

Planning local container drayage operations given a port access appointment system

Rajeev Namboothiri, Alan L. Erera*

*The Logistics Institute, School of Industrial and System Engineering, Georgia
Institute of Technology Atlanta, GA, U.S.A*

Abstract

This paper studies the management of a fleet of trucks providing container pickup and delivery service (drayage) to a port with an appointment-based access control system. Responding to growing access congestion and its resultant impacts, many U.S. port terminals have implemented appointment systems, but little is known about the potential impact of such systems on drayage fleet efficiency. To address this knowledge gap, we develop a drayage operations planning approach based on an integer programming heuristic that explicitly models a port access control system. The approach determines pickup and delivery sequences for daily drayage operations with minimum transportation cost. We use the framework to develop an understanding of the potential productivity impacts of access control systems on drayage firms. Most importantly, we find that it is critical for terminal operators to provide enough access capacity for drayage, since vehicle productivity can be increased by 10 to 24 percent when total access capacity is increased by 30 percent. Furthermore, poor (but not unreasonable) selection of access appointment time slots by drayage firms may result in substantial customer service deficiencies, reducing the number of customers that can be served by up to 4 percent for a fixed level of total access capacity.

Key words: Drayage, Pickup and Delivery.

* Corresponding author.

Email addresses: rnamboot@isye.gatech.edu (Rajeev Namboothiri),
alerera@isye.gatech.edu (Alan L. Erera).

1 Introduction

Continuous growth in global trade volumes has placed significant new burdens on the freight transportation infrastructure. In the United States, substantial growth in the volume of intermodal containers moving through seaports has especially strained the capacity of landside operations. During peak periods, it is not uncommon for queues of container drayage trucks to form both at port entrance gates and also within the facility at container pickup and drop off points. While on-facility rail connections help alleviate truck congestion at some ports, demand for trucking service is likely to remain at high levels.

Growth in U.S. intermodal container trade is not expected to slow; government and industry sources predict that U.S. international trade tonnage may double between 2001 and 2010, along with a resultant doubling of total container trade from 16 million twenty-foot equivalent units (TEUs) to 32 million TEUs (Herberger, 2001). Managing this growth in traffic, and the additional congestion it brings, will clearly be critical to preserve the productivity of maritime supply chains. New security mandates, such as the Transportation Worker Identification Credentialing (TWIC) program, may compound port congestion problems if they lead to increased processing times at port access points.

1.1 Port access control systems and efficiency

Until recently, most U.S. seaport terminals allowed unscheduled access by drayage trucks. At unscheduled terminals, trucks may arrive to pick up and drop off containers any time within gate operating hours. Since the terminal operator has little control over arrivals, such systems may be inefficient; there may be certain time periods during the day when resources are idle, and others when capacity is exceeded. Unscheduled systems can also lead to drayage firm inefficiency. Drivers may experience unnecessary waiting time if they arrive during peak periods, which reduces their available time to perform loaded moves.

In response to growing truck congestion problems both within and outside port gates, many U.S. port terminal operators are now deploying access control systems. Under such systems, drayage firms can make reservations that allow truck access during specific time windows. The terminal operator then limits the total number of appointments available within each time window in order to manage congestion. Predictable workloads enable better resource scheduling and gate productivity for terminal operators, and reduced access wait time allows drayage firms to better utilize tractors and drivers.

While scheduled terminal access systems may lead to efficiency gains, it was primarily environmental concerns that led to their adoption in the U.S. Excessive drayage truck queueing and idling leads to higher diesel engine emissions, a major concern especially for seaports in large metropolitan areas. In California, the legislature addressed this issue in 2003 by enacting Assembly Bill (AB) 2650, commonly known as the “Lowenthal Bill” after key proponent

Assemblyman Alan Lowenthal. AB 2650 restricts the allowable time trucks can idle in queue in and around port terminals to 30 minutes, and fines terminal operators for violations. However, an important additional provision of the bill allows operators to avoid fines if they deploy a drayage appointment system meeting certain specifications (Lowenthal, 2002).

Since the passing of AB 2650, many large container port terminals in California have implemented appointment systems with similar features. Typically, drayage firms can book appointment requests for hour-long time windows up to two weeks in advance of the access day. The only information that must be provided when booking is an identification code for the trucking firm. Therefore, drayage firms can later decide which move requests (either inbound or outbound) to serve with each appointment.

1.2 Research objectives

For drayage firms, the benefits of a port access appointment system hopefully lead to an increase in daily “turns” (revenue-generating moves) per vehicle and enhanced profitability. However, given the additional constraints such systems place on operations, careful planning may be required to attain the maximum benefit. In this paper, we focus on scheduling problems faced by drayage firms whose operations are constrained by a port access appointment system. We develop an optimization-based scheduling framework for determining high-quality truck routes and schedules for a single drayage firm, specifically incorporating constraints that restrict port access during certain time slots. The framework has two primary applications: (1) selection of the best set of appointment reservations for a fleet serving a set of container move requests; and (2) determination of best routes and schedules for a fleet given a pre-selected set of appointments. Using the framework, we then analyze a set of representative test problems to develop an understanding of how the design of a port access control systems may impact drayage firm productivity.

The primary contributions of this research include:

- the development of integer programming optimization models for drayage operational problems with port access time slot capacities, including a first phase model that identifies the maximum revenue set of container move tasks that can be served and a second phase routing and scheduling model;
- the development of a fast heuristic based on column generation that generates near-optimal solutions for the routing and scheduling model; and
- a computational study that indicates that substantial drayage productivity impacts may arise from small variations in access capacities. Impacts are measured by the percentage of customer requests that can be feasibly served and the number of requests that can be served per vehicle in the fleet.

The remainder of this paper is organized as follows. In Section 2, we formally define the mathematical optimization problems to be considered in this research. Next, in Section 3 we

place this work in the context of existing research on routing and scheduling. In Section 4, we develop formulations and solution techniques, and in Section 5, we present computational results. Conclusions are provided in Section 6.

2 Problem Definition

The primary problem considered in this research is now presented. Consider a single drayage firm operating a truck fleet based at a depot location, serving a single port location and a set of surrounding customer locations. Each day, the firm attempts to serve a set of container move requests to and from the port with its available fleet. Furthermore, suppose that the port operates an access control system where its operating hours are divided into a set of equal-duration time slots. During each time slot, the drayage firm is limited by an upper bound on the number of truck accesses to the port, hereafter known as the *slot capacity*. To maximize its long-run profitability, suppose that the firm then attempts to serve as many move requests as possible while minimizing operating costs.

In this research, we assume that operating costs are minimized by minimizing the number of drivers required to perform a set of moves. In local trucking operations where drivers are paid hourly or daily wages and benefits, this is a natural assumption. While drayage firms typically pay drivers (who are often owner-operators) by the move, we believe this objective remains appropriate since it maximizes the number of daily moves per driver which in turn leads to driver satisfaction and reduced turnover costs.

We now formalize the problem definition. To do so, we first introduce a simplified problem that does not consider port access constraints that we denote the *unconstrained drayage problem* UDP. Suppose that a drayage provider has received n_e export container move requests and n_i import requests for a single operating day. Let C be the set of all container move requests, where $C^E \subseteq C$ is the set of export moves and $C^I \subseteq C$ is the set of import moves. Let $E = \{E_1, E_2, \dots, E_{n_e}\}$ represent the set of export customer locations and $I = \{I_1, I_2, \dots, I_{n_i}\}$ represent the set of import customer locations. Note that these locations are not necessarily unique. Each export move request $j \in C^E$ requires a full truck move from an origin customer location E_j to port location P . Similarly, each import move request $k \in C^I$ requires a full truck move from P to a destination customer location I_k . Thus, each truck can move only a single container at a time. Note that if a customer requests multiple container moves, a separate move request is used to represent each container.

The drayage company operates a fleet of vehicles based at a single depot location D , and develops routes for its vehicles given the set of move requests. Each vehicle departs the depot, serves a sequence of requests, and then returns to the depot. For example, a vehicle route serving import request 1 followed by export requests 4 and 2 would visit the following ordered sequence of locations: $\{D, P, I_1, E_4, P, E_2, P, D\}$. No vehicle may depart D before a common start time 0, and each vehicle must return to D by a common deadline time β .

Let t_{ij} represent the travel time between any two locations i and j , where travel times are symmetric ($t_{ij} = t_{ji}$). The UDP is then to determine a set of vehicle routes serving all move requests, minimizing the required vehicle fleet size¹.

We now extend UDP by adding port access constraints to create the access-controlled drayage problem (ACDP). Let T be a discrete set of non-overlapping port access time slots τ_j . Each $\tau_j \in T$ can be described as an interval $[a_j, b_j]$ where $0 \leq a_j < b_j \leq \beta$. Suppose without loss of generality that the access slots are ordered such that $a_1 < a_2 < \dots < a_{|T|}$. Then, the non-overlapping property implies that $b_j \leq a_{j+1}$ for all $j = 1, \dots, |T| - 1$. Each slot τ_j has a corresponding access capacity of κ_j that constrains the number of vehicle arrivals to the port within that slot. A vehicle may arrive to the port without a container (to pick up an import container), or with an export container; either scenario is considered a vehicle arrival and requires an available unit of slot capacity. On the other hand, vehicles that arrive with an export container and leave after picking up an import container only require a single access. Note that we also assume that a vehicle can wait idle at a customer location either before or after picking up or dropping off a container, and thus no waiting at the port for access is required.

Given this input, the ACDP has two distinct phases. The first phase, denoted ACDP(I), is to determine the best set of container move requests to serve given the available access slot capacities by minimizing the sum of penalty costs p_j for each unserved request j . Given a subset of requests that can be feasibly served, the second phase ACDP(II) is to determine a set of scheduled routes serving the subset, minimizing the required fleet size. In this case, each port access during a vehicle route is *scheduled* for a specific time slot. It should be clear that UDP is a special case of ACDP(II), where each access slot τ_j has infinite capacity.

Solving instances of ACDP provides two types of decision support to drayage providers. In the first application, suppose that a set of access appointments has been made in advance for a particular operating day. This appointment set defines a set of access slot capacities, and the drayage firm must optimize its operations given this set. In the second application, suppose that a firm is planning operations for a future operating day with complete information describing the available remaining appointment time capacities as well as full knowledge of the container moves it will likely need to serve. Solving an ACDP instance will determine the appropriate access time slots for which the firm should request appointments.

3 Related Literature

To our knowledge, no research has addressed routing problems with constraints on the number of times vehicles may access a specific location (or locations) during different time win-

¹ We assume that each move request is feasible. For example, export request j is feasible if and only if $t_{D,E_j} + t_{E_j,P} + t_{PD} \leq \beta$.

dows, like the proposed ACDP. However, a large body of research has focused on routing problems similar to the unconstrained drayage problem. The UDP is a special case of a class of routing problems denoted pickup and delivery problems with time windows (PDPTW) (see, *e.g.*, Bent and Van Hentenryck, 2006, for a typical PDPTW definition with vehicles based at a common depot). The UDP is simpler, since it includes a restrictive time window only at the depot node, each pickup-and-delivery demand fills the vehicle to capacity, and either the pickup point or the delivery point for each request is the port node.

Much research has focused on the PDPTW, especially in the context of dial-a-ride transit. Savelsbergh and Sol (1995) and Desaulniers et al. (2001) provide excellent overviews of research on these problems. Excellent examples of high quality heuristic approaches to such problems are provided by Toth and Vigo (1997) which proposes a tabu thresholding heuristic, and Bent and Van Hentenryck (2006) which proposes a two-phase approach combining simulated annealing and large neighborhood search. The approach in the latter paper is shown to be particularly effective on large-scale problems with up to 600 demand requests. Alternatively, Dumas et al. (1991) develops an optimal algorithm for the multiple-vehicle PDPTW that utilizes column generation and an integer programming branch-and-bound algorithm, and presents computational results for problems with up to 55 demand requests. The approach works well primarily on problems with tight capacity or time window constraints that limit the number of feasible routes. More recently, Lu and Dessouky (2004) propose a branch-and-cut solution approach for problems in which such constraints are not tight. Such problems are more difficult to solve, and the approach can handle instances with up to 5 vehicles and 25 demand requests.

The simpler structure of the UDP enables the problem to be formulated as a multiple traveling salesperson problem with time windows (m -TSPTW). To do so, each container move request is represented by a single node, and $|C|$ depot nodes are included, each with time window $[0, \beta]$. The nodes are fully connected generating a complete network, and costs on all arcs are zero except arcs connecting a depot to a move request which have cost $\frac{1}{2}$. Travel times between the depot nodes are defined to be zero. Travel times on arcs inbound to a move request node include the time required to perform the request (*e.g.*, travel time on the arc connecting export request 1 to import request 2 would be $t_{P,P} + t_{P,I_2}$ while travel time on the arc connecting import request 2 to export request 1 would be $t_{I_2,E_1} + t_{E_1,P}$); note that such travel times are asymmetric. Although UDP can be formulated as an m -TSPTW, it is actually more similar to discrete item packing problems since minimizing the fleet size is the objective and the vehicle depot return deadline is the primary constraint.

In the specific area of drayage routing and scheduling, Jula et al. (2005) considers a problem similar to the UDP with multiple port locations as well as customer pickup and delivery time windows. Instead of a depot time window, the problem limits the total duration of each route. A dynamic programming solution approach is proposed which works well for determining optimal solutions to problems with up to 25 move requests, and a genetic algorithm for determining near-optimal solutions to problems with up to 100 move requests. Smilowitz (2006) develops a branch-and-bound solution approach with embedded column generation

for a problem with multiple truck depots, container yards, and port locations. An additional consideration of the model is the pickup and dropoff of empty containers, not necessarily at predefined locations; such moves are modeled as flexible tasks.

4 Solution Methodology

The UDP and, by extension, ACDP(II) are shown to be *NP*-hard optimization problems by transformation to bin packing in Namboothiri (2006). These problems are similar to other routing problems that have been treated effectively with metaheuristics (see *e.g.*, Bent and Hentenryck, 2004, and Toth and Vigo, 2003, in addition to the references provided in the previous section). In this paper, we choose to develop an optimization-based heuristic that utilizes column generation. Since ACDP(II) instances do not have customer-specific time windows, they are more loosely constrained than other drayage problems considered in the literature. Thus, we utilize a very fast (but suboptimal) pricing heuristic within the column generation. The methodology is now described.

4.1 ACDP(I): Determining Feasible Customer Requests

Given T , it may not be possible to serve all requests in C even with an arbitrarily large vehicle fleet. To determine a best subset C' of feasible requests to serve, we formulate and solve an integer programming model that minimizes the sum of penalty costs for unserved requests.

Let R^0 be a set of scheduled vehicle routes, where each $r \in R^0$ is a feasible sequence of customer requests to be served by a single vehicle departing and returning to the depot, with each request scheduled such that the slot access capacities as well as the depot time window constraint are not violated. Let α_{ir} be a $\{0, 1\}$ parameter equal to one if request i is served by route r , and let β_{jr} represent the number of port accesses in time slot τ_j required by route r . The decision variables y_i indicate which tasks in C will not be selected for service, and x_r indicate which routes in R^0 are chosen for the optimal subset.

A formulation for ACDP(I) that determines the best set of requests to serve given R^0 is now:

$$\text{minimize } \sum_{i \in C} p_i y_i$$

subject to:

$$\begin{aligned}
\sum_{r \in R^0} \alpha_{ir} x_r + y_i &\geq 1 & \forall i \in C \\
\sum_{r \in R^0} \beta_{jr} x_r &\leq \kappa_j & \forall \tau_j \in T \\
x_r &\in \{0, 1\} & \forall r \in R^0
\end{aligned}$$

We note that the objective of ACDP(I) only considers the penalty costs for uncovered requests, and does not consider fleet size costs. Thus, one can determine the minimum penalty subset of requests to serve regardless of fleet size by including in R^0 only the set of all time-feasible *single-access* routes. A single access route is one where the operating vehicle visits the port exactly one time (and therefore consumes one unit of some time slot access capacity). The single-access route set R^0 includes all single container move request routes (e.g., $\{D, E_i, P, D\}$ or $\{D, P, I_k, D\}$) for each feasible time slot, and all exporter-importer paired routes ($\{D, E_i, P, I_k, D\}$) for each feasible time slot.

4.2 ACDP(II): Determining Best Scheduled Vehicle Routes

Let C' ($n = |C'|$) be the set of feasible move requests determined after solving ACDP(I). To formulate an integer programming model for ACDP(II) given C' , let R be the set of all feasible scheduled single-vehicle routes serving subsets of C' , where we again note that a route departs the depot, serves a set of container move requests (and potentially includes multiple port visits at various times), and then returns to the depot. Note that R may contain multiple routes that serve the same move requests in the same sequence, but access the port at different times and thus utilize port access capacity differently.

The following integer programming model for ACDP(II) selects a subset of R with the smallest size, hence minimizing the number of required vehicles to ensure that each move request in C' is served:

$$\text{minimize } \sum_{r \in R} x_r$$

subject to:

$$\sum_{r \in R} \alpha_{ir} x_r \geq 1 \quad \forall i \in C' \tag{1}$$

$$\sum_{r \in R} \beta_{jr} x_r \leq \kappa_j \quad \forall \tau_j \in T \tag{2}$$

$$x_r \in \{0, 1\} \quad \forall r \in R \tag{3}$$

Note that this formulation is a slight modification of standard set covering models used in vehicle routing, where constraints (2) are added to allow modeling of access slot capacities.

Let $R^* \subseteq R$ represent the set of scheduled routes that appear in an optimal solution to this problem.

4.2.1 Minimizing vehicles versus minimizing total accesses

ACDP(II) as defined minimizes the number of vehicles required to serve all of the requests in C' . Since port access slots are capacity-constrained, it is also natural to consider the alternative objective of minimizing the number of required slot accesses. This alternative objective function can be written as $\sum_{r \in R} (\sum_{\tau_j \in T} \beta_{jr}) x_r$; denote this problem ACDP(II)-A. The following theorem holds:

Theorem 4.1 *The set of all optimal solutions for ACDP(II) need not contain the optimal solution for ACDP(II)-A, and vice versa.*

Proof: The proof is via counter-example. Consider an instance of ACDP(II) with three unique customer locations, where each location submits one export move request and one import move request. Let the travel time between the depot and any of the customer locations or the port be one time unit. The travel time between the port and any of the customer locations is two time units. Let $\beta = 10$, and suppose that there is only one port access slot, τ_1 , where $[a_1, b_1] = [0, 10]$ with infinite capacity. Given this setting, a vehicle operating a route serving a single request will require 4 time units, while a vehicle serving any export-import pair will require 6 units.

Note that a lower bound on the total required number of port accesses for any instance of ACDP(II) or ACDP(II)-A is given by $\min\{|E|, |I|\} + ||E| - |I||$, since accesses can be reduced only by pairing an export request with an immediately following import request. In the counter-example instance, this lower bound is three and can be achieved in a solution, for example, with a unique vehicle serving first the export then the import task for each customer. Note then that any optimal solution to ACDP(II)-A requires three vehicles.

Alternatively, an optimal solution to ACDP(II) for this instance requires only two vehicles, but will always require four port accesses and thus is not optimal for ACDP(II)-A. An example of such a solution uses the first vehicle to serve the export-import pair at customer 1, and the export request at location 2 thus requiring 10 time units and two port accesses. The second vehicle similarly serves the export-import pair at customer location 3, and the import request at location 2 also requiring 10 time units and two port accesses. \square

4.3 Heuristic Solution Approach: A Root Column Generation Heuristic

Enumeration of the set of feasible routes R^0 required by the formulation for ACDP(I) is a simple task. This is not necessarily the case when enumerating R for the ACDP(II) model. Since R will often contain a very large number of routes for instances of practical size, we

develop a solution heuristic based on column generation. The goal of the approach is to quickly identify a large candidate set of potential vehicle routes during column generation at the root linear relaxation to ACDP(II), and then conduct branch-and-bound to identify an integer solution using this route set.

The heuristic proceeds as follows. The first major step is to determine a near-optimal solution for the linear programming relaxation of ACDP(II) using a column generation procedure. At each iteration of this procedure, a restricted version of the linear relaxation is solved using a subset R' of the complete set of feasible vehicle routes R . Using optimal dual variable information from this solution, a pricing subproblem is solved heuristically to identify routes in $R \setminus R'$ with negative reduced cost that therefore would improve the solution. If such routes are identified, they are added to R' and the process is repeated until no such routes are found. The second major step of the heuristic is to use branch-and-bound to solve ACDP(II) using only the final restricted route set R' . Importantly, note that only if some set R^* is a subset of the final route set R' will the branch-and-bound process conclude with an optimal solution to ACDP(II).

4.3.1 Column Generation for ACDP(II)

We now describe the column generation problem for the root linear relaxation of ACDP(II) in more detail. A good initial set of routes is $R' = R^0$, since R^0 is relatively small but will always contain a feasible solution, as established in the following result which we provide without proof:

Theorem 4.2 *A problem instance of ACDP(II) with feasible customer request set C' has a feasible solution if and only if the problem has a feasible solution using the subset of routes and schedules R^0 .*

Consider now the pricing subproblem of identifying routes with negative reduced costs during each column generation iteration. Let π_i and σ_j represent the dual variables associated with constraints (1) and (2) after solving the linear relaxation of ACDP(II) over R' . Then, the reduced cost \bar{v}_r of route $r \in R$ is given by

$$\bar{v}_r = 1 - \sum_{i \in C} \alpha_{ir} \pi_i - \sum_{\tau_j \in T} \beta_{jr} \sigma_j \quad (4)$$

The column generation subproblem is then:

$$\min_{r \in R \setminus R'} \bar{v}_r. \quad (5)$$

If the solution to (5) has a negative objective function, then there exists a negative reduced cost route. Unfortunately, problem (5) is an *NP*-hard time-dependent elementary shortest-path problem with a duration constraint, and thus is difficult to solve efficiently. The time

dependency results from the fact that the dual variable for a port access varies with time, and therefore the cost of a route depends on the time(s) that it is scheduled to access the port.

To generate sets of negative reduced cost columns quickly, we use a very fast heuristic pricing method that, while efficient, is not guaranteed to identify an improving route if one exists. However, each time it is executed, the heuristic may generate many improving routes and schedules, which are then all added to the set R' . The method is an extension of the *layered shortest path algorithm* (LSP) first introduced in Namboothiri and Erera (2004) for the UDP. The LSP algorithm preserves the elementary path constraint, and generates only time-feasible routes. Unlike an optimal dynamic programming approach, however, the algorithm does not fully explore the state space.

The LSP algorithm is an extension of standard single-label shortest path algorithms to resource-constrained problems; Namboothiri (2006) first presents the approach, and its application in column generation for VRPTW and UDP. Unlike single-label algorithms (such as the classic label-correcting method in Bellman, 1958), the LSP algorithm may maintain multiple labels for each node, but at most one label for each potential position (or *layer*) of that node along some path. For example, the LSP approach would store a separate label (and an associated path) for node 1 when node 1 is the first node in the path, the second node in the path, etc. Since each node can appear at most once in any path, the number of potential layers is limited (at worst) by the number of nodes.

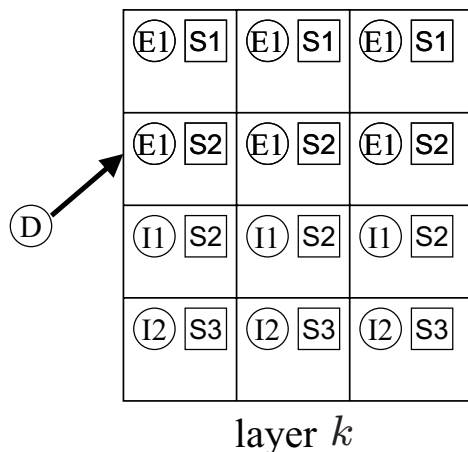


Fig. 1. Portion of the label array structure for LSP pricing heuristic; each entry represents a route that serves its final request as indicated by the circle during the time slot indicated by the square.

To apply the LSP algorithm to the ACDP pricing subproblem given by (5), we create a network of at most $n|T|$ nodes, where a visit to node (i, j) indicates that move request i is executed by a vehicle that accesses the port in time slot τ_j either immediately before (for an import request) or immediately after (for an export request) serving request i . Since the elementary path condition holds, at most n layers of labels are required; see Figure 1. Furthermore, certain move requests (for example, those further from the depot or port) may not be compatible with certain time slots. For example, accessing the port during a late time

slot may not allow an import move request to be completed and the vehicle to return to the depot by β . An early time slot may also be incompatible with an export task, for example, if the vehicle cannot travel from the depot to the export location and then to the port by the end of the slot time window. Incompatible combinations of move requests and time slots are not generated.

Given this expanded network, the LSP algorithm determines paths and their associated reduced costs layer by layer. The label stored for node (i, j) at layer k is associated with a path that performs move request i using access slot τ_j after completing $k - 1$ prior requests. The primary component of each label is the path reduced cost, \bar{v}_{ij}^k . To determine the label for node (i, j) at layer k , the algorithm finds a minimum reduced cost elementary time-feasible path extending a path at layer $k - 1$ to serve request i using time slot τ_j ; note that such a path is only feasible if the vehicle can return to the depot after serving i by the deadline β . A complete set of labels can be generated quickly layer by layer, beginning with the layer $k = 1$ entries representing single request routes. The $O(n^3|T|^2)$ procedure is now described in detail for completeness.

Let n_L be the total number of labels generated by the LSP algorithm, and let n_r denote the number of negative reduced cost routes generated; we note that $n_r \leq n_L \leq n^2|T| + 1$. Each label ℓ_{ij}^k includes the following attributes:

- A route vector \mathbf{r}_{ij}^k with each element

$$r(c) = \begin{cases} 1 & \text{if move request } c \in C' \text{ included in path for this label} \\ 0 & \text{otherwise} \end{cases}$$

- The reduced cost associated with route \mathbf{r}_{ij}^k , given by \bar{v}_{ij}^k
- A time slot vector \mathbf{ts}_{ij}^k with each element

$$ts(j) = \text{number of port accesses in time slot } \tau_j \in T \text{ required by route } \mathbf{r}_{ij}^k$$

- The feasible port access time window for request i in route \mathbf{r}_{ij}^k and time slot τ_j , denoted by $[a', b']_{ij}^k$

We maintain feasible port access time windows for the last request served by each route in order to enable simple determination of feasible route extensions to future task-slot nodes.

LSP PRICING HEURISTIC

- 1: Create initial label ℓ_0 at depot representing an empty route; set all attributes of ℓ_0 to 0 except $b_{00}^0 \leftarrow \beta$ and $\bar{v}_{00}^0 \leftarrow \infty$
- 2: Create empty initial labels ℓ_{ij}^k for all $i = 1 \dots n$, $j = 1 \dots |T|$, and $k = 1 \dots n$; set $\bar{v}_{ij}^k \leftarrow \infty$
- 3: Initialize $n_r \leftarrow 0$
- 4: $k \leftarrow 0$
- 5: **while** $k \leq n - 1$ **do**

```

6:  for all  $\ell_{ij}^k \mid \bar{v}_{ij}^k < \infty$  do
7:    for all  $u \in C$  not in route  $\mathbf{r}_{ij}^k$  do
8:      for all  $\tau_v \in T$  not before  $\tau_j$  do
9:        if request  $u$  can be feasibly served using port access slot  $\tau_v$  by satisfying FEASIBLEEXTEND given below then
10:          $\bar{v}_{uv}^{k+1,old} \leftarrow \bar{v}_{uv}^{k+1}$ 
11:         Calculate  $\bar{v}_{uv}^{k+1,new}$  using UPDATEREDUCEDCOST
12:         if  $\bar{v}_{uv}^{k+1,new} \leq \bar{v}_{uv}^{k+1,old}$  then
13:            $\bar{v}_{uv}^{k+1} \leftarrow \bar{v}_{uv}^{k+1,new}$ 
14:           Use UPDATALABEL to finish the update of the new label
15:           If  $\bar{v}_{uv}^{k+1} < 0$  and  $\bar{v}_{uv}^{k+1,old} \geq 0$ , then  $n_r \leftarrow n_r + 1$ 
16:         end if
17:       end if
18:     end for
19:   end for
20: end for
21:    $k \leftarrow k + 1$ 
22: end while

```

FEASIBLEEXTEND

Label ℓ_{ij}^k can be extended to label ℓ_{uv}^{k+1} only if

- The earliest port access time for task u at the end of the route is not greater than the end time of slot τ_v , b_v ; and
- The latest port access time for task u at the end of the route is not less than the begin time of slot τ_v , a_v , where the latest port access time is constrained by the total travel time required after port access for the vehicle to return to the depot by time β ; and
- The earliest port access time is not greater than the latest port access time.

UPDATEREDUCEDCOST

Updating the reduced cost $\bar{v}_{uv}^{k+1,new}$ depends on the tasks i and u and slots j and v :

```

1: if  $i \in E$  and  $u \in I$  and  $\tau_j = \tau_v$  then
2:    $\bar{v}_{uv}^{k+1} \leftarrow \bar{v}_{ij}^k - \pi_u$ 
3: else
4:    $\bar{v}_{uv}^{k+1} \leftarrow \bar{v}_{ij}^k - \pi_u - \sigma_v$ 
5: end if

```

UPDATALABEL

The attributes of label ℓ_{ij}^k is extended to label ℓ_{uv}^{k+1} depending on i and u , as shown below:

- Update $\mathbf{r}_{uv}^{k+1} \leftarrow \mathbf{r}_{ij}^k$; $\mathbf{r}_{uv}^{k+1}(u) \leftarrow 1$

- Update $\mathbf{ts}_{uv}^{k+1} \leftarrow \mathbf{ts}_{ij}^k$; $\mathbf{ts}_{uv}^{k+1}(v) \leftarrow \mathbf{ts}_{uv}^{k+1}(v) + 1$ unless $i \in E$ and $u \in I$

Case I : $i \in E$ and $u \in E$

- $a'_{uv}^{(k+1)} \leftarrow \max(a'_{ij}^k + 2t_{Pu}, a_v)$
- $b'_{uv}^{(k+1)} \leftarrow \min(\beta - t_{PD}, b_v)$

Case II: $i \in I$ and $u \in E$

- $a'_{uv}^{(k+1)} \leftarrow \max(a'_{ij}^k + t_{Pi} + t_{iu} + t_{uP}, a_v)$
- $b'_{uv}^{(k+1)} \leftarrow \min(\beta - t_{PD}, b_v)$

Case III: $i \in E$ and $u \in I$

- $a'_{uv}^{(k+1)} \leftarrow \max(a'_{ij}^k, a_v)$
- $b'_{uv}^{(k+1)} \leftarrow \min(\beta - t_{Pu} - t_{uD}, b_v)$

Case IV: $i \in I$ and $u \in I$

- $a'_{uv}^{(k+1)} \leftarrow \max(a'_{ij}^k + 2t_{Pi}, a_v)$
- $b'_{uv}^{(k+1)} \leftarrow \min(\beta - t_{Pu} - t_{uD}, b_v)$

At the conclusion of the heuristic, we add all n_r routes corresponding to labels with negative reduced costs to R' . If $n_r = 0$, we have found no improving routes during the pricing subproblem, and the column generation heuristic is terminated.

5 Computational Experiments on Impact of Appointment Systems

In this section, we use the heuristic solution approach for the ACDP to assess the potential impacts of port access appointment systems on drayage firm productivity. To do so, we develop a set of hypothetical problem instances with characteristics that we believe are representative of real-world port drayage operations. Next, given varying assumptions regarding both the design of the appointment system and the set of preselected access slot capacities, we develop near-optimal operating plans using the heuristic and compare solutions. Specifically, this study will investigate both customer service impacts measured by the fraction of customer move requests which can be feasibly served, and cost impacts measured by the fleet size required to execute a set of requests.

It is important to note that this assessment will be based on operating plans determined by our heuristic (suboptimal) algorithm. Namboothiri (2006) provides some validation of the effectiveness of a root column generation approach using an LSP pricing heuristic for the UDP problem with the objective of minimizing total vehicle travel time. When compared

to an exact method based on enumeration, average optimality gaps no greater than 2 percent were observed for a set of 32 test problems with 20-30 randomly-scattered customers where each vehicle route served approximately five daily requests. Although we will report no similar validation of the solution quality obtained for ACDP instances, we believe the approach should still be effective. Furthermore, we will demonstrate that the heuristic is computationally efficient for reasonably-sized problems.

5.1 Data Generation

To generate sample instances representative of real-world port drayage scenarios, assume that a fleet serves a set of customers located in a square region, with the port located at the midpoint of one of the sides of the square and the vehicle depot located one hour inland from the port. The square service region has sides of length four hours, and assume that travel times are specified by an L_1 distance function; thus, the furthest customer may be located six hours from the port. Assume further that the fleet depot of the drayage firm operates from 6 AM until 8 PM; *i.e.*, vehicles can depart the depot as early as 6 AM and can return to the depot as late as 8 PM. Customer facilities, both import and export, allow pickups or deliveries from 8 AM to 6 PM.

The port facility also opens at 8 AM and closes at 6 PM. Letting time $t = 0$ correspond to 8 AM, the port access control system is modeled by partitioning the time interval $[0, 10]$ into $|T|$ equal-duration slots. In this study, we consider experiments with three different slot durations SD , 30 minutes, 60 minutes and 120 minutes with corresponding $|T|$ values of 20, 10 and 5 respectively. The total access capacity of all slots, parameter SC , is varied in the study. We compare three different assumptions regarding the distribution of SC across the slots. Figure 2 depicts the general form of the three distributions. The first distribution, denoted *uniform*, models the case where capacity is evenly distributed (except for rounding) over the course of the day. The second distribution, denoted *morning/afternoon heavy*, represents the case where the drayage firm has requested more capacity in the morning and afternoon slots, and less around the midday. It is generated as follows. First, total capacity SC is divided by the number of slots $|T|$ to yield the average per period capacity AC (note that this may be non-integer). Next, a slope s representing the per period change in capacity is determined by assuming that the average capacity is $\frac{AC}{2}$ at time zero and $\frac{3AC}{2}$ at time $\frac{|T|}{2}$. Using these points, and slope s before time $\frac{|T|}{2}$ and $-s$ after time $\frac{|T|}{2}$, capacities are generated for each time slot. Slot capacities are then rounded to integer values such that the total capacity of all slots sums to SC . A similar procedure is used to generate the upside-down V-shaped distribution denoted *midday heavy*; in this case the capacity of $\frac{3AC}{2}$ is at time zero, and $\frac{AC}{2}$ at time $\frac{|T|}{2}$.

Each problem instance includes 100 customer requests to be performed on a given operational day, representing a medium to large operating day for a drayage firm. Exactly half of the requests are export requests (representing containers to be returned to the port, loaded or

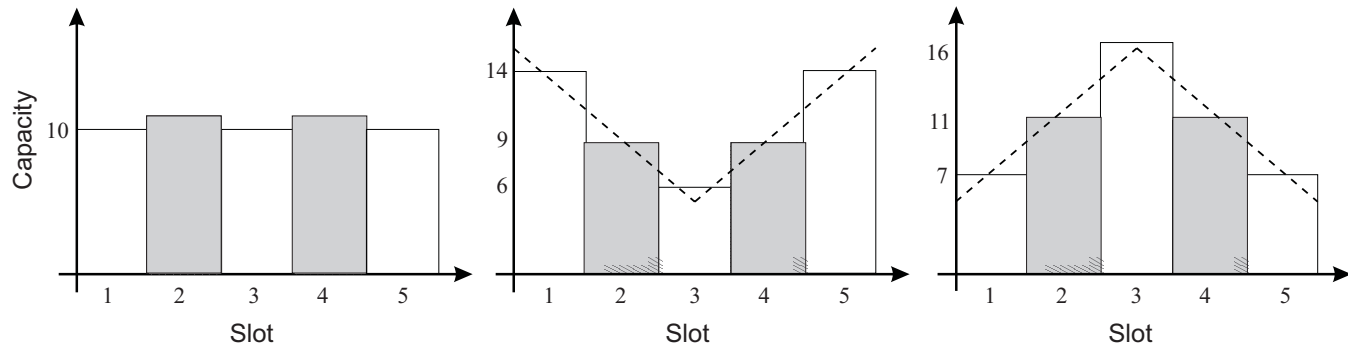


Fig. 2. Graphical depiction of the three port access time slot capacity functions (from left to right, Uniform, Morning/Afternoon Heavy, and Midday Heavy); example with total capacity $TC = 52$ and slot duration $SD = 120$ minutes.

empty) and half are import requests. The penalty cost p_j for not serving customer request j is set equal to its expected revenue, which is assumed to be proportional to the travel time from the port to the customer location.

We consider three different distributions of customers over the service region. Since most requests are likely to arrive from locations near to the port, the service region is divided into two subregions (see Figure 3). The square *depot subregion* centered on the depot location has sides of length two hours. The first geographic distribution (denoted 50/50) assumes that 50 out of 100 customers are located in the depot subregion, which comprises 25 percent of the total service region area. The second distribution (denoted 80/20) is more heavily weighted to the depot subregion, where 80 customers are assumed to be located. Finally, the third distribution (denoted 20/80) considers a less realistic distribution in which only 20 customers are located within the depot subregion, primarily for comparative purposes. Once allocated to a subregion, customer coordinates are generated with a two-dimensional uniform distribution. For each geographic distribution, 10 distinct instances are created.

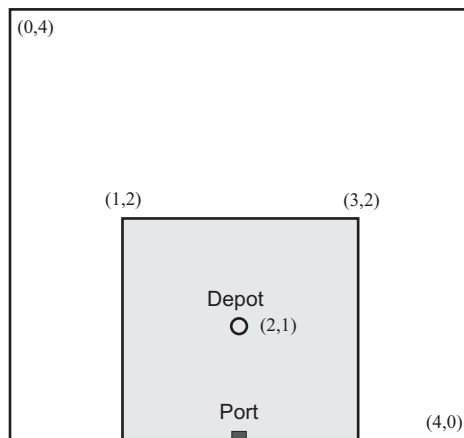


Fig. 3. Map of the test service region, with port and depot locations. Shaded area has higher customer density in some instances.

Total Capacity(SC)	No. of Columns	CPU Time			
		Phase I	LSP ColGen	B & B	Total
46	36539	183.8	238.6	345.7	768.1
48	38577	230.4	297.7	458.6	986.8
50	43674	210.3	305.5	587.7	1103.5
52	46639	25.7	137.4	259.2	422.3
54	48825	1.1	109.9	818.1	929.2
56	51290	0	94.2	674.8	769.1
58	49604	0	97.3	755.1	852.5
60	48943	0	90.0	915.7	1005.8

Table 1

Solution Computation Time: 50/50 Customer Distribution, Uniform Capacity Distribution, $SD = 30$ minutes

5.2 Solution Computation Time Performance

The solution approach is implemented in the C programming language, and utilizes the CPLEX Version 8.1 callable libraries for the solution of linear and binary integer programs when necessary. All tests were run on a dual-CPU 2.4 GHz Pentium with 2 GB of memory running Linux.

For the test problems considered, the approach generates both Phase I and Phase II solutions within 20 to 25 minutes of CPU time. Such times were deemed reasonable for the purposes of this study. Additionally, since problems with 100 daily container drayage requests should represent examples of medium to large size problems found in practice, the approach appears to require reasonable computation times that would enable its use in practical application as well.

Tables 1, 2, and 3 summarize computational performance for problem instances with the 50/50 customer distribution and the uniform capacity distribution. Each row provides average computation times in seconds for 10 sample instances. The first column reports the total slot access capacity SC available. The second column reports the average number of columns generated by the LSP subproblem heuristic when solving the root linear program of Phase II. The third, fourth and fifth columns report the average CPU execution time required to solve the Phase I integer program to optimality, the heuristic column generation using LSP for the Phase II root linear program, and then the Phase II branch-and-bound. The sixth column provides the total execution time.

Comparing the results in the tables, note that solving the Phase I integer programs to optimality never requires more than an average of 4 minutes of computation time. As expected,

Total Capacity(SC)	No. of Columns	CPU Time			
		Phase I	LSP ColGen	B & B	Total
46	20799.4	14.3	27.2	50.2	91.7
48	22241.4	21.4	39.6	110.4	171.4
50	24832.3	12.3	47.6	224.0	283.9
52	27451.3	0.5	27.6	103.1	131.2
54	27860.6	0.1	27.7	165.6	193.5
56	28144.0	0	25.3	113.8	139.2
58	27782.8	0	23.8	120.0	143.8
60	27909.6	0	24.6	127.3	151.9

Table 2

Solution Computation Time: 50/50 Customer Distribution, Uniform Capacity Distribution, $SD = 60$ minutes

Total Capacity(SC)	No. of Columns	CPU Time			
		Phase I	LSP ColGen	B & B	Total
46	12492.7	1.8	7.0	9.7	18.5
48	14483.7	2.6	11.6	24.1	38.3
50	15938.6	0.1	11.8	56.9	68.8
52	17392.7	0	12.2	50.0	62.2
54	17634.2	0	11.9	39.9	51.8
56	17751.0	0	16.1	62.1	78.2
58	17271.2	0	13.9	35.9	49.9
60	17348.1	0	15.6	52.8	68.4

Table 3

Solution Computation Time: 50/50 Customer Distribution, Uniform Capacity Distribution, $SD = 120$ minutes

the Phase I problems become easier to solve as total capacity SC increases and for larger capacity values, Phase I computation time is negligible. The time required by the heuristic column generation and the branch-and-bound for creating Phase II solutions does not vary significantly or predictably with SC . On the other hand, computation times do vary significantly with slot duration SD . In general, shorter slot durations led to more columns generated by the LSP heuristic, and thus longer required branch-and-bound times. This effect is most likely due to the fact that shorter slot durations allow each task to be completed with a visit to the port during many more unique time slots.

5.3 Impact of Available Access Capacity

First, we investigate how drayage firm productivity may be impacted by the total amount of available access capacity, and its distribution during the day. To do so, consider the 50/50 customer geographic distribution and assume that the port operates appointment slots with duration of one hour ($SD = 60$). This scenario is perhaps most closely aligned with real-world operating conditions at many urban seaports. In order to guarantee available capacity, suppose a drayage firm has decided to book appointments in advance. A natural question is then to determine how many appointments to schedule in each slot. In an ideal world, the best strategy for the firm would be to determine the productivity-maximizing set of appointments based on the customer demand requests it is likely to face. However, this may be difficult to do since many firms are likely to be competing for appointment capacity.

It is clear that different sets of available appointment capacities may lead to different best operating plans, and therefore may impact productivity. To quantify potential productivity impacts, we compare the total number of tasks that can be feasibly served and the total number of vehicles required to serve the feasible tasks for set of scenarios in which the total available capacity for the firm (SC) and its distribution across the day is varied. Figure 4 summarizes the results. The x -axis in both parts of the figure corresponds to the total capacity (SC) available over the course of the day to the drayage firm. The upper part of the figure reports the average total number of tasks that can be feasibly served given that SC is distributed over the day according to the midday heavy, uniform, or morning/afternoon heavy distributions. The lower part of the figure reports the average total number of vehicles needed to serve these feasible tasks in the best operating plans.

As expected, the average numbers of feasible tasks vary somewhat with SC , especially at smaller values. Since these problems each contain 50 export tasks and 50 import tasks, the minimum capacity required to serve all tasks is $SC = 50$. Below this value, the midday heavy curve increases at a rate of 2 tasks served per unit of added capacity, while the uniform and morning/afternoon heavy curves increase at slightly slower rates. When capacity is added beyond the absolute minimum of 50, the benefit depends on the capacity distribution. For the midday heavy distribution, 99 percent of customer requests can be served with $SC = 50$ and 100 percent can be served when $SC = 52$. For the uniform distribution, 98 percent can be served with $SC = 50$ and 100 percent when $SC = 54$. Finally, for the morning/afternoon heavy distribution, 95 percent can be served when $SC = 50$ and 100 percent when $SC = 56$.

Under all three capacity distributions, the number of vehicles tends to decrease (as expected) as SC increases. In absolute terms, the average number of vehicles required decreases about 10 percent as total capacity is increased by 20 percent from 50 to 60. A better measure of drayage productivity, however, is average customer tasks served per vehicle. Table 4 presents these statistics in this case. Interestingly, under all distributions, vehicle productivity increases roughly linearly with total capacity. When capacity is increased 30 percent from 46 to 60, vehicle productivity increases by about 24 percent under the morning/afternoon heavy

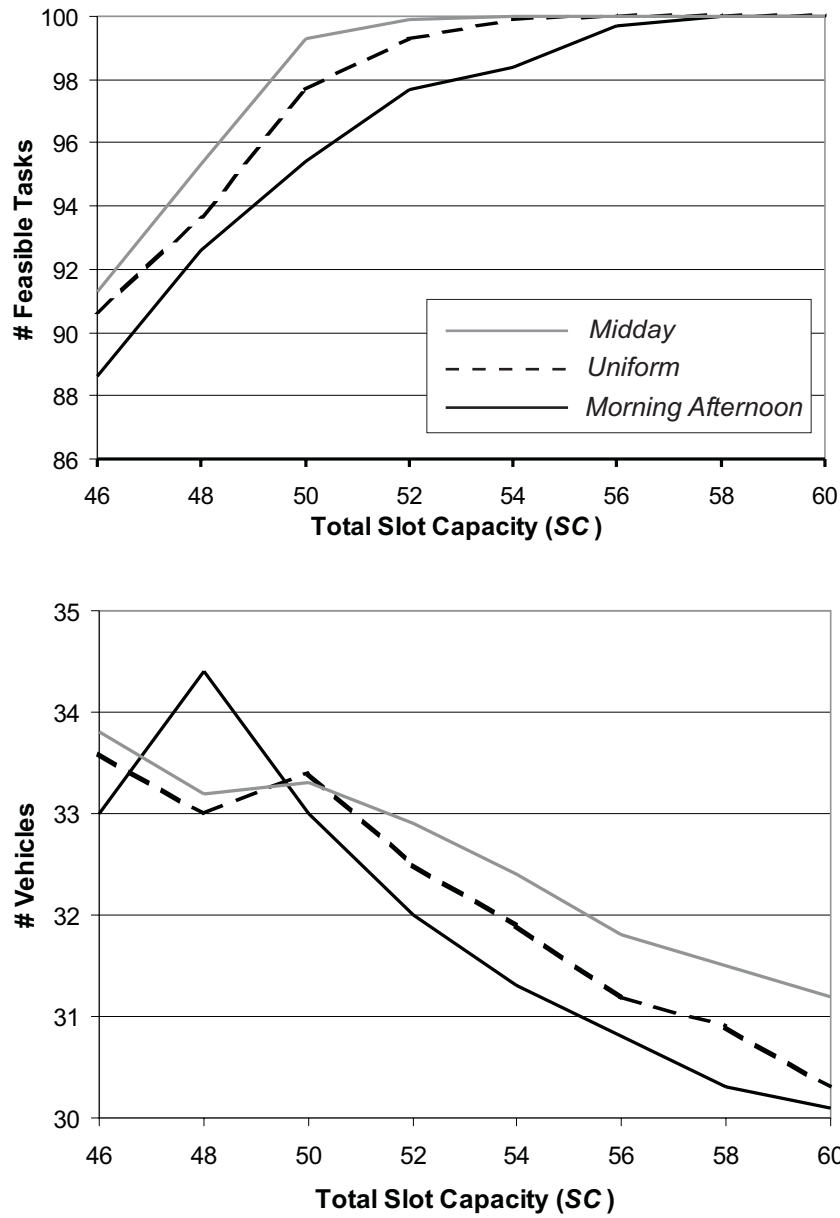


Fig. 4. Productivity Impacts of Available Capacity (SC) and its Distribution: 50/50 Customer Distribution and Slot Duration 60 minutes

distribution and by about 22 percent under the uniform distribution.

It is also clear from these results that differences in the distribution of available capacity across the operating day may have significant productivity impacts on drayage firms. From the perspective of feasible tasks, the midday heavy capacity appears best for these instances. Suppose the firm wishes to serve 99 percent of its customer requests. Then, if capacity were instead distributed uniformly across the day, 4 percent additional total capacity would be required. If capacity were distributed according to the morning/afternoon heavy distribution,

Total Capacity (SC)	Morning/Afternoon	Uniform	Midday
46	2.68	2.70	2.70
48	2.69	2.84	2.87
50	2.89	2.93	2.98
52	3.05	3.06	3.04
54	3.14	3.13	3.09
56	3.24	3.21	3.14
58	3.30	3.24	3.17
60	3.32	3.30	3.21

Table 4

Average Tasks Served Per Vehicle: 50/50 Customer Distribution and Slot Duration 60 minutes

this figure rises to 12 percent. Results in Figure 4 indicate clearly that firms must make good port slot selections (for fixed values of SC) to maintain high levels of customer service; differences between best and worst distributions can result in decreases in customer service level of up to 4 percent.

From the perspective of vehicle productivity, results in Table 4 indicate that the midday heavy distribution outperforms the alternatives slightly when total capacity is insufficient to serve all customer demand ($SC \leq 50$). However, at larger capacity values, the morning/afternoon heavy distribution leads to higher vehicle productivity in this setting, although the increases do not exceed 3 percent. The boldfaced value in each column highlights the capacity level for which the respective distribution allows 99 percent of the customer requests to be served. Although the morning/afternoon heavy distribution requires more total capacity, its vehicles are nearly 9 percent more productive than those under the midday heavy distribution when serving 99 percent of the requests.

5.4 Impact of Slot Duration

When terminal operators implement an access appointment system, consideration should be given to determination of an appropriate appointment slot duration. Shorter slot durations should allow the terminal operator to better predict when drayage vehicles may arrive, but may lead to negative impacts on drayage firm productivity. In this section, we study these potential impacts. Again, we assume that customers are distributed according to the 50/50 distribution, and furthermore that total access slot capacity for the drayage firm is uniformly distributed across the operating day.

Figure 5 summarizes the results. From the top part of the figure, it is clear that the drayage firm is able to serve more customers with less total access capacity with the slots have 120-

minute durations. In order to serve 99 percent of the customer requests, the drayage firm would need 50 total accesses with 120-minute slots, 52 total accesses with 60-minute slots, and 54 total accesses with 30-minute slots. At this service level, therefore, about 4 percent additional total capacity is needed for each reduction by half of the slot duration.

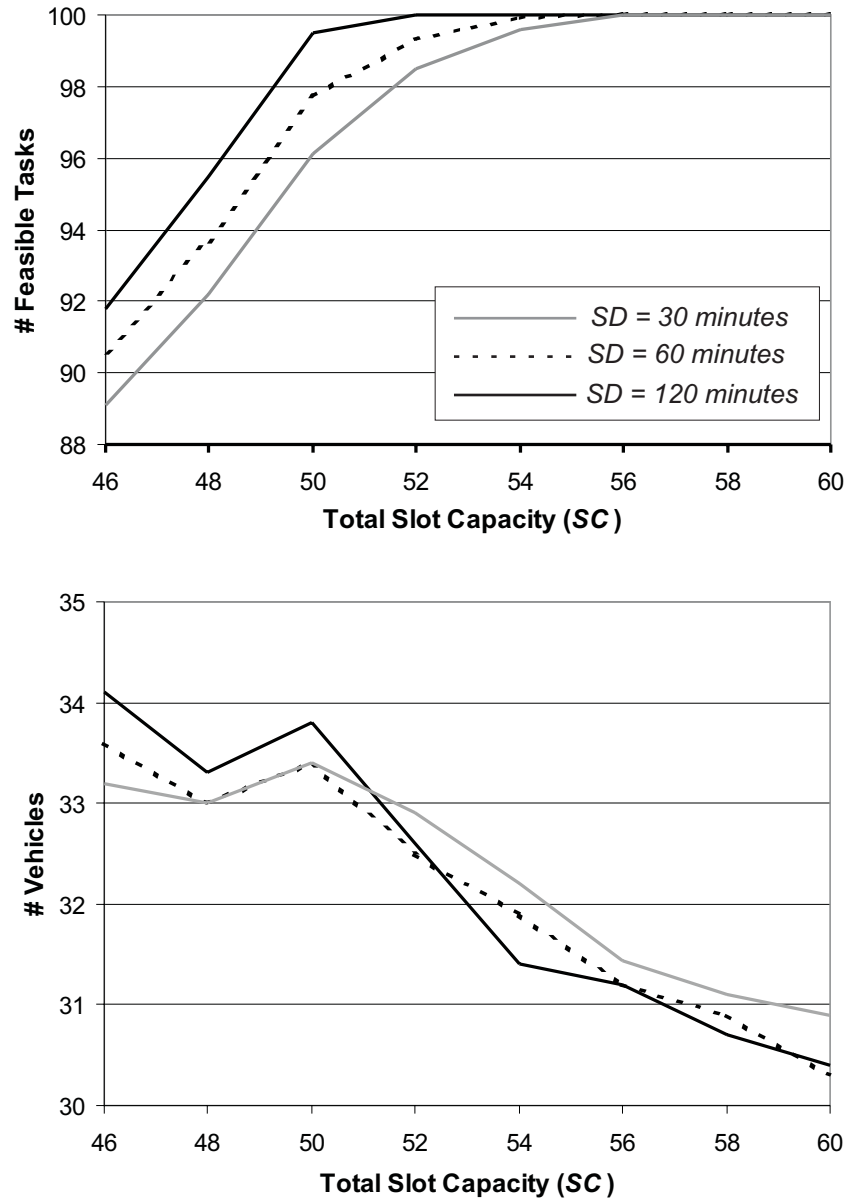


Fig. 5. Productivity Impact of Slot Duration: 50/50 Customer Distribution and Uniform Capacity Distribution

The bottom part of Figure 5 depicts how the number of vehicles varies with total slot capacity for the different slot duration assumptions. Note that when $SC \geq 56$ and all tasks can be feasibly served, the 120-minute durations lead to the fewest required vehicles as expected. However, the 60-minute duration curve is nearly identical to the 120-minute curve suggesting

Total Capacity (SC)	$SD=120$	$SD=60$	$SD=30$
46	2.69	2.70	2.68
48	2.87	2.84	2.79
50	2.94	2.93	2.88
52	3.07	3.06	2.99
54	3.18	3.13	3.09
56	3.21	3.21	3.18
58	3.26	3.24	3.22
60	3.29	3.30	3.24

Table 5
Average Tasks Served Per Vehicle: 50/50 Customer Distribution and Uniform Capacity Distribution

that the vehicle productivity benefit may be small. Table 5 summarizes the average number of tasks served per vehicle for these scenarios. Note that while the 120-minute duration statistics are almost always better than the others, the difference in productivity never exceeds 3 percent. Similarly, the difference in productivity when slot durations are reduced from 60 minutes to 30 minutes do not exceed 2 percent for these test problems. Thus, for the test problems considered here, the vehicle productivity benefit of longer time slots is somewhat small.

5.5 Impact of Customer Geography

Finally, since different drayage firms may serve different customer sets, we provide a short analysis of how productivity impacts may vary with total available access capacity for different customer geographical distributions. Figure 6 depicts the comparative results. Capacity is assumed to be uniformly distributed across the operating day, and the slot duration is 60 minutes. The gray curves in both parts of the figure depict results for the 80/20 customer distribution where customers are more closely clustered around the port and vehicle depot. The black curves depict results for the 20/80 distribution. For reference, the dashed curves repeat the results for the 50/50 distribution.

As expected, when more customers are clustered in the depot subregion, more requests can be feasibly served with less access capacity. For example, when $SC = 50$, 99 percent of customer requests can be served for the 80/20 distribution while only 95 percent for the 20/80 distribution. The vehicle productivity results are a bit more interesting. It is clear from the figure that the number of vehicles required to serve the feasible tasks is most sensitive to changes in total access capacity for the 50/50 distribution. This is also reflected in the vehicle productivity results. When SC increases from 46 to 60, vehicle productivity increases only 10 percent for the 80/20 distribution and 17 percent for the 20/80 distribution,

versus 22 percent for the 50/50 distribution. One interpretation for such results is as follows. Matching import requests to export requests is the only way to serve two requests with a single port access. When most customers are clustered close to the port and depot, there are many reasonable alternative matching opportunities, and thus it is relatively easy to make use of access capacity whenever it is available. Alternatively, when most customers are far from the port, there may be relatively few such matching opportunities and thus little room to increase vehicle productivity regardless of available access capacity.

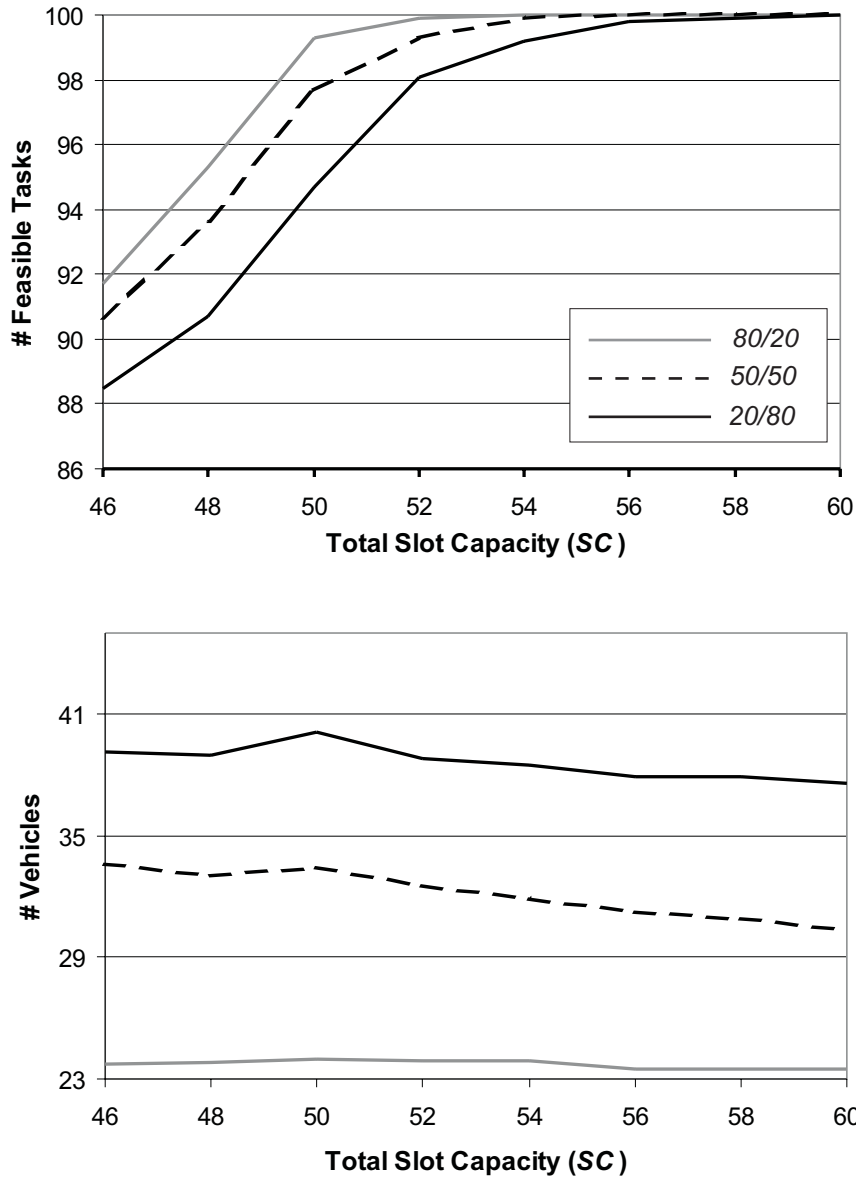


Fig. 6. Impact of Varying Customer Geography: Uniform Capacity Distribution and 60-minute Slot Duration

6 Conclusions

Optimization of port drayage operations gains added complexity when port access is restricted by available appointment slot capacity. Importantly, this research shows that productivity of drayage firms serving many daily requests can be significantly impacted by relatively minor changes in the characteristics of the allowable port accesses. Specifically, results for test problems indicate the following:

- It is critical that terminal operators provide drayage firms with enough access capacity. Results show that vehicle productivity can be increased by 10 to 24 percent when total access capacity is increased by 30 percent.
- Drayage firms must make good port appointment selections in order to maintain high levels of customer service; differences between the best and worst selections for a capacity distribution resulted in decreases in number of customers served by up to 4 percent for a fixed level of total access capacity.
- The duration of the appointment windows also may affect the ability of drayage firms to provide high levels of customer service. Test results indicate that up to 4 percent additional total capacity may be needed to maintain the same level of customer service if the slot duration is reduced by half.

Terminal operators need to carefully consider such productivity impacts when designing a port access system. Further, drayage companies operating under such a system should seriously consider using a decision support approach such as the one described in this paper to aid in the selection of access appointments, and the optimization of operations given a selection of appointments.

References

- R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.
- R. Bent and P. Van Hentenryck. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers and Operations Research*, 33:875–893, 2006.
- R.W. Bent and P. Van Hentenryck. A two-stage hybrid local search for the vehicle-routing problem with time windows. *Transportation Science*, 38:515–530, 2004.
- G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon, and F. Soumis. Vrp with pickup and delivery. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, pages 225–242. Society for Industrial and Applied Mathematics, Philadelphia, 2001.
- Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54:7–22, 1991.
- A.J. Herberger. EXPANDING OPPORTUNITIES IN COASTWISE SHIPPING: A Multi-Modal, Integrated Coastal Transportation System for the 21st Century. Technical report,

- Testimony before the House Transportation and Infrastructure Committee, Washington, D.C., May 2001. <http://www.house.gov/transportation/cgmt/05-23-01/herberger.html>.
- H. Jula, M. Dessouky, P. Ioannou, and A. Chassiakos. Container movement by trucks in metropolitan networks: modeling and optimization. *Transportation Research Part E*, 41: 235–259, 2005.
- A. Lowenthal. H&S 40720 Marine Terminal Operation; Truck Idling. Technical report, California State Assembly, State of California, September 2002. <http://www.arb.ca.gov/bluebook/bb03/HS/40720.htm>.
- Q. Lu and M. Dessouky. An exact algorithm for the multi-vehicle pickup and delivery problem. *Transportation Science*, 38:503–514, 2004.
- R. Namboothiri. *Planning Container Drayage Operations at Congested Seaports*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2006.
- R. Namboothiri and A.L. Erera. A set partitioning heuristic for local drayage routing under time-dependent port delay. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pages 3921–3926, The Hague, Netherlands, October 2004.
- M.W.P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29:17–29, 1995.
- K. Smilowitz. Multi-resource routing with flexible tasks: an application in drayage operations. *IIE Transactions*, 38:577–590, 2006.
- P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 15:333–346, 2003.
- P. Toth and D. Vigo. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31:60–71, 1997.