Computers & Industrial Engineering 97 (2016) 137–145

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

The vehicle loading problem with a heterogeneous transport fleet

Jiansheng Liu [a], Alice E. Smith [b,*], Dan Qian [a]

[a] School of Mechanical and Electronic Engineering, Nanchang University, Nanchang 330031, China
[b] Department of Industrial and Systems Engineering, Auburn University, Auburn 36849, AL, USA

ARTICLE INFO

Article history:
Received 17 August 2015
Received in revised form 28 April 2016
Accepted 30 April 2016
Available online 2 May 2016

Keywords:
Vehicle loading problem
Finished vehicle logistics
Heterogeneous multi-type fleet
Improved branch and bound algorithm
Greedy search

ABSTRACT

As an emerging variant of the vehicle loading problem, the heterogeneous multi-type fleet vehicle loading problem in finished vehicle logistics (HVLP–FVL) is modeled and solved. The HVLP–FVL maximizes the total profit of a vehicle fleet where different models of finished vehicles result in different profits and routing costs are considered based on distance traveled. Addressing the vehicle structures, simplified geometric models of both the finished vehicle and transport vehicle are defined. The optimization considers which finished vehicle orders to transport and then makes the loading assignment to the transport vehicles. To improve the computational performance of the traditional branch and bound algorithm, an enhancement using greedy search based on oscillation analysis is proposed. A real case study is used to evaluate the effectiveness of the improved algorithm and a series of experiments are conducted over a set of finished vehicle loading problems. The results demonstrate the proposed approach has superior performance and satisfies users in practice. Contributions of the paper are the modeling and solving of a real complex problem in vehicle manufacturing logistics and a simple branch-and-bound speed up that could be used in other problem classes.

1. Introduction

In recent years the auto industry was witnessed greater development of finished vehicle logistics. It is now becoming standard practice to use third party logistics providers for transportation, storage, and services. Finished vehicle logistics is a bridge between auto manufacturing enterprises and auto dealers, and provides efficient service for both. Finished vehicle logistics is broadly defined as a series of associated processes from assembly plants to final destinations. Problems and trends arising in the finished vehicle logistics supply chain motivate researchers and practitioners to study these processes with the aim to improve them. Among them, the loading process is a distribution logistic operation that is of vital importance. It is not only the beginning of a series of finished vehicle logistics operations, but also the quality of the loading solution is a crucial economical factor. To take a simple example, two finished vehicles with a length of 4 m, a width of 1.5 m and a height of 1.4 m cannot be loaded in a transport vehicle space with a length of 6 m, a width of 2 m and a height of 2 m although the volume of both finished vehicles makes up just 70% of the vehicle loading volume. Issues such as this motivate the vehicle loading problem in finished vehicle logistics (VLP–FVL). In practice, the VLP–FVL is done by a manager's experience or through simple heuristics. However, with increasing industry competition, the logistics enterprise's objective is to maximize the possible profit with a reasonable computation time. With the number of factors involved it is difficult to achieve this by personal experience or simple heuristics. If the transport fleet is heterogeneous, it makes the problem more challenging. By providing an effective and practical solution approach to this complex problem this paper contributes to the automotive logistics field. And by improving the branch and bound algorithm with a greedy search based on oscillation analysis this paper contributes in general to the optimization literature.

The paper is organized as follows. The next section describes the problem. Section 3 cites the related literature. Section 4 builds geometric models for both the finished vehicle and the transport vehicle. Section 5 introduces a solution method for the HVLP–FVL. Section 6 presents a branch and bound approach using greedy search. Real test instances, search process comparisons, and performance analyses are discussed in Section 7. Finally, conclusions are drawn in Section 8.

2. Problem overview

In Fig. 1, the problem considers a set of ordered items (finished vehicles) characterized by length, width, height, weight and profit and a transport vehicle fleet types characterized by length, width,

* Corresponding author.
E-mail address: smithae@auburn.edu (A.E. Smith).

http://dx.doi.org/10.1016/j.cie.2016.04.021
0360-8352/

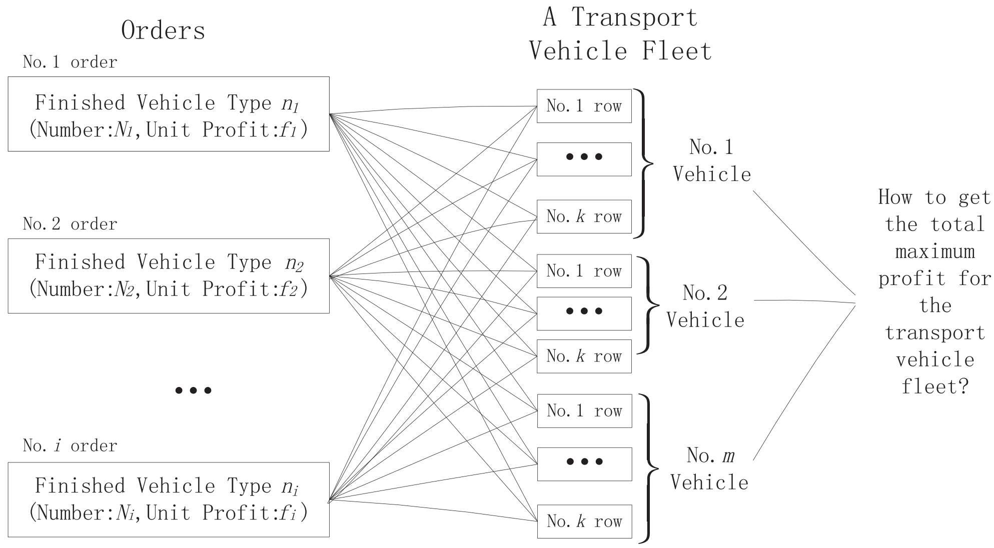

Fig. 1. The description of the HVLP–FVL.

height, floors and allowable loading weight of each vehicle type. The objective is to select the finished vehicles to load and the transport vehicles into which to load them to maximize the total profit of a fleet of transport vehicles. We term this problem the *Heterogeneous Multi-type Fleet Vehicle Loading Problem in Finished Vehicle Logistics* (HVLP–FVL). Although the HVLP–FVL is found widely in real automotive supply chains, it is very seldom studied in the literature. A typical example is given in the following description:

(1) The automotive factories produce various types of finished vehicles, such as sedan, pickup truck and SUV. A logistics company has its own vehicle fleet consisting of different kinds of transport vehicles, for example, 4-axis truck, 5-axis truck and 6-axis truck. In practice, the loading space of each transport vehicle is divided into multiple floors, and each floor may be divided into row loading space with specified length, width and height. There are also some private trucks contracted by the logistics company, and they can be hired if its own vehicle fleet cannot satisfy all orders.
(2) The logistics company receives orders from the automotive factories (considered suppliers) to transport different kinds of finished vehicles to dealers (considered customers). The set of finished vehicles is initially located at a main depot.
(3) The dealers are located in different cities, and might be far away from each other. Therefore, it is not economical to deliver finished vehicles along a single route. The logistics company defines one center city for each cluster of dealers based on their locations. A branch depot is located the center city. Then, routing is planned from the main depot (located in the supplier's city) to the branch depots (located in the cluster center cities). There is a transport vehicle fleet assigned to each route, however, the vehicle fleet is not fixed.
(4) Drivers will return to the main depot after they finish delivery. Having necessary rest, they report to the scheduling department of the logistics company, so the schedulers know how many transport vehicles are available and their types. These available vehicles form the vehicle fleet that day.
(5) Different vehicle loading solutions result in different profit depending on the type and number of finished vehicles selected because profit varies according to type of finished vehicles. The problem aims at determining a heterogeneous vehicle fleet loading solution maximizing profit.

According to the description above, there are some assumptions and constraints, as follows:

(1) Split delivery for orders is allowed. That is, an order can be serviced by more than one transport vehicle.
(2) Any type of finished vehicle can be loaded in any kind of transport vehicle. For the same reason, any type of transport vehicle can transport any kind of finished vehicle.
(3) The delivery from manufacturers to the main depot is not considered. Similarly, the delivery from the branch depot to each dealer is not considered. Loading and unloading costs are not considered.
(4) Routing cost includes highway toll, labor cost, and power cost; these are assumed to relate linearly to distance.
(5) The total weight of all finished vehicles loaded in a transport vehicle must not exceed an upper limit of the load weight capacity.
(6) Volume constraint: Each finished vehicle has a positive volume with length, width and height. Since the transport vehicles have a complex loading structure with floors and rows (see Section 4.2), the items loaded in any vehicle must meet the loading volume limit of the relevant row on the relevant floor.
(7) If the transport fleet consists of identical vehicles it is denoted homogeneous, while it is heterogeneous if different types of transport vehicles are present, as herein.

3. Literature review

If the transport vehicle loading space were one-dimensional, then the HVLP–FVL is reduced to the classic Knapsack Problem (KP). From an optimization objective view, the HVLP–FVL is analogous to the KP because both maximize profits. From an operation process view, the HVLP–FVL is like a bin packing problem (BPP) since both pack into a number of containers. In principle, the HVLP–FVL derives from the vehicle loading problem (VLP).

3.1. Knapsack problem

In the classic KP (Martello & Toth, 1990a; Kellerer, Pferschy, & Pisinger, 2004), each item has an associated value, and the objective is to choose a set of items that fit into a single knapsack to maximize profit. There exist many approaches for the KP and its variants (Pisinger, 1995b), such as the bounded single constraint KP (BKP) (Martello, Pisinger, & Toth, 1999), the Multiple Knapsack Problem (MKP) (Chekuri & Khanna, 2005), the Multi-Dimensional Knapsack Problem (MDKP) (Chu & Beasley, 1998), the Multiple-Choice Knapsack (MCKP) (Pisinger, 1995a), and the multiple-choice multi-dimensional knapsack problem (MMKP) (Hifi, Michrafy, & Sbihi, 2006). In the classic KP, each item can only be selected not at all or once (no duplicates allowed). In the BKP, each type of item can be chosen within a limited amount. The MKP is a special case of the generalized assignment problem where the profit and the size of an item can vary based with the specific knapsack that it is assigned to. The MCKP is to choose one item from each class such that the profit is maximized. In contrast, in the HVLP–FVL, any type of item can be selected in any number, and the loading vehicle fleet is heterogeneous, so it has shared characteristics with KP, BKP, and MCKP.

3.2. Bin packing problem

Similar with the three-dimensional characteristics of the HVLP–FVL, in the BPP all items are to be packed into a minimum number of containers with three dimensional constraints (Faroe et al., 2003; Martello, Pisinger, & Vigo, 2000; Scheithauer, 1991). There are BPP variants; for example, the variable size bin packing problem (VSBPP), the variable cost and size bin packing problem (VCSBPP) (Correia, Gouveia, & Saldanha-da-Gama, 2008; Crainic et al., 2011; Pisinger & Sigurd, 2005), and the generalized bin packing problem (GBPP) (Baldi, Crainic, Perboli, & Tadei, 2012). In the BPP all bins have the same capacity, in the VSBPP they have different capacities, and in the VCSBPP they differ in both capacity and cost. Different from the HVLP–FVL, in the BPP, VSBPP and VCSBPP, all items are mandatory. Compared with the BPP family, the HVLP–FVL has more complex inner loading structures (see Section 4). In addition, the optimization objective of the BPP family is to minimize the total number of used bins, not the maximum profit of selected items, which is the optimization objective of the HVLP–FVL.

3.3. Vehicle loading problem

Although the VLP is widespread in real logistics, it has been little studied in the literature. Ren, Sun, and Wang (2010) studied a hybrid genetic algorithm to improve the loading weight and capacity utilization rate for the multi-vehicle and multi-cargo loading problem. Romao, dos Santos, and Arroyo (2012) presented a two phase method to minimize the unused capacity or the number of vehicles for solving both the 2D and 3D vehicle loading problems considering mandatory and optional items. Neither considers the objective of maximum profit. In some papers, the loading problem is only taken as one additional constraint (Doerner, Fuellerer, Hartl, Gronalt, & Iori, 2007; Fuellerer, Doerner, Hartl, & Iori, 2010; Iori, Salazar-González, & Vigo, 2007). Furthermore, in other literature (Bortfeldt & Homberger, 2013; Gendreau, Iori, Laporte, & Martello, 2006; Moura & Oliveira, 2009) although the vehicle loading problem and vehicle routing problem are considered together, the objective is to optimize the vehicle numbers and the total travel distance. Different from the traditional VLP, the HVLP–FVL has its own characteristics, for example, (1) the vehicles in the traditional VLP have one bin or loading space. The vehicles for loading finished vehicles in the HVLP–FVL are special trucks with more complex inner loading structures, (2) there are fewer finished vehicles at each delivery, so schedulers pay more attention to the profit of each delivered finished vehicles, and (3) in most case, all items in the traditional VLP are mandatory, while items are optional in HVLP–FVL.

In conclusion, due to the intrinsic nature of the HVLP–FVL, there are no existing methodologies that can be directly used to solve the problem. Furthermore, the HVLP–FVL is NP-hard since it can be reduced to the general VLP, which is a well-known NP-hard problem (Yüceer & Özakça, 2010). Consequently, we develop a branch and bound algorithm embedded with a greedy strategy which enables a feasible loading plan maximizing the total profit in a short computational time.

4. Geometric models

The HVLP–FVL depends on some aspects: the types of finished vehicles and their destinations, which are both included in orders, the types of transport vehicles available, and locations of the main depot and cluster depot destinations. From a geometric view, the volume constraint in the HVLP–FVL is to match the three-dimensional size of the loaded finished vehicles to the transport vehicles.

Unlike general packing goods, the contours of finished vehicles are not simple planes. It will be a very complex process to build a precise geometric structure model. However, we can reduce the complex structure of both finished vehicles and transport vehicles by ignoring relatively minor geometric structures, and keeping the main structure characters. In the following, the geometric structure models for the finished vehicles and transport vehicles are built, respectively.

4.1. Finished vehicle model

The geometric model can be obtained by extending the maximum length, maximum width, and maximum height from a closed three-dimensional geometric contour of finished vehicles. Taking a sedan car family as an example, its contour sketch, three-dimensional view, and database representation are illustrated in Fig. 2.

Besides these geometric data, there is a control parameter, the loading angle. In most previous cases, all finished vehicles are placed horizontally. However, it has become popular for finished vehicles to be placed with a positive angle (see Fig. 3). It is obvious that equivalent length is less than the actual length. By this way, the loading capacity can be improved to some extent, and more finished vehicles might be loaded.

4.2. Transport vehicle model

Taking one six-axis transport vehicle as an example, its contour sketch, three-dimensional view, and database representation are illustrated in Fig. 4.

Depending on the difference between the upper and lower floors, transport vehicles are classified into two types: a single structure transport vehicle and a complex structure transport vehicle. In a single structure transport vehicle, the loading area on the upper floor of the vehicle has not been extended, so the loading area is the same as the lower floor (see Fig 4.2). In a complex structure transport vehicle, the loading space on the upper of the vehicle has been extended (see Fig 4.3), which makes the loading space on the top floor larger. Here, the upper loading area is divided into two rows for the top floor. Since there are two floors, a height limit for the lower floor, is imposed which means that not all types of finished vehicles can be assigned to the lower floor.

In the KP and BPP, the whole loading space is taken as one loading unit, but in the HVLP–FVL the transport vehicle space is divided into rows. In a single structure transport vehicle, there are two rows (the upper floor and lower floor are defined as one row each). In a complex structure transport vehicle, there are three rows

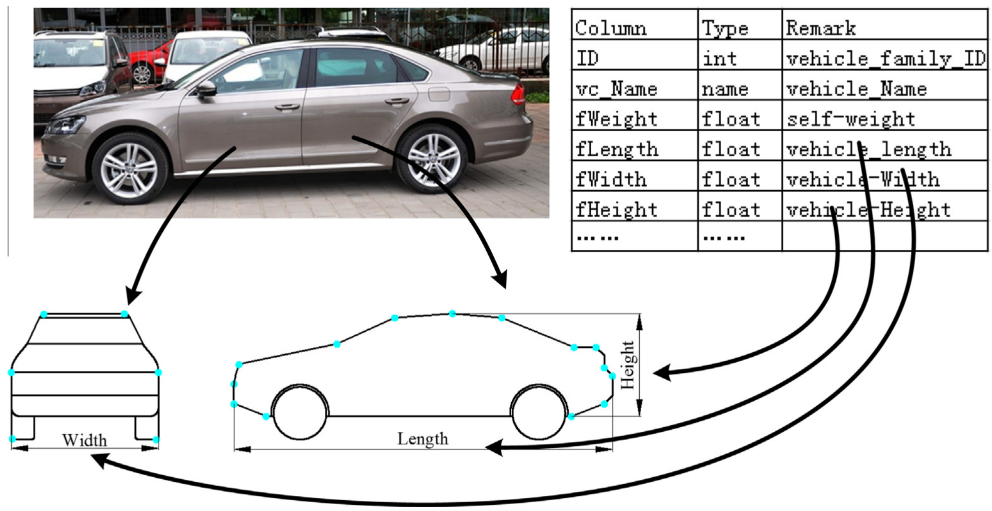

Fig. 2. An example of a model description for finished vehicles.

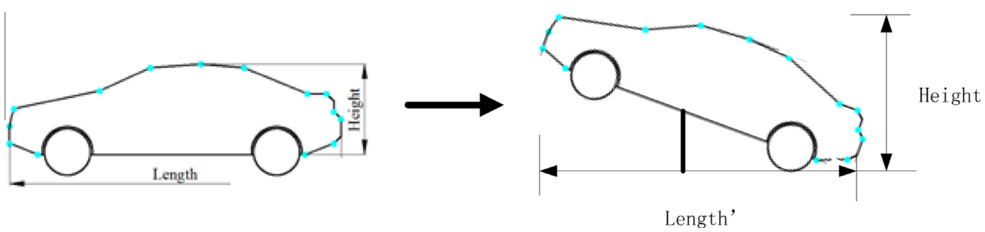

Fig. 3. Traditional and angled placed method of loading finished vehicles.

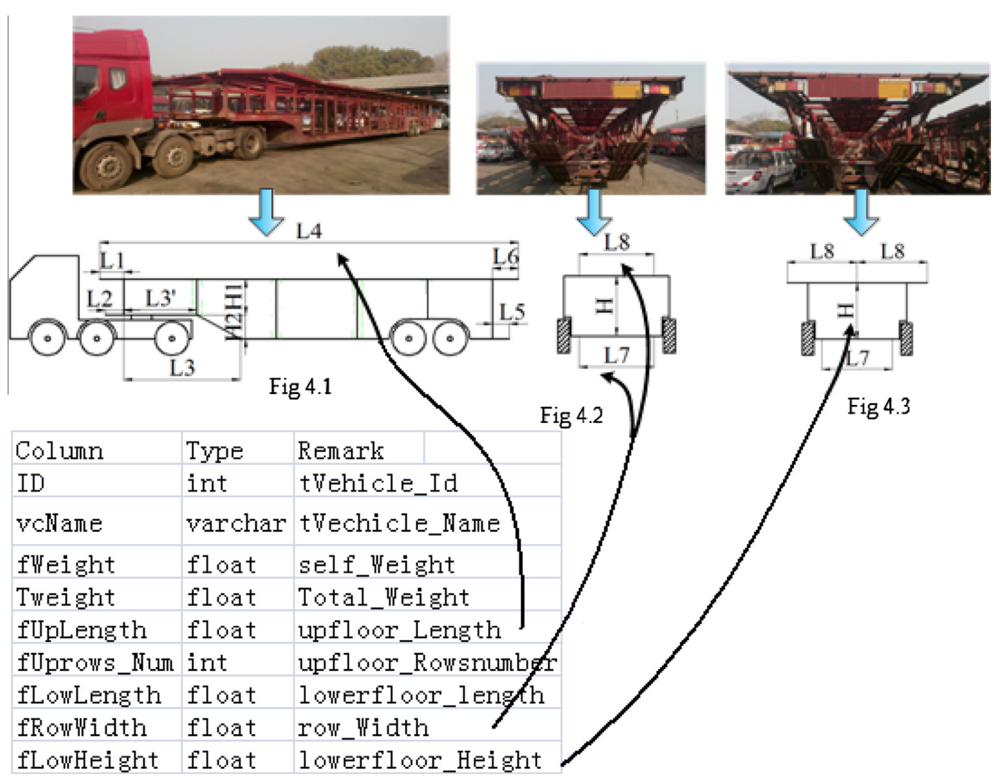

Fig. 4. An example of the model description for transport vehicles.

(the lower floor is defined as one row, and the upper floor is defined as two rows). Different from the KP and the BPP, the optimization model is built based on rows, not the whole loading space, and the final optimization solution will be obtained by combining rows.

5. Optimization model

The HVLP–FVL objective consists of selecting finished vehicle orders to fill and specifying the position in each transport vehicle of each finished vehicle within a heterogeneous fleet to maximize profit. Although minimizing the number of transport vehicles used is one of a logistics company's objectives, the main objective is to maximize the total profit by utilizing the total capacity of the selected transport vehicles. To do that, the transport vehicles to be used and the finished vehicles to make up the loading space on each must be selected.

As mentioned in the Section 1, the company has its own vehicle fleet and this will be preferentially chosen for current orders. If it cannot meet the requirements of existing orders, the remaining orders will be delivered by private transport vehicle fleets. In this case, all orders are considered optional for own vehicle fleet, and the remaining orders are considered mandatory for the private vehicle fleet. In other words, after its own vehicle fleet loading is fully loaded, the company may use the private vehicle fleet. Each selected finished vehicle must be positioned inside the fleet vehicles. Of course, the finished vehicles placed must respect the transport vehicle capacity (length, width, height and loading weight). Each finished vehicle has self-weight (w_i), and each transport vehicle has also self-weight (W_k) and a limit on total weight (G_k).

After the fleet vehicles are determined, the total self-weight ($\sum W_k$) and total standard loading weight ($\sum G_k$) of the vehicle fleet are calculated. Then the orders (each order specifies the kind and quantities of finished vehicles) are selected from the orders list. Two ways of order selection are considered.

One alternative is that schedulers manually choose orders according to customer priority. For this, an order point formulation can be formulated as follows:

$$\sum_{k=1}^{m}\sum_{j=1}^{R_k}\sum_{i=1}^{n} w_i x_{ijk} + \sum_{k=1}^{m} W_k \geqslant f\sum_{k=1}^{m} G_k \tag{1}$$

Usually, the total weight (including the finished vehicle weight and the transport vehicle self-weight) is less than $1.3G_k$ (here, = 1.3). During the process of manual choosing orders, the total weight steadily increases. When Eq. (1) is satisfied, then the process of order selection terminates. Then these orders will be optimized for loading as below.

If there is no priority for any order, then the other alternative, auto loading, occurs. This considers all orders as possibilities. For both order selection alternatives (manual and auto) the optimization model below is applied:

$$\max \sum_{k=1}^{m}\left[\left(\sum_{j=1}^{R_k}\sum_{i=1}^{n} f_i x_{ijk}\right) - (Cr_k + Co_k + Cf_k)\right] \tag{2}$$

s.t.

$$x_{ijk} \in Z, \quad \forall\, i = 1,2,\ldots,n;\ \forall\, j = 1,2,3;\ \forall\, k = 1,2,\ldots,m \tag{3}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{R_k} x_{ijk} \leqslant N_i, \quad \forall\, i = 1,2,\ldots,n \tag{4}$$

$$\sum_{i=1}^{n} l_i x_{ijk} \leqslant L_{jk},\ \forall\, k = 1,2,\ldots,m; \quad \forall\, j = 1,2,3 \tag{5}$$

$$x_{i3k}(p_i - 1) = 0,\ \forall\, i = 1,2,\ldots,n; \quad \forall\, k = 1,2,\ldots,m; \tag{6}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{R_k} w_i x_{ijk} + W_k \leqslant fG_k, \quad \forall\, k = 1,2,\ldots,m; \tag{7}$$

Notation:

n: The type number of finished vehicles,
m: The number of transport vehicles,
R_k: Row number of transport vehicle k, $k = 1,2,3$,
f_i: Profit of finished vehicle type i,
x_{ijk}: Final number of finished vehicle type i in row j of transport vehicle k,
Cr_k: Labor fee of transport vehicle type k,
Co_k: Power fee of transport vehicle type k,
Cf_k: Highway toll of transport vehicle type k,
N_i: The number of finished vehicle types $i = 1,2,\ldots,n$,
l_i: Length of finished vehicle type i,
w_i: Weight of finished vehicle type i,
h_i: Height of finished vehicle type i,
L_{jk}: Loading length in the row j of transport vehicle type k,
H: Height of the lower floor of the transport vehicle,
W_k: Self-weight of transport vehicle k, $k = 1,2,\ldots,m$,
G_k: Standard total transport weight of vehicle type k,
f: An overloading coefficient.
p_i: $p_i = \begin{cases} 0, & \text{if } h_i \geqslant H \\ 1, & \text{if } h_i < H \end{cases}$

The objective function (2) maximizes the total profit of the vehicle fleet which given by the difference between the total profit of the selected items and the routing cost. As we mentioned in Section 2, the routing cost has a linear relation with the routing distance.

Constraint (3) enforces the non-negative integer nature of the decision variables. Constraint (4) ensures the finished vehicles chosen do not exceed those ordered. Constraint (5) limits the length of each row of transport vehicles. Constraint (6) can prevent loading too high finished vehicles into the lower row with height limitation. That is x_{i3k} will be zero when p_i takes value zero. Constraint (7) limits the total weight assigned to each transport vehicle not to exceed its limits considering the overloading factor.

For automated order selection and loading optimization, a branch and bound algorithm is provided in the next section.

6. Solution approach

A complex heuristic algorithm, such as genetic algorithm or neural network, may incur calculation time which does not meet operational needs. There is also a need for specialized coding and tuning of these heuristics. The branch and bound method (Lawler & Wood, 1966) is an algorithm for finding optimization solutions. Starting with the original problem, the problem is divided into sub problems so that the union of feasible solutions for these sub problems is the entire set of feasible solutions. Sub problems are further split until they are solved. It has been successfully used to solve many integer programming problems. Several branch and bound algorithms have been developed to solve the multiple-choice knapsack problem (Dyer, Kayal, & Walker, 1984), the packing problem (Spieksma, 1994), the multidimensional knapsack problem (Gallardo, Cotta, & Fernández, 2005), and the bin packing problem (Martello & Toth, 1990b). However it has seldom been applied to the VLP.

6.1. Branch and bound algorithm

Often traditional branch and bound procedures are used to solve integer linear programming problems whose canonical form can be expressed as follows (Papadimitriou & Steiglitz, 1998):

$$\max z = c^T x \\ s.t.\ Ax \leqslant b \\ x \geqslant 0,\ x \in Z^n \tag{8}$$

where the entries of x, b and c are vectors and A is a matrix having integer values.

The improved branch and bound algorithm for solving the HVLP–FVL presented in this paper is a fairly standard type and consists of solving a structured hierarchy of sub problems. Each branch corresponds to a sub problem, and the nodes of the HVLP–FVL are generated from the predecessors by adding an appropriate constraint ($x_i \leqslant [x_i]$ and $x_i \geqslant [x_i] + 1$). The corresponding sub problem relaxation node is solved to obtain an upper bound z_i^+ on the original optimal value z_0.

Termination of a branch occurs when either:

(a) The current node solution is integer, and the optimal value is updated.
(b) The current node solution is infeasible.

The principle of branch and bound is to increase the upper bound and decrease the lower bound to finally get the optimal integer value. How to branch and which strategy should be taken with constraints are important during the procedure. A disadvantage of traditional branch and bound is that to find all non-integer variable values, the method tries directly to branch from the first variable value in the sequence. When the number of constraints and variables is relatively small, the branch levels are manageable, and the calculation time is reasonable. In contrast, in the case of a large scale problem, the branch process becomes very complex. The time of linear programming will obviously increase, as will with the iteration times. Because the HVLP–FVL is an operational problem that must be solved daily, a solution is needed in a short time. In the following section, the tradition branch and bound with a greedy strategy is used to accomplish this.

6.2. Greedy search strategy

From the viewpoint of decision theory, how to branch, how to choose sub-branches, and which decision rules to apply are important. Here, a greedy strategy is adopted. Greedy approaches have been applied to solve the KP (Gorski, Paquete, & Pedrosa, 2012), the multi-knapsack problem (Kan, Stougie, & Vercellis, 1993), and the rectangles packing problem (Gu, Dong, & Zheng, 2009). Here, first, all non-integer variable values are identified, and then they are analyzed by comparing z with $z_{(i,\min)}$, so the method preferentially begins to branch with the x_i that has maximum oscillation. The steps are as follows:

(1) Selection: Calculate all variables x_i and the original solution z_0, terminate if each x_i is integer.
(2) Pre-Branch: A new solution will be obtained after each x_i branches upwards and downwards by linear programming, the new objective values are z_i^+ and z_i^-, respectively. Then $z_{(i,\min)} = \min\{z_i^+, z_i^-\}$.
(3) Decision: Perform oscillation analysis by comparing $z_{(i,\min)}$ with z_0, and then find $\max\{|z_{(1,\min)} - z_0|, \ldots, |z_{(i,\min)} - z_0|, \ldots\}$, and $\bar{x}_i$.
(4) Branch: First branch from $\bar{x}_i$, and repeat the steps above. Then the final solution is obtained.

6.3. Implementation

The process of building and solving the HVLP–FVL can be complex for people without a background in optimization, but user interfaces can be developed. The branch and bound algorithm, and Matlab, JAVA, and the Oracle database management system provide a timely solution to the HVLP–FVL in an online environment. The implementation process consists of the three steps that follow.

Step 1. User interface is developed in JAVA.
(a) Maintenance interface: users can insert, update and delete order information, transport vehicle data and finished vehicle data. These data are stored in the database and can be shared during the optimization process.
(b) Operational interface: users can choose orders from the data base of orders. The orders list is ordered by priority, if any priority exists. Meanwhile, the available transport vehicles that form the vehicle fleet are also given.
(c) Data transportation: Relational tables are defined in the database for some entities, such as orders, finished vehicles and transport vehicles. In each relational table column-'ID' is defined as the primary key. Here, the value of primary key is sent to the HVLP–FVL algorithm by the data interface. Because the value of the primary key is unique, when the algorithm begins to run, the other related data of selected orders, finished vehicles and transport vehicles are automatically obtained from the database.

Step 2. The improved branch and bound algorithm is programmed in Matlab so the operational interface can call it. The m document of the algorithm in Matlab is programmed as follows:
(a) Initialize all decision variables, let them be zero, and the goal is ∞.
(b) Linear relaxation, get x_i and evaluate. Program the termination condition of criteria.
(c) Terminate if the value is integer. Program the termination condition of each branch.
(d) Calculate the solution which has the largest oscillation. Find the non-integer decision variable which has priority to branch.
(e) Construct the first branch, and assess if the solution is better.
(f) Construct the second branch, compare with the first branch, and assess if the solution is better.

Step 3. The output standard is defined. For the developed system the detailed loading solution is returned as an XML document to the operational interface.

In the XML document, the final loading solution shows for each transport vehicle the types of the finished vehicles, the quantities and the destinations. Each finished vehicle is assigned to a row on the transport vehicle and the finished vehicle which has the furthest destination will be the first loaded on that row and so on.

7. Computational experience

The algorithms are coded in Matlab R2010b and tested on a PC with a Pentium 2.4 GHz processor, 4.00 GB RAM, and the Microsoft Windows 7 operating system. To our knowledge, there are no instances publicly available, therefore a real situation in China is given to illustrate our approach.

7.1. A case study

Using information of a medium-sized logistics company in China, we ran our algorithm on some instances. In the database, there are about 100 transport vehicles and 600 types of finished vehicles. The logistics company transports various kinds of finished vehicles from its own depots to dealers. There is a transport vehicle fleet for each route between city pairs. Every day the available transport vehicles and the orders are collected, and schedulers need to specify the vehicles loadings. The orders from one city to another city are listed in Table 1. Each order record includes the vehicle family type, quantity, and profit (signed in the contract and updated each year) of the finished vehicle. The capacity information of the transport vehicles are listed in Table 2.

Using the enhanced branch and bound algorithm, the optimized solution is obtained in 2 s, and shown in Table 3. The height of finished vehicle C is higher than the lower floor's height of any transport vehicle. So finished vehicle C cannot be loaded in the lower floor, and must be loaded in an upper floor. Moreover, because of capacity limits, the 1st and 3rd orders are completed, while the 2nd order is not. There are two finished vehicles left. The results in the table satisfy the maximum profit expectation based on real constraints and demonstrate the feasibility and effectiveness of the approach.

7.2. Search process comparison

To clearly illustrate the difference in the search process between the traditional and the improved branch and bound algorithm, we take another small instance. As above, the orders and available vehicles are listed in Tables 4 and 5.

(1) Variants:

There are three types of finished vehicles, which will be loaded in the three rows of the transport vehicle, so there are nine variants $(x_1, x_2, \ldots, x_9)$.

(2) Constraints:

There are three quantity constraints on finished vehicles, three length constraints on each row, and a maximum-total weight limitation of the transport vehicle.

(3) Objective function

$$\max \quad z = fx$$

$$f = [1012.5 \quad 904.5 \quad 840.32 \quad 1012.5 \quad 904.5 \quad 840.32 \quad 1012.5 \quad 904.5 \quad 840.32]$$

Constraints: $Ax \leqslant b$

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 4.9 & 3.85 & 3.665 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.9 & 3.85 & 3.665 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4.9 & 3.85 & 3.665 \\ 2.17 & 1.045 & 0.94 & 2.17 & 1.045 & 0.94 & 2.17 & 1.045 & 0.94 \end{bmatrix} \quad b = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 33 \\ 33 \\ 33 \\ 30 \end{bmatrix}$$

Table 2
Transport vehicle information.

Transport vehicle type	Lower floor length (m)	Lower floor height (m)	Upper floor length (m)	Number of rows	Loading weight (t)	Limit total weight (t)
a	29	2	33	3	27	49
b	27	2	28	3	22	43

Table 3
Loading optimization solution.

Truck rows		Finished vehicles quantity			Finished vehicle quantity/row	Finished vehicle quantity/truck
		A	B	C		
a	Lower row	6	0	0	6	20
	Upper row (Left)	0	0	7	7	
	Upper row (right)	0	0	7	7	
b	Lower row	4	2	0	6	18
	Upper row (Left)	0	0	6	6	
	Upper row (right)	0	6	0	6	
Total		10	8	20	38	38

Table 4
Finished vehicle information.

Finished vehicle type	Weight (t)	Length (m)	Width (m)	Height (m)	Quantity	Profit (RMB)
A	2.170	4.900	1.875	1.900	10	1012.50
B	1.045	3.850	1.505	1.910	10	904.50
C	0.940	3.665	1.616	1.689	10	840.32

Table 5
Transport vehicle information.

Family type	Lower floor length (m)	Lower floor height (m)	Upper floor length (m)	Number of rows	Self-weigh (t)	Limit total weight (t)
c	33	2	33	3	25	55

The search process analysis and optimized solution of the traditional branch and bound are in Tables 6 and 7, respectively. The search process analysis of the improved branch and bound is shown in Table 8.

The optimized solution of the traditional algorithm is $5 * A$, $10 * B$ and $9 * C$, and the detailed layout in the transport vehicle is listed in Table 7. The optimal solution has: total income = 21,670, total loading weight = 54.76, total run time = 14.59 s.

The optimization result of the improved algorithm has the same loading solution and the total run time = 1.37 s, a reduction of over an order of magnitude.

Table 1
Finished vehicle order information.

Finished vehicle type	Length (mm)	Width (mm)	Height (mm)	Weight (kg)	Quantity	Profit/vehicle (RMB)	Order destination
A	4325	1830	1730	1683	10	1400	D1
B	4340	1700	1485	1295	10	1350	D2
C	4666	1974	2200	1805	20	1500	D3

Table 6
The search process of traditional branch and bound.

No	Search process	Branch process	Objective value	Solution	Search Sequence
1	LP1	Start	22,184	$[0, 0, 8.9384, 2.3387, 5, 0.5308, 2.3387, 5, 0.5308]^T$	X_3 Branches downwards
2	LP2	$X_3 \leqslant 8$	21,616	$[0, 0, 8, 2.0582, 5, 1, 2.0582, 5, 1]^T$	X_4 Branches downwards
3	LP3	$X_4 \leqslant 2$	21,616	$[0, 0, 8, 1.4097, 6.274, 0.5287, 2.7067, 3.726, 1.4713]^T$	X4 Branches downwards
4	LP4	$X_4 \leqslant 1$	21,616	$[0, 0, 8, 0, 7.1806, 1.461, 4.1163, 2.8194, 0.539]^T$	X_5 Branches downwards
...	...	...	...	...	...
590	LP590	$X_6 \leqslant 0$	21,674	$[0, 0, 9.0041, 3, 4, 0, 2, 6, 0]^T$	X_3 Branches downwards
591	LP591	$X_3 \leqslant 9$	21,670	$[0, 0, 9, 3, 4, 0, 2, 6, 0]^T$	End

Table 7
HVLP–FVL optimization solution based traditional algorithm.

Finished vehicle	Lower row			Upper row (left)			Upper row (right)		
	A	*B*	*C*	*A*	*B*	*C*	*A*	*B*	*C*
Quantity	0	0	9	2	6	0	3	4	0

7.3. Performance analysis

A series of experiments was conducted to fully evaluate the proposed algorithm. The performance comparison between the traditional and the improved branch and bound algorithm is showed in Table 9. The improved approach reduces CPU time to a great extent. It is clear that the efficiency of the improved algorithms is significantly better which is important in this quick turnaround and low computational capacity situation.

As the problem size increases (number of finished vehicle types and number of transport vehicles) the benefits of the improved branch and bound algorithm become more apparent. The computational times of the improved algorithm are very modest and can work well in practice even where computational power is limited. Solution quality is not compromised.

8. Conclusion

The HVLP–FVL is an interesting problem in terms of both theoretical complexity and practical applications. In this paper, the problem is modeled and solved using a traditional branch-and-bound and a computationally improved branch-and-bound. The process and application of the proposed method are demonstrated using actual cases. Results show the efficiency and effectiveness of the proposed approach.

Although the article uses HVLP–FVL as the application, the improved branch and bound algorithm can also be useful for other problem types where a similar decision among a subset of the alternatives is required. As enhancements to this work, rather than a single depot from which shipments emanate, we may consider multiple depots. Also, the geometry of the transport truck and the finished vehicles are simplified. There might be benefits to consider more precise geometric models.

Table 8
The search process of the improved branch and bound.

No	Search process	Branch process	Objective value	Solution	Search sequence
1	LP1	Start	22,184	$[0, 0, 8.9384, 2.3387, 5, 0.5308, 2.3387, 5, 0.5308]^T$	X_4 Branches downwards
2	LP2	$X_4 \leqslant 2$	22,184	$[0, 0, 8.9646, 2, 5.6173, 0.3437, 2.6774, 4.3827, 0.6918]^T$	X_7 Branches upwards
3	LP3	$X_7 \geqslant 3$	21,905	$[0, 0, 8.8267, 2, 5.2514, 0.6174, 3, 4.0787, 0.5559]^T$	X_9 Branches downwards
4	LP4	$X_9 \leqslant 0$	21,905	$[0, 0, 8.6589, 2, 4.6243, 1.3411, 3, 4.7059, 0]^T$	X_8 Branches downwards
5	LP5	$X_8 \leqslant 4$	21,697	$[0, 0, 9.0041, 2, 6, 0.0273, 3, 4, 0]^T$	X_6 Branches downwards
6	LP6	$X_6 \leqslant 0$	21,674	$[0, 0, 9.0041, 2, 6, 0, 3, 4, 0]^T$	X_3 Branches downwards
7	LP7	$X_3 \leqslant 9$	21,670	$[0, 0, 9, 2, 6, 0, 3, 4, 0]^T$	End

Table 9
Performance comparison of the two algorithms.

Instance no	Types of finished vehicles	Number of transport vehicles (Three rows)	Number of variables	IP iterations of traditional	CPU Sec of traditional	IP iterations of improved	CPU Sec of improved
1	3	1	9	529	16	11	2
2	4	2	24	546	19	30	3.8
3	4	3	36	655	23	49	5.5
4	5	4	60	1834	62	76	7
5	6	5	90	5149	188	93	15
6	8	6	144	5386	202	116	18
7	9	7	189	7779	284	143	21
8	10	8	240	13,385	480	145	21.2
9	11	9	297	18,154	605	263	25.9
10	12	10	360	24,794	777	329	69.4
11	13	11	429	32,959	972	472	79.1
12	14	12	504	42,811	1188	825	100.7
13	15	13	585	54,513	1427	1650	125.4
14	16	14	672	68,229	1687	3422	153.0
15	17	15	765	84,122	1970	6951	182.8
16	18	16	864	102,355	2275	13,547	213.9
17	19	17	969	123,090	2601	25,244	245.0
18	20	18	1080	146,491	2950	45,095	274.3

Acknowledgement

This work is partly supported by grants from the China Scholarship Council.

References

Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2012). The generalized bin packing problem. *Transportation Research Part E: Logistics and Transportation Review, 48*(6), 1205–1220.

Bortfeldt, A., & Homberger, J. (2013). Packing first, routing second – A heuristic for the vehicle routing and loading problem. *Computers & Operations Research, 40*(3), 873–885.

Chekuri, C., & Khanna, S. (2005). A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing, 35*(3), 713–728.

Chu, P. C., & Beasley, J. E. (1998). A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics, 4*(1), 63–86.

Correia, I., Gouveia, L., & Saldanha-da-Gama, F. (2008). Solving the variable size bin packing problem with discretized formulations. *Computers & Operations Research, 35*(6), 2103–2113.

Crainic, T. G., Perboli, G., Rei, W., & Tadei, R. (2011). Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Computers & Operations Research, 38*(11), 1474–1482.

Doerner, K. F., Fuellerer, G., Hartl, R. F., Gronalt, M., & Iori, M. (2007). Metaheuristics for the vehicle routing problem with loading constraints. *Networks, 49*(4), 294–307.

Dyer, M. E., Kayal, N., & Walker, J. (1984). A branch and bound algorithm for solving the multiple-choice knapsack problem. *Journal of Computational and Applied Mathematics, 11*(2), 231–249.

Faroe, O., Pisinger, D., & Zachariasen, M. (2003). Guided local search for the three-dimensional bin-packing problem. *INFORMS Journal on Computing, 15*(3), 267–283.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research, 201*(3), 751–759.

Gallardo, J. E., Cotta, C., & Fernández, A. J. (2005). Solving the multidimensional knapsack problem using an evolutionary algorithm hybridized with branch and bound. In *Artificial intelligence and knowledge engineering applications: A bioinspired approach* (pp. 21–30). Berlin Heidelberg: Springer.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science, 40*(3), 342–350.

Gorski, J., Paquete, L., & Pedrosa, F. (2012). Greedy algorithms for a class of knapsack problems with binary weights. *Computers & Operations Research, 39*(3), 498–511.

Gu, P., Dong, H., & Zheng, Y. (2009). An improved greedy algorithm for rectangles packing problem. In *2009 4th international conference on computer science & education* (pp. 131–134).

Hifi, M., Michrafy, M., & Sbihi, A. (2006). A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem. *Computational Optimization and Applications, 33*(2–3), 271–285.

Iori, M., Salazar-González, J. J., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science, 41*(2), 253–264.

Kan, A. R., Stougie, L., & Vercellis, C. (1993). A class of generalized greedy algorithms for the multi-knapsack problem. *Discrete Applied Mathematics, 42*(2), 279–290.

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Knapsack problems*. Springer Science & Business Media.

Lawler, E. L., & Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research, 14*(4), 699–719.

Martello, S., Pisinger, D., & Toth, P. (1999). Dynamic programming and strong bounds for the 0–1 knapsack problem. *Management Science, 45*(3), 414–424.

Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research, 48*(2), 256–267.

Martello, S., & Toth, P. (1990a). *Knapsack problems: Algorithms and computer implementations*. John Wiley & Sons.

Martello, S., & Toth, P. (1990b). Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics, 28*(1), 59–70.

Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *OR Spectrum, 31*(4), 775–800.

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity*. Courier Corporation.

Pisinger, D. (1995a). A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research, 83*(2), 394–410.

Pisinger, D. (1995b). *Algorithms for knapsack problems* PhD thesis.Copenhagen, Denmark: University of Copenhagen.

Pisinger, D., & Sigurd, M. (2005). The two-dimensional bin packing problem with variable bin sizes and costs. *Discrete Optimization, 2*(2), 154–167.

Ren, C., Sun, J., & Wang, X. (2010). Study on hybrid genetic algorithm for multi-vehicle and multi-cargo loading problem. In *2010 IEEE international conference on intelligent computing and intelligent systems* (vol. 1, pp. 554–558).

Romao, O.C., dos Santos, A.G., & Arroyo, J.E.C. (2012, November). A two phase hybrid method to support 2D and 3D vehicle loading. In *2012 IEEE 12th international conference on intelligent systems design and applications (ISDA)* (pp. 320–325).

Scheithauer, G. (1991). A three-dimensional bin packing algorithm. *Journal of Information Processing and Cybernetics, 27*(5/6), 263–271.

Spieksma, F. C. (1994). A branch-and-bound algorithm for the two-dimensional vector packing problem. *Computers & Operations Research, 21*(1), 19–25.

Yüceer, Ü., & Özakça, A. (2010). A truck loading problem. *Computers & Industrial Engineering, 58*(4), 766–773.