



Discrete Optimization

The freight consolidation and containerization problem

Hu Qin^a, Zizhen Zhang^{b,*}, Zhuxuan Qi^c, Andrew Lim^b^a School of Management, Huazhong University of Science and Technology, No. 1037, Luoyu Road, Wuhan, China^b Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong^c School of Labor and Human Resources, Renmin University of China, Beijing 100872, China

ARTICLE INFO

Article history:

Received 19 August 2012

Accepted 12 September 2013

Available online 21 September 2013

Keywords:

Packing

Freight consolidation

Containerization

Memetic algorithm

Third-party logistics

ABSTRACT

In today's global free market, third-party logistics providers (3PLs) are becoming increasingly important. This paper studies a problem faced by a 3PL operating a warehouse in Shanghai, China, under contract with a major manufacturer of children's clothing based in the United States. At the warehouse, the 3PL receives textile parcel shipments from the suppliers located in China; each shipment is destined for different retail stores located across the United States. These shipments must be consolidated and loaded into containers of varying sizes and costs, and then sent along shipping routes to different destination ports. An express company, such as UPS and FedEx, unloads the shipments from the containers at the destination ports and distributes them to their corresponding stores or retailers by parcel delivery. The objective is to find an allocation that minimizes the total container transportation and parcel delivery costs. We formulate the problem into an integer programming model, and also propose a memetic algorithm approach to solve the problem practically. A demonstration of a good solution to this problem was a decisive factor in the awarding of the contract to the 3PL in question.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The use of third-party logistics providers (3PL) has become an increasingly integral part of modern supply chain management (Marasco, 2008). Major business enterprises commonly span across several geographical locations at different sides of international borders, which makes central logistics planning difficult and inefficient. Consequently, companies prefer to employ the services of 3PLs for the storage and transportation of goods along parts of the supply chain. There are several benefits for doing so: (1) the company avoids the setup costs involved in managing the logistics in a new location; (2) 3PLs specialize in logistics and are likely to perform the required tasks more efficiently than the customer by taking advantage of economies of scale; and (3) 3PLs possess regional expertise and are therefore better able to take local conditions into account during logistics management. Furthermore, in addition to the storage and transportation of goods, 3PLs often provide other value-added services such as inbound operations, inspection, sorting, labeling, containerization, tracking

and outbound operations. These and other services position 3PLs as one-stop shops for the logistical needs of their customers.

We consider the case of a major manufacturer of clothing for babies, toddlers and children. Products bearing the manufacturer's brand are sold at over 400 company-owned retailer stores as well as thousands of national department stores and some of the largest retailers across the United States. The company has outsourced some of its manufacturing needs to suppliers located in China, taking advantage of the lower costs in this rapidly developing region. As a result, they have contracted with 3PLs based in China for transporting these manufactured products to their target markets. This enables the company to focus on its core business while saving on costs without sacrificing product quality.

This study is motivated by a project awarded to our team by a 3PL that services the manufacturer via a warehouse hub in Shanghai, China. The 3PL industry in China has experienced significant and rapid growth since the economic reforms of 1978 when its borders were opened to foreign investors, and especially since its entry into the World Trade Organization in 2001. Historically, the main concern of investors with regards to logistics in China is its poor transport infrastructure (Ta, Choo, & Sum, 2000), a concern that has been addressed by the rapid development and modernization of the country in the past decades. There is therefore tremendous growth potential in the Chinese logistics market, resulting in intense competition that requires 3PLs in China to constantly seek new sources of competitive advantage (Wang, Zantow, Lai, & Wang, 2006).

* Corresponding author. Address: Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Kowloon, Hong Kong. Tel.: +852 69250203; fax: +852 34420189.

E-mail addresses: tigerqin@hust.edu.cn, tigerqin1980@gmail.com (H. Qin), zzzhang@cityu.edu.hk, zhangzizhen@gmail.com (Z. Zhang), zhuxuan.qi900330@aliyun.com (Z. Qi), lim.andrew@cityu.edu.hk (A. Lim).

The transportation process that concerns the 3PL in question is briefly described as follows. First, the suppliers send the finished goods from their manufacturing plants to the warehouse hub operated by the 3PL via domestic truck transportation. Next, the 3PL loads the goods into containers of various sizes and ships these containers to distribution hubs situated in the United States, each of which is operated by an express delivery company such as FedEx or UPS. Finally, the express company unloads the goods from the containers and distributes them to the corresponding stores or retailers by parcel delivery.

Fig. 1 illustrates the transportation network, which can be viewed as a type of hub-and-spoke network. At the warehouse hub, the 3PL provides the manufacturer with value-added services; no extra services beyond the delivery of the goods are required from the express company at the distribution hubs. The goods can be shipped to any of several distribution hubs from the warehouse hub. Therefore, there are several possible transportation routes that a shipment of goods destined for a particular store can take, as illustrated in Fig. 2. This study examines the scenario that is relevant to our client, where there is only a single warehouse hub with multiple distribution hubs. We refer to the warehouse hub in Shanghai as the *origin hub*, and the distribution hubs in the United States as *destination hubs*.

The inventory of the stores is replenished periodically, and each store may demand goods from one or several suppliers. For each period, the manufacturer aggregates the demands of its stores and places orders with suppliers. In most cases, the total volume of demanded goods for a single store in one time period is much less than the capacity of a single container, which is one Twenty-foot Equivalent Unit (TEU); hence, consolidating goods for different stores into one container becomes necessary for reducing transportation cost. In order to facilitate the consolidation process, the manufacturer stipulates that the suppliers must send their goods to the origin hub within a predetermined time window that ranges from one day to several days. Since there is only one origin hub in our problem, the cost of truck transportation is fixed and we can assume that all goods have already arrived at the origin hub. The goods heading to the same store are combined into one shipment at the origin hub, so a single shipment may consist of one or several items. To simplify some store operations such as tracking and receiving goods, it is required that all items in one shipment must be transported along a single route, but they are allowed to be loaded into different containers.

In this study, we investigate the *freight consolidation and containerization problem* (FCCP). It models the task faced by the 3PL, which requires the assignment of shipments to routes as well as items to containers, with the objective of minimizing the total cost

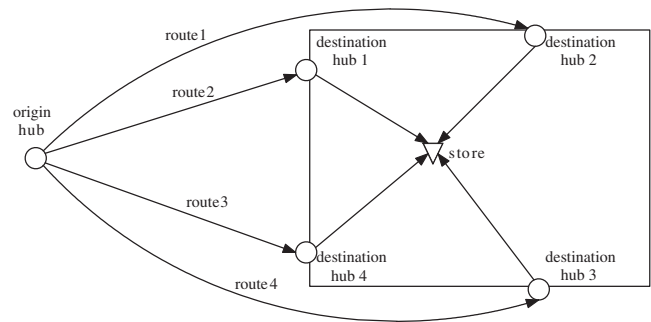


Fig. 2. Multiple routes for a single store.

of two transportation modes, namely container transportation and parcel delivery. We assume that each item transported by the 3PL is an indivisible textile package that must be loaded into only one container. Moreover, the flexibility and non-fragility of textiles allow us to treat the process of loading items into containers as a variable sized bin packing problem (VSBPP). The main difficulty lies in managing the tradeoff between container transportation cost and parcel delivery cost: on one hand, if we send every shipment to the destination hub with the cheapest associated parcel delivery cost, the containers may be underutilized; on the other hand, if we try to minimize container cost by loading each container as fully as possible, then higher parcel delivery costs may be incurred for some shipments.

The FCCP models an actual problem that combines goods consolidation and containerization (i.e., container loading). In contrast to other typical services such as inspection and tracking, a good solution to this problem translates directly into cost savings that can be passed onto the customers, which provides the 3PL with a competitive advantage. In fact, a demonstration of this capability was a decisive factor in the awarding of the contract by the manufacturer to the 3PL in question. Furthermore, consolidation and containerization often exist in tandem in the logistics industry, and the FCCP is likely to be applicable in a variety of practical scenarios. To the best of our knowledge, this problem has not been investigated in the existing literature.

The contributions of this paper are twofold. Firstly, we formally define the problem examined in this article, and also provide an integer programming (IP) model for it. Secondly, the problem is easily shown to be NP-hard. To solve it practically, we focus our efforts on devising an efficient heuristic approach that provides high quality solutions within a reasonable amount of computing time. Consequently, we propose a memetic algorithm (MA) for the FCCP that combines genetic algorithm operations, local search, and a heuristic for the VSBPP. The effectiveness of our algorithm was verified via extensive experiments on a large number of generated test instances.

The remainder of this paper is structured as follows. In Section 2, we give an overview of the relevant research in existing literature. We provide a formal description of the problem in Section 3, along with a mixed integer formulation and a brief discussion of the computational complexity of the problem. Section 4 describes the details of our MA approach. We evaluate our approach using a large set of generated test instances, which are described in Section 5 along with the computational results. Finally, we conclude our article in Section 6 and suggest some possible directions for future research.

2. Literature review

Freight consolidation, which is an important practice in logistics management, has been investigated extensively. The related

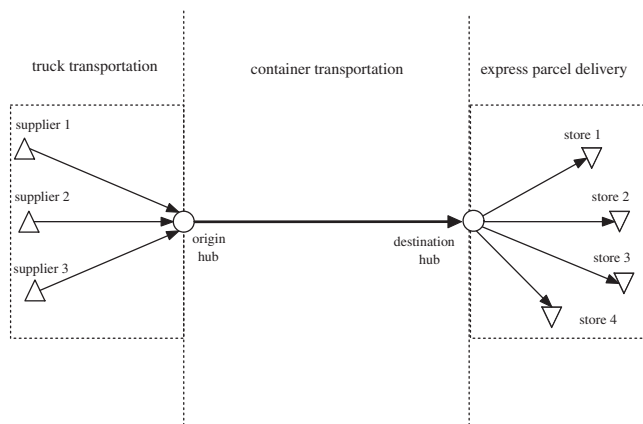


Fig. 1. Overall transportation process.

research can be categorized into two main categories depending on whether the decisions are made at the strategic level or the operational level. The strategic level is concerned with the design of networks, involving decisions on factors such as the number and locations of consolidation centers and the capacity of each center. The aim is to minimize the long-term overall cost, including transportation cost, consolidation center setup cost, inventory cost and labor cost. We refer the reader to Ahuja, Magnanti, and Orlin (1993) for a thorough review of these network design problems. *Hub-and-spoke* is a typical network topology for freight consolidation that has been widely adopted in different transport sectors. Consequently, there is a large body of literature studying the design of hub-and-spoke networks, such as air transportation (Martín & Román, 2004), truck transportation (Cunha & Silva, 2007), railway transportation (Jeong, Lee, & Bookbinder, 2007), sea transportation (Takano & Arai, 2009), and express delivery service (Kara & Tansel, 2001).

The operational level deals with the planning and execution of product distribution on an existing network configuration. Klinkiewicz (1990) presented a model in which each shipment can be sent from source to destination either directly or via a consolidation terminal. At the terminal, the products bound for the same destination are combined into one shipment and incur linear inventory holding cost. To embody the value of consolidation, i.e., economics of scale, the authors assumed that the shipping costs are piecewise linear concave functions of the shipping volume. Heuristics incorporating facility location techniques were developed to solve the model. Liu, Li, and Chan (2003) studied a transportation model allowing two delivery methods: one is direct shipment, where goods are collected from a supplier and directly sent to multiple customers, and the other is hub-and-spoke shipment in which various types of goods are collected from suppliers and consolidated at hubs, and then redistributed to multiple customers. Homogeneous vehicles are used to carry out all goods movements, and the objective is to minimize the total traveling distance of the vehicles. The authors designed a heuristic that utilizes a solution improvement procedure to determine the delivery method for the goods demanded by each customer and to schedule the vehicles.

Song, Hsu, and Cheung (2008) introduced a consolidation problem that aims at coordinating shipments between suppliers and customers through a consolidation center. This problem requires decisions to be made on the inbound and outbound times of the shipments while considering multiple factors including product release and latest arrival times, different consolidation policies, multiple transportation options, and inventory costs. A dual-based heuristic was devised to solve the problem. Leung, Hui, Wang, and Chen (2009) proposed a problem of determining the optimal integrations and consolidations of air cargo shipments. A shipment commonly goes through a number of sequential activities, such as pickup, truck transportation, warehousing and air transportation. Integration and consolidation are defined as assigning consecutive activities of a shipment and the similar activities of different shipments to one agent, respectively. A solution procedure that includes heuristics and a branch-and-bound algorithm was designed for the problem. Other freight consolidation models at the operational level can be found in Popken (1994), Croxton, Gendron, and Magnanti (2003), Dror and Hartman (2007), etc.

This study addresses a freight consolidation problem at the operational level with a hub-and-spoke distribution network. The most significant difference between our research and previous work is the fact that the loading of items into containers of various sizes is explicitly considered; this reflects the use of containers with different standard sizes in industry, such as the standard 20-ft, the standard 40-ft and the 40-ft high cube containers.

It has been shown that since the VSBPP is an extension of the one-dimensional bin packing problem (1DBPP), it is NP-hard (Friesen & Langston, 1986). Three types of solution approaches have been applied to the VSBPP, namely exact algorithms (Alves & Valério de Carvalho, 2007; Correia, Gouveia, & Saldanha-da Gama, 2008; Haouari & Serairi, 2011), approximation algorithms (Epstein & Levin, 2008; Kang & Park, 2003), and heuristics (Haouari & Serairi, 2009). Over the course of our MA, a large number of VSBPPs must be solved. Therefore, an efficient and effective solution approach to the VSBPP is crucial. After comparing the solution qualities and running times of several solution approaches, we came to the conclusion that a heuristic approach is most suitable for handling the VSBPPs in our MA.

3. Problem definition and formulation

We assume that each shipping route can provide a sufficient number of containers of different sizes. Each container is characterized by three attributes: shipping route, price, and capacity. In practice there is a fourth attribute of weight limitation, but since the products in our problem are light textiles that commonly do not exceed the container weight limitations even when being fully loaded, so we do not include this aspect in the model.

The FCCP can be modeled in the following manner. Let S and R be the sets of shipments and shipping routes, respectively. Each shipment $k \in S$ may consist of one or multiple items, and set I comprises all items, each having a size v_i . The shipment containing item $i \in I$ is denoted by $S(i)$. The collection of all containers is denoted by set B , and the attributes of each container $j \in B$ are shipping route $R(j)$, price p_j and capacity V_j . We assume that all containers with the same size and shipping route have the same price. The parcel delivery cost associated with assigning item i to container j is given by $r_{i,j}$. Note that once an item is assigned to a shipping route, its parcel delivery cost is determined since this cost is only related to the locations of the destination hub and its store. In particular, assigning an item to any container in a given shipping route would incur the same parcel delivery cost $r_{i,j}$.

We define three types of binary decision variables in our model. Let $x_{i,j}$ be a binary decision variable that is equal to 1 if item i is loaded into container j , and 0 otherwise; let $z_{k,l}$ be equal to 1 if shipment k is assigned to route l , and 0 otherwise; and let y_j be equal to 1 if container j is used, and 0 otherwise. The following is an IP model for the FCCP:

$$(IP) : \min \sum_{j \in B} p_j y_j + \sum_{i \in I} \sum_{j \in B} r_{i,j} x_{i,j} \quad (1)$$

$$\text{s.t. } \sum_{j \in B} x_{i,j} = 1, \quad \text{for all } i \in I \quad (2)$$

$$x_{i,j} \leq y_j, \quad \text{for all } i \in I \quad \text{and } j \in B \quad (3)$$

$$x_{i,j} \leq z_{k,l}, \quad \text{for all } i \in I : S(i) = k \quad \text{and } j \in B : R(j) = l \quad (4)$$

$$\sum_{l \in R} z_{k,l} = 1, \quad \text{for all } k \in S \quad (5)$$

$$\sum_{i \in I} v_i x_{i,j} \leq V_j, \quad \text{for all } j \in B \quad (6)$$

$$x_{i,j}, y_j \in \{0, 1\}, \quad \text{for all } i \in I \quad \text{and } j \in B$$

$$z_{k,l} \in \{0, 1\}, \quad \text{for all } k \in S \quad \text{and } l \in R$$

The objective (1) is to minimize the total transportation cost. Constraints (2) guarantee that each item is loaded into only one container. The container into which any item is loaded must be marked as “used”, which is realized by Constraints (3). Constraints (4) state that if item i is loaded into container j , the shipment that includes item i must be assigned to the shipping route that provides container j . Constraints (5) ensure that each shipment

must be assigned to only one shipping route. Constraints (6) require that the capacity limitations of the containers are not violated.

Observe that a special case of the FCCP that involves only one shipping route is essentially a generalized cost variable-sized bin packing problem (Epstein & Levin, 2008). From another perspective, if we assume that each container is large enough to accommodate all items in I , then only a single (cheapest) container will be needed on each shipping route. Consequently, by viewing container prices as facility setup costs, the classical uncapacitated facility location problem (Drezner & Hamacher, 2004) can also be reduced to the FCCP. These observations prove the NP-hardness of the FCCP, which implies that it is impossible to find a polynomial-time algorithm to solve the problem unless $P = NP$. In addition, preliminary experiments revealed that although small instances of the FCCP can be solved by the leading commercial IP solver ILOG CPLEX 11.1, the computation time required increases astronomically as the size of the problem increases. Hence, to obtain high quality solutions to the FCCP in a practical time frame, we developed an MA for the problem.

4. Memetic algorithm

Genetic algorithm (GA) is a search meta-heuristic based on the process of natural evolution, such as natural selection and sexual reproduction. Since the pioneering work of Holland (1975), GA has proven to be a highly successful technique for solving a wide variety of combinatorial optimization problems (see Goldberg, 1989; Mitchell, 1998; Sivanandam & Deepa, 2008). A natural modification of the GA approach is to include a local search component in order to find better solutions over the course of the evolutionary process. This technique has been called a memetic algorithm (MA), which has been successfully employed in a variety of applications. The reader is encouraged to refer to Moscato (1999), Lu and Hao (2010), Ngueveu, Prins, and Wolfer Calvo (2010) for the details and applications of MA.

Our MA approach begins by generating an initial chromosome set, each member of which corresponds to a feasible solution to the FCCP. A fitness function is used to evaluate chromosomes. We maintain two chromosome sets of fixed sizes: a *population* set P_I for generating offspring, and an *offspring* set P_O that includes all the newly generated offsprings. In each generation, we choose chromosomes from the population set using an elitist scheme favoring fitter members to be parents, and then use the chosen chromosomes to generate offspring set by reproduction. We call the set $P_I \cup P_O$, which is the combination of the population and offspring sets, the *candidate* set; the same elitist strategy is employed to select chromosomes from the candidate set to construct the population for the next generation. The above process runs iteratively until a termination criterion is satisfied and a near-optimal solution can be retrieved from the fittest member of the population.

In summary, the MA proposed in this paper consists of six components: (1) chromosome representation; (2) generation of the initial population; (3) fitness function; (4) selection process; (5) offspring generation, including crossover, mutation and local refinement of chromosomes; and (6) termination criterion. The

process of our MA approach is shown in Algorithm 1. The remainder of this section presents the detailed description of its components.

Algorithm 1. Process of the memetic algorithm.

```

1:  $P_I \leftarrow$  generate the initial population;
2: Evaluate the chromosomes in  $P_I$ ;
3: while the termination criterion is not satisfied do
4:    $P_O \leftarrow \emptyset$ 
5:   repeat
6:     Randomly select two parent chromosomes from  $P_I$ ;
7:     Produce two offsprings from the parent
      chromosomes using crossover;
8:     Insert offsprings into  $P_O$ ;
9:   until  $P_O$  is full;
10:  for each chromosome in  $P_O$  do
11:    Perform the mutation operation;
12:    Perform the refinement operations;
13:  end for
14:  Evaluate the chromosomes in  $P_O$ ;
15:   $P \leftarrow P_I \cup P_O$ ;
16:   $P_I \leftarrow \emptyset$ ;
17:  Insert the best chromosome of  $P$  into  $P_I$ ;
18:  Select chromosomes using roulette-wheel selection
    from  $P$  to insert into  $P_I$  until  $P_I$  is full;
19: end while
20: Return the best chromosome in  $P_I$ .
```

4.1. Chromosome encoding

In our approach, the information coded into a chromosome simultaneously specifies the assignment of shipments to shipping routes and the loading sequence of items into containers. A gene consists of a 2-tuple and the number of genes in a chromosome is equal to the number of items, i.e., $n = |I|$. Formally, we express a chromosome as a vector $\chi = ((\tau_1, \sigma_1), \dots, (\tau_i, \sigma_i), \dots, (\tau_n, \sigma_n))$, where τ_i and σ_i represent the shipping route and loading order of item i , respectively. The genes in a chromosome are grouped into $|S|$ blocks, each corresponding to a shipment. Since our chromosome encoding does not contain information on how the items are loaded into containers, to transform a chromosome into a feasible solution of the FCCP requires the solutions of several one-dimensional VSBPPs with a pre-determined item loading sequence; the VSBPP requires that a set of items is loaded into a number of containers with different volumes and costs such that the total cost of containers used is minimized.

An example chromosome is shown in Fig. 3. This chromosome involves shipments 1, 2 and 3, which contain item sets {1, 2}, {3, 4, 5, 6} and {7, 8}, respectively. Items 1–6 and 7–8 are assigned to shipping routes 1 and 2, respectively, and the loading sequence of items 1–8 is (3, 8, 2, 4, 7, 6, 5, 1). Note that the item loading sequence is relative; for instance, on shipping route 1 we sequentially load items 3, 1, 4, 6, 5, 2 into containers since their loading orders have the relationship $2 < 3 < 4 < 6 < 7 < 8$. As the items of each shipment must be transported along a unique shipping route, the first element of all 2-tuples in the same block are identical. To obtain a feasible solution from this chromosome, we need to solve two VSBPPs with item sequences (3, 1, 4, 6, 5, 2) and (8, 7).

To highlight the benefits of our chromosome encoding scheme, consider the alternative scheme where the number of genes is equal to the number of shipments and each gene specifies a shipping route. Using the previous example, this encoding scheme is

Item	Shipment 1		Shipment 2				Shipment 3	
	1	2	3	4	5	6	7	8
Chromosome	(1,3)	(1,8)	(1,2)	(1,4)	(1,7)	(1,6)	(2,5)	(2,1)

Fig. 3. An example of chromosome encoding scheme.

illustrated in Fig. 4, where shipments 1 and 2 are assigned to shipping route 1, and shipment 3 is assigned to shipping route 2. This encoding scheme does not explicitly specify the item loading sequence. Therefore, to calculate the cost of the containers associated with a chromosome, we need to solve at most $|R|$ VSBPPs without fixed item loading sequences. For example, for the chromosome shown in Fig. 4, we must solve two VSBPPs, one for shipping route 1 with item set $\{1, 2, 3, 4, 5, 6\}$, the other for shipping route 2 with item set $\{7, 8\}$.

Over the course of the MA, a large number of chromosomes are generated. Consequently, since we require high quality solutions to the corresponding VSBPPs within a reasonable amount of computation time, we can only employ very simple heuristics for this purpose, e.g., adaptations of the first-fit, best-fit, first-fit decreasing, or best-fit decreasing heuristics for the 1DBPP (Simchi-Levi, 1994), and such heuristics only generate one packing pattern for a given input item set. In contrast, the first encoding scheme allows the MA to produce multiple packing patterns (Haouari & Serairi, 2009), which greatly increases the potential of identifying high quality solutions.

4.2. Initial population

The first chromosome in the initial population corresponds to the greedy assignment of all items in a shipment such that the shipping route τ_i for item i is the route with the smallest parcel delivery cost $r_{i,j}$; this greedy heuristic approximates the approach employed by the decision-makers at the 3PL prior to this research. For the remaining chromosomes, the values of τ_i are randomly selected from the set S . The item loading sequences $(\sigma_1, \dots, \sigma_n)$ for all chromosomes are randomly generated permutations.

4.3. Fitness function

The MA ranks chromosomes based on a value that is calculated using a fitness function. Given a chromosome, let the vector $\xi_l = (\xi_l(1), \dots, \xi_l(n_l))$ denote the loading sequence of the n_l items assigned to shipping route l . For the chromosome shown in Fig. 3, we have $\xi_1 = (3, 1, 4, 6, 5, 2)$ and $\xi_2 = (8, 7)$.

Each shipping route $l \in R$ in a chromosome χ is given a fitness value $\Theta_l(\chi)$, computed as: $\Theta_l(\chi) = D_l(\chi) + C_l(\chi) + \alpha(1 - F_l(\chi))$, where $D_l(\chi)$ is the parcel delivery cost, $C_l(\chi)$ is the container cost, $F_l(\chi)$ is a measure of the utilization rate of the containers, and α is a user-defined parameter. When no items are assigned to shipping route l , $\Theta_l(\chi) = 0$. The overall fitness of a chromosome χ is the sum of the fitness values of its component shipping routes, i.e., $\Theta(\chi) = \sum_{l \in R} \Theta_l(\chi)$.

Recall from Section 3 that once an item is assigned to a particular shipping route, its parcel delivery cost is determined. Therefore, $D_l(\chi)$ can be directly computed from the information in ξ_l .

To compute $C_l(\chi)$, we need to solve a VSBPP with a predetermined item loading sequence (we refer to this problem as VSBPPS). The VSBPPS assumes the following procedure: starting from the first item, select a container and sequentially load the items into it according to the loading sequence. If the current container cannot accommodate the current item, close it and select another container for loading. When all items are loaded into containers, we have found a feasible solution to the VSBPPS. The task is to find

the best selection of containers so as to minimize the total container cost.

Haouari and Serairi (2009) introduced a technique to optimally solve the VSBPPS. The authors defined an acyclic digraph $G = (V, A)$, where V consists of a node for each item and an additional dummy node, and A is constructed based on the item sizes, item loading sequence and bin sizes. The optimal solution for the VSBPPS can be obtained by identifying a shortest path in $G = (V, A)$, which is achieved using an algorithm with time complexity $O(|A|)$. In a worst case, $O(|A|)$ can be equal to $O(|V|^2)$. For the practical problem instances faced by the 3PL, the cardinality of A is generally very close to $|V|^2$ when solving most of VSBPPSs using this technique, i.e., the time complexity approaches $O(n_l^2)$ for a given chromosome and shipping route l . Since the MA will generate a large number of chromosomes and the evaluation of each chromosome requires the solution of up to $|R|$ VSBPPSs, this technique is too slow for our purposes.

Consequently, we devised the following dynamic programming approach to solve the VSBPPS to near optimality. Let the price and capacity of a type q container be c_q and V_q , respectively. We define the vector $\Pi_m = (1, \dots, m)$ to be the collection of m container types for a certain shipping lane, sorted in ascending order of capacity (i.e., $1 \leq q_1 < q_2 \leq m$ implies $V_{q_1} \leq V_{q_2}$). Instead of solving the VSBPPS, our approach solves a modified version of the problem with the additional restriction that a newly selected container must be no larger than any previously selected container; we call this problem the *restricted VSBPPS*. This approach is motivated by the observation that larger containers almost always have a lower cost per unit capacity in practice, i.e., $c_{q_2}/V_{q_2} \leq c_{q_1}/V_{q_1}$ for $1 \leq q_1 < q_2 \leq m$. Therefore, selecting larger containers first is a good greedy approach for minimizing total container cost.

For a given loading sequence ξ_l and a set of available containers Π_m , our approach solves this restricted VSBPPS optimally. Preliminary experiments revealed that the optimal solution to the restricted VSBPPS is usually very close to that of the corresponding VSBPPS, and in fact they are equal in certain instances.

We next describe our dynamic programming algorithm for the restricted VSBPPS. Let $\xi(s, t) = (\xi(s), \xi(s+1), \dots, \xi(t-1), \xi(t))$, $t \geq s$ be the loading sequence from item $\xi(s)$ through $\xi(t)$, which involves $t - s + 1$ items. In addition, let function $f_q(\xi(s, t))$ be the minimum number of type q containers required for packing $\xi(s, t)$ if only type q containers are employed; note that the value of $f_q(\xi(s, t))$ can be computed in $O(m)$ time by following the packing procedure. We denote the minimum cost of the restricted VSBPPS on loading sequence $\xi(s, t)$ and containers Π_m by $C(\xi(s, t), \Pi_m)$, which can be defined recursively as follows:

$$C(\xi(s, t), \Pi_m) = \min \left\{ \min_{s \leq i < t} \{C(\xi(i+1, t), \Pi_m) + c_m f_m(\xi(s, i))\}, c_m f_m(\xi(s, t)), C(\xi(s, t), \Pi_{m-1}) \right\} \quad (7)$$

where $C(\emptyset, \Pi_m) = 0$, $C(\xi(s, t), \emptyset) = +\infty$ and $C(\xi(i_1, t), \Pi_m) \geq C(\xi(i_2, t), \Pi_m) > 0$, if $s \leq i_1 < i_2 \leq t$.

This recursion divides the choice of container made during the packing procedure of the VSBPPS into three cases: (1) the next several items in the sequence are loaded into a number of the largest legal containers, (2) all remaining items are loaded into the containers with the largest size, or (3) the largest legal container is not selected.

For a given loading sequence $\xi(s, t)$ and bin type q , let $i_q(s)$ be the largest index i , $s \leq i \leq t$ of an item such that all items from s to i in the loading sequence can be contained in a single container of type q , i.e., $i_q(s) = \max\{i: f_q(\xi(s, i)) = 1, s \leq i \leq t\}$.

The following theorem allows us to compute the values of $C(\xi(s, t), \Pi_m)$ efficiently using dynamic programming:

Shipment	1	2	3
Chromosome	1	1	2

Fig. 4. An alternative chromosome encoding scheme.

Theorem 1. Let $\bar{i} = i_m(s) = \max\{i : f_m(\xi(s, i)) = 1, s \leq i \leq t\}$. Then,

$$C(\xi(s, t), \Pi_m) = \begin{cases} \min\{C(\xi(\bar{i} + 1, t), \Pi_m) + c_m, C(\xi(s, t), \Pi_{m-1})\}, & \text{if } \bar{i} + 1 \leq t \\ c_m, & \text{if } \bar{i} = t \end{cases} \quad (8)$$

Proof. See Appendix A. \square

Before executing the dynamic programming algorithm, we perform a preprocessing phase to calculate $i_q(s)$ for all combinations of $\xi(s, t)$ and bin type q . In this phase, a sliding window algorithm is invoked for each bin type (see Algorithm 2). For each shipping route l in the restricted VSBPPS, Algorithm 2 runs in $O(n_l)$ time. Since the algorithm is performed once for each of the m bin types, the overall time complexity of the preprocessing phase is $O(mn_l)$. The values of $i_q(s)$ obtained from the preprocessing phase can then be employed by the dynamic programming algorithm to run in $O(mn_l)$ time. Hence, the restricted VSBPPS can be solved optimally in $O(mn_l)$ time in this manner.

Now, we have two approaches to compute $C_l(\chi)$: one is to solve the VSBPPS with time complexity $O(n_l^2)$ and the other is to solve the restricted VSBPPS with time complexity $O(mn_l)$.

Algorithm 2. Preprocessing for Type q bin.

```

1: INPUT: size of type  $q$  bin  $V_q$ , and an item sequence  $\xi(s, t)$ ;
2: OUTPUT:  $i_q(s), i_q(s + 1), \dots, i_q(t - 1), i_q(t)$ ;
3: Let  $v[\xi(i)]$  be the size of item  $\xi(i)$ ;
4: Initialize  $head = tail = s$  and  $volume = 0$ ;
5: while  $tail \leq t$  do
6:   while  $head \leq t$  and  $volume + v[head] \leq V_q$  do
7:      $volume = volume + v[head]$ ;
8:      $head = head + 1$ ;
9:   end while
10:   $volume = volume - v[tail]$ ;
11:   $i_q(tail) = head - 1$ ;
12:   $tail = tail + 1$ ;
13: end while

```

In practice, each container can accommodate many items because containers are usually much larger than the individual items to be loaded. For example, a 20-ft container can contain about 30 items with average size of 1 cubic meter. Good solutions to the FCCP tend to have a high proportion of containers with high utilization due to the high costs of containers. Thus, the MA may encounter several VSBPPSs where the number of items n_l allocated to the shipping route l is large. However, the number of container types m is comparatively much smaller. Therefore, our $O(mn_l)$ approach is much faster than the $O(n_l^2)$ approach by Haouari and Serairi (2009). This allows us to increase the sizes of the population and candidate sets as well as the number of generations in the MA, which enables the algorithm to explore more item sequences and increases the probability of identifying better solutions.

The process of computing $C_l(\chi)$ also generates a packing pattern for item sequence ξ_l , which is a set of containers $B_l(\chi)$ along with the items in each container. With this information, we can compute the utilization rate measure $F_l(\chi)$ as: $F_l(\chi) = \sum_{j \in B_l(\chi)} V_j(a_j/V_j)^2 / \sum_{j \in B_l(\chi)} V_j$, where a_j is the total size of the items in container j . Given a set of containers and a set of items loaded into these containers, the measure $F_l(\chi)$ favors a loading configuration with several well-filled containers over one with mostly equally-filled containers. Several papers on the 1DBPP (e.g., Falkenauer, 1996; Osogami & Okano, 2003; Loh, Golden, & Wasil, 2008) have applied $F_1(\chi)$ as the fitness function. Note that

since $F_l(\chi) \leq 1$ and the objective of the FCCP is to minimize cost, we include this utilization rate measure as a part of the fitness value $\Theta_l(\chi)$ using the term $\alpha(1 - F_l(\chi))$, where α is a user-defined coefficient that determines its relative importance.

4.4. Selection

In each generation, the MA selects parents from the population set for reproduction (line 6 in Algorithm 1), and also selects members from the candidate set to form the population set for the next generation (line 18 in Algorithm 1). We use the same elitist roulette-wheel (or fitness proportionate) selection criterion for both purposes. For a chromosome $\chi \in X$, its probability of being selected is given by: $prob(\chi) = \frac{1/\Theta(\chi)}{\sum_{\chi \in X} 1/\Theta(\chi)}$.

The above selection criterion defines an inverse relationship between the fitness value of a chromosome and its selection. When selecting parents for reproduction, it is possible for the same chromosome to be selected multiple times. However, each chromosome from the candidate set can only be selected once to be a member of the population set for the next generation. Note that the best chromosome in each generation is always retained for the next generation (line 17 in Algorithm 1). Other applications that have used this type of selection method include Back, Fogel, and Michalewicz (1999), Moon Kratica, Stanimirovic, Tosic, and Filipovic (2002), Kratica et al. (2007).

4.5. Crossover

After two parent chromosomes are selected, the MA produces two offsprings by performing a crossover operation. With our encoding scheme, the offsprings obtained by applying classical crossover operators would be infeasible in most cases. To deal with this issue, we devised an adapted two-point partially mapped crossover (APMX) operator for the MA. The partially mapped crossover (PMX) operator is a common operator in the approaches for the traveling salesman problem. It allows information contained in subsequences in the parent chromosomes to be transferred to offsprings. Our APMX operator is a combination of the PMX and classical crossover operators.

Our APMX is performed in three steps. (1) Two block boundaries are randomly selected from the parent chromosomes as crossover points, thereby dividing each chromosome into three segments (see Fig. 5(a)). (2) The genes in the middle segments (between the crossover points) are exchanged to generate two offsprings (Fig. 5(b)), which may be infeasible because two different items may have the same loading order. The middle segments also define mappings between the corresponding loading orders; in our example, the mappings are $2 \leftrightarrow 3$, $3 \leftrightarrow 4$, $7 \leftrightarrow 5$, $6 \leftrightarrow 6$. Note that due to transitivity, the first two mappings effectively become $2 \leftrightarrow 4$ (Fig. 5(c)). (3) The loading orders in the first and third segments that lead to conflicts are replaced according to the mapping (see Fig. 5(d)). In this case, we find that the second gene has a duplicate order number 2, so it is replaced by 4 using the mapping $2 \leftrightarrow 4$. The remaining conflicting genes are replaced in a similar manner for both chromosomes.

The effect of the APMX operator on the affected genes is to perform a classical block-based crossover on the shipping routes, and the PMX operation on the loading orders. This preserves the loading sequence information contained in the middle segments of the parent chromosomes.

4.6. Mutation

Mutation operations allow the exploration of wider regions of the solution space, generally by introducing random variations into

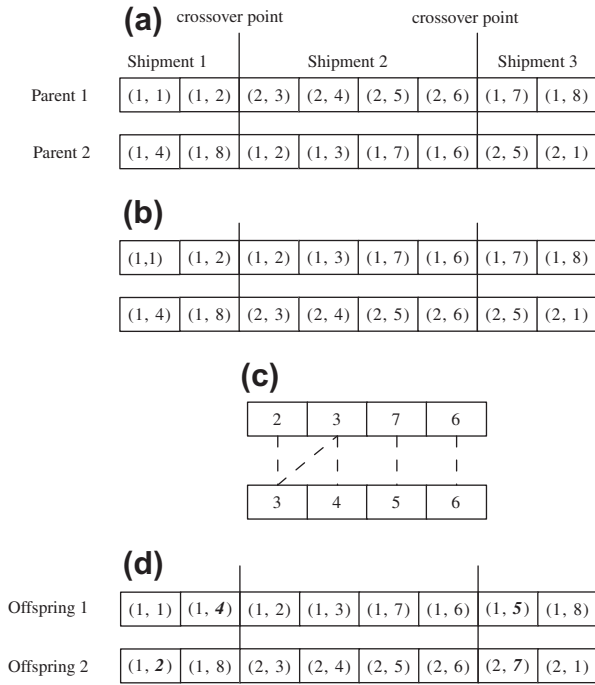


Fig. 5. (a) Randomly choose two crossover points. (b) Swap genes. (c) The gene mappings. (d) Replace loading orders based on mapping.

the chromosomes. In our MA, we perform mutation on the set of offsprings produced by the crossover operations. Two types of mutations are implemented (see Fig. 6). For each shipment in the chromosome, a Type 1 mutation occurs with probability ρ_1 , which modifies the shipping route for the shipment to another value that is uniformly and randomly chosen from R . For each gene in the chromosome, a Type 2 mutation occurs with probability ρ_2 ; another gene is uniformly and randomly selected, and the loading orders of the two genes are exchanged. If an offspring has been selected for mutation, we conduct the Types 1 and 2 mutations in order.

4.7. Local refinement

After the crossover and mutation operators, we attempt to locally improve the resultant offsprings (in terms of the fitness function) using three types of refinement operations:

1. **Consolidation.** Choose λ shipments at random and assign them to a single shipping route that results in the biggest improvement in the fitness value. If there does not exist a shipping route whereby the assignment of the shipments improves the fitness, then do nothing. In our implementation of the MA, we perform this operation on $\lambda = \{2, 3, 4\}$. Fig. 7 shows an example consolidation with $\lambda = 2$.
2. **Route swap.** Choose two shipments at random and swap their shipping routes if the resultant solution has a superior fitness value.
3. **Loading-order swap.** Choose two items at random and swap their loading orders if the resultant solution has a superior fitness value.

These operations are applied in the following order: consolidation with $\lambda = 2, 3, 4$, route swap, and finally loading-order swap. An operation is repeatedly applied until no improvement is found after β consecutive tries, whereupon the next operation is applied.

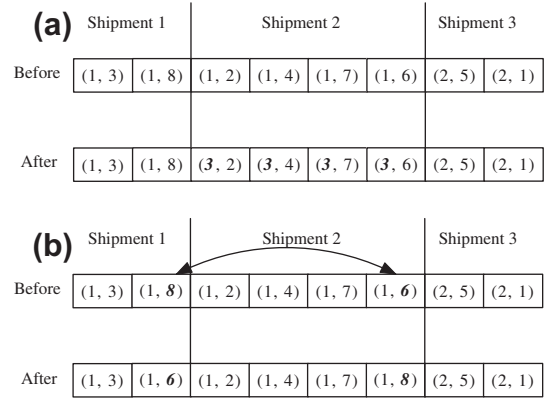


Fig. 6. (a) Type 1 mutation modifies the shipping route of shipment 2 from 1 to 3. (b) Type 2 mutation exchanges loading orders of items 2 and 6.

4.8. Termination criterion

The MA terminates when the number of generations reaches a user-defined value N_{maxGen} .

5. Computational experiments

In this section, we evaluate the merits of our proposed MA on a large number of randomly generated test instances. To provide a benchmark for the performance of our approach, we also applied the branch-and-cut search scheme in ILOG CPLEX 11.1 with default settings to solve the IP model. Both algorithms were implemented in C++ and executed on an Intel Xeon (R) 2.66 GHz server with 3 GB RAM. All computation times reported here are in CPU seconds on this server.

5.1. Test instances

The FCCP is a new problem, and therefore there are no specific benchmark test sets available in existing literature. Consequently, we generated a number of test instances using the following scheme, which was developed after discussions with experts from the 3PL in question. We take the number of shipments ($|S|$) from $\{20, 50, 80\}$ and the number of shipping routes ($|R|$) from $\{5, 10\}$. For each shipment, the number of items is selected from $U_d[3, 12]$, where $U_d[a, b]$ denotes the discrete uniform distribution in the interval $[a, b]$. The item sizes (v_i) in cubic meters are independently randomly generated using two distributions $U_c[0.3, 1]$ and $U_c[2, 6]$, where $U_c[a, b]$ denotes the continuous uniform distribution in the interval $[a, b]$. For convenience, we refer to the items with sizes generated by $U_c[0.3, 1]$ and $U_c[2, 6]$ as *small* and *big* items, respectively.

We represent the territory of the United States with a planar rectangle with Cartesian coordinates $[0, 1000] \times [0, 1000]$, while the origin hub in Shanghai can be conceptually visualized as a point outside this rectangle (see the map shown in Fig. 8). There

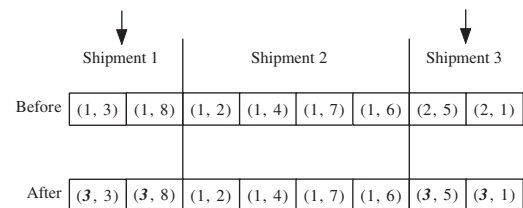


Fig. 7. An example of consolidation with $\lambda = 2$.

are 30 retailer stores randomly distributed in the planar rectangle. The destination store for each shipment is randomly selected so that some stores may receive more than one shipment while others may receive none.

All test instances make use of three types of containers: standard 20-ft, standard 40-ft and 40-ft high cube, which have capacities of 29.4, 58.8 and 67.2 cubic meters, respectively. Each shipping route corresponds to a unique destination hub whose location is randomly chosen from the boundary of the planar rectangle (resulting in $|R|$ destination hubs in total).

We illustrate our method of assigning container costs for each container type on each shipping route using Fig. 8. On the boundary of the planar rectangle, we define six points A_1, \dots, A_6 , each with the fixed value N_{A_μ} ($1 \leq \mu \leq 6$) as shown in the figure: $N_{A_1} = N_{A_5} = 2400$, $N_{A_2} = N_{A_4} = 3000$, $N_{A_3} = 3400$, and $N_{A_6} = 2000$. These values were chosen based on the shipping distance between the warehouse in Shanghai and points on the coast of North America.

Consider a destination hub on the boundary of the planar rectangle at point A_h ; observe that A_h must lie on some line segment (A_μ, A_ν) , where $\nu = (\mu + 1) \bmod 6$. Let $L(A_\mu, A_\nu)$ be the Euclidean distance between points A_μ and A_ν . We set the cost of a standard 20-ft container for the shipping route with destination hub A_h to $N_{A_\nu} + (N_{A_\mu} - N_{A_\nu}) \times L(A_\nu, A_h) / L(A_\mu, A_\nu)$. In our example, this cost would be $2000 + (2400 - 2000) \times L(A_6, A_h) / L(A_6, A_1)$. We found that the relative shipping costs for each route generated in this manner closely approximate the actual data we received from the 3PL. For each shipping route, the costs of the standard 40-ft and 40-ft high cube containers are set to be 1.2 and 1.3 times that of the standard 20-ft container.

The cost $r_{i,j}$ of delivering item i from a destination hub to its retailer store is a function of the item size and delivery distance. For example, if item i is loaded into container j heading to destination hub A_h and its store is situated at point A_s , we set $r_{i,j} = \gamma \times v_i \times L(A_h, A_s)$, where γ is a multiplicative factor for the delivery cost. A higher value of γ increases the relative cost of delivery compared to the container cost. In our test instances, the value of γ is taken from $\{0.08, 0.16, 0.32\}$.

Hence, our test instances are divided into 36 groups, corresponding to all combinations of the number of shipments (3 possible values), the number of shipping routes (2 possible values), the item sizes (2 possible distributions) and the values of γ (3 possibilities). Each instance group is identified by four numbers separated by dashes ('-'): the first two numbers are the number of shipments and shipping routes, respectively; the third number is the value of γ ; and the last number indicates item size distribution, where "1" stands for $U_c[0.3, 1]$ and "2" stands for $U_c[1, 3]$. For each instance group, we generated 10 random instances, for a total of 360 instances. All the instances and results can be found on <http://www.computational-logistics.org/orlib/fccp>.

5.2. Experimental setup

We set a time limit of 1 CPU hour for all CPLEX executions when solving the IP model. Recall from Section 4.3 that computing our

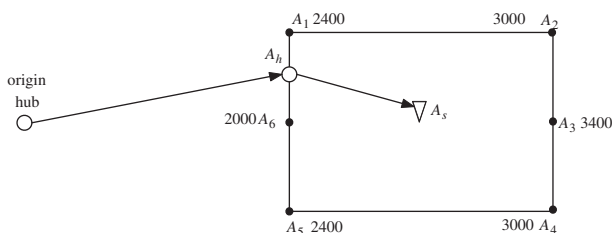


Fig. 8. Example FCCP map.

fitness function requires a solution to the VSBPPS. We implemented the MA using our near-optimal $O(mn_l)$ VSBPPS solution (denoted simply as MA1) as well as the optimal $O(n_l^2)$ VSBPPS solution by Haouari and Serairi (2009) (denoted by MA2). Since the MA is not deterministic, we executed each of the two MA implementations ten times for each instance with different random seeds and recorded the best, average and worst solutions of the ten runs.

When solving the IP model using CPLEX, an increase in the cardinality of set B (the collection of all containers) corresponds to an increase in the number of decision variables, which has a significant effect on the performance of CPLEX. Therefore, it is important to minimize the size of set B while ensuring that no optimal solution is omitted.

Consider a special instance involving only one shipping route. Suppose all items can be completely loaded into twenty standard 20-ft or ten standard 40-ft or nine 40-ft high cube containers by some approach; this implies that the optimal solution to the IP model with a set B including these thirty-nine containers is an optimal solution to the FCCP instance. This observation allows us to determine the set B for each test instance in the following manner. For all items with different sizes and for each shipping route, we solve two 1DBPPs with the first-fit decreasing heuristic (Coffman, Garey, & Johnson, 1996), one using only standard 40-ft containers, and the other using only 40-ft high cube containers. The solutions to these problems are upper bounds for the numbers of these two types of containers, which we include in set B . Note that the capacity and cost ratios of the standard 40-ft container to the standard 20-ft container are 2 and 1.3, respectively, so no optimal solution can employ two standard 20-ft containers because they can be combined into one standard 40-ft container with reduced cost. Consequently, we include only one standard 20-ft container in set B .

Our MA approach contains a number of parameters as given in Table 1. These parameters can be divided into two classes: one class consists of $N_{\max\text{Gen}}$, $|P_l|$, $|P_o|$ and β , which mainly affect the computation speed of the algorithm, and the other class contains the remaining parameters that mainly affect the quality of final solutions. Our parameter tuning process is conducted in two phases. In phase one, we set all parameters of the second class to zero and calibrated the parameters in the first class by conducting a 3^3 full factorial experimental design on the fifth instance in each of the instance groups with 50 shipments using the MA1 implementation ($|P_l| = |P_o|$ for all configurations). In this phase, we solved each instance 27 times, recorded its lowest cost and calculated a gap for each run by: $(\text{cost of the run} - \text{the lowest cost}) / \text{the lowest cost}$. After considering the tradeoff between the average gap and average computation time, we selected the following parameter values: $N_{\max\text{Gen}} = 100$, $|P_l| = |P_o| = 20$ and $\beta = 40$. In phase two, we conducted another 3^3 full factorial design to determine the values of $\alpha = 0.01$, $\rho_1 = 0.01$ and $\rho_2 = 0.01$.

5.3. Results and analysis

Our results are summarized in two tables: Table 2 summarizes the computational results for the instance groups with small items,

Table 1
Parameters for MA approach.

Symbol	Description
$N_{\max\text{Gen}}$	The number of generations of MA
$ P_l = P_o $	The sizes of the population and offspring sets
β	The maximum number of consecutive non-improving tries
α	The value of α in the chromosome fitness function
ρ_1	The probability of Type 1 mutation
ρ_2	The probability of Type 2 mutation

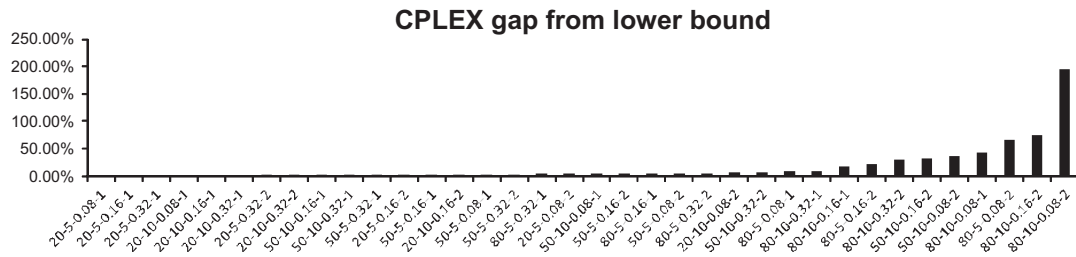


Fig. 9. The difficulty of instance groups.

Table 2

Summarized results for instance groups with small items.

Group	CPLEX		Greedy	MA1			Time	MA2			
	Gap (%)	Time		Best gap (%)	Worse gap (%)	Avg. gap (%)		Best gap (%)	Worse gap (%)	Avg. gap (%)	Time
20-5-0.08-1	0.00	226.0	30.99	0.00	0.05	0.01	14.5	0.00	0.05	0.01	89.2
20-5-0.16-1	0.00	37.1	46.01	0.00	0.00	0.00	13.9	0.00	0.00	0.00	84.2
20-5-0.32-1	0.00	2.0	52.63	0.00	0.00	0.00	15.1	0.00	0.00	0.00	75.3
20-10-0.08-1	0.00	990.3	58.36	0.09	0.15	0.12	16.7	0.09	0.15	0.12	92.3
20-10-0.16-1	0.00	56.7	51.42	0.00	0.44	0.16	17.4	0.00	0.44	0.16	83.3
20-10-0.32-1	0.00	7.0	59.19	0.00	0.46	0.26	18.2	0.00	0.46	0.26	79.9
50-5-0.08-1	2.71	3600.0	52.28	0.17	1.11	0.55	90.1	0.25	0.93	0.56	727.3
50-5-0.16-1	1.15	1554.6	57.81	−0.06	1.37	0.50	94.8	−0.07	1.48	0.59	631.8
50-5-0.32-1	0.89	2219.4	61.79	0.38	0.83	0.63	97.2	0.32	0.86	0.68	671.1
50-10-0.08-1	3.90	3600.0	67.07	−0.58	1.40	0.22	108.0	−0.64	1.07	0.28	760.0
50-10-0.16-1	0.49	2979.3	78.94	0.47	4.67	2.39	108.6	0.87	7.11	3.61	650.5
50-10-0.32-1	0.73	1973.4	78.46	0.59	2.62	1.44	112.5	0.33	2.31	1.06	630.4
80-5-0.08-1	7.87	3600.0	44.86	−4.00	−2.71	−3.42	250.0	−4.19	−2.36	−3.52	1952.7
80-5-0.16-1	4.32	3600.0	42.50	−1.79	−0.61	−1.29	244.8	−1.80	−0.95	−1.37	1878.4
80-5-0.32-1	3.73	3600.0	62.46	−1.62	−0.27	−1.06	248.7	−1.71	−0.43	−1.04	1881.2
80-10-0.08-1	42.08	3600.0	46.91	−14.60	−10.20	−12.23	284.0	−13.26	−10.64	−12.08	1999.1
80-10-0.16-1	17.67	3600.0	59.77	−7.44	−2.54	−5.26	294.4	−7.19	−1.37	−4.97	1930.7
80-10-0.32-1	8.16	3600.0	75.39	−2.83	−0.72	−1.81	300.4	−2.86	−0.79	−1.94	1809.1
Average	5.20	2158.1	57.05	−1.74	−0.22	−1.04	129.4	−1.66	−0.09	−0.98	890.4

Table 3

Summarized results for instance groups with big items.

Group	CPLEX		Greedy	MA1			Time	MA2			
	Gap (%)	Time		Best gap (%)	Worse gap (%)	Avg. gap (%)		Best gap (%)	Worse gap (%)	Avg. gap (%)	Time
20-5-0.08-2	3.88	3342.1	36.80	−0.47	−0.06	−0.29	19.4	−0.48	0.00	−0.33	66.7
20-5-0.16-2	1.05	2607.2	52.69	−0.07	0.07	−0.04	21.3	−0.08	0.17	0.01	64.1
20-5-0.32-2	0.44	1705.4	48.33	−0.05	−0.01	−0.03	20.7	−0.05	0.00	−0.01	61.0
20-10-0.08-2	6.52	3600.0	50.00	−0.61	1.92	0.53	21.0	−0.88	1.84	0.24	64.2
20-10-0.16-2	1.99	3600.0	49.10	0.37	1.47	0.98	23.4	0.27	2.69	1.00	64.8
20-10-0.32-2	0.47	1324.4	75.27	0.11	0.69	0.33	23.6	0.00	0.35	0.16	59.3
50-5-0.08-2	5.57	3600.0	34.35	−1.93	1.58	−0.28	142.2	−2.10	0.76	−0.71	416.0
50-5-0.16-2	4.02	3600.0	42.75	−1.71	−1.01	−1.40	135.9	−1.92	−1.32	−1.69	397.0
50-5-0.32-2	2.80	3600.0	64.59	−0.93	−0.77	−0.86	135.6	−0.94	−0.88	−0.91	397.4
50-10-0.08-2	36.95	3600.0	27.49	−8.32	−4.70	−6.70	155.6	−8.74	−5.25	−7.16	432.1
50-10-0.16-2	31.93	3600.0	52.23	−10.21	−7.09	−8.95	147.4	−9.98	−7.64	−8.71	395.9
50-10-0.32-2	7.12	3600.0	87.97	−4.04	−2.83	−3.41	146.2	−4.03	−3.00	−3.61	371.8
80-5-0.08-2	65.63	3600.0	8.60	−19.78	−19.00	−19.38	401.3	−19.85	−19.11	−19.48	1083.1
80-5-0.16-2	22.17	3600.0	20.50	−11.85	−11.20	−11.60	385.4	−11.76	−11.23	−11.52	1045.0
80-5-0.32-2	5.61	3600.0	51.89	−4.17	−3.86	−4.02	387.4	−4.09	−3.86	−3.99	1067.4
80-10-0.08-2	194.89	3600.0	−8.10	−33.40	−30.55	−32.05	465.7	−33.48	−30.57	−32.13	1133.6
80-10-0.16-2	75.15	3600.0	16.77	−28.82	−27.39	−28.17	408.4	−28.78	−27.67	−28.20	1049.6
80-10-0.32-2	29.78	3600.0	63.76	−13.93	−12.93	−13.42	404.2	−13.97	−12.55	−13.45	1026.4
Average	27.55	3298.8	43.06	−7.77	−6.43	−7.15	191.4	−7.82	−6.51	−7.25	510.9

while Table 3 provides the results for big items. All columns except the first one in each table give the average values of corresponding gaps or computation times of all instances of the same group. We calculated the gap between the solution and the lower bound found by CPLEX after 1 h of computation for each instance as: $(CPLEX - Lower\ bound) / Lower\ bound$ and present the average gap of all instances in each group in the column “Gap (%)” under

the block “CPLEX”. The other gaps were derived using the solutions of CPLEX as baseline, e.g., the gap for the average solution of 10 MA1 runs for some instance is calculated by: $(MA1\ average - CPLEX) / CPLEX$, where MA1 average is the average cost of executing MA1 10 times. The column “Gap (%)” under the block “Greedy” gives the average cost of the initial solutions found using the greedy heuristic described in Section 4.2, which provides

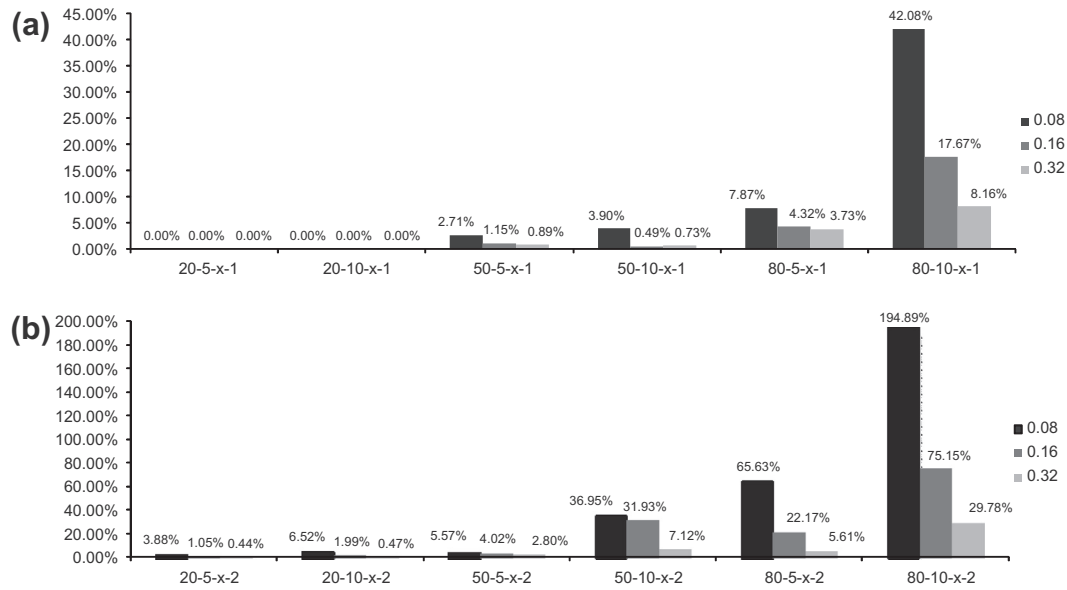
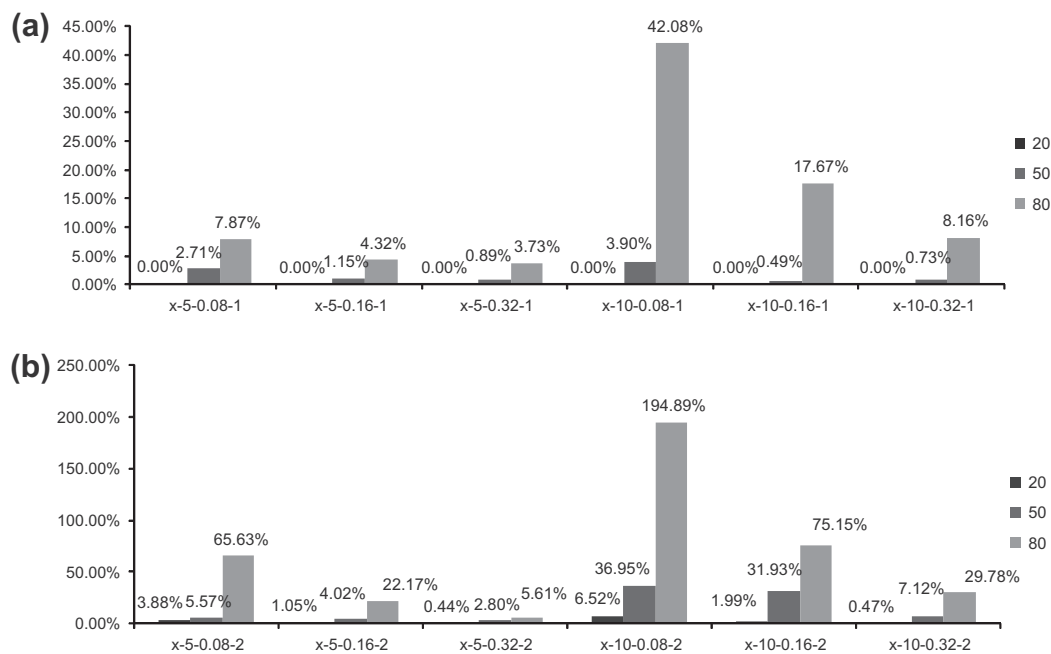
Fig. 10. The impact of γ .

Fig. 11. The impact of the number of shipments.

an indication of the quality of the human-generated solutions by the decision makers at the 3PL. This is followed by the results obtained by the two MA implementations. It is reasonable to assume that the values in the column “Gap (%)” under the block “CPLEX” reflect the difficulty of instance groups. Consequently, we sorted all instance groups by increasing value of “Gap (%)” of CPLEX and plotted the results in Fig. 9. The detailed results for all instances can be found in the Appendix B.

From the gaps of CPLEX in both Tables 2 and 3, we find that the instances with big items are more difficult to be solved by CPLEX than their counterparts with small items. We can also see that for the groups with $|S| = 20$ and small items, CPLEX is able to find the optimal solutions within the time limit. The instances with $|R| = 10$ shipping routes require more computation times to be solved and have larger CPLEX gaps (and are therefore more

difficult) than their counterparts with $|R| = 5$. Fig. 10 plots the CPLEX gaps for each instance group and is derived from these two tables; we observe an inverse relationship between the difficulty of the instances and the value of γ (the only exceptions are “50-10-0.16-1” and “50-10-0.32-1”). This implies that the problem becomes more difficult as the cost of delivery decreases relative to the container cost. Furthermore, the difficulty of instances increases with the number of shipments, which is illustrated by Fig. 11.

The greedy heuristic performed better than CPLEX for only group “80-10-0.08-2”. For the rest of the groups, it produced much poorer solutions than CPLEX, which is demonstrated by the large positive values under the block “Greedy”. We can clearly see that our MA implementations are far superior to the greedy heuristic employed by human decision makers, producing solutions that

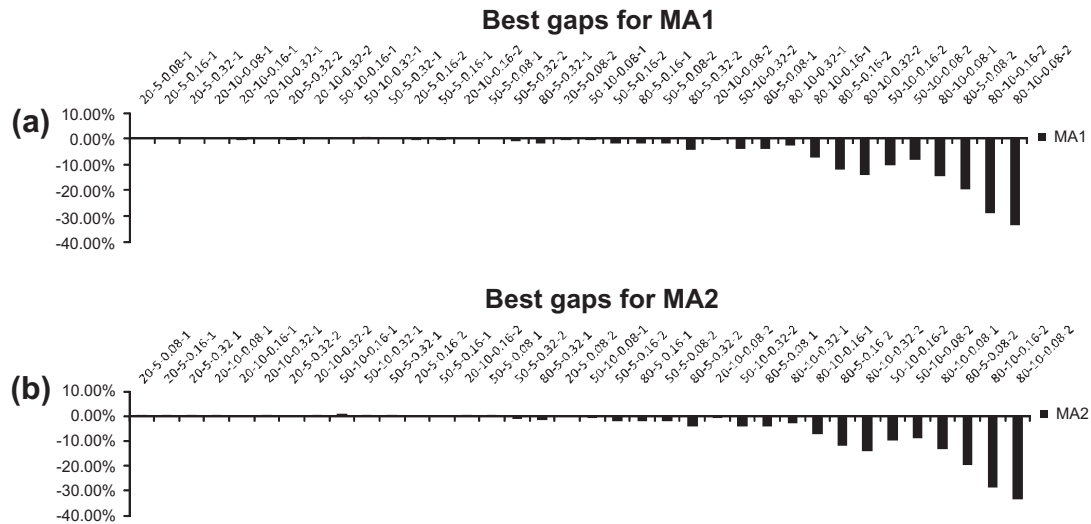


Fig. 12. The best gaps of two MA implementations plotted with the same group order in Fig. 9.

cost less than half as much in certain cases. This represents a substantial cost saving, which provides dramatic evidence of the importance of having an effective solution to the FCCP. A demonstration of this fact was one of the decisive factors when the manufacturer eventually awarded the contract to the 3PL in question.

Just like CPLEX, both MA1 and MA2 also generated optimal solutions for all instances with $|R| = 20$ and small items. The positive numbers (marked in bold) in the column “Best gap (%)” under the block “MA1” or “MA2” indicate that CPLEX was superior to MA1 or MA2 for those groups, which are all “easier” groups according to Fig. 9. For the harder groups, our MA implementations exhibited their superiority and achieved better solutions than CPLEX. Moreover, our MA implementations are more likely to outperform CPLEX as the difficulty of the instance increases; this is illustrated in Fig. 12, where the order of instance groups is the same as that in Fig. 9.

Table 3 shows that the superiority of our MA implementations over CPLEX is even greater for the instances with big items than with small items. This is because more containers are needed when the items are relatively large, which significantly increases the number of variables in the IP formulation and reduces the effectiveness of the branch-and-cut approach utilized by CPLEX. Note that our big items were randomly generated to have volumes between 2 and 6 cubic meters, and these types of items are commonly encountered by the 3PL in question.

An inspection of the columns under the blocks “MA1” and “MA2” reveals that there is not much difference in the quality of the solutions found by these two MA implementations. However, MA2 requires between approximately 2.4 and 8.1 times the amount of computation time employed by MA1 to complete.

6. Conclusions and future work

In this article, we introduced a new freight allocation and consolidation problem that involves choosing shipping routes for shipments and loading items into containers with varying sizes and costs. It models the task faced by a third-party logistics company under contract with the major children’s textiles manufacturer, where different shipments of textiles are to be consolidated so that the total cost (consisting of container and delivery costs) is minimized. Since this problem is NP-hard, we designed and implemented a memetic algorithm approach to solve it under practical settings. Extensive experimental results show that compared to CPLEX 11.1, our two MA implementations (MA1 and MA2)

outperform CPLEX in terms of both solution quality and computation time when solving medium- and large-sized instances. On average, MA1 and MA2 improved on the CPLEX solutions for instances with small items by 1.04% and 0.98%, respectively. These two MA implementations showed more advantages on solving the instances with big items since the corresponding average improvements are as much as 7.15% and 7.25%. By experiments, we also compared the performance of MA1 and MA2, and find that MA1 is more time-saving while providing equivalent solution quality.

Major business enterprises looking to expand into the emerging Chinese market often require the services of 3PLs for their transportation needs. However, many companies are hesitant to venture into the region due to a prevailing impression that local 3PLs are ill-equipped to provide cost-effective logistical support. There is a need to assure foreign investors that local 3PLs can handle complex practical problems; the project upon which this study is based is a prime example.

The FCCP represents an initial investigation into a common logistical problem where the consolidation of different shipments and the packing of items into containers are simultaneously taken into account. We modeled the packing of items as a one-dimensional bin-packing problem, which is justified since the customer of the 3PL is a textile manufacturer. An obvious extension would be to consider two-dimensional or three-dimensional packing, which may be more appropriate for customers dealing in other types of products. Our MA approach can be customized for these cases by replacing our dynamic programming based VSBPPS algorithm with an appropriate sequence-based algorithm for 2D or 3D packing.

Acknowledgements

This research was partially supported by the Fundamental Research Funds for the Central Universities, HUST (Grant No. 2012QN213) and National Natural Science Foundation of China (Grant No. 71201065). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.ejor.2013.09.015>.

References

- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Englewood Cliffs, N.J., USA: Prentice Hall.
- Alves, C., & Valério de Carvalho, J. M. (2007). Accelerating column generation for variable sized bin-packing problems. *European Journal of Operational Research*, 183, 1333–1352.
- Back, T., Fogel, D. B., & Michalewicz, Z. (1999). *Basic algorithms and operators*. IOP Publishing Ltd.
- Coffman, E., Jr, Garey, M., & Johnson, D. (1996). *Approximation algorithms for bin packing: A survey. Approximation algorithms for NP-hard problems*. PWS Publishing Co., p. 46–93.
- Correia, I., Gouveia, L., & Saldanha-da Gama, F. (2008). Solving the variable size bin packing problem with discretized formulations. *Computers & Operations Research*, 35, 2103–2113.
- Croxton, K. L., Gendron, B., & Magnanti, T. L. (2003). Models and methods for merge-in-transit operations. *Transportation Science*, 37, 1–22.
- Cunha, C. B., & Silva, M. R. (2007). A genetic algorithm for the problem of configuring a hub-and-spoke network for an LTL trucking company in Brazil. *European Journal of Operational Research*, 179, 747–758.
- Drezner, Z., & Hamacher, H. W. (2004). *Facility location: Applications and theory*. Berlin: Springer-Verlag.
- Dror, M., & Hartman, B. C. (2007). Shipment consolidation: Who pays for it and how much? *Management Science*, 53, 78–87.
- Epstein, L., & Levin, A. (2008). An APTAS for generalized cost variable-sized bin packing. *SIAM Journal on Computing*, 38, 411–428.
- Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2, 5–30.
- Friesen, D. K., & Langston, M. A. (1986). Variable sized bin packing. *SIAM Journal on Computing*, 15, 222–230.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Haouari, M., & Serairi, M. (2009). Heuristics for the variable sized bin-packing problem. *Computers & Operations Research*, 36, 2877–2884.
- Haouari, M., & Serairi, M. (2011). Relaxations and exact solution of the variable sized bin packing problem. *Computational Optimization and Applications*, 48, 345–368.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, Michigan: University of Michigan Press.
- Jeong, S.-J., Lee, C.-G., & Bookbinder, J. H. (2007). The European freight railway system as a hub-and-spoke network. *Transportation Research Part A: Policy and Practice*, 41, 523–536.
- Kang, J., & Park, S. (2003). Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147, 365–372.
- Kara, B. Y., & Tansel, B. Ç. (2001). The latest arrival hub location problem. *Management Science*, 47, 1408–1420.
- Klincewicz, J. G. (1990). Solving a freight transport problem using facility location techniques. *Operations Research*, 38, 99–109.
- Kratka, J., Stanimirovic, Z., Tosic, D., & Filipovic, V. (2007). Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, 182, 15–28.
- Leung, L. C., Hui, Y. V., Wang, Y., & Chen, G. (2009). A 0–1 LP model for the integration and consolidation of air cargo shipments. *Operations Research*, 57, 402–412.
- Liu, J., Li, C.-L., & Chan, C.-Y. (2003). Mixed truck delivery systems with both hub-and-spoke and direct shipment. *Transportation Research Part E: Logistics and Transportation Review*, 39, 325–339.
- Loh, K.-H., Golden, B., & Wasil, E. (2008). Solving the one-dimensional bin packing problem with a weight annealing heuristic. *Computers & Operations Research*, 35, 2283–2291.
- Lu, Z., & Hao, J. K. (2010). A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203, 241–250.
- Marasco, A. (2008). Third-party logistics: A literature review. *International Journal of Production Economics*, 113, 127–147.
- Martín, J. C., & Román, C. (2004). Analyzing competition for hub location in intercontinental aviation markets. *Transportation Research Part E: Logistics and Transportation Review*, 40, 135–150.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. Cambridge, Mass., USA: MIT Press.
- Moon, C., Kim, J., Choi, G., & Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, 140, 606–617.
- Moscatto, P. (1999). Memetic algorithms: A short introduction. In *New ideas in optimization* (pp. 219–223). UK: McGraw-Hill Ltd.
- Ngueveu, S. U., Prins, C., & Wolfler Calvo, R. (2010). An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37, 1877–1885.
- Osogami, T., & Okano, H. (2003). Local search algorithms for the bin packing problem and their relationships to various construction heuristics. *Journal of Heuristics*, 9, 29–49.
- Popken, D. A. (1994). An algorithm for the multiattribute, multicommodity flow problem with freight consolidation and inventory costs. *Operations Research*, 42, 274–286.
- Simchi-Levi, D. (1994). New worst-case results for the bin-packing problem. *Naval Research Logistics*, 41, 579–585.
- Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to genetic algorithms*. Berlin Heidelberg, New York: Springer.
- Song, H., Hsu, V. H., & Cheung, R. K. (2008). Distribution coordination between suppliers and customers with a consolidation center. *Operations Research*, 56, 1264–1277.
- Ta, H.-P., Choo, H.-L., & Sum, C.-C. (2000). Transportation concerns of foreign firms in China. *International Journal of Physical Distribution & Logistics Management*, 30, 35–54.
- Takano, K., & Arai, M. (2009). A genetic algorithm for the hub-and-spoke problem applied to containerized cargo transport. *Journal of Marine Science and Technology*, 14, 256–274.
- Wang, Q., Zantow, K., Lai, F., & Wang, X. (2006). Strategic postures of third-party logistics providers in mainland China. *International Journal of Physical Distribution & Logistics Management*, 36, 793–819.