1047.
[39] N.N. Stavros, Berth planning by evaluation of congestion and cost (closure), J Waterways Harbors Div 95 (1969), pp. 419–425.
[40] K.E. Stecke and J.E. Aronson, Review of operator/machine interference models, Int J Prod Res 23 (1985), pp. 129–151.
[41] K.M. van Hee and R.J. Wijbrands, Decision support system for container terminal planning, Eur J Oper Res 34 (1998), pp. 262–274.
[42] R. Zoran, Ship-berth link as bulk queuing system in ports, J Waterway Port Coastal Ocean Eng 118 (1992), pp. 474–495.