Available online at www.sciencedirect.com
ScienceDirect
Transportation Research Part E 44 (2008) 217–234
TRANSPORTATION RESEARCH PART E
www.elsevier.com/locate/tre

An attribute–decision model for cross-border drayage problem

Raymond K. Cheung [a,*], Ning Shi [b], Warren B. Powell [c], Hugo P. Simao [c]

[a] Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[b] IBM China Research Lab, Tower A, Diamond Building, Zhong Guang Cun Software Park, Hai Dian District, Beijing 100094, China
[c] Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, United States

Abstract

The drayage services between a container terminal and the origin (or destination) of a shipment account for a significant portion of the total transportation cost. They are the key sources of shipment delays, road congestions, and disruptions in the international logistics network. Such a situation is even worse when the drayage services involve cross-border issues. Using Hong Kong, the busiest port in the world, as an example, we illustrate the challenges and issues in managing drayage activities in hub cities. We show that managing cross-border drayage container transportation is a very challenging problem because not only individual resources (e.g., driver, tractor, and chassis) but also the composites of them (e.g., the driver–tractor–chassis triplets) need to be managed simultaneously. The problem is further complicated by the regulatory policies which govern the cross-border activities. We use an attribute–decision model for this problem and implement an adaptive labeling algorithm to solve it. We conduct numerical experiments to evaluate the system performances under various regulatory policies. The results show that the benefit gained by relaxing the regulatory policies is significant.

Keywords: Drayage problem; Labeling algorithm; Composite resource scheduling

1. Introduction

Due to intensifying globalization, both the volume and the complexity of the logistics flow in international port cities have rapidly increased. The port cities in the Asia Pacific region, in particular, have experienced phenomenal growth over the last few years. In 2004, the world's six busiest ports were all in the Asia Pacific region. As shown in Fig. 1, the growth of China's ports has been particularly strong. Growth as well as ever-increasing customer expectations and speed requirements have put pressure on major international port operators to improve efficiency and reduce cost. In terms of improving terminal operations and planning, there has been a substantial amount of research devoted to understanding these issues (see Steenken et al., 2004 for a review). Drayage services (that is, short-haul connecting services between a container terminal and the origin

* Corresponding author.
E-mail addresses: rcheung@ust.hk (R.K. Cheung), shining@cn.ibm.com (N. Shi), powell@princeton.edu (W.B. Powell), hpsimao@princeton.edu (H.P. Simao).

1366-5545/$ - see front matter
doi:10.1016/j.tre.2007.07.009

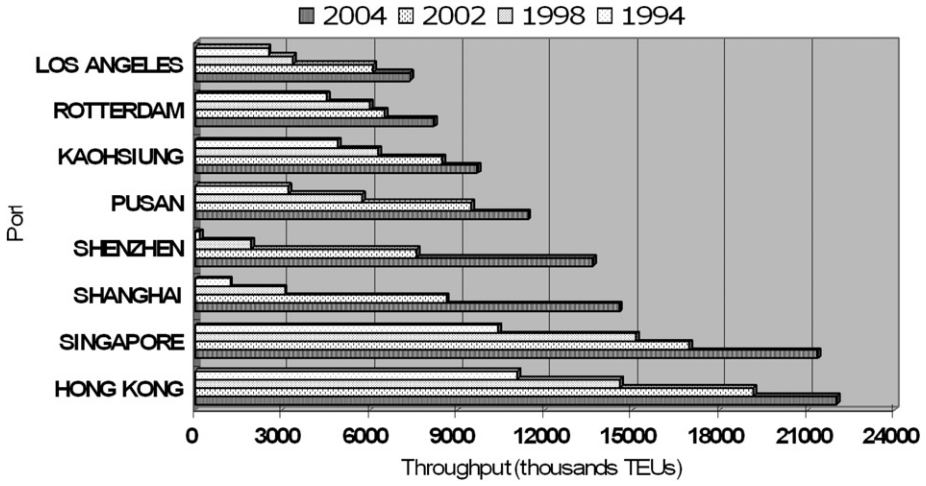

Fig. 1. Throughput of the top eight container ports from 1994 to 2004.

or destination of the container), however, have not received much research attention. As the volume of container flow is growing larger, managing the drayage activities is becoming increasingly difficult. Such a situation is even worse when the drayage activities involve cross-border issues. In this paper, we use the case of Hong Kong to illustrate how policies create challenges in managing the drayage activities, and present a labeling algorithm to solve the problems for various drayage situations.

As indicated in Fig. 1,[1] both Hong Kong and Singapore have topped the world in terms of container throughput for the last decade. In 2004, both handled over 20 million twenty-foot equivalent units (TEU) of containers. Due to the nature of these ports, however, the levels of drayage container transportation activities, called the drayage problem, are very different. Singapore is basically a transshipment port where containers from long-haul vessel services are transferred to other long-haul vessel services or regional feeder services. Hong Kong, on the other hand, has a much smaller proportion of transshipment cargo. Most containers passing through Hong Kong's port are imports, exports and re-exports from the southern part of Chinese Mainland (PRC). The sources or destinations of these containers are almost all in the Pearl River Delta (PRD) region, which is within 100 km of Hong Kong. Thus, the level of drayage activities in Hong Kong is much higher than in Singapore. Millions of containers pass through the three check-points at the Hong Kong – Shenzhen, PRC border each year. Because the licensing and regulatory policies in Hong Kong and in mainland China are different, the cross-border drayage activities are difficult to manage.

1.1. Cross-border drayage problem

Despite the productivity of container handling in Hong Kong being high and the vessel turnaround times being short at the container terminals in Hong Kong, the trucking of containers between Hong Kong and PRD has very low productivity in terms of drivers' time, trip time and tractor time. First, a typical round trip for trucking a container from Hong Kong to PRD is less than 160 km and should take less than 3 hours of transportation time. Thus, during one shift of duty, a driver could take at least two trips. In practice, however, the average number of trips per day is only 1.2 (Hong Kong Special Administrative Region, 2004). Second, in an ideal trip, a laden (loaded) container is taken from Hong Kong to PRD and then on the way back, another laden container is returned. In reality, in a trip, a laden container is taken in one direction and the same container is returned empty in the other direction. Even worse, this empty container may be moved later to PRD again for goods loading. Third, a tractor is typically operated for less than 10 h per day (including traveling and waiting at the factory and the border). Most of the time, the tractor is idle. The combined effect is that the

[1] *Source*: Dubai Ports Authority, Port of Hamburg.

productivity of container trucking is less than 40% of what it could be. One major factor that causes the low productivity is the regulatory policies governing the cross-border drayage activities.

1.2. Regulatory policies

Two regulatory policies have the largest impact on the drayage productivity. The first is the *4-up-4-down* policy. Under this policy, when a quadruple of driver–tractor–chassis–container goes from Hong Kong to PRD, the exact same quadruple has to come back to Hong Kong together. The policy was set up in the early 1990s to avoid illegal import of containers to mainland China where an empty container was considered as a commodity rather than an item of transportation equipment. With this policy, the quadruple is considered as a single resource, rather than four individual resources. This policy severely constrains how the resources are used. First, a driver who has taken a laden container from Hong Kong to a factory in PRD needs to wait a long period for unloading. Second, the driver needs to take the emptied container back to Hong Kong, thereby creating low utilization of the driver time and the trip time.

Suppose this policy is relaxed, the "pick up and deliver" operational mode can be launched as illustrated in Fig. 2: the driver-tractor picks up a chassis and a loaded container, then delivers it to one manufacturing site. Instead of waiting, it leaves the chassis and the container there for unloading. Then it goes to another manufacturing site, picks up a loaded container and delivers it to the depot. In this way, the frequency of trips is increased and the utilization of the driver-tractor is significantly improved by bypassing the loading and unloading activities.

Another policy that imposes strong limitation on resource management is the *1-driver-1-tractor* policy. As shown in Fig. 3, under this policy, a Hong Kong driver is licensed to operate on a specified tractor and this

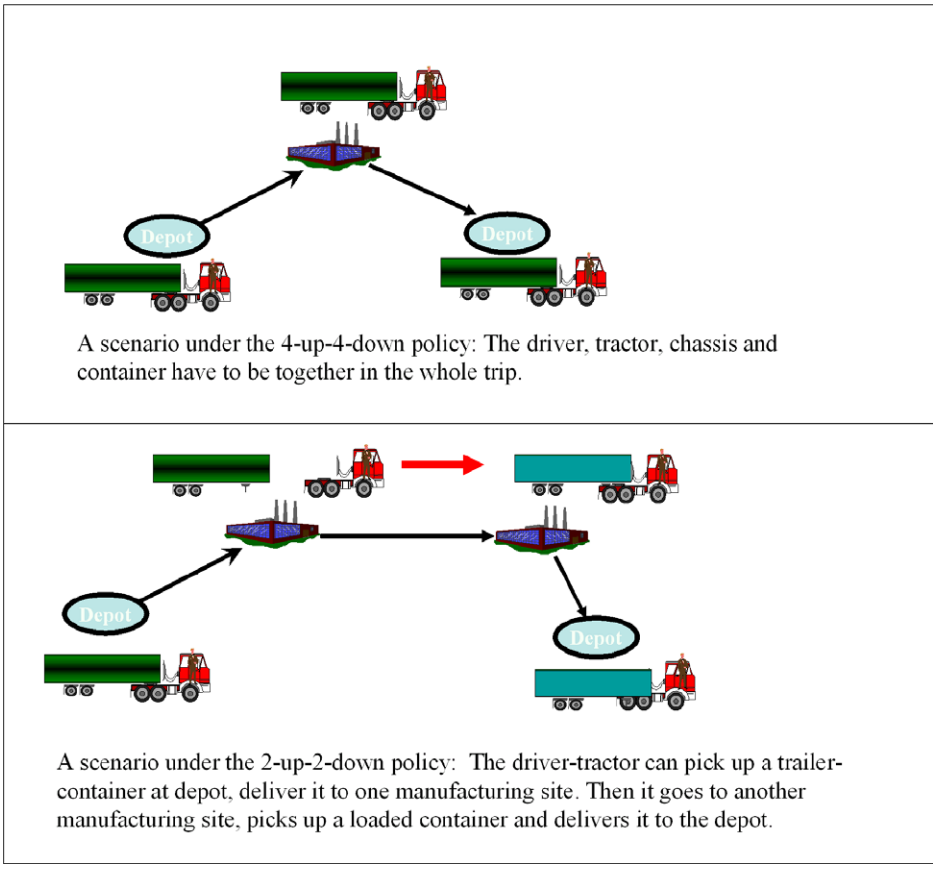

Fig. 2. The impact of the *4-up-4-down* policy.

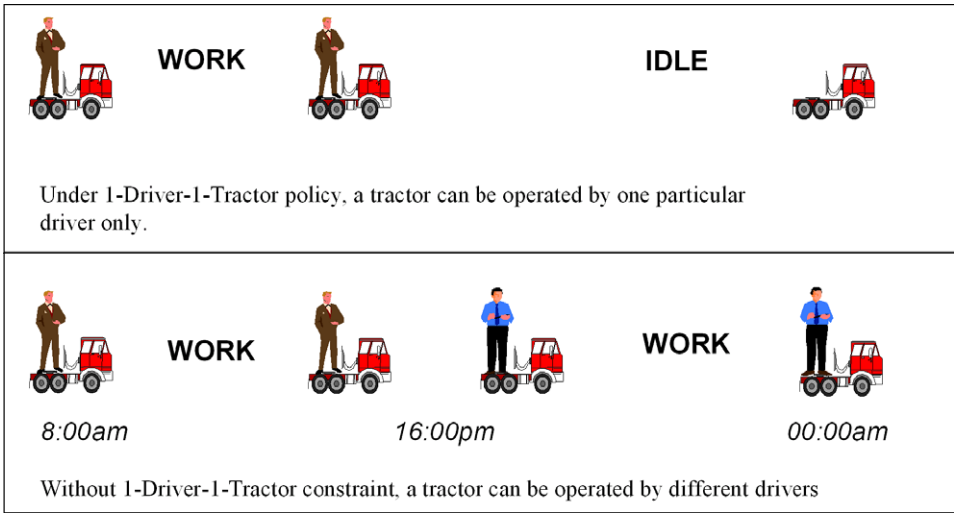

Fig. 3. The impact of the *1-driver-1-tractor* policy.

tractor can only be operated by this particular driver. Such a policy causes a low utilization of the truck time. Suppose this policy is relaxed, a tractor can be operated by multiple drivers of different shifts.

Therefore, the constraints representing the regulatory policies must be incorporated in the models.

In the remaining part of this paper, we consider three circumstances which have different levels of policy restriction on these two policies. We use the notation $\mathscr{P}$ to denote the set of circumstances:

$$\mathscr{P} = \{4\text{-}up\text{-}4\text{-}down, 2\text{-}up\text{-}2\text{-}down, Policy\text{-}free\}$$

In the circumstance of *4-up-4-down*, the quadruple of driver–tractor–chassis–container has to be together during the whole trip and the *1-driver-1-tractor* policy has to be obeyed. In the circumstance of *2-up-2-down*, only the *1-driver-1-tractor* policy has to be obeyed. Under the circumstance of *Policy-free*, both the *4-up-4-down* policy and the *1-driver-1-tractor* policy are relaxed. In the below, without loss of generality, the phases "policy" and "circumstance" are equivalent.

The cross-border drayage problem, is actually a short-haul full-truckload management problem, or, in a broader sense, a vehicle routing and scheduling problem (Bodin and Golden, 1981 and Fisher, 1995). Versions of these problems that consider uncertainty have appeared in the literature. For example, see Dror et al. (1998) for stochastic vehicle routing problems, Smilowitz (2006) for a stochastic dial-a-ride problem, and Powell (1998) for the full-truckload problems with random demands. On land transportation of ocean containers, Cheung et al. (2003) develop a dynamic stochastic approach for assigning drivers to loads. If a driver can cover multiple requests, we can consider the problem as determining the trips of the drivers. The length of the trip is typically constrained by the maximum number of hours a driver can spend on the road. To solve such a problem, tour construction and tour improvement procedures are used (see Ball et al., 1981 and Desrosiers et al., 1995). However, research on modeling the regulatory policies for cross-border drayage activities is not found in the literature.

1.3. Composite resource management

When the flexibility is introduced by relaxing the policies, the quadruple of resources cannot be considered as a single resource. We have to manage those individual resources and the composite of them simultaneously. Meanwhile, we have to respect the practical constraints on using them as well as the interactions between them. Those constraints and operational details cannot be captured by the classical vehicle routing and scheduling models.

The difficulty of drayage operations has attracted the attention of some researchers. Sinclair and Dyk (1987) give an excellent description on practical constraints and issues of this problem. Smilowitz (2006) model

this problem as a multi-resource routing problem and solve it by the column generation technique, regarding the "driver and tractor" as a single resource. Some researchers in artificial intelligence field apply multi-agents system (MAS) in the scheduling and planning of the resources (Fish and Chaibdraa, 1999; Gorodetski et al., 2003). The application of such an approach seems limited to classical vehicle routing problems.

1.4. Contribution

The contributions of this paper are three-fold:

- We discuss the challenges facing the cross-border drayage operators and show how the regulatory policies impact the system.
- We use an attribute–decision model that readily handles a high degree of operational details and incorporates the constraints representing different levels of policy restriction. Furthermore, we implement an efficient labeling algorithm for solving this problem.
- We use this approach as a policy evaluator to quantify the benefit of relaxing the *4-up-4-down* policy and the *1-driver-1-tractor* policy. The numerical experiments demonstrate that the benefit is very significant.

This paper is organized as follows. In Section 2, an attribute–decision model is presented. Section 3 discusses different solution approaches. The numerical experiments are reported in Section 4. Section 5 concludes this paper.

2. The attribute–decision model

Simao et al. (xxxx) introduce a new class of composite resource scheduling problems where different types of resources are involved. To solve this class of problems, they propose an attribute–decision model which not only considers the interactions among resources but also the flexibility of dynamically reusing resources. The fundamental idea is to represent a resource by an attribute vector associated with a decision set. With those attribute–decision pairs, a generic adaptive labeling algorithm can be applied. In this paper, we apply the attribute–decision model for the drayage problem.

2.1. Resources and attribute vectors

In the drayage problem, there are four primitive resources in the system: Driver (D), Tractor (T), Chassis (I) and Container (C). The set of primitive resource classes, is represented using:

$$\mathscr{C}^{\mathscr{R}}: \text{ the collections of primitive resources}: \{\mathrm{D},\ \mathrm{T},\ \mathrm{I},\ \mathrm{C}\}.$$

Any instance of primitive resource can be uniquely described by an attribute vector a. From all attribute vectors, we abstract a vector of common attributes, which is named as a "generic attribute vector":

$$a_g = \begin{bmatrix} a_{g,1} \\ a_{g,2} \\ a_{g,3} \end{bmatrix} = \begin{bmatrix} \text{Time Window} \\ \text{Time} \\ \text{Location} \end{bmatrix}. \tag{1}$$

The availability of one resource is characterized by the attribute "Time Window". The constraints on the working shift of one driver, the maintenance period of a tractor, and the convenient time of picking up and delivering a container, can be specified by this attribute. In our application, the attribute "Time Window" is specified by an interval, for example, [8:00am, 16:00pm]. It could be a union of those intervals in more complex applications.

Each generic attribute vector includes a "Time" attribute. It represents the instant at which the resource will have the attributes in the attribute vector. The "Location" attribute represents the geographical position of the resource.

Besides these generic attributes, each resource may have specific attributes. In our application, the attributes we are concerned with one driver are

$$a_d = \begin{bmatrix} a_g \\ a_{d,1} \\ a_{d,2} \\ a_{d,3} \end{bmatrix} = \begin{bmatrix} \text{Generic attributes} \\ \text{Domicile} \\ \text{Remaining duty hours} \\ \text{Preferred tractor's types} \end{bmatrix}. \tag{2}$$

The drivers may have different domiciles, and their maximum duty hours are protected by the government regulation. Meanwhile, their preferences on the tractors are respected by the attribute "Preferred tractor's types". Since one driver may have several preferred types of tractors, this attribute actually is an ordered list.

The tractors have different types. Each type of tractor may only match some specific types of chassis. Thus we consider the following attributes:

$$a_t = \begin{bmatrix} a_g \\ a_{t,1} \\ a_{t,2} \end{bmatrix} = \begin{bmatrix} \text{Generic attributes} \\ \text{Type} \\ \text{Preferred chassis's types} \end{bmatrix}. \tag{3}$$

The chassis have two lengths: 20 feet or 40 feet. The length determines how many containers can be carried. A 20-foot chassis can carry only one 20-foot container. A 40-foot chassis can carry either one 40-foot container or two 20-foot containers. Due to the tractors' preferences, there are two different types of chassis. We consider the following attributes for the chassis:

$$a_i = \begin{bmatrix} a_g \\ a_{i,1} \\ a_{i,2} \end{bmatrix} = \begin{bmatrix} \text{Generic attributes} \\ \text{Length} \\ \text{Type} \end{bmatrix}. \tag{4}$$

We assume that there are enough empty containers in the system and exclude the necessity of knowingly repositioning them. Therefore, the "container" actually represents the request of transporting a loaded container. It is defined by the origin, destination and its length. In some applications (for example, the inventory routing problem), we can have two resources "container" and "request" respectively since one container can be used for several different requests. Here, we prefer to have the resource "container" only. Its attributes are

$$a_c = \begin{bmatrix} a_g \\ a_{c,1} \\ a_{c,2} \\ a_{c,3} \end{bmatrix} = \begin{bmatrix} \text{Generic attributes} \\ \text{Origin} \\ \text{Destination} \\ \text{Length} \end{bmatrix}. \tag{5}$$

We can combine two primitive resources to get a composite resource. For example, one driver can couple with a tractor and form a composite resource of driver-tractor. A composite resource can further combine another primitive resource to form a new one, or it can be uncoupled into primitive resources which can be reused later. We define

$$\mathscr{C}^{\mathrm{C}}: \text{ the collections of composite resources} : \{\text{D–T, D–T–I, D–T–I–C, T–I, T–I–C, I–C}\}.$$

As primitive resources, we use attribute vectors to characterize the composite resources. Take the example in Fig. 4 for instance: a_{d-t} is the attribute vector of a D–T resource. It can be obtained by concatenating the attributes of a_d and a_t, denoted by: $a_{d-t} = a_d|a_t$. Note that the generic attributes of a_{d-t} are determined by the generic attributes of a_d and those of a_t. More specifically, the attribute "Time Window" of a_{d-t} is the intersection of the "Time window" of a_d and the "Time window" of a_t. The attribute "Time" and "Location" of a_{d-t} should be the same as "Time" and "Location" of a_d and a_t.

According to the roles of the resources in the decision making process, we categorize the resources into two groups: leading resources and passive resources. The leading resources can make decisions and take actions while the passive resources can only be passively involved. Let

$$\mathscr{C}^{\mathrm{L}} = \text{The collections of leading resources} : \{\text{D, D–T, D–T–I, D–T–I–C}\}.$$

$$\mathscr{C}^{\mathrm{P}} = \text{The collections of passive resources} : \{\text{T–I–C, T–I, I–C, T, I}\}.$$

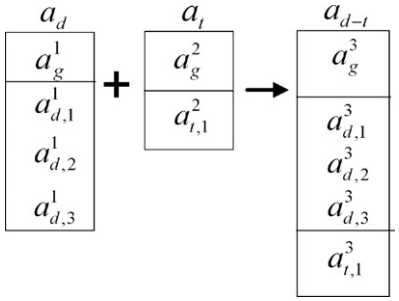

Fig. 4. An attribute vector of a D-T resource.

By definition, we know that: $\mathscr{C}^{\mathrm{C}} \cup \mathscr{C}^{\mathrm{R}} = \mathscr{C}^{\mathrm{L}} \cup \mathscr{C}^{\mathrm{P}}$. We represent the attribute vectors using:

$\mathscr{A}$ = The set of all possible attribute vectors.

$\mathscr{A}_{\mathrm{L}}$ = The set of all possible attribute vectors for all leading resources.

$\mathscr{A}_{\mathrm{P}}$ = The set of all possible attribute vectors for all passive resources.

In the remaining part of this paper, when there is no ambiguity, we will use the term "resource" to represent "an attribute vector of the resource".

2.2. Decisions

What decision can be made on a resource depends not only on the type, but also on the status of the resource. For example, a Hong Kong driver who has less than two hours of remaining duty period should not take a load crossing the border to mainland China. Let

D_a = The set of possible decisions that can be performed on a resource with the attribute vector a.

$$x_{ad} = \begin{cases} 1 & \text{if decision } d \in D_a \text{ is acted on a resource with attribute vector } a \\ 0 & \text{otherwise} \end{cases}$$

There are three types of possible decisions in a composite resource scheduling problem, namely *Couple* decisions, *Uncouple* decisions and *Modify* decisions.

Couple decisions represent the combining of one resource with another. For example we may take a driver–tractor, and act on it by coupling it with a chassis. It is important to notice that we have to model not only the change in the attributes of the resource, but also the fact that we no longer have the chassis as an individual resource.

Uncouple decisions are the reverse of the couple decisions in which a composite resource is decomposed into individual resources. If a composite resource is to be decomposed into more than two resources, we can perform the uncouple decision multiple times.

Modify decisions do not involve any interaction of resources. It only changes the attributes of resources. For example, if one driver–tractor–chassis resource moves from the depot to one manufacturing site, the decision "move" is referred to as a modify decision since it only changes the resource's "Location" attribute.

By definition, any decision is involved with a leading resource, however, only 'Couple' decisions are involved with passive resources. To describe the relationships between a decision and the involved resources, let us define:

$$\delta(a,d) = \begin{cases} 1 & \text{if } a \text{ is the leading resource which executes the decision } d \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda(p,d) = \begin{cases} 1 & \text{if the passive resource} p \text{is involved in the decision } d \\ 0 & \text{otherwise} \end{cases}$$

We can use a *transferring function* to represent the consequence of a decision d (let $d \in D_a$). The resulting leading resource a' of the decision d can be denoted as $a\prime = \mathscr{M}^{\mathrm{L}}(a,d)$. For the uncouple type decision, we have to record the resulting passive resource p', denoted by $p' = \mathscr{M}^{\mathrm{P}}(a,d)$. It is especially important to recognize that "Time" is handled as an attribute. If we couple a resource that is available at time t with a resource that is available at time t', and it takes time τ to complete the coupling, then the resulting resource is available at time $\max\{t,t'\}+\tau$. These calculations are all imbedded in the transferring function.

Mathematically, the relationships between a decision and its resulting resource attribute vectors can be represented by the following indicators:

$$\gamma(a',d) = \begin{cases} 1 & \text{if } a' \text{ is the resulting leading resource of the decision } d \\ 0 & \text{otherwise} \end{cases}$$

$$\beta(p',d) = \begin{cases} 1 & \text{if } p' \text{ is the resulting passive resource of the decision } d \\ 0 & \text{otherwise} \end{cases}$$

The relationships between the resources and the decisions can be further illustrated by the example in Fig. 5. The attribute vector a^1 represents one resource of D–T–I type. It couples a container p^1. Let d_1 be such a "couple" decision. Then, its resulting leading attribute vector a^2 can be obtained by the transferring function $\mathscr{M}^{\mathrm{L}}(a^1,d_1)$. The attribute vector a^2 then uncouples the container at the destination of the container. Let d_2 be such an "uncouple" decision. Then its resulting leading attribute vector a^3 can be obtained by $\mathscr{M}^{\mathrm{L}}(a^2,d_2)$ and its resulting passive attribute vector p^2 can be obtained by $\mathscr{M}^{\mathrm{P}}(a^2,d_2)$. According to the definitions, $\delta(a^1,d_1)=1$, $\lambda(p^1,d_1)=1$, $\gamma(a^2,d_1)=1$, $\gamma(a^3,d_2)=1$ and $\beta(p^2,d_2)=1$. In the remaining part of this paper, we use $p(d)$ to represent one passive resource such that $\lambda(p,d)=1$.

A resource can generally be acted on with more than one type of decision, which we refer to as "decision set". The idea is best illustrated with a D–T–I composite resource. A driver with a tractor and a chassis may: (1) load a full-load container at the manufacturer's site; (2) uncouple the chassis and leave it at the manufacturer's site for loading; (3) move to another location.

In our application, the decision set for one driver is

$$D_a = [d_1] = [\text{Couple a tractor}]. \tag{6}$$

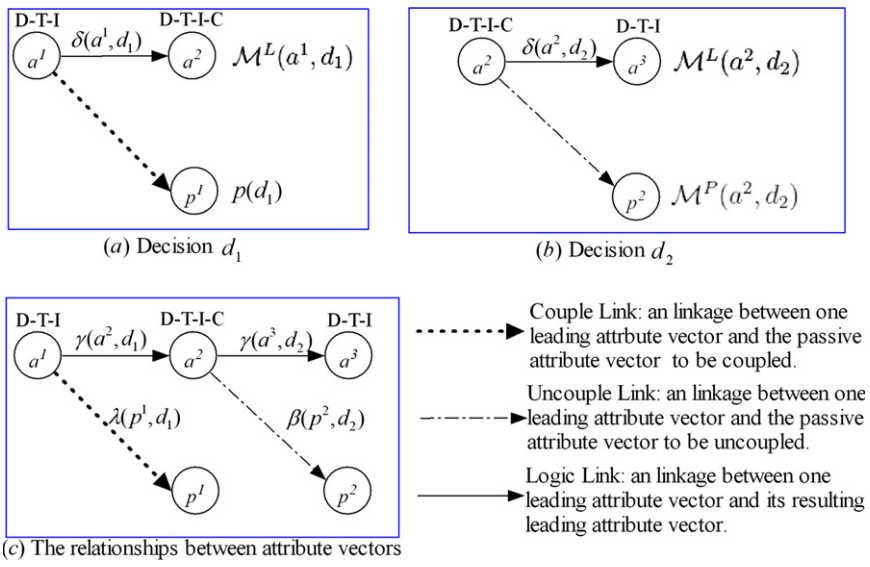

Fig. 5. The relationships between attribute vectors.

For one driver-tractor,

$$D_a = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} \text{Couple a chassis} \\ \text{Uncouple a tractor} \\ \text{Move back to domicile} \end{bmatrix}. \tag{7}$$

For one driver–tractor–chassis,

$$D_a = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} \text{Load a container} \\ \text{Empty move to another place} \\ \text{Uncouple a chassis} \end{bmatrix}. \tag{8}$$

For one driver–tractor–chassis–container,

$$D_a = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} \text{Unload a container} \\ \text{Load one more container} \\ \text{Unload the chassis together the container} \\ \text{Empty move to another place} \end{bmatrix}. \tag{9}$$

Note that the feasible decision set for any given attribute vector is policy dependent. In the following, we use $D_a^{\text{4-up-4-down}}$, $D_a^{\text{2-up-2-down}}$ and D_a^{free} be the decision sets for attribute vector a under the *4-up-4-down* policy, the *2-up-2-down* policy and the *Policy-free* situations respectively. Generating the decision set for one attribute vector is easy to implement by rule-based programs.

2.3. System dynamics

We now can express the flow conservation constraints for the system:

$$\sum_{d' \in D_{a'}} x_{a'd'} - \sum_{a \in \mathscr{A}_L} \sum_{d \in D_a} \gamma(a', d) x_{ad} = R_{a'}, \quad a' \in \mathscr{A}_{\mathrm{L}} \tag{10}$$

$$\sum_{a \in \mathscr{A}_L} \sum_{d \in D_a} x_{ad} \lambda(p, d) - \sum_{a \in \mathscr{A}_L} \sum_{d \in D_a} \beta(p, d) x_{ad} \leqslant R_p, \quad p \in \mathscr{A}_{\mathrm{P}} \tag{11}$$

where $R_{a'}$ is the number of initial leading resources with attribute vector a' and R_p is the number of initial passive resources with attribute vector p.

2.4. Objective function

The objective function can be simply stated as

$$\max \sum_{a \in \mathscr{A}_L} \sum_{d \in D_a} c_{ad} x_{ad} \tag{12}$$

where c_{ad} is the contribution of one action d acted on a. In the implementation, such an outcome is computed by rule-based functions which take account of the violation of "soft" constraints such as the time window constraints.

Through this model, all the constraints of the problem are contained in the specification of the attribute vectors, the determination of the decision set function D_a, and the computation of the outcomes of the decisions (which contain most of the engineering calculations).

3. Solution approaches

We first introduce an exact method which can be used to solve small size problems exactly. The optimal solutions obtained by this method can be regarded as the benchmarks. We then use the labeling algorithm proposed in Simao et al. (XXXX) which can solve problems of much larger size.

3.1. An exact integer programming approach

To solve the integer programming problem of (10)–(12), by the commercial integer programming solver, we have to enumerate the complete set of attributes vectors and the complete set of decisions. They can be obtained by a tree search procedure:

Tree Search Procedure

Step 1: *Initialization*Generate the attribute vectors for all drivers. Put them in a list Q.
Step 2: *Breadth-first search for one attribute vector*Remove one attribute vector a from Q. Generate its decision set D_a. For each $d \in D_a$, generate $a' = \mathscr{M}^{\mathrm{L}}(a,d)$ and $p' = \mathscr{M}^{\mathrm{P}}(a,d)$. Put a' in the tail of Q. Set up the indicators $\delta(a,d)$, $\beta(p',d)$, $\gamma(a',d)$, $\lambda(p(d),d)$ and compute the contribution c_{ad}.
Step 3: *Termination check*If Q is empty, stop. Otherwise go back to Step 2.

After running this procedure, we can write the integer programming formulations and solve the problem by CPLEX. The complication for practical problems is that the number of attribute vectors can become quite large. It is intractable to enumerate all attribute vectors for middle size problems. In our application, a PC with 1.0 G memory and 2.4 GHZ CPU cannot store the attribute vectors for a problem of 2 drivers, 2 tractors, 2 chassis and 10 containers.

To solve this problem, we, therefore, implement a labeling algorithm proposed in Simao et al. (XXXX) which intelligently searches out good solutions. In the dynamic fleet management literature, Powell et al. (2000) have applied a multi-attribute labeling method to solve a driver-task assignment problem where there is only one type of resources which have complex attributes. Powell et al. (2002) use an adaptive dynamic programming algorithm for heterogeneous resource allocation problems, which again only considers one type of resources. Simao et al. (XXXX) propose a labeling algorithm that can solve composite resource scheduling problem. We demonstrate that this algorithm works well for the drayage problem. In the next section, we introduce the solution approach in detail.

3.2. Labeling algorithm

When a decision is made, the attribute vectors of the resources involved will be changed (the values of its elements or even its dimensions). To tackle this attribute–decision model, we can build a candidate list, Q, that contains a list of attribute vectors. For a given attribute vector, a, we select and make the most beneficial decision among all decisions that can be applied. The decision, however, may result in some changes to the system such as making a previously made decision invalid, reducing or increasing the number of resources available, etc.

We will use the following notations in the description of the algorithm:

π_p = the dual price of one passive resource p
n = iteration counter
$N_{\max}$ = a preset maximum number of iterations
z_{ad} = the net contribution of the decision $d \in D_a$

The major steps are

Step 1: *Initialization:*
Generate the attribute vectors for all primitive resources.
Put the attribute vectors of the drivers in a candidate list Q.
Put the attribute vectors of the tractors, trailers and containers to the set $\mathscr{A}_{\mathrm{P}}$.
Set $\pi_p = 0$ for any $p \in \mathscr{A}_{\mathrm{P}}$.
Set $\overline{D} = \emptyset$ and $n = 0$.
Explanation: The drivers in the list Q are sorted in the sequence of ascending available times. Namely, the ones with earliest available times will be put to the head of Q. The set $\overline{D}$ is used to store all previously made decisions.

Step 2: *Choose the best decision:*
Set $n = n + 1$.
Remove an attribute vector, a, from the head of Q.
Generate the decision set D_a.
For each decision $d \in D_a$,
Set $z_{ad} = 0$.
Compute the direct contribution c_{ad}, update $z_{ad} \leftarrow z_{ad} + c_{ad}$.
If the decision d involves with the passive resource $p(d)$,
set $z_{ad} \leftarrow z_{ad} - \pi_{p(d)}$.
End If
$a' \leftarrow$ the resulting attribute vector of d.
Obtain $v(a')$ and set $z_{ad} \leftarrow z_{ad} + v(a')$.
End For
Obtain the best decision $\hat{d}$ from D_a, that is, $\hat{d} = \arg\max_{d \in D_a}\{z_{ad}\}$.

Explanation: The net contribution of a decision d consists of three cost elements. The first is the direct contribution (or cost) c_{ad} of the decision. The second is the *estimated price* $\pi_{p(d)}$ that we are prepared to pay for the resource $p(d)$ that is used by decision d. The third is the future value $v(a')$ of the attribute a' that is resulted from the decision just made.

The values $v(a')$ estimate the future contributions of the resulting attribute vectors. To calculate an estimation, we can use a parameter-controlled tree search approach. The procedure is similar with Step 2 in *Tree Search Procedure* of the exact method except that the depth and the width of the search are controlled by parameters: m_1, m_2, respectively. From this tree, we select the path with the largest contribution and set the contribution of this path as $v(a')$. The procedure can be described as follows:

Estimate $v(a')$

Step 1: *Initialization*
Put a' in a list Q. Let the breadth courter $i = 0$.

Step 2: *Breadth-first search for one attribute vector*
Remove the attribute vector a' from the top of Q. Generate its decision set $D_{a'}$. Let d be one of the best m_2 decisions in $D_{a'}$. For each d, generate $a'' = \mathscr{M}^{\mathrm{L}}(a', d)$ and $p' = \mathscr{M}^{\mathrm{P}}(a', d)$. Put a'' in the tail of Q and compute the contribution $c_{a'd}$, and if $i = m_1 - 1$, put a' in the top of another list Q'. Set $i = i + 1$.

Step 3: *Backtrack the future value*
If $i = m_1$, set $v(a'') = 0$. Retrieve one element a'' from Q'. Identify the leading attribute vector a' and its decision d that results a''. Update $v(a') \leftarrow \min\{v(a'), v(a'') + c_{a'd}\}$. Put a' at the tail of Q'. Repeat this step until Q' becomes empty.

In our short-haul truckload problem, the partial tree search strategy is affordable since for each driver the possible number of trips per day is typically no more than four.

The price $\pi_{p(d)}$ measures the level of competition for getting the resource p that is involved in d. The more intense the competition, the higher the price should be. It will be updated during the execution of the best decision.

Step 3: *Execute the best decision* $\hat{d}$:
If the decision $\hat{d}$ exists,[2]
$a' \leftarrow$ the resulting leading attribute vector of $\hat{d}$.
$p' \leftarrow$ the resulting passive attribute vector of $\hat{d}$.
Put a' to the head of Q.
Put p' to the set $\mathscr{A}_{\mathrm{P}}$ if p' exists.

[2] Due to the availability of resources, the attribute vector a may not be able to find a feasible decision.

Set $\delta(a,\hat{d})=1$, $\beta(p',\hat{d})=1$, $\gamma(a',\hat{d})=1$, $\lambda(p(\hat{d}),\hat{d})=1$.
If the decision $\hat{d}$ involves with the passive resource $p(\hat{d})$,
obtain the second best decision $\bar{d}$ by:
$\bar{d}=\arg\max_{d\in D_a,d\neq\hat{d}}\{z_{ad}\}$.
Let $\pi_{p(\hat{d})}\leftarrow\pi_{p(\hat{d})}+z_{a\hat{d}}-z_{a\bar{d}}+\epsilon$.
End if
End If

Explanation: One way to update $\pi_{p(\hat{d})}$ is to use the concept of opportunity cost. The opportunity cost for not implementing $\hat{d}$ but implementing $\bar{d}$ is $z_{a\hat{d}}-z_{a\bar{d}}$. The rationale is that if another resource wants to couple with the same resource, the benefit created by this new couple should at least offset the opportunity cost.

Besides this opportunity cost, we add a small value ϵ to break the tie. This idea is borrowed from the auction algorithm in Bertsekas et al. (1997).

Note that the resulting attribute vector a' is put to the head of the Q. In this way, the resources with the largest number of primitive resources are considered first in most cases.

Step 4: *Decision adjustment*
For each decision $d'\in\overline{D}$,
If $\gamma(p(\hat{d}),d')=1$, /*, i.e., there is some previously made decision d' used the resource $p(\hat{d})$.*/
adjust all decisions related to the decision d'.
End If
End For
Explanation: When a previously made decision is being canceled, the downstream decisions based on this decision need to be canceled. Using the example in Fig. 6, if the chassis with attribute vector p^1 is recoupled with a D-R resource with attribute vector a^7, then all the decisions related to a^1, which previously couple with p^1, should be removed.

Step 5: *Secondary dual price adjustment*
Add $\hat{d}$ to $\overline{D}$.
If n is divisible by M
For each $p\in\mathscr{A}_P$,
If $\sum_{d'\in\overline{D}}\lambda(p,d')=0$, /*, i.e., the resource p has not been used. */
Update $\pi_p\leftarrow\pi_p*\beta$, /* β is a parameter within (0,1).*/
End If
End For
End If
Explanation: After a number of iterations, the dual prices of some passive resources can be increased too high such that they may be ignored by the leading resources. Thus, we reduce the price by a factor β every M iterations as long as these passive resources remain unused.

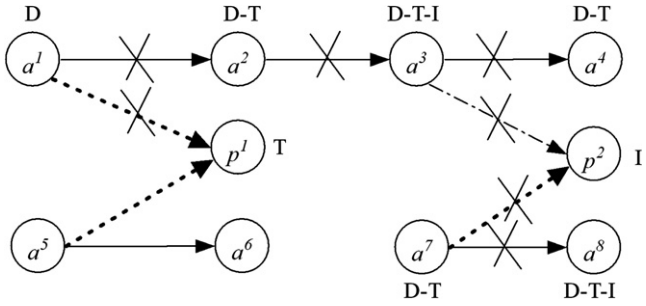

Fig. 6. The couple decision, which was previously made on a^1, is adjusted. All links with "×" represent the decisions should be adjusted.

Step 6: *Termination check*

If $Q = \emptyset$ or $n = N_{\max} + 1$
terminate,
Else
If $n = N_{\max}$,
For each $p \in \mathscr{A}_P$,
If $\sum_{d' \in \overline{D}} \lambda(p, d') = 0$, /*, i.e., the resource p has not been used. */
Set $\pi_p = 0$.
End If
End For
End If
Go to Step 2.
End If

Explanation: If Q is null or $n = N_{\max} + 1$, terminate the algorithm. It is possible that after $N_{\max}$ iterations, some containers have not been covered. Thus, we reduce the dual prices of all unused containers to 0. In this way, some of the unserved containers may be covered.

4. Numerical experiment

We conduct numerical experiments to estimate the impact of different policies on the system performance.

4.1. Problem setting

We generate a set of test problems and use a time horizon of 720 periods (representing 24 h with 720 two-minute intervals). The origin and the destination of a load are randomly located in a 100-by-100 unit square. The travel speed is 1 unit per period. The middle point of the time-window of picking up a container is generated uniformly between 0 and 360 while the width of the window is generated uniformly between 120 and 180. The number of 40-foot loads and the number of 20-foot loads are controlled by a ratio λ:

$$\lambda = \frac{\text{Number of 40-foot loads}}{\text{Number of 40-foot loads} + \text{Number of 20-foot loads}} \tag{13}$$

We assume that the tractors and the chassis are available during the whole planning horizon such that they can be reused.

The major cost items include:

- *Matching cost*: When we couple a "driver" with a "tractor", the matching cost depends on whether the driver "prefers" the tractor. The cost can be set as a big-M if the driver is not able to operate the "tractor". Similarly, the coupling between a "driver-tractor" with "chassis" also involves such a matching cost.
- *Transportation cost*: The major part of transportation cost is the oil consumption fee that is proportional to the transportation time/distance. As a simplification, the transportation cost from location i to location j is independent with the resources' attributes and can be calculated by the following formula:

$$c_{ij} = d^*_{ij} 15 \tag{14}$$

  where d_{ij} is the distance.
- *Penalty of violating the time window*: The time window constraints for drivers, tractors and chassis are hard constraints which cannot be violated. While the time window constraints for picking up and delivering a container can be violated by paying a penalty. The penalty is set as 10 per time unit for per cargo.
- *Reward*: The reward will be gained when the full-load container arrives at the destination. In our experiments, it is set to be a function of the size of container. The 40-foot container has a reward of 4000, and the 20-foot container has a reward of 2000.

In the below, we use N_D, N_T, N_I, N_C to represent the numbers of drivers, tractors, chassis and containers respectively.

The parameters that are used in our numerical experiments are listed in Table 1.

4.2. Solution quality

We conduct a set of experiments to evaluate the solution quality of the labeling algorithm. We create a set of test problems whose parameters are listed in Table 2.

We compare the solutions obtained by the labeling algorithm with the optimal solutions obtained by the exact method introduced in Section 3. The results are shown in Table 3. In this table, columns 3 and 5 record the solutions obtained by CPLEX and the labeling algorithm respectively. Columns 4 and 6 record the computational times (measured by seconds), while column 7 records the solution gap.

The results show that for most test problems, the labeling algorithm obtains the optimal solutions. The largest gap between the solutions is within 2.5%. On the other hand, the computation times of the labeling algorithm are less than those of the exact method.

Now, we are ready to use the labeling algorithm as a "policy evaluator" to evaluate the system performances under different policies. We conduct two set of experiments. The first set of experiments is to compare the system performance under the *4-up-4-down* policy with the correspondence under the *2-up-2-down* policy. The second set of experiments is to compare the system performance under the *2-up-2-down* policy with the correspondence under the *policy-free* policy.

Table 1
Parameters

N_{max}	$400N_D$
M	$40N_D$
β	0.8
m_1	4
m_2	3
ϵ	50

Table 2
Test problems

Problem	Policy	N_D	N_T	N_I	N_C
1	*4-up-4-down*	1	1	1	4
2	*4-up-4-down*	1	1	1	6
3	*4-up-4-down*	1	1	1	8
4	*4-up-4-down*	2	2	2	4
5	*4-up-4-down*	2	2	2	6
6	*4-up-4-down*	2	2	2	8
7	*2-up-2-down*	1	1	1	4
8	*2-up-2-down*	1	1	1	6
9	*2-up-2-down*	1	1	1	8
10	*2-up-2-down*	2	2	2	4
11	*2-up-2-down*	2	2	2	6
12	*2-up-2-down*	2	2	2	8
13	*Policy-free*	2	1	1	4
14	*Policy-free*	2	1	1	6
15	*Policy-free*	2	1	1	8
16	*Policy-free*	4	2	2	4
17	*Policy-free*	4	2	2	6
18	*Policy-free*	4	2	2	8
19	*4-up-4-down*	2	2	2	9
20	*2-up-2-down*	2	2	2	9
21	*Policy-free*	4	2	2	9

Table 3
Comparisons between the exact method with labeling algorithm for small problems

Problem	Exact method		Labeling algorithm		Solution gap (%)
	Profit	Time	Profit	Time	
1	3537	3.3	3537	0.7	0
2	7071	4.9	7071	1.1	0
3	7071	16.2	7071	1.1	0
4	6764	4.1	6764	2.4	0
5	8771	4.3	8771	2.5	0
6	10,569	31.3	10,569	3.3	0
7	7076	1.4	7076	1.2	0
8	7076	3.3	7076	1.5	0
9	11,064	27	11,064	1.5	0
10	10,211	3.4	10,210	2.1	0
11	12,306	4.3	12,169	2.6	1.1
12	18,140	45.0	18,140	3.6	0
13	10,211	3.4	10,211	0.3	0
14	12,306	4.3	12,306	0.8	0
15	18,140	45.0	18,140	3.6	0
16	11,770	6.2	11,468	4.9	2.5
17	16,822	13.3	16,818	6.8	0.2
18	22,391	70.8	22,387	7.8	0.2
19	7071	195.4	7071	5.7	0
20	18,308	467.9	18,308	3.2	0
21	24,327	867.3	24,327	9.1	0

4.3. 4-up-4-down policy vs. 2-up-2-down policy

We create two sets of test problems. In the first set of test problems, the number of containers is fixed as 48 and we consider three possible ratios λ: 25%, 50% and 75%. For each case, we increase the number of drivers from 4 to 10. Using this set of problems, we can see the marginal benefit of adding one driver in the system. All drivers are available in $[0, 360]$. We assume there are enough tractors and chassis, namely, $N_D = N_T = N_I$. In such a setting, the *Policy-free* policy is no difference with the *2-up-2-down* policy since there are enough tractors.

Table 4 illustrates the result for the first set of test problems. Columns 3 and 5 record the contributions under the *4-up-4-down* policy and the *2-up-2-down* policy respectively. Columns 4 and 6 record the computational times (measured by seconds), while column 7 records the contribution improvement under the *2-up-2-down* policy compared with the *4-up-4-down* policy. The result shows that as more drivers are put in the

Table 4
System performances under the *4-up-4-down* policy and the *2-up-2-down* policy, when the number of drivers increases

N_D	λ (%)	*4-up-4-down*		*2-up-2-down*		Improvement (%)
		Profit	Time	Profit	Time	
4	25	29,661	9.9	48,087	12.9	62
6	25	40,061	11.5	83,154	11.8	107
8	25	43,425	13.1	97,827	11.7	125
10	25	57,480	18.3	100,221	25.8	74
4	50	30,457	8.8	54,516	11.7	78
6	50	38,562	11.1	67,349	9.5	74
8	50	50,532	17.3	75,693	27.4	49
10	50	52,610	20.0	91,592	34.6	74
4	75	20,794	6.5	48,222	12.0	131
6	75	30,470	7.4	52,964	24.4	73
8	75	44,701	9.8	64,956	42.4	45
10	75	50,803	18.0	82,400	47.2	62

system, more contribution will be earned. The drivers can earn more money when λ is smaller since there are more 40-foot containers in the system.

In the second set of test problems, we increase the number of containers from 60 to 240. The number of drivers is increased from 10 to 40 accordingly. Namely, $N_D = N_C/6$. We assume that there are enough tractors and chassis in the system, namely, $N_T = N_I = N_D$. The format of Table 5 is the same as Table 4. The result shows that under the *2-up-2-down* policy, we can always achieve better results than under the *4-up-4-down* policy. In addition, the algorithm can compute the result for a problem with 40 drivers and 240 containers in reasonable time.

4.4. 2-up-2-down policy vs. Policy-free policy

The purpose of this set of experiments is to compare the *2-up-2-down* policy and the *Policy-free* policy. The drivers in the system work in two different shifts. The first group of drivers work in the period [0, 360] while the second group of drivers work in the period [360, 720]. Under the *2-up-2-down* policy, one tractor can only be licensed to one driver. While under the *Policy-free* policy, one tractor can be operated by two drivers of different shifts.

We first consider a set of test problems where there are 48 containers. We increase the number of drivers from 4 to 10. The tractors are regarded as scarce resources and their number is set as $N_T = N_D/2$. The results are shown in Table 6. In Table 6, the term "*2U*", "*4U*", "*PF*" represent the *2-up-2-down* policy, the *4-up-4-down* policy and *Policy-free* policy respectively.

Table 5
System performances under the *4-up-4-down* policy and the *2-up-2-down* policy, when the number of containers increases

N_C	λ (%)	*4-up-4-down*		*2-up-2-down*		Improvement (%)
		Profit	Time	Profit	Time	
60	25	57,533	24.8	112,036	47.6	95
120	25	110,638	202.1	254,523	270.5	130
240	25	221,502	1345.4	504,654	803.7	128
60	50	36,015	24.1	101,421	30.9	182
120	50	115,239	236.3	252,154	414.2	119
240	50	221,463	1272.2	512,440	1887.3	131
60	75	48,582	49.1	91,832	53.2	89
120	75	106,121	230.7	221,772	646.9	109
240	75	210,609	1199.2	441,714	1807.1	110

Table 6
System performances under the *4-up-4-down* policy, the *2-up-2-down* policy and the *Policy-free* policy, when the number of tractors increases

N_T	λ (%)	*4-up-4-down*		*2-up-2-down*		*Policy-free*		Improvement (%)	
		Profit	Time	Profit	Time	Profit	Time	*2U/4U*	*PF/2U*
4	25	14,660	3.7	47,320	6.2	83,546	9.6	233	77
6	25	27,320	5.8	58,934	11.8	91,334	10.6	116	55
8	25	31,321	7.3	66,926	9.6	119,387	11.4	114	78
10	25	34,798	6.3	82,525	14.2	170,916	14.1	137	107
4	50	13,998	3.4	43,187	19.9	69,039	16.4	209	60
6	50	20,785	3.8	47,065	22.0	80,870	13.7	126	72
8	50	22,576	12.5	58,034	18.6	92,229	22.1	157	59
10	50	27,434	8.8	66,103	28.8	112,108	41.7	141	70
4	75	13,908	3.9	34,961	38.0	56,000	16.8	151	60
6	75	20,726	11.8	44,593	17.4	65,899	13.7	115	48
8	75	20,916	18.2	46,027	30.4	73,998	36.7	120	61
10	75	23,454	13.8	49,799	31.7	79,999	65.0	112	61

Table 7
System performances under the *4-up-4-down* policy, the *2-up-2-down* policy and the *Policy-free* policy, when the number of containers increases

N_C	λ (%)	*4-up-4-down*		*2-up-2-down*		*Policy-free*		Improvement (%)	
		Profit	Time	Profit	Time	Profit	Time	*2U/4U*	*PF/2U*
60	25	35,448	14.1	86,781	13.5	145,261	13.1	144	67
120	25	72,299	71.3	164,988	92.7	281,788	28.1	128	71
240	25	145,532	701.8	380,517	747.1	620,846	2766.0	161	63
60	50	29,308	27.7	75,765	27.8	113,631	30.2	158	50
120	50	56,801	110.9	143,787	185.8	278,333	527.6	153	94
240	50	115,426	1323.3	333,048	1638.4	493,418	4444.6	188	48
60	75	23,728	25.3	53,621	48.3	88,855	47.2	125	66
120	75	50,903	203.0	131,615	309.5	205,734	720.9	158	56
240	75	98,007	1514.4	185,287	2512.1	317,799	9747.7	89	72

Secondly, we consider a set of test problems where the number of containers increases from 60 to 240, the number of drivers increases from 10 to 40, respectively, (namely, $N_R = N_C/6$). We also assume that there are enough chassis. The results are shown in Table 7.

According to our numerical experiments, if the *4-up-4-down* policy is relaxed, the system performance can be improved by more than 100%. If the *1-driver-1-tractor* policy is relaxed, the system performance can be further improved by more than 70%. The experimental results also show the efficiency of the labeling algorithm. Our approach can handle the problem with 40 drivers, 40 tractors, 40 chassis and 240 containers in reasonable time.

5. Conclusion

We have described the challenges in managing cross-border drayage operations. We show that two regulatory policies, *4-up-4-down* and *1-driver-1-tractor*, significantly restrict the trucking efficiency. Meanwhile, it is shown that the attribute–decision modeling perspective together with the adaptive labeling algorithm can provide good solutions for such a complex problem. It is interesting to apply this attribute–decision model to other applications such as handicapped person transportation problem where different vehicles can be used to serve different types of patients. It is also very interesting to investigate the theoretical foundation for the labeling algorithm.

Acknowledgement

The authors thank the Research Grants Council of Hong Kong to support the research through Grant 612206.

References

Ball, M., Golden, B., Assad, A., Bodin, L., 1981. Planning for truck fleet size in the presence of a common carrier option. Decision Sciences 14 (1), 103–120.

Bertsekas, D.P., Polymenakos, L.C., Tseng, P., 1997. An epsilon-relaxation method for convex network optimization problems. SIAM Journal on Optimization 7, 853–870.

Bodin, L., Golden, B., 1981. Classification in vehicle routing and scheduling. Networks 11, 97–108.

Cheung, R., Powell, W., Hang, D., 2003. Multi-attribute label matching algorithms for vehicle routing problems with time windows and backhauls. IIE Transaction 35, 191–205.

Desrosiers, J., Solomon, M., Soumis, F., 1995. Time constrained routing and scheduling. In: Monma, C., Magnanti, T., Ball, M. (Eds.), Handbook in Operations Research and Management Science Volume on Networks. North Holland.

Dror, M., Laporte, G., Trudeau, P., 1998. Vehicle routing with stochastic demands: properties and solution frameworks. Transportation Science 23, 166–176.

Fish, K., Chaibdraa, B., 1999. A simulation approach based on negotiation and cooperation between agents: a case study. IEEE Transactions on Systems, Man, and Cybernetics Part C 29 (4), 531–545.

Fisher, M., 1995. Vehicle Routing. Network routing. In: Ball, M. et al. (Eds.), Handbooks in Operations Research and Management Science 8, 1–33.

Gorodetski, V., Karsaev, O., Konushy, V., 2003. Multi-agent system for resource allocation and scheduling. In: Proceedings of Multi-Agent Systems and Applications III: Third International Central and Eastern European Conference on Multi-Agent Systems, 236–246.

Hong Kong Special Administrative Region, 2004. Study on Hong Kong Port – Master Plan 2020, Economic Development and Labour Bureau.

Powell, W., 1998. A stochastic model of the dynamic vehicle allocation problem. Transportation Science 20, 117–129.

Powell, W., Snow, W., Cheung, R., 2000. Adaptive labeling algorithms for the dynamic assignment problem. Transportation Science 34, 50–66.

Powell, W., Shapiro, J., Simao, H.P., 2002. An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem. Transportation Science 36, 231–249.

Simao, H.P., Powell, W., Cheung, R, xxxx. A multi-layered resource scheduling problem. Working Paper.

Sinclair, M., Dyk, E.V., 1987. Combined routing and scheduling for the transportation of containerized cargo. The Journal of the Operational Research Society 44 (6), 951–963.

Smilowitz, K., 2006. Multi-resource routing with flexible tasks: an application in drayage operations. IIE Transactions 38, 555–568.

Steenken, D., Voß, S., Stahlbock, R., 2004. Container terminal operation and operations research – a classification and literature review. OR Spectrum 26 (1), 3–49.